Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

UNIVERSITÀ DEGLI STUDI DI MILANO

NOVA FACULDADE DE CIÊNCIAS E TECNOLOGIA

DOCTORAL THESIS IN COTUTELLE

UNIVERSITÀ DEGLI STUDI DI MILANO

Doctorate in Intersectoral Innovation

NOVA SCHOOL OF SCIENCE AND TECHNOLOGY

Doctorate in Computer Science

BRIDGING INTERPRETABILITY AND PERFORMANCE IN 3D DEEP LEARNING THROUGH GEOMETRIC INDUCTIVE BIASES

Advancing Power Grid Inspections with 3D Data and Geometric Insights

DIOGO RESTOLHO MATEUS MARÇALO LAVADO

Department of Environmental Sciences
Matriculation n. R14089
cycle XXXVIII
Scientific Area INFO-01/A and MATH-03/B
ORCID n. 0009-0005-5302-0659

Supervisor: Prof. Alessandra Micheletti
Co-supervisor: Prof. Cláudia Soares
Coordinator: Prof. Ernesto Damiani

A.Y. 2024/2025

Acknowledgements

I would like to start by thanking Professors Cláudia Soares and Alessandra Micheletti for their incredible guidance, patience, and constant support throughout this work. Your insight and constant support have shaped not only this thesis but also my way of approaching research with curiosity and rigor. Thank you both for believing in my ideas, challenging me to refine them, and helping me grow into a more confident researcher.

I also want to thank EDP NEW and Labelec for the opportunity to develop this project. To Manuel Pio, thank you for always being ready to help, for your guidance, and for trusting me with the freedom to learn, try, and sometimes fail. Your patience and encouragement made a huge difference.

To my research group buddies: Ruben Belo, Frederico Metelo, Pedro Valdeira, Filipa Valdeira, and Ricardo Ferreira, thank you for all the discussions, support, and laughs along the way. You made the work lighter and the lab a place I actually wanted to be in. And to some other honorable mentions who made this journey easier and more fun: Carolina, Diogos, and Manel. Thank you for the good energy, for the shared frustrations, and for always showing up. A special thank you to the best administrative queen of FCT, Gracinda Caetano, for saving me from endless paperwork, for always finding a way to make exceptions, and for answering hundreds of my emails with a smile. You truly keep everything running.

I have to give my biggest thanks to my friends Carolina Lopes, Tiago Soares, and Xavier Pacheco. Thank you for sticking around all these years. It has been eight loooong but good years together, filled with chaos, screams, sleepless nights, but mostly cackling howls of laughter. I know I can be unbearable, dramatic, and a bit too much sometimes, but it's comforting to know that you'll always be there, regardless. You've been my family away from home, my safe place, and my biggest source of laughter when I need it most. I could never have done this PhD without you, and I'll never forgive you for not stopping me from doing it in the first place.

To Carolina Vares, the best divorce gift ever. Thank you for the endless talks and hot tea, for the cigarette breaks, for the varanda breakfasts, and for the timeless vacations at your house that always reset my mind. You've celebrated every small victory with me

and called me out on my BS when I needed it. This PhD journey as been a lot better (and funnier) with you around.

To my love, Rodrigo, thank you for being who you are, for listening to me, for putting up with me, and for loving me exactly as I am. I've spent a long time convinced that I was hard to love, but you might be proving me wrong. I'm still cynical, still a bit stubborn, but there is nothing truer or better than what we have. Maybe you're right after all. I love you, truly and completely.

And finally, to my mom, Marina Mateus. You've been telling me since I was a little boy that I have no idea of the strength I carry inside. Thank you for being a fighter, for every sacrifice, for every word of encouragement, and for always being my home. Everything I achieve is yours as well.

”

"And here, poor fool! With all my lore, I stand no wiser than before."

— Goethe, Faust
(Writer)

”

"Perseverance is not a long race; it is many short races one after the other."

— Walter Elliot, Speech
(Scottish politician)

”

"It's not that I'm so smart, it's just that I stay with problems longer."

— Albert Einstein, Attributed
(Physicist)

Abstract

This thesis investigates how geometric inductive biases can address fundamental limitations in deep learning for 3D point clouds, particularly for safety-critical applications such as power grid inspection. Despite remarkable progress in 3D scene understanding, most state-of-the-art models operate as black-boxes, requiring substantial computational resources and large datasets to rediscover basic geometric relationships that could be encoded as priors. This gap becomes especially problematic in infrastructure monitoring, where interpretability, efficiency, and reliability are paramount.

We propose a novel research direction into geometric inductive biases (GIBs) through three distinct paradigms that span the interpretability-performance spectrum. Our first contribution, SCENE-Net, demonstrates that fully interpretable white-box models based on Group Equivariant Non-Expansive Operators (GENEOs) can achieve competitive performance with five orders of magnitude fewer parameters than black-box alternatives. Building on this foundation, SCENE-Net V2 introduces a gray-box approach that combines interpretable geometric feature extraction with standard black-box classification, bridging the gap between transparency and expressiveness for multiclass segmentation tasks. Finally, GIBLy addresses scalability limitations through lightweight geometric bias layer that operates directly on raw point clouds, enabling seamless integration with any 3D backbone while eliminating the computational bottlenecks of voxelization.

To support evaluation in power grid inspection applications, we introduce TS40K, the first large-scale benchmark for rural power grid inspection. Comprising over 40,000 kilometers of densely annotated UAV-acquired LiDAR data, TS40K captures the unique challenges of infrastructure monitoring. In this benchmark, we establish new performance records with 72% IoU for towers and 97% IoU for power lines. While our work confirms the existence of an interpretability-performance trade-off, we show that this trade-off need not be steep when geometric priors align well with task structure. To show the practical application of our work, we develop an inspection tool that utility operators can use for real-time monitoring and analysis of power grid infrastructure. This tool also includes a cost-benefit analysis and human-in-the-loop validation, demonstrating the viability of our approach in real-world deployment.

This research contributes to advancements in geometric 3D deep learning by showing that explicit geometric knowledge enhances learning, offering a path toward 3D scene understanding methods that are more robust, efficient, and with higher performance to guarantee real-life impact in critical venues such as power grid inspection.

Keywords: 3D Scene Understanding, Geometric Inductive Biases, Interpretable Machine Learning, Power Grid Inspection, 3D Semantic Segmentation, LiDAR Point Clouds

Resumo

Esta tese investiga como os *geometric inductive biases* podem abordar limitações fundamentais em aprendizagem profunda 3D, particularmente para aplicações críticas de segurança como a inspeção de redes elétricas. Apesar do progresso notável em *3D scene understanding*, a maioria dos modelos de estado da arte operam como *black boxes*, requerendo bastantes recursos computacionais e grandes quantidades de dados para redescobrir relações geométricas básicas que poderiam ser codificadas como *priors*. Esta lacuna torna-se especialmente problemática na monitorização de infraestruturas, onde a interpretabilidade, eficiência e fiabilidade são fundamentais.

Propomos uma exploração de *geometric inductive biases* (GIBs) através de três paradigmas distintos que abrangem o espectro interpretabilidade-performance. A nossa primeira contribuição, SCENE-Net, demonstra que modelos *white-box* completamente interpretáveis baseados em *Group Equivariant Non-Expansive Operators* (GENEOs) podem alcançar uma performance competitiva com menos cinco ordens de magnitude do número de parâmetros. Baseando-se nisto, SCENE-Net V2 introduz uma abordagem *gray-box* que combina extração de características geométricas de forma interpretável com classificação feita por modelos *black-box* standard, preenchendo a lacuna entre transparência e expressividade para segmentação multiclasse. Finalmente, GIBLy aborda limitações de escalabilidade através de *geometric inductive bias layers* leves que operam diretamente em *point clouds*, permitindo uma integração fluída com qualquer *backbone* 3D enquanto elimina por completo as desvantagens computacionais da voxelização.

De forma a realizármos uma avaliação rigorosa no domínio das redes elétricas, introduzimos TS40K, o primeiro *benchmark* de larga escala para inspeção de redes elétricas em ambientes rurais. Este dataset inclui mais de 40.000 quilómetros de dados LiDAR completamente anotados e demonstra os desafios únicos da monitorização de infraestruturas. Com o nosso *benchmark*, estabelecemos novos recordes de performance com 72% IoU para torres e 97% IoU para linhas de energia. Embora o nosso trabalho confirme a existência de um *trade-off* entre interpretabilidade e performance, demonstramos que este *trade-off* não precisa de ser acentuado quando os *priors* geométricos se alinham bem com a tarefa em questão. De forma a demonstrar a utilidade e relevância da nossa investigação,

desenvolvemos um programa para inspeções elétricas de fácil utilização que inclui uma análise custo-benefício e uma validação por responsáveis de manutenção enquanto o programa corre. Isto demonstra a viabilidade da nossa abordagem em cenários reais.

Por último, a nossa investigação contribui para a aprendizagem geométrica profunda ao mostrar que o conhecimento geométrico explícito melhora em vez de restringir a aprendizagem. Isto abre um caminho para sistemas de segmentação 3D que são simultaneamente interpretáveis, eficientes e operacionalmente viáveis para aplicações críticas de segurança.

Palavras-chave: 3D Scene Understanding, Geometric Inductive Biases, Interpretable Machine Learning, Power Grid Inspection, 3D Semantic Segmentation, LiDAR Point Clouds

Sommario

Questa tesi indaga come i *geometric inductive biases* possano risolvere limitazioni fondamentali nel *deep learning* 3D, in particolare per applicazioni critiche di sicurezza come l'ispezione delle reti elettriche. Nonostante i progressi significativi nel campo del *3D scene understanding*, la maggior parte dei modelli allo stato dell'arte funziona come *black boxes*, richiedendo ingenti risorse computazionali e grandi quantità di dati per riscoprire relazioni geometriche di base che potrebbero invece essere incorporate come *priors*. Questa lacuna si rivela particolarmente problematica nel contesto della monitorizzazione delle infrastrutture, dove interpretabilità, efficienza e affidabilità sono elementi fondamentali.

In questa tesi, proponiamo un'esplorazione dei *geometric inductive biases* (GIBs) attraverso tre paradigmi distinti che coprono l'intero spettro interpretabilità–performance. Il nostro primo contributo, SCENE-Net, dimostra che modelli *white-box* completamente interpretabili, basati su *Group Equivariant Non-Expansive Operators* (GENEOs), possono raggiungere performance competitive, con un numero di parametri inferiore di cinque ordini di grandezza rispetto ai metodi *black-box*. Su questa base, SCENE-Net V2 introduce un approccio *gray-box* che combina un'estrazione interpretabile di caratteristiche geometriche, con una classificazione effettuata da modelli *black-box* standard, colmando il divario tra trasparenza ed espressività nella segmentazione multiclasse. Infine, GIBLy affronta le limitazioni di scalabilità attraverso *geometric inductive bias layers* leggeri che operano direttamente su nuvole di punti, consentendo un'integrazione fluida con qualsiasi *backbone* 3D ed eliminando completamente gli svantaggi computazionali legati alla voxelizzazione.

Per condurre una valutazione rigorosa nel campo delle reti elettriche, introduciamo TS40K, il primo *benchmark* su larga scala per l'ispezione di reti elettriche rurali. Questo dataset comprende oltre 40.000 chilometri di dati LiDAR completamente annotati e cattura le sfide tipiche della monitorizzazione infrastrutturale: diversità strutturale negli oggetti critici, artefatti di acquisizione che si confondono con gli elementi della linea elettrica e *labels* semanticamente imprecise. La nostra validazione sperimentale su questo e altri quattro *benchmarks* evidenzia come i *geometric inductive biases* forniscano vantaggi consistenti quando i dati presentano sufficiente dettaglio geometrico.

Nel caso specifico dell'ispezione delle reti elettriche, abbiamo stabilito nuovi record di

performance con il 72% di IoU per le torri e il 97% di IoU per le linee elettriche. Sebbene il nostro lavoro confermi l'esistenza di un *trade-off* tra interpretabilità e performance, mostriamo che questo *trade-off* non deve necessariamente essere marcato quando le *priors* geometriche sono ben allineate con il compito da svolgere. Per dimostrare l'utilità e la rilevanza della nostra ricerca, abbiamo sviluppato un programma per ispezioni elettriche di facile utilizzo che include un'analisi costo–beneficio e una validazione da parte dei responsabili della manutenzione durante l'esecuzione del programma. Questo dimostra la fattibilità della nostra proposta in scenari reali.

In conclusione, la nostra ricerca contribuisce al *geometric deep learning* mostrando che la conoscenza geometrica esplicita non limita ma potenzia la capacità di apprendimento. Ciò apre la strada a sistemi di segmentazione 3D allo stesso tempo interpretabili, efficienti e operativamente validi per applicazioni critiche di sicurezza.

Parole Chiave: 3D Scene Understanding, Geometric Inductive Biases, Interpretable Machine Learning, Power Grid Inspection, 3D Semantic Segmentation, LiDAR Point Clouds

Contents

List of Figures

Acronyms

BEV Bird's Eye View *(p. 43)*

CNN Convolutional Neural Network *(pp. 78–80, 84, 89, 90, 105)*
CV Computer Vision *(pp. 59, 146)*

DL Deep Learning *(pp. 123, 124)*

EDP Energias de Portugal *(p. 147)*

FPS Farthest Point Sampling *(p. 47)*

GENEO Group Equivariant Non-Expansive Operator *(pp. xv, xvi, 77–86, 88–90, 92–99, 101–109, 111–121, 129, 131)*
GIB Geometric Inductive Bias *(pp. xvi, 123, 125–133, 136, 138, 140–142, 144)*
GIBLy Geometric Inductive Bias Layer *(pp. xvi, 122–125, 132–140, 142–144)*

IoU Intersection over Union *(pp. 48–50, 56, 61–65, 67–69, 72, 76, 78, 90, 91, 93, 95, 98, 102, 115–118, 137, 138, 146–148)*

Labelec EDP's Energy R&D Laboratory *(p. 147)*
LiDAR Light Detection and Ranging *(pp. xv, 38–43, 54, 56, 58, 59, 80, 81, 128, 146)*

mIoU Mean Intersection over Union *(pp. 47, 49, 55, 61, 68, 114–118, 120, 123, 125, 137–141, 143)*
ML Machine Learning *(pp. 39, 44, 59, 104)*
MLP Multi-Layer Perceptron *(pp. 108, 113, 120, 122, 124, 126, 132, 134)*

RGB Red, Green, Blue *(pp. 39, 60)*
RGB-D Red, Green, Blue, Depth *(p. 128)*

UAV Unmanned Aerial Vehicle *(pp. 38–41, 45, 47, 50, 53, 56, 58, 59, 80, 81)*

Symbols

α	Weight scaling hyperparameter in class balancing *(pp. 87, 112)*
β	Cone slope or inclination parameter *(pp. 86, 110, 111, 128, 129)*
η	geometric inductive bias *(pp. 126, 127, 130, 131)*
$\urcorner$	composite biases *(p. 131)*
Σ	covariance matrix for ellipsoid *(pp. 129, 130)*
$\mathcal{D}$	training dataset *(pp. 87, 111, 112)*
Δ^{m-1}	(m-1)-dimensional simplex *(pp. 87, 111)*
ϵ	Small hyperparameter to avoid zero weights in class weighting function *(pp. 87, 112)*
g	kernel function *(pp. 81, 82, 84, 108)*
$f_w(\alpha, \epsilon; y)$	Class imbalance weighting scheme function *(pp. 87, 112)*
$g_{\mathrm{Ar}}(x)$	Arrow GENEO kernel response at x *(pp. 86, 110)*
$g_{\mathrm{Cn}}(x)$	Cone GENEO kernel response at x *(p. 111)*
$g_{\mathrm{Cy}}(x)$	Cylinder GENEO kernel response at x *(pp. 85, 110)*
$g_{\mathrm{Dk}}(x)$	Disk GENEO kernel response at x *(p. 110)*
$g_{\mathrm{El}}(x)$	Ellipsoid GENEO kernel response at x *(p. 111)*
$g_{\mathrm{NS}}(x)$	Negative Sphere GENEO kernel response at x *(pp. 86, 110)*
$\mathcal{L}_{\mathrm{seg}}$	Segmentation loss *(pp. 87, 111, 112)*
Λ	convex coefficient matrix *(pp. 107, 108, 111, 112)*
Φ	admissible measurement space *(pp. 85, 108)*
φ	an admissible measurement function *(pp. 81, 82, 85, 106, 108)*

$\mathcal{N}_q$	local neighborhood of query point q *(pp. 123, 127, 130)*
$\mathcal{N}_x$	local neighborhood of point x *(p. 126)*
$h(x) = \max(0, -x)$	non-negative penalty *(p. 112)*
C	number of additional features in a point cloud *(p. 81)*
m	number of GENEO kernels *(pp. 81, 82, 106, 107, 112)*
n	number of observers *(pp. 107, 111, 112)*
N	number of points in a point cloud *(p. 81)*
$\mathcal{H}$	GENEO observer *(pp. 82, 83, 107)*
$\mathcal{H}_i$	i-th GENEO observer *(p. 107)*
$\mathcal{H}'$	transformed observer features *(p. 108)*
ω	Suppression or weighting strength (e.g., Negative Sphere) *(pp. 86, 110, 111)*
Γ	general GENEO operator used in this work *(pp. 81, 82, 85, 86, 106, 108)*
Ψ	output GENEO space *(p. 108)*
ϑ	shape parameters *(pp. 81, 82, 84–87, 106–108, 110–112, 126, 127, 132)*
$\boldsymbol{\phi}$	Orientation / rotation parameters for kernels *(pp. 123, 127–130)*
$\mathcal{P}$	point cloud *(pp. 7, 81, 82, 106, 126)*
$\mathcal{M}$	probability map *(pp. 82–84, 87)*
$\mathcal{M}_{\Lambda,\vartheta}$	model output with parameters *(p. 112)*
$\widetilde{\mathcal{M}}$	thresholded prediction map *(p. 84)*
r	Radius parameter (cylinder, cone, disk, ellipsoid) *(pp. 85, 86, 110, 111, 123, 127–130)*
$\mathbb{R}^3$	3D Euclidean space *(pp. 81, 85, 106, 108, 126, 130)*
$\mathbb{R}^C$	C-dimensional real space *(p. 126)*
$(t)_+ = \max\{0, t\}$	ReLU activation function *(p. 83)*
$\mathbb{R}^{N\times(3+C)}$	point cloud space with N points and 3+C features *(pp. 7, 81, 106, 126)*
$z_{\boldsymbol{\phi}}(x)$	rotated coordinates in canonical reference frame *(pp. 128, 129)*
$R_{\boldsymbol{\phi}}$	rotation matrix *(p. 127)*
$\mathbb{R}^T_+$	non-negative orthant *(pp. 87, 111, 112)*
σ	Gaussian spread parameter of a kernel *(pp. 85, 86, 110, 111)*
t	Thickness / shell tolerance for hollow variants *(pp. 128–130)*
τ	threshold parameter *(p. 84)*
$v(x)$	Height of point x relative to a reference plane *(pp. 86, 110)*

w	Vertical threshold / plane height in disk kernels *(p. 129)*
W	weight matrix for linear combination of GIBs *(pp. 131, 132)*
λ	convex combination weights *(pp. 82, 87, 107)*

1 Introduction

3D scene understanding is a rapidly evolving field in computer vision, with applications spanning autonomous driving, robotics, or augmented reality. At its core, the field involves interpreting 3D data, whether point clouds, meshes, or volumetric representations, to extract meaningful insights about the environment. The central problems tackled include semantic segmentation, object detection, and scene reconstruction, where models must assign semantic labels to points or regions, identify and localize distinct objects, and build structured 3D representations from 2D images.

While traditional signal and image processing have laid strong foundations, introducing a third spatial dimension brings both unique advantages and significant challenges. On the positive side, 3D data captures spatial geometry more faithfully than 2D projections, enabling accurate depth perception, size and shape estimation, and precise spatial localization, all crucial in dynamic environments such as traffic scenes. Moreover, 3D data is less susceptible to environmental conditions like lighting changes or motion blur, which often degrade 2D image quality. However, these advantages come at a cost. Point clouds, despite being one of the most common formats of 3D data, lack regular structure, suffer from non-uniform sampling, and are permutation invariant, making them difficult to process with standard convolutional architectures that thrive on grid-like data. Volumetric representations impose structure through voxelization but face significant memory and computational costs, especially in large-scale scenes, due to their cubic scaling. Meshes introduce yet another layer of complexity, with irregular topologies that require expensive operations such as remeshing or normal estimation and customized convolutions on graph-like structures.

Despite these obstacles, the field has progressed rapidly over the past decade. The emergence of deep learning models has dramatically advanced 3D scene understanding, from the seminal PointNet [78] to transformer-based approaches [110, 111, 121], alongside large annotated datasets such as SemanticKITTI [6], ScanNet [23], and NuScenes [12]. These developments have driven remarkable improvements in performance and robustness across diverse applications. However, a clear trend has emerged. As models continue to grow in depth and scale – for example, Point Transformer V3 [111] nearly quadruples the

parameter count of its predecessor (from 12.8M to 46.2M) – the performance gains have become increasingly marginal, often limited to improvements of only 1–3% in mean IoU across major benchmarks. These incremental advances depend on heavy computational resources, massive datasets, and multi-dataset training pipelines that are accessible only to large research groups or industry laboratories.

This pattern raises a critical question: *can we design better-performing models without simply scaling up data and model size?* To understand this challenge, it is useful to revisit the evolution of 2D vision. The rise of convolutional neural networks (CNNs) transformed the field not by increasing the expressive power of neural networks, since fully connected networks are universal function approximators, but by embedding strong inductive biases. CNNs exploit two fundamental assumptions: that local regions contain meaningful information and that similar features may appear across spatial locations. Through locality and translation equivariance, these models learn efficiently, generalize better, and require fewer parameters than fully connected architectures that treat all input pixels as independent and unstructured. In short, the success of CNNs stems from the introduction of inductive biases that mirror the geometric properties of images.

In 3D scene understanding, however, such inductive biases are still largely absent. Many early methods attempted to impose structure on 3D data to reuse well-understood 2D operations. Multi-view approaches projected point clouds into 2D images from multiple viewpoints [26, 54, 91], allowing the use of CNNs trained on projected depth or intensity maps. Volumetric methods [15, 72, 120] discretized 3D space into voxels, enabling 3D convolutions but introducing quantization errors and severe memory bottlenecks. These strategies improved early performance but remained inefficient, losing resolution and geometric relationships present in 3D data. To address these limitations, point-based models emerged [44, 47, 58, 78–80, 96], operating directly on raw point clouds. They retain spatial precision, but often struggle to define consistent local neighborhoods. Recent attention-based methods extended these ideas further. Inspired by Transformers [104], several works adapted self-attention to unordered 3D points [48, 49, 76, 106, 110, 111, 121], achieving state-of-the-art benchmark results. Yet, these architectures remain largely data-driven and agnostic to geometry: their inductive biases come from generic attention mechanisms rather than explicit spatial structure.

This brings us to a key insight: **3D data is inherently geometric**. Unlike images, which are projections of reality onto a 2D plane, 3D data directly encodes the true spatial relationships between points, objects, and surfaces in the physical world. Depth, orientation, and curvature are intrinsic to this representation, not inferred. Nonetheless, current models largely ignore this fundamental property, learning spatial reasoning implicitly rather than exploiting the geometry explicitly embedded in the data. As a result, they require vast amounts of data to learn patterns that could otherwise be modeled directly through geometric principles. In this thesis, we explore the hypothesis that embedding **geometric inductive biases** into learning architectures can improve performance and interpretability without relying solely on data or model scaling. Geometric inductive biases

(GIBs) can be seen as architectural constraints that reflect known geometric principles, such as symmetry, invariance, and locality in 3D space. By introducing these biases explicitly, models can learn more efficiently, generalize across domains, and maintain transparency in their decision-making.

In parallel to this methodological challenge, we also focus on practical applications of geometric inductive biases. Research efforts have focused primarily on benchmark-rich domains such as autonomous driving or indoor mapping; however, safety-critical real-world applications remain underexplored. Power grid inspection exemplifies this gap: despite the maturity of 3D scene understanding techniques, infrastructure monitoring still depends largely on manual visual assessment. In Portugal, this process is carried out by companies such as EDP, the country's largest energy distributor, and Labelec, its dedicated research and testing center, which performs periodic inspections of power lines both domestically and abroad. Through a close collaboration and internship with these organizations, this work directly addresses the challenges they face in automating inspection pipelines and bridges academic research with industrial needs. Specifically, human operators must inspect thousands of kilometers of power lines, a process that is costly, time-consuming, and prone to human error. Missed defects can lead to severe outcomes, including power outages, infrastructure damage, or even wildfires. Automating this task requires models that are not only accurate but also reliable, explainable, and efficient.

1.1 Thesis Overview and Contributions

This thesis addresses these fundamental challenges by exploring the integration of geometric inductive biases (GIBs) into 3D deep learning, with a particular focus on advancing both methodology and practical applications in safety-critical domains. Through a comprehensive research program spanning interpretable architectures, benchmark dataset development, and real-world deployments, this work advances our understanding of how explicit geometric knowledge can enhance both the performance and interpretability of 3D scene understanding systems. We argue that **geometric inductive biases**, similar to the domain-specific assumptions that powered CNNs in 2D vision, can and should be embedded into 3D scene understanding models. By explicitly modeling local geometric relationships, we can guide feature learning in ways that improve generalization, reduce reliance on massive datasets, and yield more interpretable and efficient models. The integration of these biases is not merely an academic exercise but a practical necessity for achieving reliable, scalable, and trustworthy 3D analysis systems tailored to high-impact domains. Our approach incorporates GIBs through three complementary methodologies, each representing a different point along the interpretability-application spectrum:

White-box Interpretable Models. Our first contribution, SCENE-Net, demonstrates that fully interpretable models can achieve competitive performance on specialized tasks

when equipped with appropriate geometric priors. By leveraging Group Equivariant Non-Expansive Operators (GENEOs) as building blocks, SCENE-Net embeds domain knowledge directly into its architecture, making every computational step traceable and meaningful. The model achieves remarkable parameter efficiency, delivering competitive results with five orders of magnitude fewer parameters than state-of-the-art baselines. This work corresponds to the published paper: Lavado, D., Bocchi, G., Frosini, P., Micheletti, A. and Soares, C., 2025, *SCENE-Net: Geometric Induction for Interpretable and Low-Resource 3D Pole Detection with Group-Equivariant Non-Expansive Operators*. In the Computer Vision and Image Understanding journal.

Gray-box Hybrid Approaches. Recognizing the limitations of purely white-box approaches in handling multiclass setups, SCENE-Net V2 introduces a hybrid paradigm that combines interpretable geometric feature extraction with black-box classification. By using GENEOs as feature extractors followed by traditional neural network classifiers, this approach preserves the interpretability of geometric priors while gaining the expressiveness needed for complex segmentation tasks. This work was published as Lavado, D., Soares, C. and Micheletti, A., 2024, *SCENE-Net V2: Interpretable Multiclass 3D Scene Understanding with Geometric Priors*. In ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling.

Lightweight Bias Layers for 3D Backbones. Our final methodological contribution, GIBLy (Geometric Inductive Bias Layer), addresses the scalability limitations of voxel-based approaches by introducing lightweight geometric inductive bias layers that operate directly on raw point clouds. GIBLy can be seamlessly integrated into any 3D backbone, adding only minimal parameters while consistently improving performance across diverse benchmarks by up to +11.48% mean IoU. This work was submitted to the Winter Conference on Applications of Computer Vision (WACV) 2026 with the title: *GIBLy: Improving 3D Semantic Segmentation through an Architecture-Agnostic Lightweight Geometric Inductive Bias Layer*.

Power Grid Inspection Dataset. Beyond methodological innovations, we recognized the need for a comprehensive dataset with meaningful labels to modernize rural power grid inspections with 3D Deep Learning. TS40K, our large-scale LiDAR dataset for rural power grid inspection, fills a significant gap in 3D computer vision benchmarks. Comprising over 40,000 kilometers of densely annotated transmission corridors, TS40K captures the unique challenges of infrastructure monitoring: extreme class imbalance, high-density noise, structural diversity, and realistic label noise. This work was published as Lavado, D., Santos, R., Coelho, A., Santos, J., Micheletti, A. and Soares, C., 2025, *Learning Under Noisy Labels, Spurious Points, and Diverse Structures: TS40K, a 3D Point Cloud Dataset of Rural Terrain and Electrical Transmission Systems*. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).

End-to-End Inspection Pipeline. Finally, we demonstrate the practical viability of our approaches through a complete inspection pipeline that integrates computer vision predictions with human-in-the-loop validation and cost estimation. This system achieves industry-relevant performance, with IoU exceeding 72% for towers and 97% for power lines when enhanced with our geometric inductive biases. This work was submitted for publication to the IEEE journal of Power and Energy with the custom title: *A Cost-Aware Machine Learning Decision Support System for LiDAR-Based Power Grid Inspections*.

1.2 Research Methodology and Novelty

Our research methodology centers on the integration of geometric inductive biases through the mathematical framework of Group Equivariant Non-Expansive Operators (GENEOs) [8, 13]. GENEOs provide a theoretical foundation that can, for instance, generalize traditional convolutions to arbitrary topological spaces. This enables the design of operators that respect geometric transformations while maintaining mathematical guarantees of non-expansiveness and convergence. The innovation lies in our recognition that GIBs need not be fixed architectural constraints but can be learned, differentiable modules that adapt to task-specific requirements. This insight enables three distinct integration strategies:

Full Integration: SCENE-Net demonstrates how geometric operators can form the complete computational backbone of a neural network. Every stage performs an interpretable operation, resulting in a white-box system where feature extraction directly corresponds to meaningful geometric measurements. This approach proves particularly effective for specialized tasks where domain knowledge can be precisely encoded.

Hybrid Integration: SCENE-Net V2 explores the middle ground by using geometric operators for initial feature extraction while employing traditional neural networks for final classification. This gray-box approach maintains interpretability in the geometric processing stages while providing flexibility for complex decision boundaries that may not have simple geometric interpretations.

Augmentative Integration: GIBLy represents the most pragmatic approach, introducing geometric processing as lightweight enhancement layers within existing architectures. This strategy preserves the power of state-of-the-art models and consistently improves upon it across diverse scenarios. However, it comes at the cost of losing most of its interpretability. Here, GIBs primarily serve as high-level feature extraction layers for geometric cues, rather than providing transparent, step-by-step reasoning.

On the other hand, from a practical perspective, this thesis establishes new standards for automated infrastructure inspection. The TS40K dataset provides the research community with its first large-scale, publicly available benchmark for rural power grid monitoring, while our inspection pipeline demonstrates operational viability with quantitative cost-benefit analysis. We reach industry-relevant performance thresholds that guarantee a safe

and robust inspection process with minimal human intervention. This directly translates into a significant cost reduction, a quicker inspection turnaround, and improved overall system reliability.

1.3 Document Structure

The remainder of this document is organized to provide a comprehensive account of our research contributions and their implications for the field of 3D scene understanding:

Chapter 2 provides a thorough review of related work, examining the evolution of 3D deep learning architectures, the mathematical foundations of geometric inductive biases, and current approaches to infrastructure monitoring. This review establishes the theoretical and practical context for our contributions.

Chapter 3 presents the TS40K dataset, our foundational contribution that enables evaluation of 3D scene understanding methods in the context of power grid inspection. We detail the data collection process, annotation methodology, and comprehensive benchmark evaluation that establishes baseline performance for this challenging domain.

Chapter 4 describes the development and evaluation of our practical inspection pipeline. This chapter demonstrates how academic research can be translated into operational tools, including detailed cost-benefit analysis and integration strategies for existing maintenance workflows.

Chapter 5 introduces SCENE-Net, our first approach to incorporating geometric inductive biases through fully interpretable architectures. We demonstrate their effectiveness in creating transparent, efficient models for supporting tower 3D segmentation.

Chapter 6 extends this work with SCENE-Net V2, exploring gray-box architectures that balance interpretability with performance. This chapter examines the trade-offs inherent in hybrid approaches and demonstrates that a set of carefully designed GIBs can enhance multiclass segmentation.

Chapter 7 presents GIBLy, our architecture-agnostic approach to integrating geometric inductive biases into existing 3D models. We demonstrate consistent performance improvements across diverse benchmarks and analyze the scalability advantages of operating directly on point clouds rather than voxelized representations.

Finally, Chapter 8 synthesizes our findings, discusses their implications for the broader field of 3D scene understanding, and outlines promising directions for future research. We examine how geometric inductive biases might evolve and consider their potential impact on other domains requiring interpretable, efficient 3D analysis.

2

Related work

This chapter surveys the main theoretical and methodological background that underpin this thesis. Section 2.1 traces the development of 3D scene understanding, highlighting the main challenges of working with point clouds and reviewing the progression from early projection and voxel-based techniques to current point-based and hybrid models, along with a summary of important benchmark datasets. Section 2.2 discusses the concept of inductive bias in deep learning, outlining its significance, the adaptation of convolutional operators, and the emergence of group equivariant approaches. Section 2.3 introduces the Group Equivariant Non-Expansive Operators (GENEOs) framework, explaining its topological underpinnings, the shift from raw data to functional representations, and the formal definition of GENEOs as a foundation for geometric inductive biases used in this work.

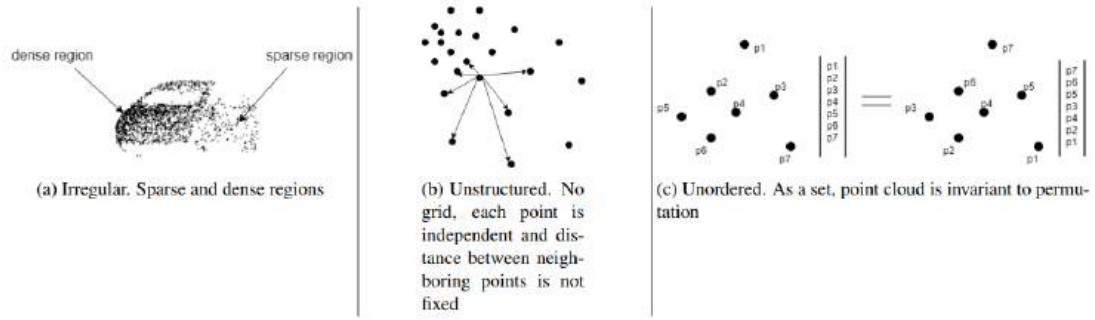
2.1 Overview of 3D Scene Understanding

2.1.1 Introducing 3D Point Clouds

Point clouds have become a standard representation for capturing and analyzing three-dimensional environments. At their core, point clouds are unordered sets of points defined in Euclidean space, typically represented as $\mathcal{P} \in \mathbb{R}^{N\times(3+C)}$, where N is the number of points, each described by spatial coordinates (x, y, z) and C are optional additional features such as color (RGB), surface normals, intensity, or semantic labels.

The widespread adoption of depth-sensing technologies, such as LiDAR and RGB-D cameras [1, 89], has facilitated the collection of large-scale point clouds. Consumer devices like smartphones and autonomous vehicles now routinely generate high-resolution point clouds [59], fueling the creation of large-scale annotated datasets such as ScanNet [23], S3DIS [3], KITTI [6], among others. These resources have become foundational benchmarks for advancing 3D scene understanding using data-driven approaches.

Even though point clouds have a compact format and are extremely detailed, they also pose a unique set of challenges for deep learning methods, primarily due to their irregular, unordered, and variable-density nature:



(a) Irregular. Sparse and dense regions

(b) Unstructured. No grid, each point is independent and distance between neighboring points is not fixed

(c) Unordered. As a set, point cloud is invariant to permutation

Figure 2.1: Key challenges in processing 3D point clouds [7]. (a) Heterogeneous density: objects may contain both densely and sparsely sampled regions, complicating local feature extraction. (b) Lack of structure: unlike images, point clouds do not lie on a regular grid, requiring custom neighborhood and aggregation strategies. (c) Permutation invariance: the order of points does not alter the underlying geometry, demanding specialized model designs to ensure robustness.

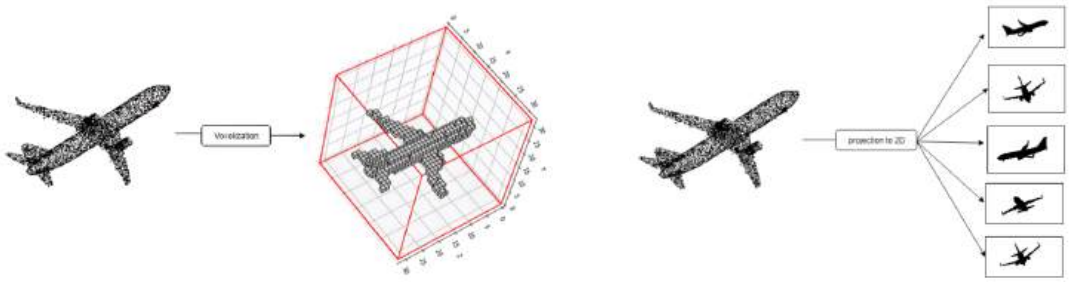
Irregular Sampling and Heterogeneous Density: Unlike images that present uniform pixel spacing, point clouds may exhibit uneven sampling densities across different parts of a scene. Objects can have both dense and sparse regions depending on occlusions, sensor perspective, or reflectivity, as shown in Fig. 2.1 (a). This irregularity complicates local feature extraction and challenges models reliant on fixed-size neighborhoods (e.g., k-NN), which can either miss fine-grained details or incorporate irrelevant context.

Lack of Structure: Point clouds are inherently unstructured. They do not lie on a regular grid, and there is no implicit ordering or spatial adjacency like in image pixels, as illustrated in Fig. 2.1 (b). This absence of structure means that traditional convolutional architectures are not directly applicable. Instead, learning effective point-wise representations requires custom aggregation functions or graph-based reasoning to capture local geometry and long-range dependencies.

Permutation Invariance: Point clouds are independent of the order in which they are stored, as it does not change the represented scene. This is depicted Fig. 2.1 (c). Deep learning methods often correlate the order of the input to the intended output. However, invariance to permutations clashes with this behavior. As a result, state-of-the-art methods often show additional strategies to mitigate this challenge.

2.1.2 Projection-based Methods

A straightforward strategy to apply well-established 2D learning techniques to 3D data is to project point clouds into multiple 2D views. Projection-based methods render 3D scenes from different camera perspectives and apply 2D convolutional neural networks (CNNs) to the resulting images, typically RGB or depth maps, as illustrated in Figure 2.2b. This paradigm benefits from leveraging powerful pretrained 2D backbones, such as ResNet [42], and has shown strong performance in early 3D classification and retrieval tasks [26, 54, 68, 91, 117].



(a) Voxelization. (b) Multi-view projection.

Figure 2.2: Structured-based learning techniques for 3D scene understanding [7]. (a) Voxelization introduces regular grids for 3D convolutions, supporting efficient spatial reasoning but limited by quantization artifacts and memory inefficiency. (b) Multi-view projection enables the use of powerful 2D CNNs on 3D data, leveraging pretrained image models but losing depth information, struggling with occlusions, and missing fine 3D details critical for scene understanding.

Despite these strengths, the inherent loss of depth information and the difficulty of occlusion handling limit their expressiveness in more complex tasks. For instance, multi-view pipelines may struggle to accurately capture spatial context or reconstruct fine 3D structures, which are essential for scene-level tasks such as 3D semantic segmentation or object detection [40]. Moreover, generating and fusing multiple 2D projections can introduce additional computational overhead. Consequently, projection-based methods have been mostly superseded by more expressive volumetric or point-based approaches in current research.

2.1.3 Voxel-based Methods

To introduce structure into the inherently unstructured nature of point clouds, voxel-based methods discretize 3D space into regular grids, typically denoted as $N \times M \times K$ voxel volumes. Each voxel aggregates local geometric information, often through occupancy or density statistics, allowing the use of 3D convolutional kernels over the structured tensor. Early works such as VoxNet [72] and more recent advances like MinkowskiNet [18] demonstrate the viability of this approach for large-scale scene understanding.

Figure 2.2a illustrates how voxelization transforms a 3D point cloud into a structured voxel grid. In this format, each voxel at aggregates points from the original data, and a representation function encodes the presence or density of points within each voxel. Encoding the density of point in a 3D voxel often leads to extremely small values representing sparse regions, which can affect the training process of deep learning models. Thus, a common practice is to use a binary occupancy function, where each voxel is assigned a value of one if it contains points and zero otherwise. As depicted in Figure 2.2a, voxelization reduces the sharpness of objects discretized, which results in loss of geometric detail at lower resolution voxel grids. Increasing the grid resolution can mitigate this, but comes at the cost of significantly higher memory usage and computational overhead,

as most voxels remain empty and 3D convolutions are inherently more expensive than their 2D counterparts [40] (with a relative cost 3 times higher). This trade-off between resolution and efficiency means that low-resolution grids are fast but imprecise, whereas high-resolution grids are accurate but resource-intensive [7, 40].

To address these limitations, octree-based methods [55, 107] partition the space hierarchically, refining only occupied regions and enabling higher effective resolutions (up to 256^3 voxels) without excessive memory consumption. In octree representations, the space is recursively subdivided, focusing computation on non-empty voxels and reducing the impact of sparsity. Despite these advances, point-based approaches have largely surpassed voxel-based methods in both accuracy and computational efficiency for many 3D scene understanding tasks.

2.1.4 Point-based Methods

Rather than altering the input structure, point-based methods operate directly on raw point clouds without imposing a voxel grid or projecting the data. This preserves the geometric fidelity of the original input and avoids quantization artifacts. The seminal PointNet architecture [78] pioneered this idea by applying shared Multi-Layer Perceptrons (MLPs) to each point individually, followed by a symmetric global aggregation function to ensure permutation invariance. Building on this, PointNet++ [79] introduced hierarchical feature learning by grouping points into local neighborhoods, enabling the extraction of local geometric features. Since then, a rich landscape of point-based methods has emerged, employing point convolutions [96], attention mechanisms [121], or newer version of MLP-based networks [80] to better capture inter-point relationships. Thus, we can subdivide point-based methods into three main categories: MLP-based, convolutional, and attention-based methods.

2.1.4.1 MLP-based Methods

MLP-based methods, such as PointNet [78] and PointNet++ [79], rely on shared MLPs to process point clouds. They introduced the concept of **hierarchical feature learning**, where local neighborhoods are formed and processed by multiple layers of MLPs and iteratively expand the receptive field to capture both local and global context. Additionally, they take advantage of **symmetric functions**, such as max or average pooling, to aggregate features across points, ensuring permutation invariance.

More recently, PointNeXt [80] revisited the MLP-based paradigm with a deeper and modular architecture that incorporates residual connections, local normalization, and an advanced data augmentation regime to improve performance and training stability. First, it employs an initial MLP block to project the feature space of the input points into a higher-dimensional space, akin to traditional transformer-based methods. Second, it introduces a **local coordinate normalization** step that normalizes the coordinates of each point relative to its local neighborhood instead of the global coordinate system. This

leads to larger coordinate values that are more suitable for network training. Lastly, it incorporates a **data augmentation** strategy that takes advantage of recent advances in 3D data augmentation, such as random rotations, scaling, cropping, random point dropping, color dropping and jittering, among others. This leads to a performance boost of up to 5% on standard benchmarks such as ScanObjectNN [102]. This work demonstrated that with the right design principles, MLPs can match or even outperform more complex convolutional and transformer-based models.

Other approaches, such as PointMLP [69], further challenge the necessity of explicit local geometric modeling by introducing a **geometric affine module**, which combines point features from a local neighborhood using learnable affine transformations, rather than, for instance, convolutional operations. This allows the network to adaptively weight and blend features from neighboring points.

2.1.4.2 Convolutional Methods

Convolutional methods for point clouds define local neighborhoods and learnable kernels directly in 3D space, avoiding the need for voxelization or 2D projections. PointCNN [58] introduced an χ**-transformation** that learns a canonical order of local neighborhoods, enabling permutation-aware convolutional operations on unordered point sets. While effective, this approach is sensitive to density variations and local geometric irregularities.

KPConv [96] marked a turning point by introducing **Kernel Point Convolutions**, where filters are defined as sets of kernel points positioned in Euclidean space, each with an associated weight. Feature aggregation is performed via distance-weighted interpolation, allowing the kernel to operate directly in continuous space. This design preserves spatial structure, supports arbitrary point distributions, and provides **translation-equivariant** filters without relying on grids. Building on this, several works have improved kernel flexibility and robustness. Spherical convolution methods [56] proposed discretizing neighborhoods in angular bins to achieve a traditional convolution kernel in spherical coordinates, while Monte Carlo convolutions [43] introduced probabilistic integration over non-uniform neighborhoods in order to better handle sample variability. PointConv [109] further advanced point cloud convolutions by approximating the convolutional kernel with an efficient shared MLP, analogous to a 1×1 convolution, which reduces both computational and memory overhead. Additionally, PointConv introduced an **inverse density scaling** mechanism that weights the MLP outputs according to local point density, effectively mitigating the effects of anisotropic sampling and improving robustness to non-uniform point distributions.

Overall, these methods and techniques demonstrate that convolutional operations can be adapted to the irregular structure of point clouds, enabling local feature extraction with spatial kernels. This highlights the value of incorporating inductive baises, namely translation equivariance and locality, into the design of point-based networks. A principle we extend through the use of interpretable, modular and simple geometric inductive

biases in our work.

2.1.4.3 Attention-based Methods

The attention mechanism was introduced in the Transformer architecture [104] as a means to model pairwise interactions between elements in a sequence. Given a set of input tokens, each is mapped to a query vector $\mathbf{q}$, key vector $\mathbf{k}$, and value vector $\mathbf{v}$ via learned linear projections. The core operation, known as scaled dot-product attention, is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \tag{2.1}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N\times d}$ are matrices of queries, keys, and values for all N input tokens, and d is the dimensionality of the embeddings. Intuitively, each point asks a *query* ($\mathbf{q}$) about the scene, compares it against the *keys* ($\mathbf{k}$) of all other points to measure relevance, and then aggregates the associated *values* ($\mathbf{v}$) of those relevant points. Each output is computed as a weighted sum of values, where the weights are determined by the similarity between queries and keys. This formulation enables each element to attend to every other, allowing global context modeling with minimal architectural assumptions.

In the context of point clouds, each point is treated as a token, and self-attention can be applied to model point-wise dependencies. However, two key challenges arise: first, **computational cost** grows quadratically with the number of points, which limits scalability to large scenes; second, the **positional encoding in 3D** is not straightforward since the standard NLP or vision strategies (e.g., sinusoidal embeddings) cannot be directly applied to unordered point sets.

Several works have been proposed to adapt attention mechanisms to the specific challenges of 3D point cloud processing. Point Transformer [121] was one of the first to successfully integrate self-attention into point-based architectures by constraining attention computation to local neighborhoods and incorporating relative positional encodings as follows:

$$\mathbf{y}_i = \sum_{\mathbf{x}_j \in \mathcal{X}(i)} \rho\left(\gamma\left(\varphi(\mathbf{x}_i) - \psi(\mathbf{x}_j) + \delta\right)\right) \odot \left(\alpha(\mathbf{x}_j) + \delta\right). \tag{2.2}$$

Here, $\mathbf{y}_i$ is the output feature vector for point i, and $\mathcal{X}(i) \subseteq \mathcal{X}$ represents the set of points in a local neighborhood (specifically, k nearest neighbors) of point $\mathbf{x}_i$. The functions φ, ψ, and α are pointwise feature transformations implemented as linear projections or MLPs that map input features to query, key, and value representations, respectively. The subtraction operation $\varphi(\mathbf{x}_i) - \psi(\mathbf{x}_j)$ serves as a relation function that captures the feature difference between points, enabling the model to reason about relative feature relationships. The position encoding is defined as $\delta = \theta(\mathbf{p}_i - \mathbf{p}_j)$, where θ is typically encoded as a MLP and $\mathbf{p}_i$ and $\mathbf{p}_j$ are the spatial coordinates of points i and j, respectively. This formulation incorporates spatial information and is added to both the attention

computation and the transformed features. The mapping function γ is implemented as an MLP with two linear layers and one ReLU nonlinearity that produces attention vectors for feature aggregation. Finally, ρ is a normalization function (typically softmax) that ensures the attention weights sum to one, and $\odot$ denotes element-wise multiplication that allows the attention vectors to modulate individual feature channels.

Such a design allows Point Transformer to effectively capture local geometric relationships and achieved state-of-the-art performance on several 3D benchmarks. In turn, its sucessor Point Transformer v2 [110] improves upon this architecture by adopting grouped vector attention and partition-based pooling, leading to enhanced scalability and improved training stability. The grouped design reduces memory usage while preserving context awareness, making the model more suitable for large-scale scenes. In a complementary direction, OctFormer [106] proposes a scalable design by organizing point clouds into octree-based hierarchies, thereby enabling multi-resolution attention across spatial partitions. Rather than attending over all points or fixed neighborhoods, OctFormer applies local self-attention at each octree level, which significantly reduces computational cost.

More recently, Point Transformer v3 [111] introduced several key methodological innovations that significantly advance the state of point-based attention mechanisms. The most notable contribution is the introduction of **serialized attention**, which leverages space-filling curves (specifically, Z-order curves) to impose a spatial ordering on inherently unordered point clouds. This serialization enables the application of efficient 1D attention mechanisms while preserving spatial locality, as nearby points in 3D space remain close in the serialized sequence. This approach dramatically reduces inference latency and memory consumption by 3.3 and 10.2 times when compared to Point Transformer v2 respectively, making attention feasible for large-scale scenes without sacrificing geometric awareness. From an architectural perspective, Point Transformer v3 simplifies the network design by removing complex grouped attention mechanisms in favor of streamlined attention blocks that operate on serialized point sequences. The training methodology also received significant improvements through enhanced data augmentation strategies. Point Transformer v3 employs **multi-scale training** with geometric augmentations, including random scaling, rotation, and elastic deformation that better simulate real-world sensor noise and environmental variations. Additionally, the model introduces **mix-up strategies** specifically adapted for point clouds, where multiple scenes are blended at the point level to improve generalization and robustness. The combination of serialized attention, simplified architecture, and advanced training strategies allowed Point Transformer v3 to achieve state-of-the-art results on major 3D benchmarks while maintaining computational efficiency.

2.1.5 Hybrid-based Methods

Hybrid methods seek to unify the advantages of multiple representations by fusing voxel-based, point-based, and sometimes image-based inputs into a single framework. These

models aim to overcome the limitations of individual paradigms by leveraging structured representations for efficient computation while retaining the geometric precision of raw points [17, 63, 64, 115]. The core motivation behind hybrid approaches stems from the complementary nature of different 3D representations. While point clouds preserve geometric detail and avoid quantization artifacts, they lack the structured neighborhood relationships that facilitate efficient convolutions. Conversely, voxel grids enable regular convolutions but suffer from memory inefficiency and resolution limitations. Image projections leverage powerful 2D networks but lose crucial depth information.

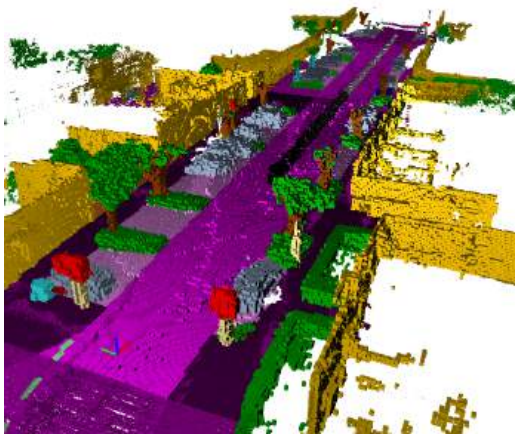
Several works have explored the integration of point and voxel representations. Point-Voxel CNN (PVCNN) [64] introduced a dual-branch architecture that processes point clouds through both voxel-based 3D convolutions and point-based operations in parallel. The voxel branch captures structured spatial context through regular convolutions, while the point branch preserves fine-grained geometric details. Features from both branches are fused through MLP layers. Building on this, 2DPASS [115] extends the fusion paradigm to outdoor large-scale scenarios by incorporating multi-modal sensor data. The method combines LiDAR point clouds with corresponding 2D camera images, leveraging both geometric and photometric information for enhanced scene understanding. Then, UniSeg [63] takes a different approach by unifying multiple representation learning paradigms within a single framework. The method employs a multi-branch encoder that simultaneously processes the same input through voxel-based, point-based, and image-based pathways. A sophisticated fusion module then combines features from all branches to produce a unified representation.

However, while hybrid methods offer the theoretical promise of combining the best aspects of multiple 3D representations, they face significant practical challenges in terms of architectural complexity, computational requirements, and feature alignment. The trade-off between improved representational capacity and increased system complexity means that hybrid approaches are most beneficial for applications where the performance gains justify the additional computational overhead and engineering effort.

2.1.6 Benchmarking Datasets

Benchmark datasets play a crucial role in the development and evaluation of 3D deep learning methods. They offer standardized scenes and ground truth annotations for tasks such as object classification, semantic segmentation, part segmentation, and 3D object detection.

For **indoor-scene mapping**, datasets such as ScanNet [23] and S3DIS [3] are dominant. ScanNet provides over 1,500 RGB-D scans of indoor environments with dense annotations for 20 object classes, captured using a handheld RGB-D sensor, as shown in Figure 2.3c. S3DIS, on the other hand, contains 3D scans of six large-scale indoor areas, with detailed point-wise labels, making it particularly useful for evaluating segmentation performance under clutter and occlusion.



(a) SemanticKITTI [6]

(b) ShapeNet-Part [14]

(c) ScanNet [23]

Figure 2.3: Representative examples from major 3D point cloud benchmarks. **Top Left:** SemanticKITTI features large-scale outdoor LiDAR scans with point-wise annotations, challenging models with sparsity and motion artifacts. **Top Right:** ShapeNet-Part contains synthetic 3D objects with fine-grained part segmentation labels, supporting part-level geometric reasoning. **Bottom:** ScanNet offers dense RGB-D indoor scenes with object-level annotations, making it a benchmark for cluttered semantic segmentation. Together, these datasets span diverse environments, sensor modalities, and labeling granularities.

For **outdoor large-scale segmentation**, the benchmark of choice is SemanticKITTI [6], which offers point-wise labels over full LiDAR sequences from the KITTI Odometry dataset. Captured with a Velodyne HDL-64E sensor, it introduces challenges such as motion blur, sparsity, and occlusions, pushing models to handle dynamic and noisy data, as illustrated in Figure 2.3a. Datasets like nuScenes [12] and Waymo Dataset [92] further extend these ideas by integrating multimodal sensors (e.g., radar, camera) and offering annotations for 3D detection and tracking across diverse driving scenarios.

In **object part segmentation**, datasets such as ShapeNet [14] and PartNet [74] have

become the standard for part segmentation over individual objects. ShapeNet provides over 16,000 3D models from 16 categories with point-wise part labels (e.g., airplane: wings, tail, engines), and is widely used in benchmarking point-based networks, with typical examples shown in Figure 2.3b. PartNet builds on this by introducing hierarchical, instance-level part decompositions across thousands of models, enabling the evaluation of both coarse and fine-grained segmentation.

2.2 Inductive Biases in 3D Deep Learning

This section details the role of inductive biases in deep learning, with an emphasis on their impact in computer vision tasks. We begin by defining inductive biases in the context of machine learning. We then review canonical operators like convolution in 2D and 3D. Finally, we survey group equivariant methods and assess the current landscape of successes and limitations in this space.

2.2.1 Understanding Inductive Biases

Inductive biases are the assumptions underlying a model's design that guide its ability to generalize beyond the training data. While machine learning aims to learn representations from data with minimal manual engineering, the learning process is necessarily constrained by a prior (i.e., an inductive bias) that determines which solutions the model is predisposed to favor even before training begins. This necessity stems from the no-free-lunch theorem [5, 35], which states that no learner can perform optimally across all possible data distributions without incorporating prior knowledge. Goyal and Bengio [35] formalize inductive bias as a form of preference or constraint on the learned function space, explicitly distinguishing between architectural, input-dependent, and optimization-induced biases. These biases manifest at different stages of the model pipeline, from architectural choices like convolutional layers, to data preprocessing strategies, and even to the dynamics of gradient-based optimization itself. Such biases are not optional artifacts of design, but essential mechanisms to guide learning in high-dimensional, under-constrained spaces.

In computer vision, convolutional neural networks (CNNs) show this principle. Their success is attributed not to raw expressiveness but to their *inductive structure*. CNNs embed translation equivariance and local connectivity into their architecture, reflecting strong prior knowledge about natural images: nearby pixels are more correlated than distant ones, and objects do not change semantics under translation. These biases yield models that are more data-efficient, robust to shifts, and capable of learning meaningful representations even in the presence of noise or limited supervision. In 3D vision, the problem is amplified. Point clouds lack the regular grid structure of images and exhibit high variability in density, orientation, and sampling. As a result, the strong architectural priors used in 2D (e.g., convolutions on a grid) are no longer directly applicable. This motivates the introduction of *geometric inductive biases*, which exploit spatial, topological,

and symmetry properties of 3D data to guide representation learning. Such biases not only improve generalization but also promote interpretability and robustness, especially in domains where data is scarce, or structurally complex.

2.2.2 The Convolution Operator

The convolution operator has been a foundational tool in deep learning, particularly in computer vision, due to its locality, parameter sharing, and equivariance properties.

2.2.2.1 Theoretical Formulation

At its core, convolution is a mathematical operator that combines two functions by integrating the product of one function with a reversed version of the other over space. In practical implementations, such as convolutional neural networks, this operation is typically implemented as cross-correlation, which measures the similarity between a signal (e.g., an image) and a filter. The continuous convolution operator between an input function $f(x)$ and a kernel $g(x)$ is defined as:

$$(f * g)(x) = \int_{\mathbb{R}^n} f(t) \cdot g(x - t)dt. \tag{2.3}$$

In the context of deep learning, this operation is discretized and implemented as a series of matrix multiplications. In image processing (typically $n = 2$), f represents the image and g is the convolutional kernel. The kernel is shifted across the domain of the input, producing a response at each location. In discrete 2D convolution, the continuous convolution in (2.3) is approximated by a correlation operator, for an image $I \in \mathbb{R}^{H\times W\times C}$ and a kernel $K \in \mathbb{R}^{k\times k\times C}$, the operation becomes:

$$(I * K)[i, j] = \sum_{m}^{k-1} \sum_{n}^{k-1} I[i + m, j + n] \cdot K[m, n]. \tag{2.4}$$

This sums over a local neighborhood centered at pixel $[i, j]$, producing an output that emphasizes certain features, such as edges or textures, depending on the kernel's learned weights.

2.2.2.2 Extension to 3D Point Clouds

The lack of structure and irregular density in point clouds sharply contrasts with the regular grid structure of images. To address this, projection-based and voxel-based methods force point clouds into a structured grid, allowing 2D and 3D CNNs methods to be employed. The convolution operator extends naturally to 3D tensors:

$$(I * K)[i, j, k] = \sum_{m}^{k-1} \sum_{n}^{k-1} \sum_{p}^{k-1} I[i + m, j + n, k + p] \cdot K[m, n, p]. \tag{2.5}$$

This 3D convolution operates over a cubic neighborhood and is used in models that take voxelized inputs. Due to the cubic growth of the input space, 3D convolutions are computationally expensive and memory-intensive, especially for high-resolution data.

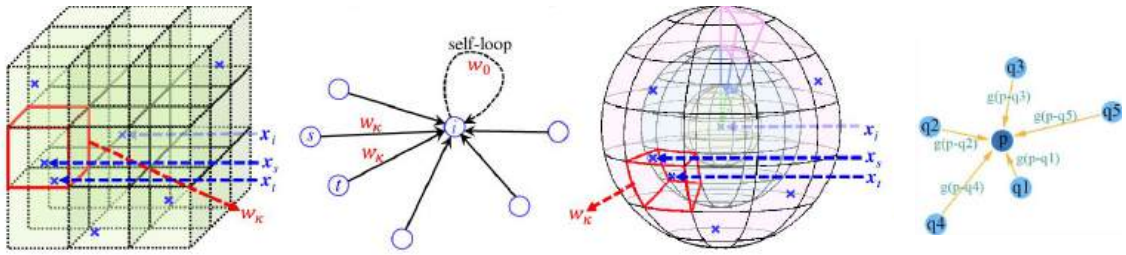


Figure 2.4: Different approaches to defining convolution kernels for point cloud processing [56, 113]. **Far Left:** Voxel-based methods (e.g., 3D CNNs) apply regular grid kernels to voxelized point clouds, enabling standard convolutions but introducing quantization errors and high memory cost. **Center Left:** Graph-based convolutions treat points as nodes with learned edge weights over local neighborhoods, enabling flexibility in topology. **Center Right:** Spherical kernels (e.g., SPH3D) discretize local neighborhoods into angular bins in spherical coordinates, enabling rotation-aware filtering with efficient spatial partitioning. **Far Right:** Continuous convolution methods (e.g., SpiderCNN) learn spatially-varying filters using learned weights (e.g., MLPs) over relative coordinates, offering adaptability but limited interpretability.

To effectively apply convolutions to point clouds, the convolution operator must be adapted to handle their irregular sampling and lack of grid structure. To this end, point-based methods have introduced several strategies that reinterpret point clouds as geometric graphs, where each point is a node and edges are defined via local neighborhoods [43, 88]. This graph-based formulation allows the definition of convolutional operations over local regions, preserving spatial relationships without imposing a voxel grid.

These methods can be broadly categorized based on how the convolution kernel is defined: **discrete** versus **continuous** kernels:

Discrete-kernel methods [55, 56, 96, 105, 123] define the convolution operator over local neighborhoods by discretizing the space around each point into bins (e.g., spherical or cylindrical coordinates) [56] or by placing a set of kernel points in Euclidean space that act as learnable weights. For instance, KPConv [96] defines a kernel as a set of K points in 3D space, and computes a weighted sum over features of neighboring points, where the weights depend on the distance between the neighbors and the kernel points. The key advantage of this approach is the adoption of a classical convolution-based framework with useful properties such as locality and translation equivariance. However, the fixed kernel layout can limit flexibility in highly irregular regions.

Continuous-kernel methods [37, 58, 108, 109, 113], on the other hand, model the convolution kernel as a continuous function over space. Instead of using predefined bins, these methods learn a function $h(x_j - x_i)$ that maps the relative position between a center point

x_i and a neighbor x_j to a weight. This function is typically implemented as a multi-layer perceptron (MLP), enabling the kernel to adapt to arbitrary geometric configurations. However, continuous-kernel approaches approximate the convolution weights using a learned MLP over relative offsets, rather than directly modeling the convolutional kernel as a continuous function. Additionally, MLPs fail to account for the geometric prior of 3D point clouds without it being explicitly large in terms of parameters, which can lead to overfitting. SpiderCNN [113] is the first work in continuous-kernel methods to recognize this limitation. The authors depart from this by explicitly parameterizing the convolutional kernel as a product of a step function and a Taylor expansion.

In contrast, our work begins with a 3D voxel-based formulation and arrives at an operator rooted in the formal convolution definition, where kernels are explicitly modeled as continuous functions of distance.

2.2.3 Group Equivariant Methodologies

3D scene understanding often entails reasoning about objects that appear in arbitrary orientations and positions. This poses a challenge for traditional models because they are typically sensitive to such transformations unless explicitly trained to account for them. A possible approach to overcome this limitation is the incorporation of group equivariance into neural networks.

Equivariance is a property of a function or an operator that preserves the structure of the input space under a specific group of transformations. Formally, an operator $f : \mathcal{X} \to \mathcal{Y}$ is said to be *equivariant* with respect to a group of transformations G if for every transformation $g \in G$, the following holds:

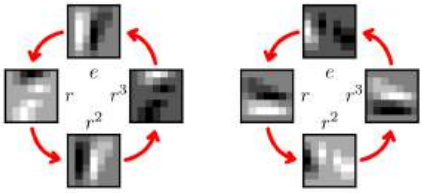
$$f(g \cdot x) = \rho(g) \cdot f(x), \quad \forall x \in \mathcal{X}, \tag{2.6}$$

where $g \cdot x$ denotes the action of g on the input x, and $\rho(g)$ is a (possibly different) representation of g acting on the output space $\mathcal{Y}$. For example, consider standard 2D convolutions in image processing, these are translation-equivariant, meaning that if the input image is shifted in space, the resulting feature maps the shift in the same way:

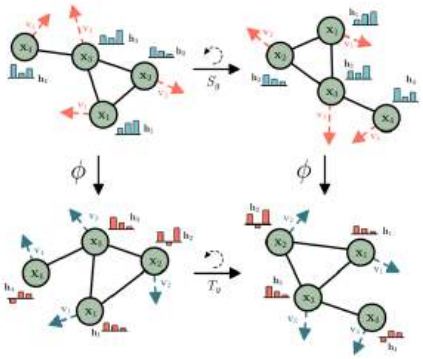
$$\text{Conv}(T_{\Delta x} x) = T_{\Delta x}(\text{Conv}(x)), \tag{2.7}$$

where $T_{\Delta x}$ denotes a translation by Δx.

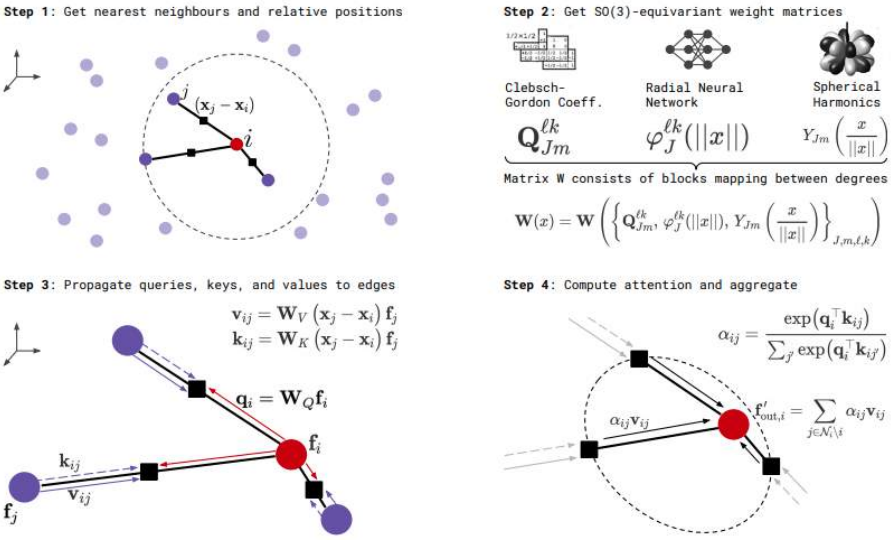
In 3D scene understanding, transformations of interest include translations, rotations and reflections, typically represented by the Euclidean group E(n) or the special Euclidean group SE(n). Designing models that are equivariant to such groups allows them to reason about objects and structures in a way that mirrors the physical symmetries of the world. This means, for instance, that rotating a 3D object in space will result in a corresponding, interpretable rotation of the model's internal features, rather than forcing the network to learn the rotation through brute-force data augmentation.



G-Convolution ($p4$ group) [19]



Equivariant GNN (EGNN) [84]



SE(3)-Transformer [33]

Figure 2.5: Visual comparison of group equivariant methodologies for 3D deep learning. **Top Left:** G-convolutions extend classical convolutions to symmetry groups such as $p4$ or $p4m$, enabling equivariance to translations and discrete rotations. **Top Right:** EGNN introduces a lightweight, continuous formulation that preserves SE(3)-equivariance through coordinate-based message passing. **Bottom:** SE(3)-Transformers generalize attention mechanisms to be equivariant under 3D rigid-body motions by leveraging group representations and directional geometry between points. The process is illustrated here in detail.

G-convolutions [19] generalize the conventional translation-equivariant convolution to arbitrary groups (see Fig. 2.5, top left). However, in practice, they are often restricted to discrete planar symmetry groups such as *p4* and *p4m*, which include translations, 90-degree rotations, and reflections. This work was later expanded by LieConv [29], which generalizes G-convolutions to arbitrary Lie groups, allowing for equivariance to more complex transformations. In turn, Tensor Field Networks (TFN) [97] represent point cloud features as geometric tensors: objects that transform under group actions according to specific rules. TFNs use spherical harmonics to construct rotation-equivariant convolution operators that maintain the properties of the input throughout the network. This approach

enables fine-grained modeling of physical phenomena, particularly in molecular and physical systems. SE(3)-Transformers [33] extend self-attention to be equivariant under 3D rigid-body transformations (see Fig. 2.5, bottom). They incorporate geometric information through relative positional encodings based on distances and directions between point pairs, using irreducible representations of SE(3) to preserve equivariance across layers. This framework has proven effective in tasks requiring spatial reasoning, such as molecular property prediction and protein structure modeling. Equivariant Graph Neural Networks (EGNN) [84] adopt a lighter approach by designing message passing layers that preserve equivariance without relying on spherical harmonics or high-order tensors (see Fig. 2.5, top right). They use simple geometric operations such as relative distances to propagate information while maintaining SE(3)-equivariance.

2.2.4 Effectiveness and Limitations

Inductive biases, when carefully chosen and implemented, can lead to substantial gains in performance, generalization, and interpretability. However, limitations remain. First, designing equivariant architectures often involves a trade-off between expressiveness and computational cost [28]. Second, while group equivariant methods generalize well to known transformations, they may struggle when the data distribution violates assumed symmetries [11]. Third, explicit priors can constrain learning if they misalign with the task, and tuning such priors requires domain expertise [35]. Moreover, the injected biases do not always behave as intended. They are often designed based on preconceived notions of the data's structure, which may not align with the data or how models utilize them [11, 34, 35]. In some cases, these assumptions can suppress useful signal or reinforce spurious patterns [34]. Finally, despite growing interest, the field lacks standardized benchmarks to evaluate the contribution of inductive biases across diverse 3D tasks. This makes it difficult to compare approaches and understand which biases are most beneficial under which conditions.

2.3 The GENEOs Framework

2.3.1 Motivation

Group Equivariant Non-Expansive Operators (GENEOs) are a general and rigorous mathematical framework designed to represent function operators that preserve symmetries under group actions [8]. Developed with a foundation in topological data analysis, GENEOs offer a broad abstraction that encompasses many well-known neural network components, including convolutional kernels, group-equivariant convolutions [19], and SE(3)-equivariant transformers [33]. Essentially, GENEOs unify these approaches by viewing them as special cases of function operators that act equivariantly on data represented as real-valued functions over a domain.

In this work, we explore GENEOs as a principled mechanism for instantiating *geometric inductive biases* (GIBs) in the context of 3D scene understanding. This serves two purposes: first, it provides a theoretically grounded way to embed spatial symmetry assumptions (e.g., translation, rotation) into our architectures; second, it lays the foundation for developing interpretable and robust operators that can be analyzed and composed modularly. We argue that GENEOs provide not only a formal theory for many of the inductive priors we seek in geometric deep learning, but also a practical blueprint for incorporating new ones whenever a problem domain presents a clear symmetry or structural constraint.

Furthermore, by designing our geometric priors as GENEOs, we benefit from important properties of the space of GENEOs that are desirable in a learning context. Specifically, these operators are *non-expansive* by definition, and under mild assumptions on the function spaces involved, the set of GENEOs forms a *compact and convex* subset [8, 13]. This leads to practical guarantees in terms of robustness, approximation, and optimization: GENEOs can be sampled, interpolated, and even learned over smooth Riemannian manifolds [13]. As we show later, these properties allow us to construct operators that are expressive, meaningful, and inherently aligned with the geometry of the data.

2.3.2 Topological Foundations of Data Representation

The GENEO framework begins with a simple but powerful shift in perspective: it proposes that data should be interpreted not as isolated samples, but as the result of measuring physical or abstract phenomena over structured domains. This viewpoint is inspired by topological data analysis (TDA), where datasets are modeled as topological spaces endowed with a set of continuous, real-valued functions that encode meaningful information about their structure.

Topological spaces are general mathematical objects that allow us to reason about nearness, continuity, and deformation. A key concept in this setting is *homeomorphism*: a continuous bijection with a continuous inverse. Homeomorphisms preserve all topological properties of a space, such as connectivity, compactness, and the number of holes or components. Two objects are considered topologically equivalent (homeomorphic) if one can be deformed into the other via a homeomorphism. For example, a donut and a coffee mug are homeomorphic; each has a single hole, and one can be smoothly morphed into the other without tearing or gluing.

The goal of employing topology in the GENEO framework is to construct a space of real-valued functions that can meaningfully represent measurements over a domain. This enables the design of operators that act not on individual data points, but on entire functions, transforming them in ways that preserve symmetries and structural properties. To this end, GENEOs rely on function spaces endowed with a topology that captures how these measurements vary across the domain. In practice, this allows us to encode data as bounded real-valued functions over a domain X, such as an image grid. Each function corresponds to a *measurement* (a way of observing the world) and the set of all

such functions defines an agent's observation space.

This approach offers several advantages. First, it naturally supports the modeling of invariances and symmetries through the use of group actions on X. Second, it enables the definition of function-space distances and topologies that reflect the similarity between observations. And third, it lays the foundation for defining structured operators (GENEOs, that is) that transform measurements while preserving key properties like equivariance and non-expansivity.

2.3.3 From Raw Data to Functional Representations

At the heart of the GENEO framework lies a fundamental shift in perspective: instead of modeling raw data directly, we model the space of *admissible measurements* over the data domain. That is, input samples are represented as functions defined on an input space X, mapping each data point to a real value that encodes a meaningful property (e.g., intensity, curvature, probability), which we denote as the space of admissible measurements Φ.

Formally, let X be the domain of observation (e.g., a 2D grid, 3D manifold, or point cloud), and let $\Phi \subseteq (\mathbb{R}_b^X, d_\infty)$ be the set of admissible measurements, where the topological space $(\mathbb{R}_b^X, d_\infty)$ denotes the space of bounded real-valued functions on X induced by the infinity norm:

$$d_\infty(\varphi_1, \varphi_2) = \|\varphi_1 - \varphi_2\|_\infty = \sup_{x \in X} |\varphi_1(x) - \varphi_2(x)|, \qquad \varphi_1, \varphi_2 \in \Phi. \tag{2.8}$$

Thus, each function $\varphi \in \Phi$ can be interpreted as a function measuring a property of the data at each point in X. For example, an image may be viewed as a function assigning RGB values to pixel locations, or a LiDAR scan as a function assigning intensity values to 3D coordinates.

To reason about similarity between points in X, we induce a pseudo-metric D_X via the measurements in Φ:

$$D_X(x_1, x_2) := \sup_{\varphi \in \Phi} |\varphi(x_1) - \varphi(x_2)|, \qquad x_1, x_2 \in X. \tag{2.9}$$

This quantity measures how distinguishable two points are according to the measurements in Φ. If x_1 and x_2 always yield the same values for all admissible functions, then $D_X(x_1, x_2) = 0$, and the points are indistinguishable from the observer's perspective. The space X is then endowed with the topology induced by D_X, which encodes the observer's perception of geometry through the lens of Φ. This construction allows us to shift focus from the ambient space X to the measurement space Φ, which formalizes the idea that the agent perceives the world through a restricted set of sensors or views.

2.3.4 Transforming Data

Once the data representation space Φ is defined, the GENEO framework introduces prior knowledge through the notion of symmetry, formalized by the action of a group of

transformations on the domain X. These transformations should preserve the structure of Φ, ensuring that the set of admissible measurements remains invariant under such operations. To capture this, we define the set of Φ-preserving homeomorphisms on X:

$$\mathrm{Homeo}_\Phi(X) := \left\{ g \in \mathrm{Homeo}(X) \;\middle|\; \varphi \circ g \in \Phi \text{ and } \varphi \circ g^{-1} \in \Phi,\ \forall \varphi \in \Phi \right\}. \tag{2.10}$$

This set consists of all bijective, continuous maps $g : X \to X$ with continuous inverse, such that pre-composing or post-composing any admissible measurement $\varphi \in \Phi$ with g yields another function in Φ.

Let $G \subseteq \mathrm{Homeo}_\Phi(X)$ denote the subgroup of transformations under which we wish to enforce equivariance. In practice, G captures the geometric invariances we expect from the problem domain, for example, translations or different orientations.

We can then define a pseudo-metric on G that quantifies the discrepancy between transformations in terms of their effect on the space of measurements Φ. This is given by:

$$D_G(g_1, g_2) := \sup_{\varphi \in \Phi} \|\varphi \circ g_1 - \varphi \circ g_2\|_\infty. \tag{2.11}$$

It induces a topology on G which is compatible with the action of G on Φ, and under mild conditions (e.g., compactness and completeness), the space (G, D_G) forms a topological group [8].

Natural Pseudo-Distance. The notion of similarity between two data functions is encoded by the *natural pseudo-distance*. This metric, originating from TDA [32], serves as the "ground truth" distance between functions in Φ under the equivalence induced by G:

Definition 1 (Natural Pseudo-Distance)**.** *Let $G \subseteq \mathrm{Homeo}_\Phi(X)$ be a group acting on Φ via precomposition. The **natural pseudo-distance** $d_G : \Phi \times \Phi \to \mathbb{R}$ is defined by:*

$$d_G(\varphi_1, \varphi_2) := \inf_{g \in G} \|\varphi_1 - \varphi_2 \circ g\|_\infty. \tag{2.12}$$

Intuitively, $d_G(\varphi_1, \varphi_2)$ measures how close two measurements $\varphi_1, \varphi_2 \in \Phi$ are after optimally aligning them via a transformation in G. If φ_1 and φ_2 encode the same information up to symmetry, that is, they are G-equivalent, then $d_G(\varphi_1, \varphi_2) = 0$. In other words, if φ_1 and φ_2 are supposed to be equivalent when acted on by transformations in G, meaning that such measures are equivariant with respect to G, then $d_G(\varphi_1, \varphi_2) = 0$.

Invariance Properties. The metric d_G enjoys an important invariance property: it is stable under the action of G on either argument. This is formalized below:

Definition 2 (Strong G-Invariance)**.** *A pseudo-metric $d : \Phi \times \Phi \to \mathbb{R}$ is said to be strongly G-invariant if for all $\varphi_1, \varphi_2 \in \Phi$ and all $g_1, g_2 \in G$, we have:*

$$d(\varphi_1, \varphi_2) = d(\varphi_1 \circ g_1, \varphi_2) = d(\varphi_1, \varphi_2 \circ g_2) = d(\varphi_1 \circ g_1, \varphi_2 \circ g_2). \tag{2.13}$$

Proposition 1. *The natural pseudo-distance d_G is strongly G-invariant. (Proof in Appendix C of [8])*

This property ensures that d_G measures remain unaltered by the action of G on the functions in Φ. In other words, the distance between two measurements is invariant to the transformations G applied to them.

Interpretation and Challenges. The natural pseudo-distance offers a way to compare data in the presence of known symmetries, and is widely considered a theoretical ideal for symmetry-aware learning. However, computing d_G exactly is often infeasible in practice, especially when G is infinite, continuous, or large [8]. This motivates the need for approximations, such as through the use of GENEOs and persistent homology, which serve as tractable surrogates.

2.3.5 Group Equivariant Non-Expansive Operators (GENEOs)

Having introduced the data representation and the role of symmetry-preserving transformations, we are now prepared to formally define Group Equivariant Non-Expansive Operators (GENEOs). These operators constitute the fundamental building blocks of the GENEO framework, acting as structure-preserving maps between spaces of admissible measurements.

Let us define a *perception pair* as the tuple (Φ, G), where Φ is a set of admissible real-valued functions $\varphi : X \to \mathbb{R}$ defined on a compact topological space X, and $G \subseteq \mathrm{Homeo}_\Phi(X)$ is a group of homeomorphisms acting on X that preserve the structure of Φ under composition. In what follows, we consider two perception pairs (Φ, G) and (Ψ, H), with G and H acting on Φ and Ψ, respectively. To relate these symmetry groups, we require a group homomorphism $T : G \to H$, which satisfies:

$$T(g_1 \circ g_2) = T(g_1) \circ T(g_2), \qquad \forall g_1, g_2 \in G. \tag{2.14}$$

This mapping translates symmetry transformations from the input space to the output space in a way that preserves their compositional structure.

Definition 3 (Group Equivariant Non-Expansive Operator)**.** *Let (Φ, G) and (Ψ, H) be perception pairs, and let $T : G \to H$ be a group homomorphism. A map $F : \Phi \to \Psi$ is said to be a Group Equivariant Non-Expansive Operator (GENEO) if the following two conditions are satisfied:*

1. ***Equivariance:*** *For all $\varphi \in \Phi$ and all $g \in G$,*

$$F(\varphi \circ g) = F(\varphi) \circ T(g). \tag{2.15}$$

2. ***Non-expansivity:*** *For all $\varphi_1, \varphi_2 \in \Phi$,*

$$d_\Psi(F(\varphi_1), F(\varphi_2)) \leq d_\Phi(\varphi_1, \varphi_2), \tag{2.16}$$

where d_Φ and d_Ψ denote the sup-norm pseudo-distances induced on Φ and Ψ, respectively.

The equivariance condition guarantees that the action of a transformation $g \in G$ on the input is preserved after applying F, up to the mapped transformation in the output space. The non-expansivity condition ensures that the operator does not increase the distance between functions, that is, inputs that are close in Φ remain close in Ψ.

Interpretation. GENEOs can be seen as structured observers that map input measurements into higher-level representations while preserving the symmetries encoded in the data. In the context of machine learning, such operators enable the design of modules that are inherently symmetry-aware, interpretable, and robust by design.

Approximation of d_G via GENEOs. An important theoretical result in the GENEO framework is the approximation of the natural pseudo-distance d_G using GENEOs.

Proposition 2 (Bergomi et al. [8])**.** *Let $F : \Phi \to \Psi$ be a GENEO associated with a group homomorphism $T : G \to H$. Then F is a contraction with respect to the natural pseudo-distances d_G and d_H, that is,*

$$d_H(F(\varphi_1), F(\varphi_2)) \leq d_G(\varphi_1, \varphi_2), \qquad \forall \varphi_1, \varphi_2 \in \Phi. \tag{2.17}$$

In this sense, GENEOs offer a tractable means of approximating the symmetry-aware geometry of the function space, even when the natural pseudo-distance d_G is computationally intractable.

2.3.6 The Space of GENEOs

The set of all GENEOs between two perception pairs (Φ, G) and (Ψ, H), denoted $\mathcal{F}_{\Phi,\Psi}^{G,H}$, inherits a topological structure that plays a fundamental role in the applicability of GENEOs in learning contexts. In particular, under mild assumptions on the underlying function spaces, this set is compact, convex, and closed under convex combinations. These properties enable efficient sampling, interpolation, and even optimization over $\mathcal{F}_{\Phi,\Psi}^{G,H}$.

Topology of the space. Let $\mathcal{F}_{\Phi,\Psi}^{G,H}$ be endowed with the topology induced by the sup-norm distance D_Ψ:

$$D_{\mathcal{F}}(F_1, F_2) := \sup_{\varphi \in \Phi} D_\Psi(F_1(\varphi), F_2(\varphi)), \qquad F_1, F_2 \in \mathcal{F}_{\Phi,\Psi}^{G,H}. \tag{2.18}$$

This metric captures how differently two GENEOs act on all admissible measurements $\varphi \in \Phi$. When Φ and Ψ are both compact subsets of $(\mathbb{R}_b^X, d_\infty)$ and $(\mathbb{R}_b^Y, d_\infty)$ respectively, it follows that the space $\mathcal{F}_{\Phi,\Psi}^{G,H}$ is also compact in the topology induced by $D_{\mathcal{F}}$ [8, 27]. Compactness ensures that any sequence of GENEOs has a convergent subsequence in $\mathcal{F}_{\Phi,\Psi}^{G,H}$. Indeed, for any $\varepsilon > 0$, there exists a finite set of GENEOs $\{F_1, \ldots, F_n\} \subset \mathcal{F}_{\Phi,\Psi}^{G,H}$ such that:

$$D_{\mathcal{F}}(F_i, F_j) < \varepsilon, \quad \forall i, j \in \{1, \ldots, n\}. \tag{2.19}$$

Convexity of the GENEO space. Another key property is that $\mathcal{F}_{\Phi,\Psi}^{G,H}$ is convex whenever the output space Ψ is convex. Let $F_1, \ldots, F_n \in \mathcal{F}_{\Phi,\Psi}^{G,H}$ be a finite collection of GENEOs, and let $(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$ be coefficients satisfying:

$$\sum_{i=1}^{n} \lambda_i = 1, \quad \lambda_i \geq 0 \quad \forall i = 1, \ldots, n. \tag{2.20}$$

Then, the convex combination operator $F : \Phi \to \Psi$ defined as:

$$F(\varphi)(x) := \sum_{i=1}^{n} \lambda_i F_i(\varphi)(x), \quad \forall \varphi \in \Phi,\ x \in X, \tag{2.21}$$

is itself a GENEO, i.e., $F \in \mathcal{F}_{\Phi,\Psi}^{G,H}$. This is in consequence of the fact that both equivariance and non-expansivity are preserved under convex interpolation (Given Ψ is closed under such operations).

Implications for Learning. The compactness and convexity of the GENEO space make it particularly well-suited for machine learning applications. From a theoretical perspective, compactness guarantees that the space is totally bounded and that any function in it can be approximated arbitrarily well by a finite collection of operators. This allows for discrete approximations of continuous operator families.

From an optimization standpoint, convexity enables gradient-based or manifold-based methods to explore the GENEO space efficiently. Recent work [13] further shows that $\mathcal{F}_{\Phi,\Psi}^{G,H}$ can be endowed with a Riemannian structure, enabling the use of geodesic descent algorithms over smooth manifolds of equivariant, non-expansive operators. Thus, the GENEO framework provides a rich and flexible space for designing and learning structured operators that respect the symmetries of the data.

2.3.7 Convolutional Operators as GENEOs

Once a space of GENEOs $\mathcal{F}_{\Phi,\Psi}^{G,H} = \mathrm{GENEO}((\Phi, G), (\Psi, H), T)$ is established, one may construct parametric families of such operators by introducing a parameter space $\Theta \subseteq \mathbb{R}^p$. Each element $\theta \in \Theta$ defines a specific operator $F_\theta \in \mathcal{F}_{\Phi,\Psi}^{G,H}$, and the resulting family:

$$\mathcal{F}_\Theta = \{F_\theta \colon \theta \in \Theta\} \subseteq \mathcal{F}_{\Phi,\Psi}^{G,H} \tag{2.22}$$

inherits the equivariance and non-expansivity properties of the GENEO framework by construction. This formulation opens the door to designing neural architectures as modular compositions of GENEOs, where each layer or component is structured according to known transformation symmetries. An especially instructive case is the interpretation of standard convolutional layers in CNNs as instances of GENEOs.

Example: Convolution as a GENEO. Let the observation space be $X = \mathbb{R}^2$, and consider the space of admissible measurements $\Phi \subseteq \mathbb{R}_b^X$, consisting of bounded grayscale images.

Let $G = \mathbb{R}^2$ denote the group of planar translations acting on X via $T_\delta(x) = x + \delta$ for $\delta \in \mathbb{R}^2$. The group acts on Φ by precomposition: for $\varphi \in \Phi$ and $T_\delta \in G$, we define $(\varphi \circ T_\delta)(x) := \varphi(x + \delta)$.

Now, we fix a continuous, bounded kernel function $k : \mathbb{R}^2 \to \mathbb{R}$ satisfying $\int_{\mathbb{R}^2} |k(y)|\, dy < \infty$. The corresponding convolutional operator $F_\theta : \Phi \to \Phi$ is defined as:

$$F_\theta(\varphi)(x) := \frac{1}{Z_\theta} \int_{\mathbb{R}^2} \varphi(y) \cdot k_\theta(x - y)\, dy, \tag{2.23}$$

where Z_θ is a constant ensuring normalization.

We now verify the two GENEO properties.

Equivariance. Let $\varphi \in \Phi$ and $T_\delta \in G$. Then:

$$\begin{aligned}
F_\theta(\varphi \circ T_\delta)(x) &= \int \varphi(y + \delta) \cdot \frac{k_\theta(x - y)}{Z_\theta}\, dy && (2.24)\\
&= \int \varphi(z) \cdot \frac{k_\theta(x - (z - \delta))}{Z_\theta}\, dz \quad \text{(change of variables } z = y + \delta) && (2.25)\\
&= \int \varphi(z) \cdot \frac{k_\theta((x + \delta) - z)}{Z_\theta}\, dz = F_\theta(\varphi)(x + \delta) && (2.26)\\
&= \left[F_\theta(\varphi) \circ T_\delta\right](x), && (2.27)
\end{aligned}$$

proving that $F_\theta(\varphi \circ T_\delta) = F_\theta(\varphi) \circ T_\delta$, i.e., F_θ is equivariant under translations.

Non-expansivity. Let $\varphi_1, \varphi_2 \in \Phi$, and consider the pointwise difference at any $x \in X$:

$$\begin{aligned}
|F_\theta(\varphi_1)(x) - F_\theta(\varphi_2)(x)| &= \left| \int \big(\varphi_1(y) - \varphi_2(y)\big) \cdot \frac{k_\theta(x - y)}{Z_\theta}\, dy \right| && (2.28)\\
&\le \int \big|\varphi_1(y) - \varphi_2(y)\big| \cdot \frac{|k_\theta(x - y)|}{Z_\theta}\, dy && (2.29)\\
&\le \|\varphi_1 - \varphi_2\|_\infty \cdot \int \frac{|k_\theta(x - y)|}{Z_\theta}\, dy && (2.30)\\
&= \|\varphi_1 - \varphi_2\|_\infty, && (2.31)
\end{aligned}$$

since the normalized kernel integrates to one. Thus, we obtain:

$$\|F_\theta(\varphi_1) - F_\theta(\varphi_2)\|_\infty \le \|\varphi_1 - \varphi_2\|_\infty,$$

which confirms that F_θ is a non-expansive operator.

Hence, $F_\theta \in \mathrm{GENEO}((\Phi, G), (\Phi, G), \mathrm{Id}_G)$ is a translation-equivariant, non-expansive operator. This provides a formal justification for the robustness of convolutional neural networks under shifts, a foundational inductive bias in computer vision. More generally, the GENEO framework generalizes this example and extends it to other transformation groups (e.g., rotations, scalings) that are not available in standard CNNs, allowing for the design of equivariant operators that respect the particular symmetries of a given dataset.

2.3.8 Networks of GENEOs

The GENEO framework naturally supports the construction of deep architectures by composing operators that act on function spaces. In this context, a network can be seen as a chain of GENEO layers, each composed of multiple parametric operators, which may attend to different transformation groups and encode complementary geometric biases.

Let Φ and Ψ be function spaces defined over a shared topological domain, and consider a collection of M parametric GENEO families:

$$\{\mathcal{F}_{\Theta^{(i)}}^{G^{(i)},H^{(i)}}\}_{i=1}^{M}, \quad \text{with} \quad \mathcal{F}_{\Theta^{(i)}}^{G^{(i)},H^{(i)}} := \{F_\theta^{(i)} : \theta \in \Theta^{(i)}\} \subseteq \mathcal{F}_{\Phi,\Psi}^{G^{(i)},H^{(i)}},$$

where each family defines a set of GENEOs with respect to its own group action $G^{(i)} \curvearrowright \Phi$ and corresponding homomorphism $T^{(i)} : G^{(i)} \to H^{(i)}$. Despite potentially attending to different symmetry groups, these operators act on the same input space Φ, enabling them to encode diverse geometric priors while observing the same data.

Each GENEO layer then applies a fixed number of operator instances, such as one from each parametric family. These are combined via a convex combination to form what we call a *complex observer*:

$$\mathcal{H} := \sum_{i=1}^{M} \lambda_i F_{\theta_i}^{(i)}, \quad \text{where } \sum_{i=1}^{M} \lambda_i = 1, \quad \lambda_i \geq 0.$$

Here, each weight λ_i quantifies the importance of the corresponding GENEO instance $F_{\theta_i}^{(i)}$ in constructing the overall observer. This design has both functional and interpretative advantages: it allows for us to build operators that can adaptively respect different transformation groups, while also providing a clear interpretation of the role of each operator in the context of the overall observation.

Once defined, such complex observers can be composed sequentially across layers. Let $\Phi^{(1)}, \Phi^{(2)}, \ldots, \Phi^{(L)}$ be admissible spaces across L layers, then each GENEO layer maps:

$$\mathcal{H}^{(l)} : \Phi^{(l)} \to \Phi^{(l+1)},$$

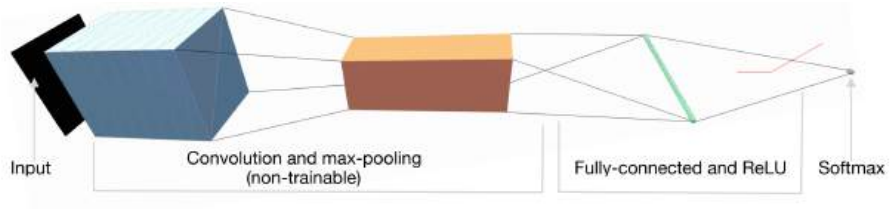
where $\mathcal{H}^{(l)}$ is a complex observer composed from families acting on $\Phi^{(l)}$. The full network is given by:

$$\mathcal{N} := \mathcal{H}^{(L)} \circ \cdots \circ \mathcal{H}^{(1)} : \Phi^{(1)} \to \Phi^{(L+1)}.$$

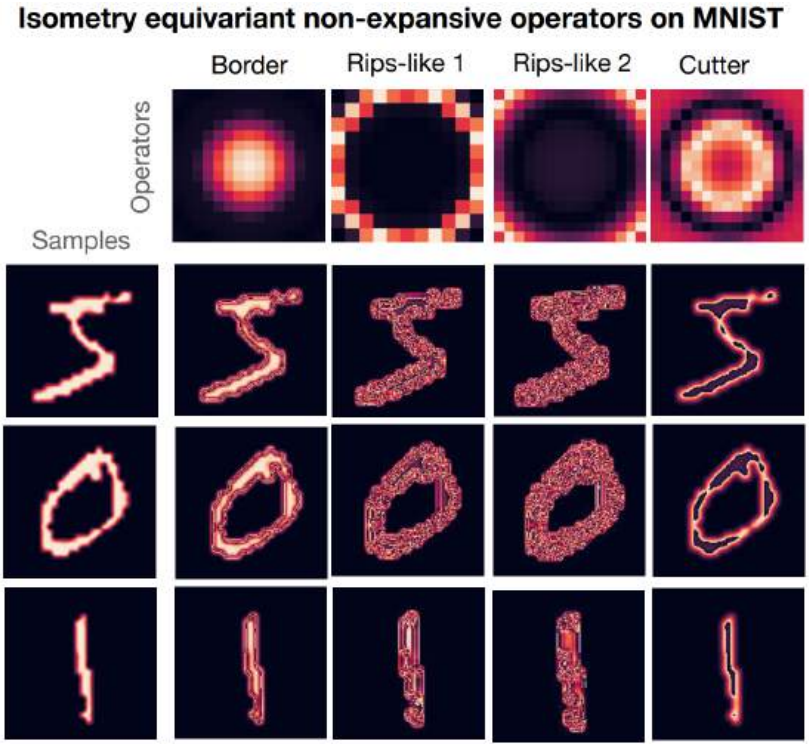
This modular construction enables networks to enjoy multiple symmetries, exploit interpretable convex coefficients λ and shape parameters θ at each layer, and maintain provable stability through non-expansivity.

2.3.9 Applications

This section presents concrete applications of the GENEO framework in two distinct contexts: classical image classification and geometric biology. The former assesses the utility of GENEOs when embedded into convolutional networks for learning from images, while the latter focuses on pocket detection in 3D protein structures, showcasing how domain knowledge can be encoded through group equivariant non-expansive operators.



Network architecture



Selected IENEOs

Figure 2.6: Overview of the GENEO-CNN classification pipeline proposed in [8]. The first image (top) illustrates the CNN architecture where the initial convolution and pooling layers are fixed and composed of selected Isometry Equivariant Non-Expansive Operators (IENEOs), while only the final fully connected layers are learned. This structure ensures equivariance to planar isometries, allowing the network to generalize better across transformed inputs. The second image (bottom) shows examples of IENEOs selected for MNIST digit classification. These operators were sampled from a parametric family based on their ability to discriminate instances within the same class, as determined by pseudo-distances over the operator space. Despite constraints imposed by equivariance, the selected IENEOs exhibit meaningful topological behavior, such as edge detection, void filling, and structural cutting.

2.3.9.1 Image Classification with GENEOs

The foundational works [8, 13] introduce the GENEO formalism as a robust alternative to conventional machine learning methods by leveraging topological data analysis. In particular, they show that convolutional layers in neural networks can be reinterpreted

as structured sets of GENEOs, providing built-in guarantees of equivariance and non-expansivity. To illustrate this, Bergomi et al. [8] implemented convolutional neural networks (CNNs) in which the trainable convolution kernels were replaced by fixed GENEO operators selected from a parametric family of isometry equivariant and non-expansive operators. The goal was to assess whether the introduction of structured priors, rather than learned filters, could still lead to competitive or even improved performance in standard visual classification tasks.

Experimental Setup. The experiments targeted MNIST, Fashion-MNIST, and CIFAR-10, three widely adopted benchmarks in image recognition. In each setting, two models were trained: (1) A baseline CNN with standard convolutional kernels initialized randomly and learned via backpropagation. (2) A GENEO-CNN using convolutional layers composed of GENEOs sampled from a parametric family (e.g., Gaussians mixtures), which were not updated during training (illustrated in Figure 2.6).

The key property of the GENEOs used in this context is equivariance with respect to isometries, specifically planar rotations and reflections. Data augmentation was performed using random translations, reflections, and rotations to empirically evaluate the impact of equivariance on generalization.

Operator Selection and Sampling. The construction of the GENEO-layer required careful sampling from a parametric family of operators. The authors employed a two-step strategy: (i) selecting operators that best discriminate between classes under a pseudo-distance d_{match}, and (ii) eliminating redundant operators based on a threshold in function space Δ_{GENEO} to ensure diversity in the learned representations. This sampling strategy is crucial for the performance of the GENEO-CNN. It acts as a form of structured prior injection, ensuring that the set of GENEOs forms a maximally diverse and label-relevant basis for the first layer of the network.

Results and Analysis. Empirical results showed that GENEO-based models consistently outperformed or matched their learned counterparts across all datasets. In particular, the GENEO-CNN achieved: higher accuracy and lower validation loss, especially in data regimes with limited samples; increased robustness to transformed inputs, attributed to the built-in equivariance; and a stable convergence due to the fixed nature of the GENEO kernels.

This study suggests that the explicit encoding of geometric priors into neural architectures via GENEOs offers multiple advantages. By enforcing symmetries through operator design, GENEO-based models can reduce the reliance on heavy data augmentation, extensive regularization, and large-scale parameter optimization. Moreover, the properties of GENEOs, namely equivariance and non-expansivity, lead to more interpretable models that can generalize better from fewer training examples. Crucially, the sharp reduction in

the number of trainable parameters translates to a significantly lower sample complexity, enabling these models to generalize well even in data-scarce regimes.

Limitations. Nevertheless, the GENEO framework presents limitations that merit attention. The construction of suitable parametric families and the manual selection or sampling of GENEOs often relies on domain knowledge, which may not always be readily available. Moreover, while fixed GENEOs offer strong inductive biases, they may constrain expressiveness if the assumptions embedded in their design misalign with the true distribution of the data.

2.3.9.2 Protein Pocket Detection with GENEOs

Equivariant operator design is particularly beneficial in molecular modeling tasks, where 3D spatial structure and symmetry are essential to understanding biological function. In this context, Bocchi et al. [10] introduce GENEOnet, a learning architecture grounded in the GENEO framework tailored to the task of protein pocket detection, which is a critical step in structure-based drug discovery.

Given a protein's 3D atomic configuration, the goal is to identify ligand-binding pockets, i.e., regions that exhibit suitable geometric cavities and physicochemical affinity. This task is naturally equivariant to the group of rigid-body transformations, $\mathrm{Isom}(\mathbb{R}^3)$: rotating or translating a protein should induce an equivalent transformation on its predicted binding pockets. To accommodate this, GENEOnet defines a set of $n = 8$ admissible scalar functions $\varphi_i : \mathbb{R}^3 \to \mathbb{R}$, each representing a molecular property (e.g., atomic occupancy, electrostatic potential, hydrophobicity). These measurements are evaluated over a voxelized embedding of the protein structure. Each function is then processed by a GENEO F_i selected from a parametric family $\mathcal{F}_\Theta$ based on convolutional kernels parameterized by $\theta_i \in \Theta$:

$$\psi_i := F_i(\varphi_i), \qquad i = 1, \ldots, 8. \tag{2.32}$$

The resulting observations $\{\psi_i\}_{i=1}^8$ are combined into a global pocket prediction ψ using a convex combination:

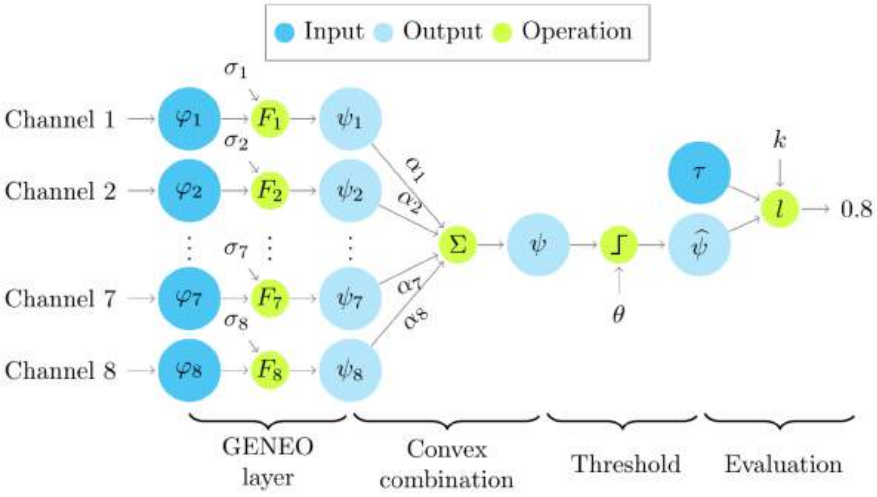
$$\psi = \sum_{i=1}^{8} \lambda_i \psi_i, \qquad \text{with } \lambda_i \in [0,1],\ \sum_{i=1}^{8} \lambda_i = 1. \tag{2.33}$$

A global threshold $\tau \in \mathbb{R}$ is applied to obtain a binary prediction map:

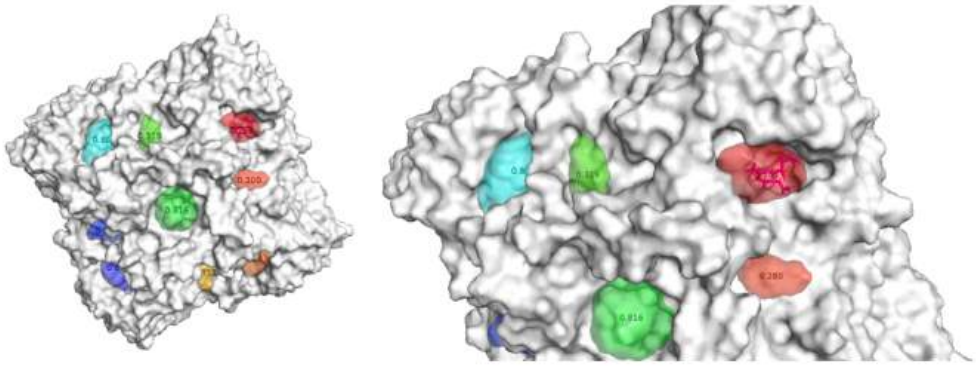
$$\hat{\psi}(x) := \mathbf{1}\{\psi(x) \geq \tau\}, \qquad x \in \mathbb{R}^3, \tag{2.34}$$

which marks spatial regions likely to constitute valid binding pockets.

Figure 2.7 illustrates the complete workflow of the model (top), along with an example of pocket prediction over a protein structure from the PDBbind dataset (bottom). Unlike earlier GENEO-based models such as GENEO-CNN [8], which rely on a curated selection



GENEOnet Architecture



Pocket Detection Example

Figure 2.7: Overview of GENEOnet [10], a GENEO-based model for protein pocket detection. The architecture (top) processes eight input channels, each encoding a structural or physicochemical property of the protein, through a corresponding GENEO F_i with learnable shape parameter σ_i. Each operator produces an output $\psi_i = F_i(\varphi_i)$, capturing task-relevant features. These outputs are combined into a unified representation $\psi = \sum_i \alpha_i \psi_i$ using a convex combination with learned weights α_i reflecting the importance of each channel. A global threshold θ is then applied to produce a binary classification map $\hat{\psi}$, indicating the predicted binding pockets. The prediction example (bottom) visualizes the output of the model for a representative protein (PDB ID: 2QWE), showing distinct pocket candidates rendered in color and ranked by their predicted binding scores. Each region reflects the likelihood of serving as an optimal ligand-binding site, supporting tasks such as drug design and protein-ligand interaction analysis.

of operators from a precomputed space, GENEOnet learns its GENEOs end-to-end through backpropagation. The total number of parameters is remarkably small, just 17 in total (8 kernel shape parameters θ_i, 8 convex weights λ_i, and 1 threshold τ), and yet the model achieves high predictive power with strong geometric priors and interpretability.

This work also marks the first application of GENEOs to 3D data. To enable convolutional operators over irregular protein structures, the authors employ a voxelization

strategy that embeds the input into a regular grid. This allows the use of differentiable 3D convolutions as a basis for defining GENEO families while preserving the necessary topological and symmetry constraints.

Empirical evaluation was conducted on a curated subset of the PDBbind dataset, containing over 12,000 protein-ligand complexes with annotated binding regions. GENEOnet achieves state-of-the-art performance, obtaining a top-3 accuracy $T_3 = 0.941$, which measures the probability that the correct binding site is among the three most confident predictions. It compares favorably against established methods such as Fpocket, DeepPocket, and P2Rank, while using dramatically fewer parameters and training examples. In contrast to typical 3D convolutional networks, GENEOnet offers several advantages:

- **Interpretability**: Each GENEO corresponds to a domain-specific measurement, and the learned weight λ_i reflects its relative importance.
- **Low Number of Parameters**: The model is highly parameter-efficient, with only 17 trainable parameters, which is significantly lower than most deep learning models for similar tasks.
- **Sample efficiency**: The compactness of the GENEO space allows generalization from limited supervision, even in noisy or imbalanced datasets.

2.4 3D Scene Understanding for Power Grid Inspection

2.4.1 Motivation and Context

Power grid inspection has long relied on manual techniques such as ground patrols and helicopter-based surveys. These traditional practices, while effective in specific operational settings, are increasingly inadequate given the rising complexity and geographic expansion of modern transmission networks. Expanding infrastructure requires more frequent and thorough inspection cycles, and the associated costs, time demands, and safety risks to personnel have become significant concerns [30]. The challenge is further compounded by the fact that power lines traverse diverse terrains, including rural and remote areas, where access can be difficult and hazardous. Manual inspections are labor-intensive and often fail to capture the full scope of potential issues, such as structural degradation, vegetation encroachment, or environmental damage. These vulnerabilities can lead to catastrophic failures, including large-scale outages or even wildfires caused by downed lines [75]. Climate change and extreme weather events further exacerbate these vulnerabilities. Studies have demonstrated that hurricanes, wildfires, and other hazards can severely disrupt power transmission and distribution systems [75, 118]. Preventing large-scale outages and ensuring resilience depend critically on the early identification of structural degradation, vegetation encroachment, and other risk factors along thousands of kilometers of lines.

To alleviate the burden on manual inspection teams, utilities have turned to unmanned aerial vehicles (UAVs) as a more scalable alternative [4, 98]. Equipped with high-resolution cameras and onboard sensors, drones can survey infrastructure from various viewpoints without requiring human operators to physically access hazardous environments. This shift significantly improves inspection coverage and operator safety. Nonetheless, it introduces new technical demands, such as autonomous navigation, multi-sensor fusion, and the post-processing of large volumes of data.

2.4.2 Manual and Automated Inspection Practices

Historically, inspection routines have involved technicians walking or driving along transmission routes to visually assess towers, insulators, and conductors. For hard-to-reach locations, helicopter flights are used to capture imagery and thermal data. Although these methods offer fine-grained control and human judgment, they suffer from a slow throughput and environmental limitations [99].

In recent years, UAV-based inspection has emerged as a powerful supplement to conventional methods. Platforms such as the LineScout system have demonstrated success in navigating high-voltage corridors while acquiring detailed imagery of conductors and components [99]. Improvements in GPS, inertial navigation, and visual odometry have enhanced drone positioning accuracy, allowing operators to collect data even in remote and GPS-challenged environments [4].

Beyond collection, the integration of machine learning has enabled automatic processing of visual data. Deep learning models, particularly convolutional neural networks, have been applied to classify defects, detect corrosion, or segment vegetation in aerial images [46]. However, these solutions remain predominantly image-based and are constrained by the inherent ambiguity of 2D projections, limited depth perception, and occlusions.

2.4.3 Domain-Specific Challenges

Deploying machine learning systems in the context of power grid inspection involves challenges that surpass typical academic or industrial benchmarks. First, the consequences of model errors are critical: false negatives can result in missed faults or fire risks, while false positives lead to unnecessary field interventions and maintenance dispatches.

Power grid scenes are characterized by a wide variety of conditions, including changing terrain, dense vegetation, atmospheric noise, and non-standard tower geometries. These factors make generalization difficult and demand models that are robust to occlusion, noise, and structural diversity. Moreover, infrastructure classes such as power lines and insulators occupy a small portion of the data, creating severe class imbalance. Most of the scanned area consists of background elements like vegetation and ground, which can easily dominate learning algorithms if not properly addressed. Another challenge is the need for high-resolution 3D data. While 2D images can provide useful information, they lack depth and spatial context, making it difficult to accurately assess the condition of

structures or vegetation. This is particularly important for tasks such as detecting vegetation encroachment, where the spatial relationship between power lines and surrounding foliage is crucial. Traditional 2D methods often struggle to capture this context [45, 70]. In contrast, LiDAR sensors mounted on UAVs can capture detailed 3D point clouds, but processing these datasets requires intensive labor efforts to annotate the data and develop specialized algorithms.

Finally, the operationalization of machine learning models in this domain requires compatibility with existing Geographic Information Systems (GIS) and asset management platforms. Models must support human-in-the-loop validation to ensure reliability and trustworthiness. This may impose constraints on model architecture, output resolution, and uncertainty estimation, as well as need for explainability features to facilitate adoption by utility operators.

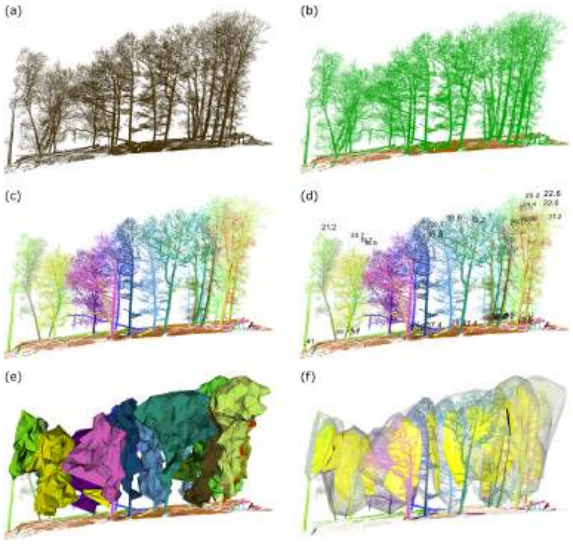
2.4.4 Existing Datasets and Their Limitations

While several datasets have driven progress in 3D scene understanding, most are tailored to urban driving, indoor navigation, or synthetic object classification. SemanticKITTI [6] and nuScenes [12] provide valuable benchmarks for autonomous driving but feature vehicle-mounted LiDAR from road-level perspectives. Datasets like ScanNet [23] and S3DIS [3] target indoor reconstruction and object segmentation. ShapeNet [14] and PartNet [74] support synthetic 3D shape analysis.

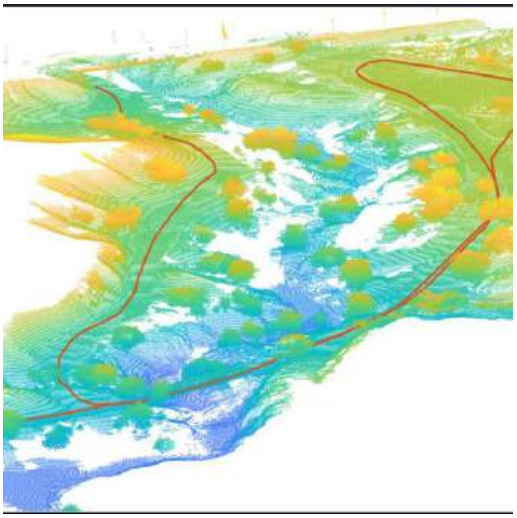
For vegetation and forestry, Forest3D [101] and NEON [71] offer airborne LiDAR data but do not contain infrastructure components. GTASynth [22] provides synthetic terrain with simulated LiDAR, yet lacks the fidelity and variability of real-world rural power lines. DALES [103] includes utility towers but only in urban settings and focuses on large high-voltage structures, which are typically isolated and easier to detect.

These datasets fail to capture the intricate challenges of rural utility inspection, where mid- and low-voltage towers are embedded in cluttered environments and are often occluded or surrounded by dense vegetation. There is a pressing need for annotated, high-resolution 3D datasets specifically curated for power grid inspection tasks.

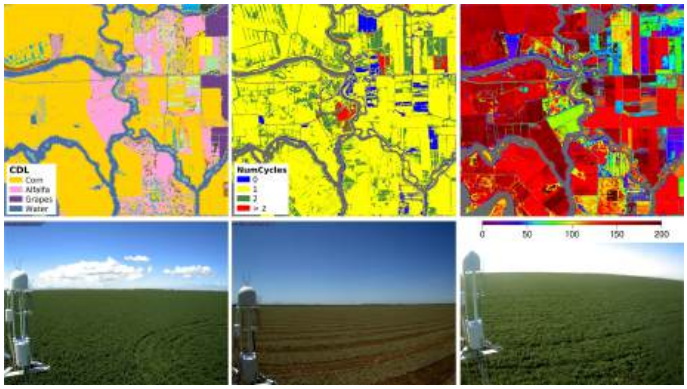
In this work, we present a benchmark dataset that addresses this gap. It consists of UAV-acquired point clouds covering realistic rural power line scenarios, including various tower types, terrain conditions, and vegetation profiles. The dataset is annotated with fine-grained semantic labels suitable for segmentation and detection tasks, providing a foundation for developing robust and interpretable 3D learning models for power grid inspection.



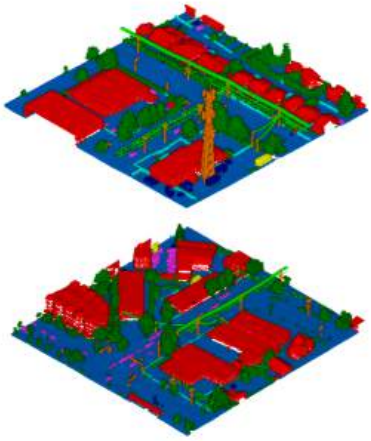
Forest3D



GTASynth



NEON



DALES

Figure 2.8: Visual comparison of 3D datasets frequently referenced in infrastructure scene understanding. **Top left:** Forest3D [101] provides high-resolution LiDAR over forest plots for biomass and canopy analysis. In the image: (a) TLS data imported into 3D Forest (Base cloud) before segmentation; (b) automatic octree-based segmentation into Terrain (brown) and Vegetation (green), refined manually; (c) individual Tree clouds shown in random colors; (d) DBH and tree height annotated for each tree; (e) tree crown concave hulls; (f) 3D convex hulls of crowns and their intersections (yellow). **Top right:** GTASynth [22] features synthetic LiDAR from simulated off-road driving environments, but exhibits limited variability and realism compared to true rural domains. The path shown in the image represents the trajectory of a simulated vehicle. **Bottom left:** NEON [71] offers rich ecological data from airborne sensors, with a focus on vegetation dynamics and no coverage of transmission systems. The image shows the evolution of the same terrain over the span of a year. **Bottom right:** DALES [103]-style datasets include large-scale utility towers in urban areas, where structures are isolated and relatively easy to detect.

3

TS40K: A Benchmark Dataset on Rural Power Grid Infrastructure

This chapter introduces **TS40K**, a novel benchmark dataset designed to support machine learning research in rural power grid inspection. The dataset comprises high-resolution 3D LiDAR scans acquired from Unmanned Aerial Vehicle (UAV) flights conducted over medium-voltage electrical infrastructure in rural, vegetation-rich environments. Unlike existing 3D benchmarks, which primarily focus on autonomous driving or indoor robotics, TS40K captures the unique challenges of infrastructure monitoring, including sparse structural elements, dense vegetation, and strong class imbalance. This chapter is based on work published at the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) [51]. We begin by describing the data acquisition protocol, including UAV hardware specifications. We then detail the annotation methodology and how the labels were tailored to support common inspection tasks in semantic segmentation and object detection. Following this, we present dataset statistics such as class distributions, offering an overview of the dataset's scope. Lastly, we establish baselines for 3D semantic segmentation and 3D object detection. We conclude by discussing the unique challenges posed by TS40K, including high-density noise artifacts, noisy labels, and structural diversity.

3.1 Introduction

Rural power grids are a vital component of electrical infrastructure, ensuring the distribution of energy across remote areas and supporting communities, agriculture, and critical services. Maintaining these systems requires regular inspection to prevent faults, ensure compliance, and mitigate the risk of wildfires. However, traditional inspection methods such as ground patrols and helicopter flyovers are labor-intensive, costly, and pose safety risks for field personnel. As electrical transmission systems expand across thousands of kilometers, these inspections become not only more necessary but also increasingly difficult to carry out at scale.

In recent years, UAV-based remote sensing has emerged as a promising alternative,

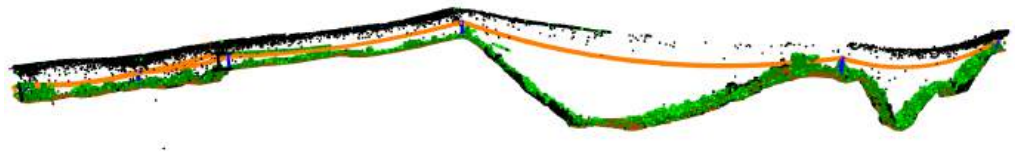
offering scalable and efficient data acquisition through high-resolution LiDAR. Despite this progress, the automation of downstream processing remains limited. Most of the acquired 3D data must still be manually inspected or annotated before use, delaying critical decisions and blocking the full potential of UAV-based inspection. Machine learning solutions have been proposed to automate visual inspection via classification and segmentation models, yet many of these approaches rely exclusively on Red, Green, Blue (RGB) images, which lack the spatial depth needed for accurate infrastructure analysis. 3D data offer a richer geometric perspective, which is particularly useful for identifying vegetation encroachment, structural deformation, or component failures. However, there remains a critical lack of publicly available, high-resolution 3D datasets tailored to rural power grid inspection. Existing datasets in rural areas or containing power grid elements such as DALES [103] and Forest3D [101] often focus on high-voltage transmission lines or vegetation detection. Neither of these datasets captures the full complexity of rural power grid inspection, which involves medium-voltage infrastructure, diverse terrain types, and varying vegetation densities.

To address this gap, we introduce **TS40K**, a novel LiDAR benchmark dataset designed specifically for rural power grid inspection, illustrated in Figure 3.1. Captured using UAV-mounted sensors over real-world transmission corridors, TS40K contains over 40,000 kilometers of densely annotated point clouds spanning various terrain types, vegetation densities, and infrastructures. Each scene is labeled with a task-driven class ontology suited for Machine Learning (ML)-based inspection workflows, including supporting towers, power-lines, vegetation, and terrain. The dataset presents numerous interesting challenges for learning systems: high-density noise artifacts, structural elements with high shape diversity and low point density, and realistic label noise introduced by adapting specific inspection labels to ML-driven labels.

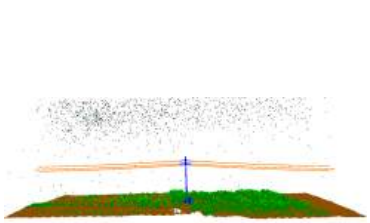
The main contributions of this chapter are:

- We introduce **TS40K**, the first large-scale, publicly available LiDAR dataset for rural power grid inspection, densely annotated with a domain-specific class ontology.
- We define benchmark tasks for 3D semantic segmentation and object detection, reflecting the operational needs of inspection workflows.
- We provide baseline experimental results to enable reproducible research and fair comparisons.
- We highlight the dataset's unique challenges, which include noisy annotations, structural diversity, and the presence of spurious points.

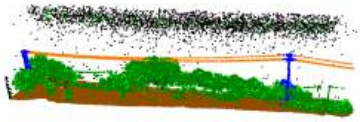
In essence, this work introduces a novel dataset and extends an invitation to the research community to explore, innovate, and address the unique challenges of rural power grid inspection in a more diverse and realistic setting.



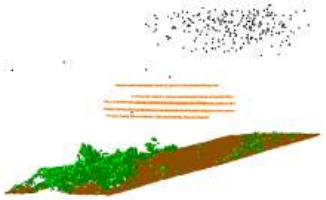
(a) Raw TS40K LiDAR scan showing a typical rural power grid corridor



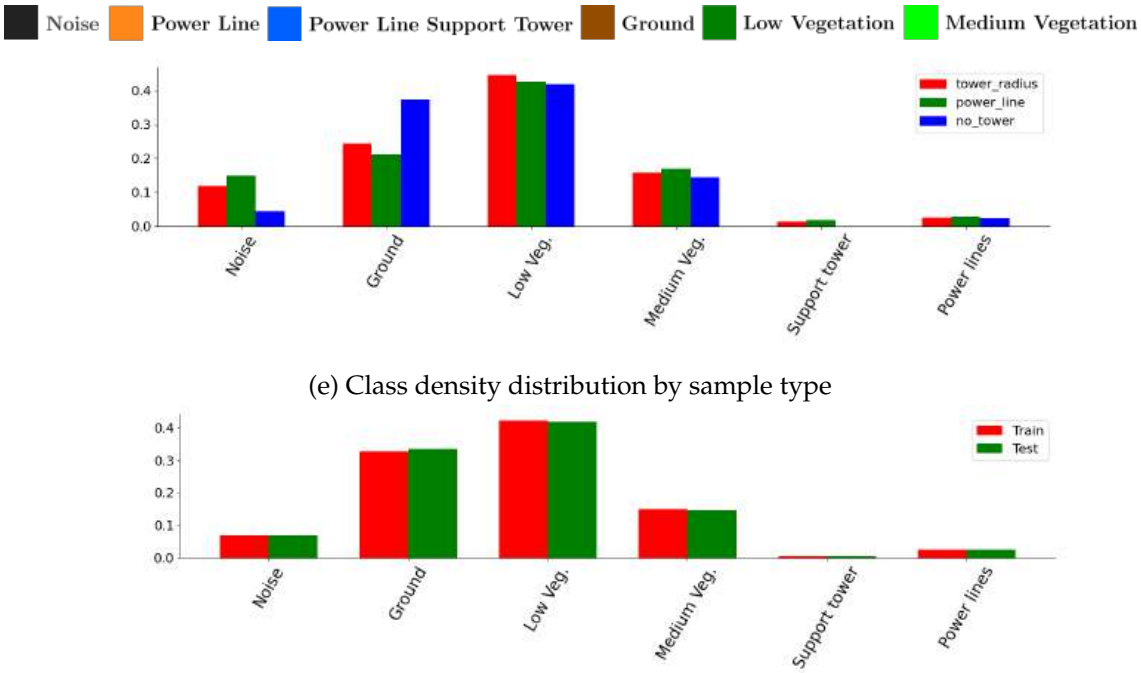
(b) Tower-radius sample type



(c) Power-line sample type



(d) No-tower sample type



(e) Class density distribution by sample type

(f) Overall train/test class distribution

Figure 3.1: **Overview of the TS40K dataset structure and sample types. (a)** Raw UAV-acquired LiDAR scan illustrating the typical linear corridor structure of rural power grid infrastructure in highly irregular terrain. The TS40K dataset is organized into three distinct sample types based on infrastructure content: **(b)** *Tower-radius samples* focus on transmission towers and their immediate surrounding environment, capturing critical structural details and nearby vegetation for clearance analysis. **(c)** *Power-line samples* emphasize the linear transmission elements between towers. **(d)** *No-tower samples* represent rural terrain segments along transmission corridors that exclude supporting structures but may contain power lines. **(e-f)** Class density analysis reveals the challenging distribution characteristics of TS40K: extreme class imbalance with terrain and vegetation dominating the point count, while infrastructure elements (towers and power lines) represent less than 2% of total points.

3.2 Data Acquisition Setup

TS40K was constructed using a UAV-based LiDAR acquisition pipeline tailored to the specific needs of rural electrical infrastructure inspection. Unlike most autonomous driving datasets that rely on vehicle-mounted sensors traversing urban grids, TS40K captures long, linear corridors that reflect the layout of transmission grids. This section details the technical configuration of the data collection platform, the strategy for selecting relevant scenes, and the types of objects represented in the dataset.

It is important to note that the data was originally acquired and is owned by the inspection company Labelec, which routinely performs these surveys as part of their operational activities. Our work leverages this existing data with the company's permission for research purposes.

3.2.1 UAV and Sensor Configuration

The data was collected using a multirotor UAV platform fitted with a high-resolution LiDAR sensor and a tightly integrated GNSS-IMU navigation system. Data acquisition campaigns were conducted across multiple countries in Europe, including Portugal, Spain, and Italy, where the company responsible for the inspections operates. This ensured geographic diversity and captured a wide range of rural infrastructure and environmental conditions. Flight missions were conducted using multirotor UAVs (namely, YellowScan Mapper+ system) optimized for long-endurance operations, typically flying at an altitude of approximately 100 meters and a speed of 5 to 20 meters per second. The system offers a horizontal swath of approximately 140 meters and a point density of up to 170 points per square meter, with a precision of 2.5 cm and an absolute accuracy of 3.0 cm. The LiDAR operates with a field of view (FOV) of 70.4°, firing 240,000 laser shots per second with up to three echoes per shot, generating up to 720,000 points per second. Captured data is synchronized using GNSS timestamps and post-processed using Applanix POSPac UAV software to refine geo-referencing and motion compensation. The resulting point clouds exhibit strong spatial coherence and high fidelity.

This UAV-LiDAR configuration strikes a balance between operational feasibility, specifically in terms of flight time, payload weight, regulatory compliance, and the vertical resolution required to inspect elevated infrastructure components from a safe distance. Overall, it allows for scalable acquisition of hundreds of meters of infrastructure in a single mission, enabling remote inspections of large rural networks.

3.2.2 Scene Selection

Each UAV mission captures several minutes of uninterrupted flight, yielding raw point clouds that may include a mixture of takeoff zones, surrounding terrain (such as rural areas with no relevant power grid elements), infrastructure segments, and other unrelated elements. In order to construct a clean, task-relevant benchmark, each mission is manually

reviewed to identify and extract inspection-worthy segments. Scenes that contain only vegetation, forest canopy, or unstructured background are discarded. The selected scenes are typically linear segments aligned with the real layout of the power grid infrastructure. Rather than generating square or tiled LiDAR frames, each raw segment corresponds to a continuous corridor that follows the transmission path. This representation is more consistent with how inspections are conducted in practice and ensures that structural relationships (e.g., the spacing between poles) are preserved across frames. These segments present various terrain types, such as flat, hilly or vegetated areas, and structural configurations, including single- and multi-pole structures with diverse shapes.

3.2.3 Main Actors of the Scene

The original labels provided by the utility operator were designed for operational inspection and maintenance purposes, often encoding application-specific categories (such as model keypoint). For the purposes of 3D scene understanding, we abstracted these into a semantically meaningful set of object classes that better align with the needs of segmentation and detection tasks. These higher-level actors reflect the main structural and environmental elements present across all scenes in the TS40K dataset:

- **Power line supporting towers**: includes wood, metal, and concrete variants with varying height and shapes.
- **Power lines**: are thin linear elements suspended between poles, often appearing as catenary curves in space. Their detection is critical for clearance assessment and vegetation encroachment analysis.
- **Vegetation**: includes trees, bushes, and other natural elements that may interfere with overhead lines. The vegetation class varies significantly in shape, density, and elevation, posing a challenge for both segmentation and detection.
- **Ground**: comprises soil, grass, and other surface elements that define the bottom layer of each scene. These points are numerous and dominant in terms of class frequency.

To characterize the geometry of key infrastructure elements, we conducted an exploratory analysis of the labeled point clouds. For power line supporting towers, we applied DBSCAN (Density-Based Spatial Clustering of Applications with Noise), a density-based clustering algorithm, to isolate instances using the provided annotations. Proper tuning of the neighborhood parameters (ϵ, n) yielded well-separated clusters, each corresponding to a single tower. Analysis of these clusters revealed that towers typically consist of 334 to 5,860 points. Their bases are usually square in shape, averaging 4.15 meters in width and 4.2 meters in length. Tower heights range from 18 to 40 meters, with a mean value of 34.74 meters. The spacing between towers varies from 45 to 500 meters, averaging

approximately 300 meters. All towers are connected to one or more power lines. A similar strategy was employed to analyze the structure of power lines. Each line segment, defined as the stretch between two adjacent towers, typically contains between three and six cables and spans 200 to 500 meters. Line segments are represented with low fidelity, containing between 500 and 900 points. The sag of the cables generally ranges from 1 to 3 meters. Vegetation classes vary widely in shape, density, and vertical extent, ranging from low vegetation to tree canopies. In some scenes, vegetation reaches or exceeds tower height, increasing the risk of line contact. Lastly, terrain points constitute the dominant class by volume and are highly irregular in shape, with varying elevations and relief.

3.2.4 Properties of the LiDAR Data

A defining characteristic of TS40K lies in its acquisition protocol: all scenes are captured from a Bird's Eye View (BEV). This top-down perspective results in point clouds that exhibit properties that differ substantially from conventional 3D datasets acquired from ground-based sensors, such as those mounted on vehicles for autonomous driving data. The BEV perspective introduces a set of data properties that are especially relevant for machine learning applications in 3D scene understanding:

- **No Occlusion**: In contrast to ground-based LiDAR datasets (such as Waymo [92] or ScanNet [23]), which frequently suffer from object occlusion, the BEV acquisition of TS40K ensures that all scene elements are unobstructed and heavily detailed (except for their interiors).
- **Homogeneous Object Density**: Objects exhibit spatially uniform density across their surface, meaning that the number of points per unit area remains consistent across the extent of an object. This consistency simplifies sampling, neighborhood construction, and feature extraction operations that are central to point-based neural networks. Although different classes have different absolute densities, intra-object density remains highly stable.
- **High Point Density**: Each scene contains a large number of points per square meter, which facilitates the application of voxelization schemes. In contrast, autonomous driving datasets [6, 23] are typically much sparser.

Although these properties make TS40K especially suitable for machine learning due to its rich representations, they also accentuate a core challenge: extreme class imbalance. The BEV setup inherently prioritizes the terrain, leading to a severe underrepresentation of infrastructure elements. Despite the careful selection and cropping of relevant scenes (detailed in Section 3.3), less than 2% of all points in TS40K belong to classes directly associated with the power grid (e.g., towers and lines). This imbalance introduces significant difficulty in both training and evaluation, as learning algorithms may struggle to extract discriminative features for rare but crucial classes.

3.3 Point Cloud Annotation and Sample Types

3.3.1 Annotation Workflow

All 3D annotations in TS40K were produced internally by Labelec's inspection team, with the primary goal of supporting maintenance workflows rather than downstream machine learning. The labeling process is driven by the needs of human inspectors and risk assessment protocols, not by dataset curation or model training considerations. To this end, a hybrid workflow is employed, combining rule-based heuristics with manual verification and correction. For example, ground points are often identified using geometric criteria and elevation-based filters, while finer structures like towers and vegetation are reviewed and annotated manually.

One notable policy is applied to supporting towers: rather than restricting labels strictly to the visible structure, all points within a small radius surrounding the base of each tower are labeled as part of the supporting infrastructure. This is a deliberate security precaution, ensuring that any object or obstruction in close proximity is accounted for in clearance and encroachment assessments. While this enhances inspection reliability, it also introduces label noise from power lines, terrain artifacts, or small objects that may not belong to the tower itself.

The full set of annotated classes and their distributions are detailed in Table 3.1.

3.3.2 Labels Designed for Inspection Tasks in Machine Learning

Although TS40K's annotations stem from practical inspection objectives, they translate effectively into a semantic structure suitable for training and evaluating ML models. In collaboration with domain experts, we consolidate the annotated labels into a set of six semantic classes that reflect the primary actors of the inspection task: *ground*, *low vegetation*, *medium vegetation*, *power line support tower*, *power line*, and *noise*. These categories serve as the foundation for both 3D semantic segmentation and object detection benchmarks. A complete mapping from original labels to semantic classes is provided in Table 3.2.

This mapping process simplifies the annotation space in a manner that supports ML training, while still preserving the practical semantics relevant to inspections. For example, distinguishing between low and medium vegetation supports models in estimating clearance risks, while identifying tower structures enables detection-based models to analyze supporting infrastructure. Noise points, which stem from sensor artifacts or incomplete labeling, are retained to reflect the natural variability and imperfection of real-world datasets.

3.3.3 Semantic Classes

In Table 3.2, we detail the mapping of the inspection annotations to the semantic classes used in TS40K. The table also indicates which original labels were excluded from the semantic task, such as model keypoints and low reliability annotations, which do not

Table 3.1: Annotated classes in the TS40K dataset and their distribution for power-grid inspection. Ground and road surfaces constitute the majority of the dataset (63%), whereas the power-grid only constitute 1.43% of the 3D points.

Label	Class	Density(%)	Label	Class	Density(%)
0	Created	0	11	Road surface	**44.752**
1	Unclassified	0.571	12	Overlap points	0.529
2	Ground	**23.403**	13	Medium Reliability	0
3	Low vegetation	**18.758**	14	Low Reliability	0
4	Medium vegetation	0.241	15	Power line support tower	**0.519**
5	Natural obstacle	1.069	16	Main power line	**0.907**
6	Human structures	0	17	Other power line	**0.002**
7	Low point	0.362	18	Fiber optic cable	0
8	Model key points	0	19	Not rated object to be consider	8.205
9	Water	0	20	Not rated object to be ignored	0
10	Rail	0.681	21	Incidents	0

Table 3.2: Mapping of annotated labels to semantic classes for power grid inspection. '—' indicates labels excluded from the semantic task.

Original Label	**Annotated Class**	**Semantic Class**	**Original Label**	**Annotated Class**	**Semantic Class**
0	Created	—	11	Road surface	Ground
1	Unclassified	Noise	12	Overlap points	Low Vegetation
2	Ground	Ground	13	Medium reliability	—
3	Low vegetation	Low Vegetation	14	Low reliability	—
4	Medium vegetation	Medium Vegetation	15	Power line support tower	Power line support tower
5	Natural obstacle	Medium Vegetation	16	Main power line	Power lines
6	Human structures	—	17	Other power line	Power lines
7	Low point (noise)	Noise	18	Fiber optic cable	—
8	Model keypoints (masspoints)	—	19	Not rated (object to be considered)	Noise
9	Water	—	20	Not rated (object to be ignored)	—
10	Rail	Ground	21	Incidents	—

Table 3.3: Distribution of semantic classes in the TS40K dataset.

Label	Semantic Class	Density (%)
0	Noise	1.348
1	Ground	55.281
2	Low Vegetation	35.520
3	Medium Vegetation	6.647
4	Power Line Support Tower	0.431
5	Power Line	0.771

contribute to the core inspection objectives. The resulting semantic class distribution, shown in Table 3.3, highlights the inherent imbalance present in UAV-acquired datasets. The vast majority of points belong to terrain or vegetation classes, with transmission system elements accounting for less than 2% of the total point cloud.

3.3.4 Sample Types

To safeguard the topology of the transmission system while preserving the utility of the data, we segment the dataset into three sample types, each reflecting distinct aspects of the power grid environment:

- **(1) Tower-radius**: Includes the environment around a power-line support tower, providing a comprehensive view of the surroundings relevant to the tower's location.
- **(2) Power-line**: Focuses on power lines as the main actors, featuring two towers at opposite sides. This sample type offers insights into the spatial relationships of power lines and their supporting structures.

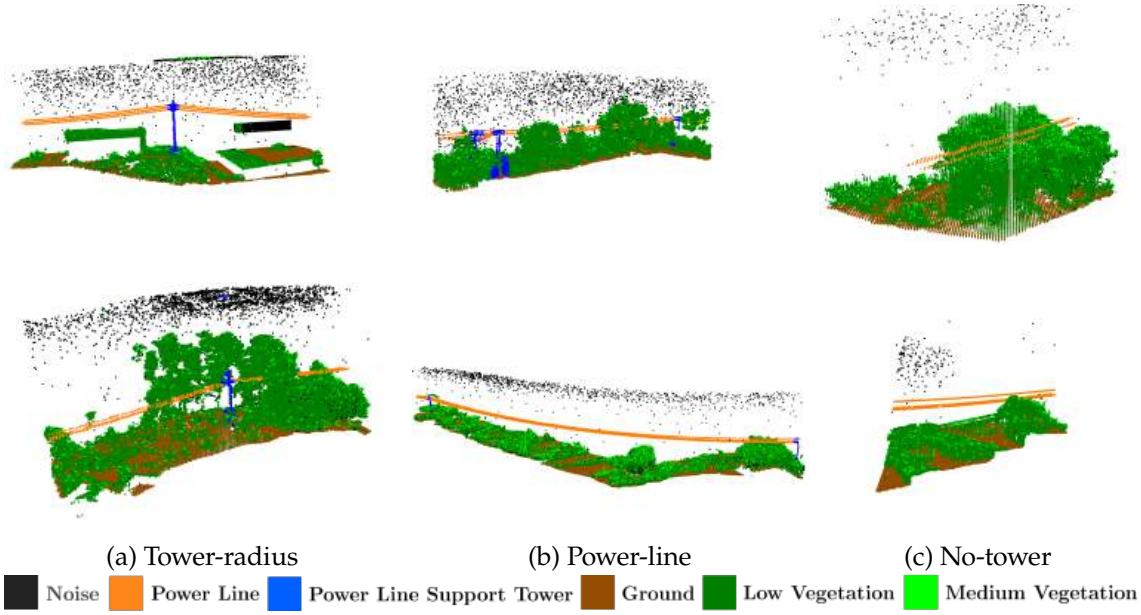


Figure 3.2: **TS40K sample types.** Each column shows two examples of a sample type described as follows: **(a) Tower-radius:** Focuses on the immediate vicinity of transmission towers, capturing detailed structural geometry and nearby vegetation, which is critical for clearance and encroachment analysis. **(b) Power-line:** Spans the region between two adjacent towers, emphasizing the linear transmission elements and their interaction with terrain and vegetation, supporting line sag and risk assessment. **(c) No-tower:** Represents corridor segments without visible towers, typically dominated by terrain and vegetation, serving as background or negative samples for detection tasks. The legend below maps semantic classes to colors, illustrating the extreme class imbalance and the spatial relationship between infrastructure and natural elements.

- **(3) No-tower**: Represents rural terrain without supporting towers but potentially includes power lines. This sample type provides context for areas where transmission infrastructure is absent.

These sample types are illustrated in Figure 3.2. On average, samples span 70, 100, and 90 meters, respectively, for tower-radius, power-line, and no-tower segments.

3.4 Benchmark Tasks

3.4.1 Data Preprocessing

To support reproducible machine learning experiments on TS40K, we define a benchmark split and establish a standardized preprocessing pipeline. The dataset consists of 24,355 point cloud samples derived from the three sample types: tower-radius (3663 samples), power-line (3590 samples), and no-tower (17,102 samples). For each type, 80% of the samples are allocated for training and validation, and the remaining 20% are held out for testing. The training-validation split is randomized at each training cycle, while the test set remains fixed. This ensures robust evaluation while allowing flexibility for model

development. To somewhat mitigate the severe class imbalance inherent to UAV-based inspection data, we apply point subsampling strategies. We compare three subsampling methods:

- **Farthest Point Sampling (FPS)** preserves geometric structure by ensuring that selected points are evenly spaced. It is particularly effective at retaining rare classes such as power lines and towers but incurs higher computational cost.
- **Inverse Density Importance Subsampling (IDISS)** samples points based on local point density. While it promotes inclusion of sparser regions, it often overemphasizes noise and may distort geometry.
- **Random Point Sampling (RPS)** offers computational simplicity but tends to eliminate underrepresented classes due to its unbiased nature, making it suboptimal for inspection-critical tasks.

As demonstrated in Figure 3.3, Farthest Point Sampling (FPS) strikes the best balance between class representation and spatial fidelity. This strategy is adopted in our semantic segmentation and object detection baselines. As a result, the power grid classes (towers and lines) increase from 1.4% in the raw data to approximately 2.9% after subsampling, improving training signal for minority classes.

3.4.2 3D Semantic Segmentation

The task of 3D semantic segmentation involves assigning a class label to each point in a scene:

$$\hat{y}_i = f_{\text{seg}}(\mathcal{P}) \quad \text{for } i = 1, \ldots, N, \tag{3.1}$$

where $\mathcal{P}$ is the input point cloud, and $\hat{y}_i$ is the predicted class for point p_i.

We evaluate segmentation performance using the standard **mean Intersection over Union (Mean Intersection over Union (mIoU))** metric:

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{3.2}$$

where TP_c, FP_c, and FN_c denote the true positives, false positives, and false negatives for class c, and C is the total number of classes.

3.4.3 3D Object Detection

3D object detection aims to localize and classify discrete structures within the point cloud by predicting oriented bounding boxes. This task is defined as:

$$B = f_{\text{det}}(\mathcal{P}), \tag{3.3}$$

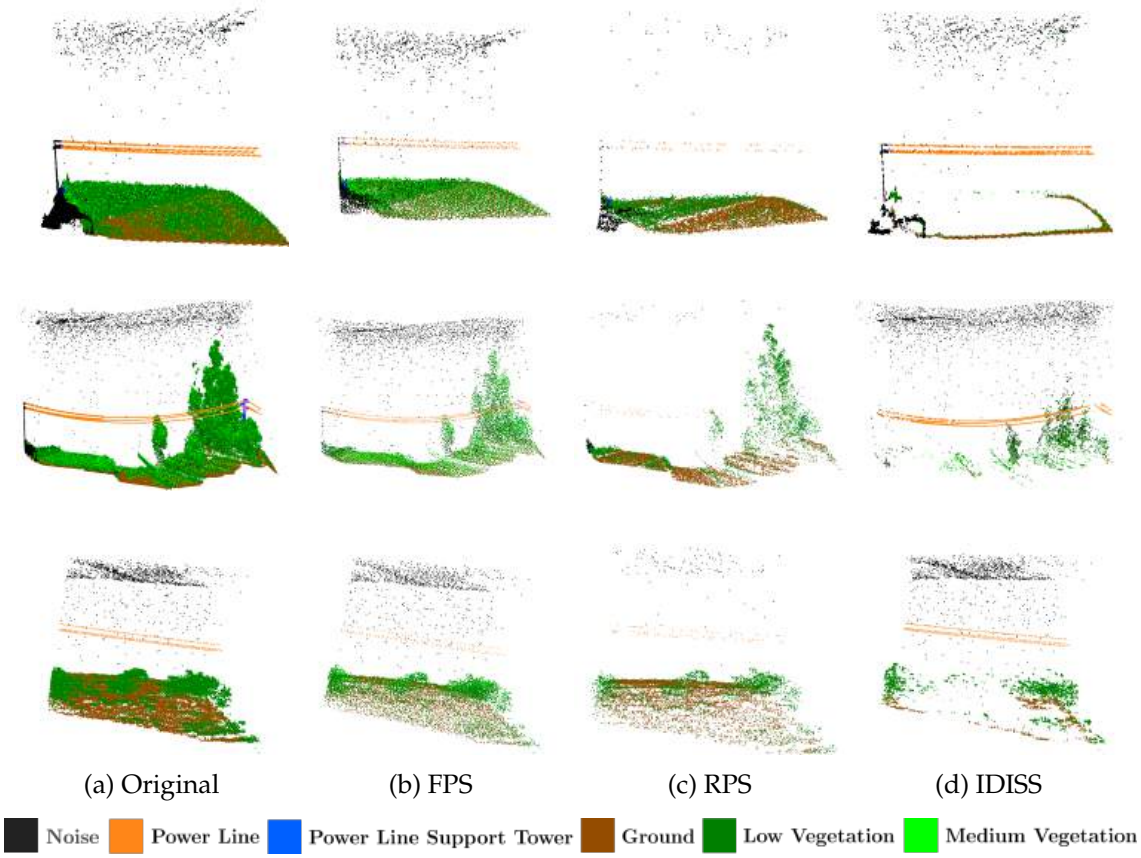


Figure 3.3: **Comparison of subsampling techniques across TS40K sample types.** Each row corresponds to a different sample type: **Top:** Tower-radius, **Middle:** Power-line, **Bottom:** No-tower. Each column shows the effect of a subsampling method: **(1) Original**, **(2) Farthest Point Sampling (FPS)**, **(3) Random Point Sampling (RPS)**, and **(4) Inverse Density Importance Subsampling (IDISS)**. FPS preserves geometric structure and improves minority class representation, RPS is efficient but may lose rare classes, and IDISS increases density for sparse regions but can distort geometry. The legend below maps semantic classes to colors.

where $B = \{B_1, \ldots, B_M\}$ is the set of predicted objects, and each 3D object B_i is represented as:

$$B_i = [x_c, y_c, z_c, l, w, h, \theta, \text{class}], \tag{3.4}$$

with (x_c, y_c, z_c) as the center, l, w, and h as the dimensions, θ as the orientation, and *class* as the object category.

We consider three target classes for detection: *power line support towers*, *power lines*, and *medium vegetation*. These are relevant for assessing clearance, sag detection, and vegetation encroachment. The primary metric is **mean Average Precision (mAP)**, computed as:

$$AP_c = \int_0^1 P_c(R)\, dR \quad \text{and} \quad mAP = \frac{1}{C} \sum_{c=1}^{C} AP_c, \tag{3.5}$$

where $P_c(R)$ is the precision at recall R for class c. We use 3D Intersection over Union (IoU) thresholds of 0.5 for towers and 0.25 for lines and vegetation, reflecting their different

Table 3.4: **3D semantic segmentation**: Benchmark results of baselines on the TS40K test set. Noise is ignored during training and evaluation. We report mean IoU (mIoU %) and per-class IoU (%) scores. Due to the extreme class imbalance, we showcase the results with both regular and weighted cross entropy.

Method	Loss Function	mIoU (%)	Ground	Low Vegetation	Medium Vegetation	Power Line Support Tower	Power Line
PointNet [78]	Cross Entropy	36.25	49.57	55.53	9.58	4.52	66.73
PointNet++ [79]		40.72	59.05	55.62	11.42	2.92	74.58
RandLaNet [44]		14.38	28.69	43.18	0.04	0	0
KPConv [96]		56.18	63.35	59.76	24.41	**40.62**	**92.75**
PTV1 [20, 121]		59.26	75.15	**66.02**	29.74	35.32	90.05
PTV2 [20, 110]		**62.27**	**77.73**	65.78	**49.45**	26.39	91.98
PointNet [78]	Weighted Cross Entropy	44.58	62.72	44.92	17.91	17.57	79.79
PointNet++ [79]		46.90	59.03	55.35	18.57	21.32	80.22
RandLaNet [44]		16.76	23.21	40.27	17.38	0.91	2.02
KPConv [96]		57.58	64.52	59.23	38.08	33.03	93.06
PTV1 [20, 121]		62.67	**77.34**	**67.90**	32.78	**43.80**	91.51
PTV2 [20, 110]		**65.58**	77.31	64.22	**48.94**	43.42	**93.99**

spatial extents.

3.5 Experimental Results

3.5.1 3D Semantic Segmentation Results

To assess the efficacy of TS40K for semantic segmentation, we benchmark several state-of-the-art 3D point cloud segmentation models: PointNet [78], PointNet++ [79], RandLA-Net [44], KPConv [96], and two variants of Point Transformer [110, 121]. Results are evaluated using mean Intersection over Union (mIoU) and per-class IoU across the six semantic classes defined in the dataset.

Performance Overview. As summarized in Table 3.4, Point Transformer V2 (PTV2) achieves the best overall performance with an mIoU of 65.58% using weighted cross-entropy. It consistently outperforms earlier methods, particularly in medium vegetation (48.94%) and power line (93.99%). KPConv also performs strongly, especially in power-grid-related classes, which may be attributed to its convolutional architecture capturing local geometric patterns.

In contrast, RandLA-Net struggles with underrepresented classes, performing poorly on both vegetation and power grid elements. Rand-LA-Net subsamples points in its encoding layers randomly, giving the model a fast inference time but leading to a significant loss of information, especially for rare classes. This is evident in its low mIoU of 16.76% with weighted loss, indicating that random sampling is not effective for imbalanced datasets like TS40K. This supports our earlier analysis in Figure 3.3.

Impact of Class Imbalance. Weighted loss functions significantly boost detection rates for rare classes. For example, PointNet++ improves tower IoU from 2.92% to 21.32% when switching from cross-entropy to weighted loss. This confirms the benefit of counterbalancing label skew during training, particularly for power grid components, which constitute less than 2% of the dataset.

Table 3.5: **Confusion matrix analysis of Point Transformer V2 performance on TS40K semantic segmentation.** Rows correspond to true labels and columns to model predictions. Noise is frequently mislabeled as support towers or power lines, likely due to proximity. Medium and low vegetation are also occasionally misclassified as towers. This real-world sensor noise provides an opportunity to evaluate denoising methods.

	Noise	Ground	Low Veg.	Med. Veg.	Tower	Power Line
Noise	–	9.05%	14.35%	20.95%	**11.14%**	**44.52%**
Ground	–	86.45%	6.53%	6.39%	0.55%	0.07%
Low Veg.	–	6.70%	73.20%	18.12%	1.77%	0.20%
Med. Veg.	–	1.89%	14.06%	81.70%	**1.92%**	0.52%
Tower	–	0.16%	0.63%	0.65%	97.84%	0.72%
Power Line	–	0.01%	0.19%	0.54%	1.80%	97.36%

Qualitative Analysis. Figure 3.4 illustrates PTV2's performance across different scenarios. The model demonstrates strong capability in detecting infrastructure elements, successfully identifying towers and associated power lines. Notably, PTV2 sometimes predicts tower structures that are absent in the ground truth annotations, potentially revealing under-labeling in the original dataset rather than model errors.

However, the qualitative results also expose some challenges: the model occasionally generates spurious ground or vegetation predictions near tower bases, likely caused by the annotation policies employed.

Deployment Assessment. While the quantitative results in Table 3.4 show promising performance, particularly for power line detection (>90% IoU), supporting towers remain below the deployment threshold imposed by our industry partner, Labelec, which requires at least 85% IoU for safety-critical elements, namely the power grid. No evaluated model achieves this for tower detection. The confusion matrix in Table 3.5 highlights that noise points are frequently misclassified as infrastructure elements, especially towers and power lines. This indicates that future methods must be more carefully designed to address spurious point artifacts, which are common in UAV-acquired datasets.

3.5.2 3D Object Detection Results

For object detection evaluation, we implement five representative 3D detection methods from the OpenPCDet framework [95]: SECOND [116], PointPillars [50], PointRCNN [85], Part-A^2 Net [86], and PV-RCNN [87]. These methods represent different architectural paradigms, from pure voxel-based approaches to hybrid point-voxel fusion strategies. We evaluate detection performance using mean Average Precision (mAP) computed over 11 recall points for three inspection-critical object classes: power line support towers, power lines, and medium vegetation.

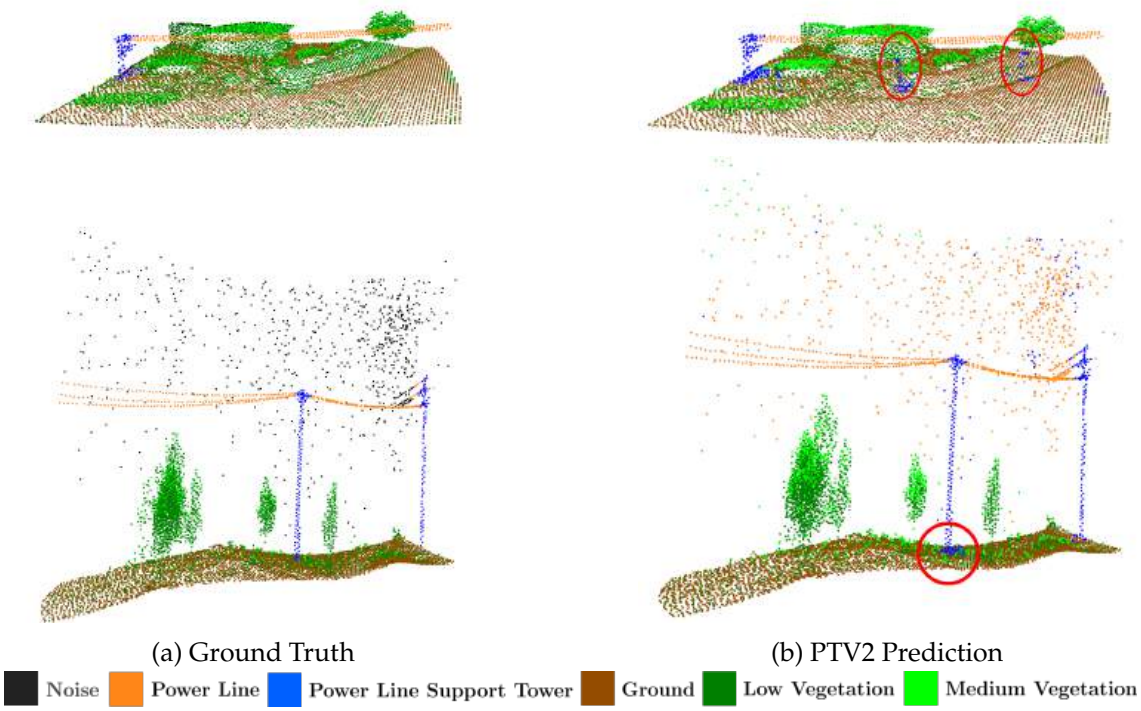


Figure 3.4: **Qualitative analysis of Point Transformer V2 (PTV2) performance on TS40K semantic segmentation.** The figure demonstrates both the capabilities and limitations of state-of-the-art point cloud segmentation methods on rural power grid infrastructure. **Top row:** Example showcasing PTV2's robust detection capabilities, where the model successfully identifies the primary transmission tower (left corner) and accurately segments smaller supporting structures that are occasionally missed in ground truth annotations. **Bottom row:** Challenging case revealing common failure modes, where PTV2 introduces spurious ground patches around tower bases that are absent in the original annotations. Additionally, PTV2 frequently misclassifies noise artifacts as power grid elements (such as towers and power lines), which undermines the reliability of ML-based inspection by compromising the accurate retrieval of infrastructure. These qualitative results complement the quantitative ones in Table 3.4, illustrating why even high-performing models require careful validation before deployment.

Performance Overview. As shown in Table 3.6, PV-RCNN achieves the best overall mAP of 61.23%, excelling across all three categories. Power line detection proves consistently strong for all models, with AP values above 83%. This can be attributed to the line's relative separation from the scene. In contrast, detecting support towers remains difficult. Even the best model (PV-RCNN) reaches only 40.32% AP for towers.

Table 3.6: **3D object detection:** Benchmark results of baselines on the TS40K test set under the 3D Average Precision (AP) metric with 11 recall points. We report mean AP and per-class AP scores.

Method	mAP (%)	Power Line Support Tower (%)	Power Line (%)	Medium Vegetation (%)
SECOND [116]	52.68	32.64	85.09	40.32
PointPillars [50]	56.63	38.63	83.74	47.53
PointRCNN [85]	57.65	36.54	88.71	44.26
Part-A^2 Net [86]	58.65	39.55	86.69	48.00
PV-RCNN [87]	**61.23**	**40.32**	**92.77**	**50.61**

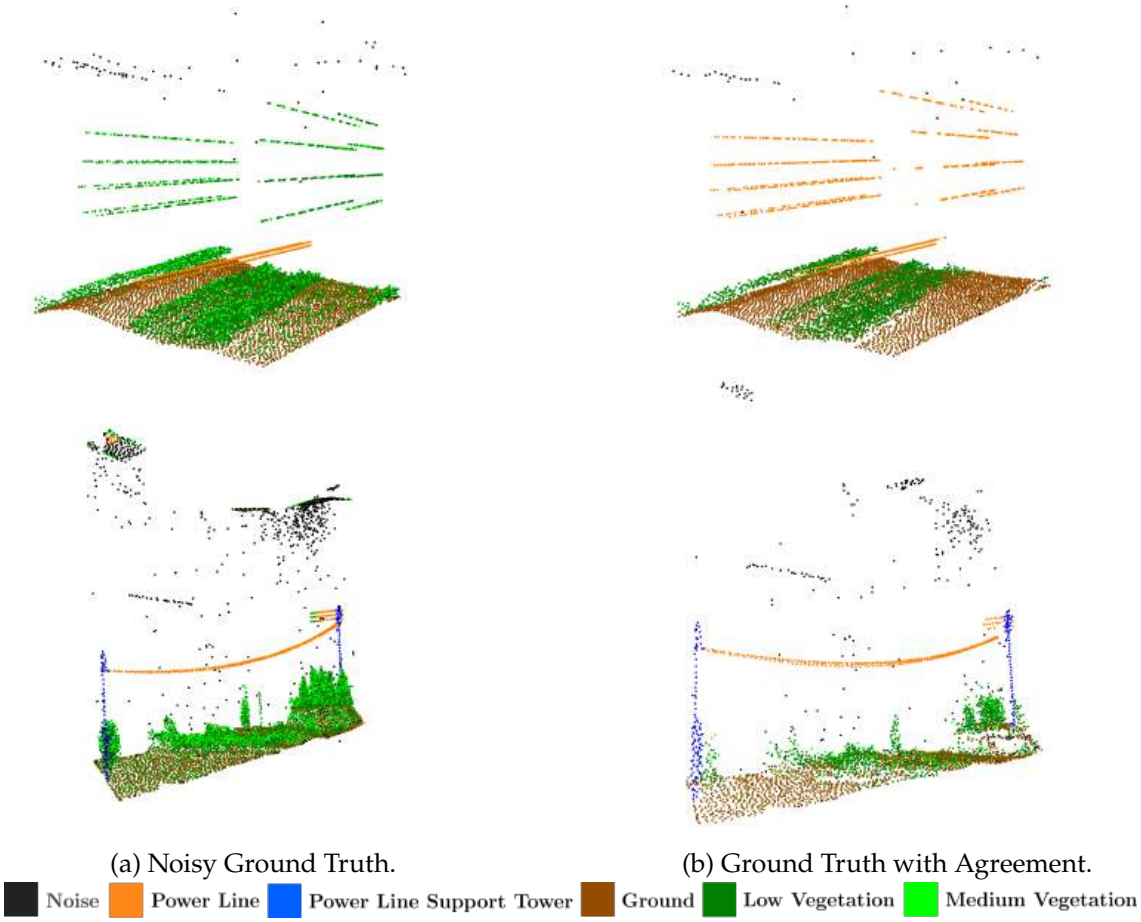
3.6 TS40K's Unique Challenges

3.6.1 Noisy Labels

The TS40K dataset inherits several annotation artifacts that are intrinsic to inspection pipelines. Annotations are achieved by maintenance personnel using a combination of heuristic filters and manual revision, with the goal of enabling faster inspections rather than machine learning. This introduces systematic noise in the ground truth, particularly near supporting structures. For instance, safety-related heuristics may expand tower labels to include nearby ground and clutter, ensuring that vegetation encroachments around infrastructure are not missed. Similarly, isolated power lines not belonging to the main transmission system may be labeled inconsistently or misclassified as medium vegetation. These mismatches between visual semantics and operational logic complicate the use of TS40K for pointwise learning tasks (as illustrated in Figure 3.5).

To mitigate this, we construct an alternative label set by computing model agreement among the five best-performing semantic segmentation baselines on the dataset. Specifically, we identify points for which at least seventy percent of the models predict the same class, and retain these labels as a proxy for higher-confidence ground truth. However, we also retain all points originally labeled as support towers regardless of model disagreement, given their low representation, low baseline performance, and critical importance. However, this results in a substantial reduction of the dataset size: approximately 40% points are pruned after filtering for consensus. To support transparent evaluation, we release both the original noisy labels and the alternative, agreement-based version, along with per-point agreement scores and majority class assignments. This enables researchers to select the labeling strategy best suited to their application.

The noisy labels in TS40K highlights the inherent difficulties of repurposing industry datasets for machine learning. At the same time, the diversity of scenarios and annotation policies ensures that TS40K remains representative of real-world inspection challenges. As a promising future direction, one could explore introducing a post-reviewing stage that combines automated and manual assessment: model predictions would be accepted when their softmax confidence exceeds a predefined threshold, while ambiguous cases would be deferred to human experts for final verification. We explore this hybrid approach in more depth in Chapter 4, as a mechanism to improve both the quality of annotations and



(a) Noisy Ground Truth. (b) Ground Truth with Agreement.

Figure 3.5: **Noisy labels in TS40K: Challenges and Mitigation.** Noisy labels represent a significant challenge in the TS40K dataset, particularly in safety-critical regions such as those surrounding supporting towers. In certain cases, non-grid power lines or spurious sensor points may be incorrectly annotated as vegetation or structural components due to heuristic-driven labeling procedures. This is especially common near towers, where conservative labeling practices are applied to capture potential encroachment risks. The top row illustrates such cases of mislabeling in the ground truth annotations. To address this, we leverage an agreement-based approach using predictions from state-of-the-art models. By selecting only the 3D points where model agreement exceeds a defined threshold, we construct a refined annotation mask that improves consistency and reduces noise. The bottom row displays the revised label distribution based on model agreement, which serves as an alternative supervision label for model training and evaluation.

the operational latency of inspection pipelines.

3.6.2 Spurious Points from High-density Noise

Although the TS40K dataset benefits from UAV-based acquisition with high point density and minimal occlusion, it is still subject to real-world sensor artifacts. Weather conditions, reflective surfaces, and trajectory perturbations can introduce spurious points that appear

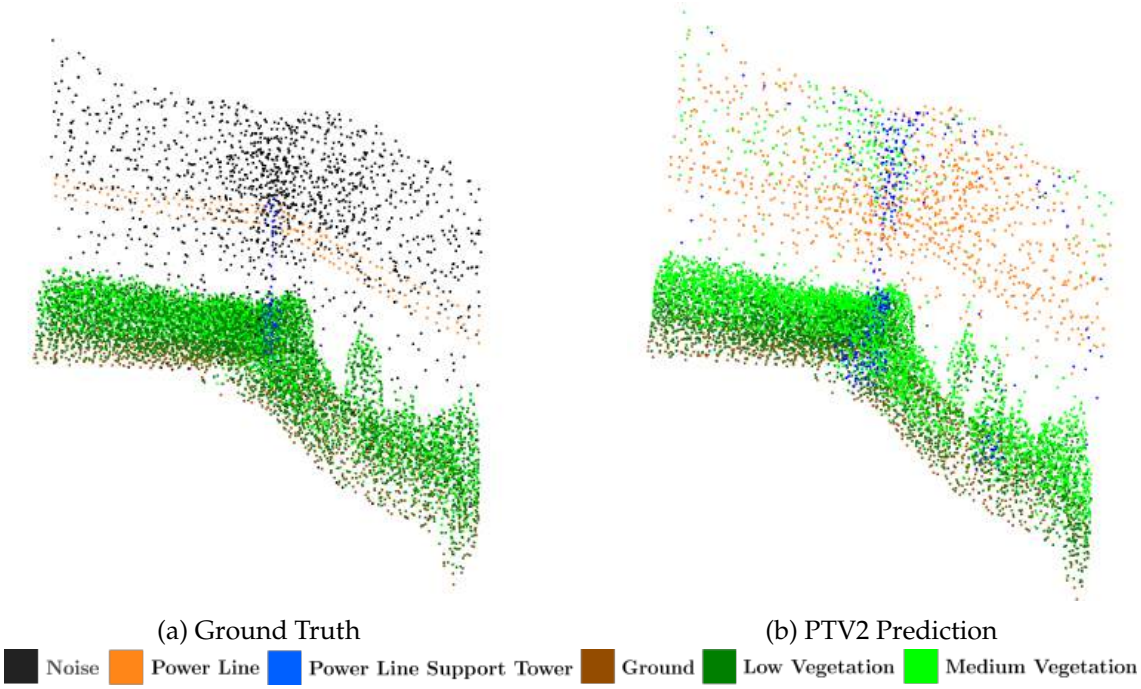


Figure 3.6: **Impact of high-density LiDAR noise on semantic segmentation in TS40K.** This figure illustrates a challenging scenario where Point Transformer V2 struggles to accurately segment power grid infrastructure due to dense spurious points. Weather conditions, reflective surfaces, and flight trajectory perturbations can generate unstructured noise clouds, particularly near towers and within vegetation. As a result, the model misclassifies noise points as power lines and towers. These artifacts blur segmentation boundaries and undermine the reliability of automated inspection. This highlights the need for denoising and uncertainty-aware learning strategies in real-world 3D datasets.

as dense noise clouds. These points often form unstructured outliers near towers or within vegetation volumes.

While the noise class is typically ignored during training, its presence during inference poses a significant challenge. In high-density regions, segmentation boundaries become blurred, and models are prone to confusion between towers, vegetation, and noise. As shown in Figure 3.6, even high-performing baselines such as Point Transformer V2 can produce unreliable predictions under these conditions. To better understand this, we compute the confusion matrix for the best-performing model (PTV2) using the original noisy labels. Table 3.5 reveals that more than forty percent of points labeled as noise are misclassified as power lines, and more than eleven percent are misclassified as towers. This level of misclassification is problematic in inspection pipelines, especially in power grid elements.

On the other hand, TS40K serves as a valuable benchmark for advancing denoising techniques and uncertainty-aware segmentation. Its real-world noise artifacts provide fertile ground for developing resilient learning strategies and are seldom found in other 3D point cloud benchmarks.

Table 3.7: **Focusing on towers**: Benchmark results of 3D semantic segmentation baselines on the TS40K trained with *tower-radius* and *power-line* sets.

Method	Test Set	mIoU (%)	Ground	Low Vegetation	Medium Vegetation	Power Line Support Tower	Power Line
PointNet [78]	Only Tower Including Test Set	34.35	50.38	52.07	0.9	7.15	61.16
PointNet++ [79]		39.55	48.47	50.54	2.37	30.06	66.51
RandLaNet [44]		4.28	5.26	0	16.12	0	0
KPConv [96]		47.46	48.40	26.82	30.85	42.69	88.53
PTV1 [20, 121]		50.29	64.12	25.98	35.46	41.71	78.10
PTV2 [20, 110]		**60.88**	75.08	**48.74**	43.24	**48.47**	**88.86**
PointNet [78]	Entire Test Set	32.77	48.63	50.51	2.18	4.73	57.82
PointNet++ [79]		34.78	45.17	50.60	1.03	16.56	60.51
RandLaNet [44]		4.13	2.07	0	18.58	0	0
KPConv [96]		42.38	49.40	18.54	29.10	**32.08**	82.77
PTV1 [20, 121]		44.89	69.47	28.53	31.90	16.43	78.10
PTV2 [20, 110]		**56.41**	77.35	55.06	40.98	25.79	**82.87**

Table 3.8: **Mitigating class imbalance**: Benchmark results of 3D semantic segmentation baselines on the TS40K trained with SMOTE and IDISS preprocess.

Method	Imbalance Technique	mIoU (%)	Ground	Low Vegetation	Medium Vegetation	Power Line Support Tower	Power Line
PointNet [78]	Oversampling: SMOTE	42.59	57.65	54.61	14.93	12.83	72.95
PointNet++ [79]		44.31	64.20	57.64	14.52	12.30	72.86
RandLaNet [44]		8.09	23.37	0	17.10	0	0
KPConv [96]		48.92	65.11	40.27	35.51	14.26	89.27
PTV1		59.65	73.52	52.30	40.97	40.16	91.32
PTV2 [20, 110]		**65.17**	79.42	62.87	47.41	**43.22**	**92.94**
PointNet [78]	Undersampling: IDIS	23.51	48.81	37.46	0.70	0.04	30.51
PointNet++ [79]		30.59	55.08	53.69	10.76	1.81	31.59
RandLaNet [44]		20.86	47.06	36.77	9.86	0.00	10.62
KPConv [96]		38.13	51.08	49.15	13.22	6.30	70.92
PTV1 [20, 121]		47.12	66.80	**58.24**	21.25	15.99	73.29
PTV2 [20, 110]		**49.80**	66.85	53.22	**30.35**	**18.66**	**79.91**

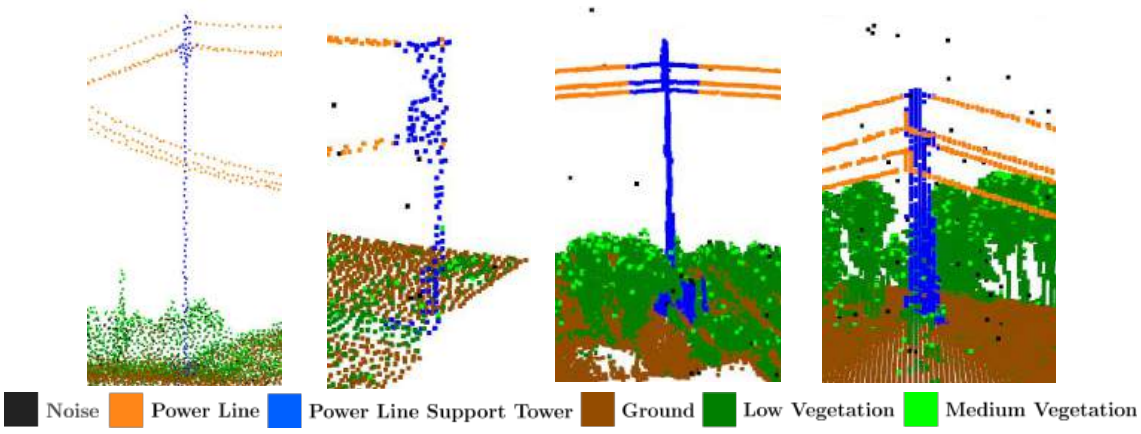


Figure 3.7: **Structural diversity of power line support towers in TS40K.** We showcase four distinct examples of supporting towers, each with markedly different shape and size and point density. This diversity reflects the variability encountered in power grid networks and highlights the challenge for machine learning models to generalize across heterogeneous structures.

3.6.3 Diverse Structures and the Impact of Extreme Class Imbalance

In addition to label noise and spurious artifacts, TS40K presents significant structural diversity across scenes. The dataset includes multiple types of supporting towers with varying geometries, materials, and configurations, as illustrated in Figure 3.7. These elements exhibit different profiles depending on voltage level, terrain, and vegetation

context. As a result, power line support towers are both rare and highly variable, which makes them particularly difficult to detect and segment reliably.

Although power lines and towers are similarly underrepresented in terms of point count, segmentation and detection models consistently perform better on power lines. This discrepancy is partially explained by shape regularity: power lines are consistently thin, elongated, and suspended in open space, while towers are bulky, embedded in vegetation, and structurally heterogeneous. Moreover, tower annotations often include nearby ground or noise, which introduces additional variance.

To evaluate whether the poor tower performance is attributable to data imbalance, we conduct two experiments. First, we train models only on samples that contain towers and lines, and evaluate performance on both tower-inclusive and full test sets. As shown in Table 3.7, this improves tower IoU significantly, but at the cost of generalization. When tested on the full dataset, all models show reduced performance, especially on vegetation classes. Second, we experiment with oversampling towers using SMOTE and undersampling dominant classes using IDISS. Results in Table 3.8 show that SMOTE yields modest gains, particularly in transformer-based models. In contrast, IDISS tends to harm performance, likely because it disrupts geometric consistency across scenes. These results indicate that imbalance alone does not fully explain poor tower recognition. The combination of structure diversity, annotation noise, and low point count provide a more comprehensive explanation.

3.7 Conclusion and Future Work

In this Chapter, we present **TS40K**, a large-scale dataset specifically designed for advancing 3D scene understanding in the context of rural electrical infrastructure. Unlike prior benchmarks which primarily address urban or synthetic settings, TS40K captures the complexity of real-world power-grid corridors, offering high-resolution LiDAR scans acquired from UAV platforms across 40,000 kilometers of diverse rural terrain. Each sample is densely annotated using an ontology derived from operational inspection pipelines, enabling research in both semantic segmentation and object detection. The motivation behind TS40K stems from the growing need to automate and scale inspection processes over expansive transmission networks, where manual practices are costly, time-consuming, and potentially hazardous. By enabling machine learning models to process high-density 3D data, stakeholders can reduce inspection latency, detect structural or environmental hazards proactively, and prioritize human interventions based on algorithmic recommendations. These capabilities are especially critical for mitigating wildfire risks and minimizing service disruptions in remote areas.

We conduct extensive experiments on TS40K across two core tasks: 3D semantic segmentation and 3D object detection using a suite of state-of-the-art baseline methods. The results, while promising in some categories such as power line segmentation, reveal

shortcomings in the detection of supporting towers and in the model's robustness to high-density noise.

Beyond its immediate experimental use, TS40K constitutes a novel benchmark for large-scale 3D scene understanding in real-world rural environments. In contrast to highly curated academic datasets or predominantly driving benchmarks, TS40K captures rural infrastructure settings where structural heterogeneity, extreme class imbalance, real sensor noise, and imperfect annotations are intrinsic properties of the data rather than controlled artifacts. As such, this dataset provides a demanding testbed for evaluating inductive biases, noise detection, and learning under an imperfect ground truth. Here, we highlight several promising avenues for future work:

- **Improving annotation quality**: We explore this further in the next chapter by proposing a hybrid post-reviewing pipeline that combines softmax-based confidence thresholds with human inspection to refine ambiguous regions and reduce noise-related errors.
- **Semi-supervised and weakly supervised learning**: Given the high annotation cost and presence of mislabeled regions, future models may benefit from architectures that can leverage unlabeled or partially labeled data, either through self-training, teacher-student models, or contrastive representations.
- **Uncertainty and noise modeling**: TS40K presents a valuable opportunity for evaluating methods that explicitly model aleatoric uncertainty, denoise spurious points, or integrate denoising mechanisms within the learning process itself.
- **Infrastructure-aware scene modeling**: As current methods struggle with highly variable infrastructure geometry, future work may incorporate priors about object topology (e.g., parametric tower models) to improve detection and generalization. We develop this idea further in Chapter 5, by introducing a white-box model that leverages the structural properties of power line support towers to detect them.

4

Cost-Aware Decision Support System for Power Grid Inspections

This Chapter introduces a cost aware decision support system for the inspection of electrical power grids. The work expands directly on the results presented in Chapter 3, where the TS40K dataset was proposed and analysed. While said chapter focused on the creation of a comprehensive benchmark for semantic segmentation and object detection in rural power transmission scenarios, this work takes a step further by investigating how the outputs of these models can be incorporated into a decision making system that addresses the specific needs of inspection teams.

We begin by presenting the motivation and context for a cost aware decision support framework for rural power grid inspections. Then, we expand on the TS40K evaluation, focusing on performance indicators and inspection relevant classes that directly impact decision making. Next, we introduce the workflow of the inspection tool, which combines computer vision predictions, cost estimation, and human-in-the-loop validation. Lastly, we develop an operational cost model, enabling us to quantify trade offs between traditional inspections and data driven approaches.

4.1 Introduction

Ensuring the safe and efficient operation of electrical transmission and distribution systems is a fundamental responsibility of power grid operators. Regular inspections are essential to maintaining grid reliability by detecting structural defects, assessing collision risks, and mitigating environmental hazards such as vegetation encroachment, structural degradation, and damage caused by severe weather events, including wildfires. Traditional inspection methods, which rely on on-site personnel or manned helicopters, are resource intensive, costly, and time consuming, and therefore struggle to meet the efficiency and scalability requirements of modern complex power networks.

Recent advances in unmanned aerial vehicle technology, particularly the integration of LiDAR sensors, have transformed the way inspections can be performed. UAVs enable the

remote acquisition of high resolution three dimensional point clouds of transmission and distribution infrastructure, thereby reducing the need for on-site deployment. However, while UAVs streamline data collection, the manual annotation and interpretation of these large scale datasets remains time consuming, expensive, and susceptible to human error. This limitation creates a unique opportunity for adopting Computer Vision (CV) strategies to speed up labeling and automatically detect potential risks.

3D semantic segmentation methods have the potential to substantially improve inspection workflows by automatically identifying important actors such as towers, power lines, and surrounding vegetation directly from LiDAR point clouds. By designing an inspection tool powered with CV, we can improve inspection efficiency, reduce costs, and support timely interventions. Compared to 2D image based approaches, which are widely used for inspection tasks such as fault detection, component identification, and vegetation encroachment monitoring, LiDAR based methods provide a richer representation of the scene. Two dimensional imagery is inherently limited by its dependence on favourable weather and lighting conditions and by the absence of depth information, both of which are critical for reliable distance estimation between vegetation and power grid elements. The TS40K dataset, introduced in the previous chapter, was specifically developed to address these limitations. It contains high density UAV captured three dimensional point clouds of power grids in rural environments, with detailed annotations for essential components. The dataset captures a diverse range of tower structures, varying terrain conditions, and surrounding vegetation, making it highly suitable for training ML models tailored to realistic inspection scenarios. In this chapter, we build upon the benchmark results obtained with TS40K to develop a cost aware decision support system for power grid inspections.

This work bridges the gap between academic research in 3D computer vision and the operational requirements of power grid operators. We present a deployable inspection framework that integrates transformer-based models trained on TS40K with a cost estimation module. The system automates core inspection tasks such as vegetation encroachment assessment and collision risk evaluation, while also incorporating an uncertainty-aware review process to maintain reliability. The main contributions of this chapter are as follows:

1. An in-depth evaluation of 3D semantic segmentation approaches tailored for power grid inspection tasks, highlighting practical advantages and deployment considerations.

2. An extensive benchmarking analysis using the TS40K dataset, leveraging features such as surface normals and color information to reflect real-world inspection requirements.

3. An inspection pipeline that combines 3D computer vision outputs with a manual uncertainty-aware validation step, ensuring both efficiency and reliability in

operational settings.

4.2 Benchmark Results on TS40K

This section provides a comprehensive overview of benchmarking results on the TS40K dataset. We begin by detailing the evaluation metrics used to assess model performance. Next, we present baseline results for state-of-the-art 3D semantic segmentation models under various training configurations. Then, we explore the use of normal vectors and RGB information, comparing their effects on model performance and discussing implications for deployment inspection scenarios.

4.2.1 Evaluation Metrics

A rigorous evaluation protocol is essential for assessing the suitability of computer vision models for power grid inspections. In this work, we adopt evaluation metrics that not only measure predictive performance but also reflect the practical implications of errors when deployed in inspection workflows. The chosen metrics are computed per class and then aggregated, allowing us to highlight performance trends for critical inspection targets such as supporting towers and power lines.

4.2.1.1 Confusion Matrix and Operational Implications

The confusion matrix is a fundamental tool in semantic segmentation evaluation. It organises the predictions of a model into four categories that quantify both correct and incorrect classifications:

- **True Positives (TP)** — points correctly predicted as belonging to a given class. In the context of inspections, high TP values for towers and power lines indicate that the model effectively detects these components, ensuring that critical infrastructure is properly monitored.
- **False Positives (FP)** — points incorrectly assigned to a class they do not belong to. For example, misclassifying vegetation as a power line can trigger unnecessary maintenance interventions, increasing operational costs and diverting resources.
- **False Negatives (FN)** — points belonging to a class that the model fails to identify. FN errors are particularly critical for inspections, as they can result in undetected vegetation encroachment or structural defects, potentially leading to outages or safety hazards.
- **True Negatives (TN)** — points correctly identified as not belonging to the class of interest. Although not the primary focus in semantic segmentation, TN values contribute to overall accuracy.

From an operational perspective, FP errors tend to degrade efficiency, while FN errors directly compromise safety. The TS40K dataset's combination of noisy labels, dense noise artifacts, and structural diversity in towers makes both FP and FN mitigation challenging. Therefore, a balanced interpretation of these metrics is necessary when selecting models for deployment.

4.2.1.2 Intersection over Union (IoU) and Mean IoU

Intersection over Union (IoU) is the standard metric for evaluating semantic segmentation. It is defined for a given class c as:

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}, \tag{4.1}$$

where TP_c, FP_c, and FN_c represent the true positives, false positives, and false negatives for class c. The mean IoU (mIoU) averages this value over all C classes:

$$\text{mIoU} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c. \tag{4.2}$$

mIoU provides a global performance measure, but in inspection scenarios, high per-class IoU for towers and power lines is often more important than the overall mIoU score. This metric is the established benchmark for comparing state-of-the-art 3D semantic segmentation models in the literature, and as such, it will also serve as the primary metric for comparing the different models evaluated in our TS40K experiments.

4.2.1.3 F_β Score and its Role in Inspection Scenarios

The F_β score is an alternative evaluation metric that combines precision and recall into a single measure, allowing different weightings between the two. It is defined as:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}, \tag{4.3}$$

where Precision measures the proportion of correctly identified positive points out of all predicted positives, and Recall measures the proportion of correctly identified positive points out of all actual positives.

The parameter β controls the trade-off:

- $\beta = 1$ (F_1) gives equal weight to precision and recall, suitable for balanced inspection objectives.
- $\beta > 1$ (e.g., F_2) prioritises recall, reducing FN errors and minimising the chance of missing critical defects.
- $\beta < 1$ (e.g., $F_{0.5}$) prioritises precision, reducing FP errors and avoiding unnecessary maintenance actions.

In operational terms, F_2 is ideal for safety-critical scenarios, such as detecting faults in towers where missing an issue could have severe consequences. $F_{0.5}$ is appropriate for cost-sensitive situations, such as vegetation clearance planning, where false alarms would waste resources. The ability to select an appropriate β value enables the inspection pipeline to be tuned to the specific priorities of a maintenance campaign.

4.2.2 Baseline Benchmarking on TS40K

This section presents the performance of state-of-the-art 3D semantic segmentation baselines on the TS40K dataset under a variety of training configurations. Given the dataset's unique challenges: noisy labels, high-density noise points, and a diverse set of tower structures, our benchmarking emphasises the impact of different training strategies. We focus particularly on segmentation performance for supporting towers and power lines, as these elements are the most critical for power grid inspection tasks.

4.2.2.1 Weighted Training to Prioritize Towers and Power Lines

Supporting towers and power lines constitute a relatively small proportion of points in TS40K, making them underrepresented compared to dominant classes such as ground or vegetation. To address this imbalance, we apply a class-weighted loss function that increases the relative contribution of these critical classes during training. The aim is to improve per-class Intersection over Union (IoU) for towers and power lines.

It is important to note that the results reported in Table 3.4 for TS40K are higher than those presented here. In that earlier benchmarking, noise points were ignored during both training and evaluation, a common practice in machine learning benchmarks. While this approach simplifies the evaluation process, it is not suitable for the TS40K dataset because noise points form a substantial and challenging component of the data. Ignoring them leads to overly optimistic results that are not representative of real inspection conditions. For a realistic operational assessment in the same conditions as Table 3.4, we report in Table 4.1 results where noise points are excluded during training but included in evaluation.

Table 4.1: Benchmark results of 3D semantic segmentation baselines on the TS40K test set using a weighting scheme to prioritise underrepresented classes. Noise is ignored during training but considered in evaluation. All values are reported in percentages (%).

Model	Mean IoU	Noise	Ground	Low Veg	Med Veg	Tower	Power Line
PTV3 [20, 111]	**58.58**	—	76.13	64.11	**48.37**	**51.12**	53.16
PTV2 [20, 110]	57.12	—	**80.65**	**67.29**	46.39	43.00	48.29
PTV1 [20, 121]	56.07	—	78.08	62.28	42.93	47.36	49.70
KPConv [96]	43.86	—	64.95	39.31	31.87	32.49	50.69
PointNet++ [79]	50.61	—	66.90	59.61	17.96	28.86	**79.72**
PointNet [78]	42.67	—	58.66	53.50	16.43	11.53	73.24
RandLaNet [44]	6.52	—	16.73	0.00	15.89	0.00	0.00

Table 4.1 shows that transformer-based architectures (PTV1–PTV3) achieve the highest mean IoU, with PTV3 reaching 58.58%. Interestingly, while PointNet++ attains the highest power line IoU (79.72%), it performs substantially worse in tower detection. From an inspection perspective, this trade-off is not ideal: a model optimised for power line detection but weak in tower detection limits its usefulness and may lead to missed structural issues during inspections.

4.2.2.2 Noise Detection and Its Impact on Performance

Noise points in TS40K are often misclassified as towers or power lines due to similarities in point distribution and height. Given their high density, such misclassifications can severely reduce operational efficiency by triggering false inspection alerts. Given that noise points are more frequent than actual power grid elements and that they are often misclassified as towers or power lines, we train models to explicitly detect noise as a separate class, aiming to improve the discrimination between actual infrastructure and sensor artifacts.

Table 4.2: Benchmark results of 3D semantic segmentation baselines on the TS40K test set with noise detection included as a separate class. We report mean IoU and per-class IoU values. All values are reported in %.

Model	Mean IoU	Noise	Ground	Low Veg	Med Veg	Tower	Power Line
PTV3 [20, 111]	63.55	59.23	70.77	50.47	43.86	**61.42**	**95.53**
PTV2 [20, 110]	**68.29**	61.16	**80.13**	**68.17**	**51.39**	54.48	94.43
PTV1 [20, 121]	64.90	57.50	77.33	60.34	46.51	54.19	93.54
KPConv [96]	52.77	57.02	64.75	37.12	34.63	37.36	89.99
PointNet++ [79]	45.99	59.27	59.99	54.36	14.55	22.61	78.41
PointNet [78]	30.01	49.36	54.52	46.00	14.23	0.00	35.28
RandLaNet [44]	6.50	7.91	0.00	0.00	21.58	0.00	10.92

As shown in Table 4.2, noise detection yields substantial improvements for power grid elements, particularly in transformer-based models. For example, PTV3 achieves an IoU of 95.53% for power lines, a gain of 42.3% compared to the baseline in Table 4.1. This demonstrates that noise detection is crucial in the TS40K dataset, as it forces models to learn to distinguish between noise and actual power grid components, thereby improving the overall segmentation performance.

4.2.2.3 Removing Ground Points for Operational Relevance

In operational practice, many inspection workflows remove ground points before analysis, as these do not typically require inspection and constitute the majority of the dataset. In TS40K, ground points account for over 55% of all points. Removing them allows models to allocate computational and representational capacity to more relevant classes.

Table 4.3 demonstrates that removing ground points results in noticeable gains in IoU for key inspection classes, with PTV3 reaching 96.25% for power lines and 65.05% for towers, which represent gains of 0.72% and 13.93% respectively compared to the baseline in

Table 4.3: Benchmark results of 3D semantic segmentation baselines on the TS40K test set with the ground class removed during training. We report mean IoU and per-class IoU values. All values are reported in %.

Model	Mean IoU	Noise	Ground	Low Veg	Med Veg	Tower	Power Line
PTV3 [20, 111]	67.46	64.67	—	64.24	47.08	**65.05**	**96.25**
PTV2 [20, 110]	**72.86**	**70.89**	—	**76.86**	**59.01**	61.69	95.83
PTV1 [20, 121]	67.89	58.25	—	74.28	54.69	57.15	95.07
KPConv [96]	47.31	61.44	—	59.03	42.13	42.73	92.64
PointNet++ [79]	42.72	61.35	—	76.08	27.94	25.88	83.70
PointNet [78]	36.68	49.32	—	45.74	19.78	8.57	45.69
RandLaNet [44]	7.25	7.63	—	3.41	19.26	0.00	12.04

Table 4.2. In the development of a decision-support system for inspections, this approach is particularly relevant, as it allows the model to focus on the most critical components of the power grid and reduces the computational burden associated with processing large amounts of irrelevant data.

4.2.2.4 Qualitative Analysis of Segmentation Outputs

Figure 4.1 illustrates the qualitative performance of PTV3. The segmentation outputs show that PTV3 is able to accurately delineate towers and power lines, even in the presence of noise artifacts and complex vegetation. Notably, noise points are now effectively separated from genuine power grid elements, whereas previously they would have been misclassified as such. This clear distinction prevents false alarms and ensures that inspection resources are focused on actual infrastructure rather than sensor artifacts.

4.2.3 Semantic Segmentation on TS40K with Normal Vectors

Normal vectors, which represent the orientation of surface elements in three dimensional space, are a commonly used feature in three dimensional semantic segmentation. They provide geometric context that can help a model differentiate between flat horizontal surfaces. In principle, this additional geometric information can improve the delineation of object boundaries and enhance the recognition of classes with distinctive geometric profiles.

Table 4.4: Benchmark results of three dimensional semantic segmentation baselines on the TS40K test set when incorporating normal vectors as additional input features. All values are reported in percentages (%).

Model	Mean IoU	Noise	Ground	Low Veg	Med Veg	Tower	Power Line
PTV3 [20, 111]	64.00	59.21	70.55	53.18	44.27	**61.16**	**95.65**
PTV2 [20, 110]	66.77	**62.62**	78.20	61.66	47.28	56.36	94.51
PTV1 [20, 121]	**67.75**	61.27	**79.14**	**65.05**	**49.69**	56.69	94.68
KPConv [96]	55.95	56.48	67.06	42.08	36.89	42.08	91.65
PointNet++ [79]	48.32	60.38	60.17	55.14	16.04	30.02	80.23
PointNet [78]	39.84	51.83	56.54	48.32	20.39	9.34	52.62
RandLaNet [44]	8.17	9.23	0.31	4.65	13.42	1.05	10.46

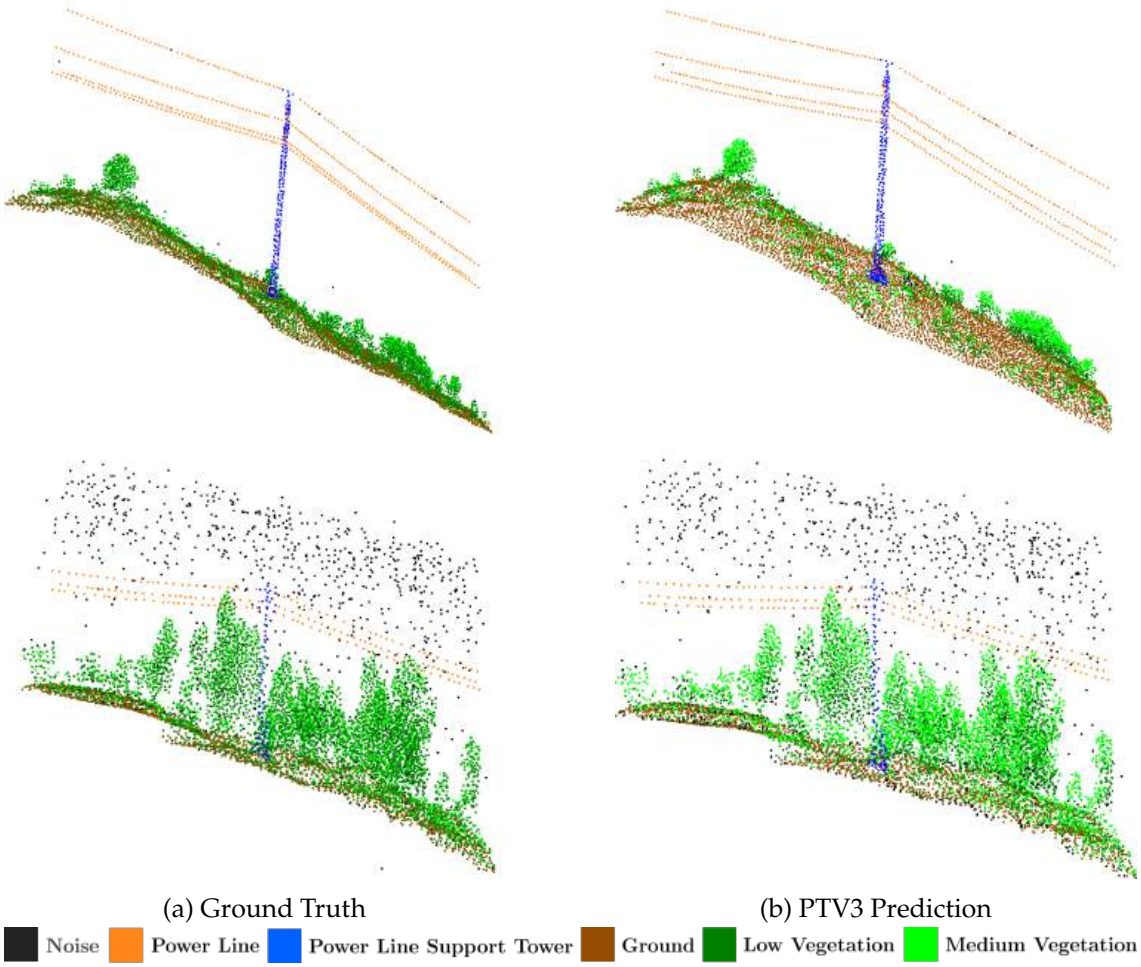


Figure 4.1: **Qualitative results of Point Transformer V3 (PTV3) on TS40K**. While not the top performer in mean IoU, PTV3 achieves the highest segmentation accuracy for towers and power lines, making it well-suited for operational scenarios prioritising these classes.

We evaluate the impact of adding normal vectors to the feature set for all baseline models, keeping the training configuration otherwise identical to the noise detection setting. This ensures that differences in performance can be attributed primarily to the inclusion of normal vectors. To ensure a fair assessment of model performance, we compare results with Table 4.2, which includes the ground class. While removing ground points (as in Table 4.3) serves as a proof of concept that such filtering can boost performance, it does not provide a complete picture of model capabilities across all classes present in the dataset. For benchmarking different segmentation models, we consider all annotated classes so that results reflect the full complexity of TS40K.

As shown in Table 4.4, the inclusion of normal vectors yields only modest improvements over the noise detection baseline in Table 4.2. For example, PTV1 improves from 64.90% to 67.75% mean IoU, while PTV3 increases from 63.55% to 64.00%. These gains are not consistent across all models, with some methods showing negligible differences. A key observation is that transformer based architectures maintain their lead over other baselines

regardless of whether normal vectors are used. This suggests that normal vectors have minimal impact on model performance, as there are no major differences when compared to evaluations without them. When comparing the results in Table 4.4 to those in Table 4.2, it becomes clear that normal vectors do not help with performance in the TS40K dataset. They provide indirect cues that may help in specific cases but do not reliably improve discrimination between noise and fine structural elements.

4.2.4 Extended Evaluation on TS-RGB

4.2.4.1 Dataset Description and Differences from TS40K

Figure 4.2: **TS-RGB Dataset Visualization.** TS-RGB is an augmented version of the TS40K dataset, incorporating RGB channels to improve 3D semantic segmentation in power grid environments. Covering approximately 8,000 kilometers of transmission network, it includes over 1,295 million points collected using LiDAR sensors. Ground points are automatically removed by heuristics and annotations exclude differentiation between low and medium vegetation.

The TS-RGB dataset is an augmented variation of TS40K, enriched with RGB colour information for each point in the LiDAR point cloud. It covers approximately eight thousand kilometres of rural transmission network and contains more than one billion two hundred and ninety five million points. While it shares the same transmission corridors as TS40K, the data in TS-RGB were collected using a more advanced LiDAR sensor, resulting in substantially higher point density. This higher density has the potential to improve the detection of thin structures such as power lines, although it also increases the volume of data to process. Several differences in annotation practice distinguish TS-RGB from TS40K. Ground points are automatically removed using well established heuristic filters that are routinely applied in operational inspections. Furthermore, the dataset does not

distinguish between low and medium vegetation, merging them into a single vegetation class. These differences mean that TS-RGB more closely resembles the data that would be processed in an actual utility company pipeline after initial pre-processing.

4.2.4.2 Performance without RGB Information

To establish a baseline, we first train models on TS-RGB using only point coordinates, without incorporating the RGB channels. This setting allows direct comparison with TS40K, since it mirrors the feature space used in the earlier experiments.

Table 4.5: Benchmark results of three dimensional semantic segmentation baselines on the TS-RGB test set using only point coordinates. All values are reported in percentages (%).

Model	Mean IoU	Noise	Vegetation	Tower	Power Line
PTV3 [20, 111]	59.15	29.24	91.48	38.14	77.72
PTV2 [20, 110]	**59.65**	31.07	91.75	**38.45**	77.34
PTV1 [20, 121]	58.28	31.25	**92.77**	31.36	**77.75**
KPConv [96]	49.12	19.83	87.09	21.94	68.34
PointNet++ [79]	43.02	**51.58**	14.89	28.03	62.04
PointNet [78]	36.59	45.57	17.63	8.59	40.80
RandLaNet [44]	7.01	8.57	11.62	1.28	9.27

As shown in Table 4.5, transformer based models again achieve the highest mean IoU scores, with PTV2 slightly outperforming the others at 59.65%. The vegetation class benefits the most from the increased point density, reaching over 92% IoU for PTV1. However, tower and power line segmentation performance is somewhat lower than in TS40K, likely due to the dominance of the vegetation class which constitutes the vast majority of points in the dataset. Figure 4.3 provides a qualitative illustration of PTV2's segmentation results on TS-RGB.

4.2.4.3 Impact of Adding RGB Channels

We next investigate the effect of incorporating RGB channels alongside the coordinate features.

The results in Table 4.6 show that RGB channels provide only marginal improvements in mean IoU. For example, PTV3 improves from 59.15% to 59.34%, with the largest gains observed in vegetation segmentation. This limited impact is likely due to the lack of strong colour contrast between vegetation and infrastructure elements, which often share similar tones in aerial imagery.

4.2.4.4 Impact of RGB and Normal Vectors Combined

Finally, we evaluate the effect of combining RGB channels with normal vectors. Table 4.7 shows that combining RGB with normal vectors does not yield consistent performance

Table 4.6: Benchmark results of three dimensional semantic segmentation baselines on the TS-RGB test set with RGB channels included. All values are reported in percentages (%).

Model	Mean IoU	Noise	Vegetation	Tower	Power Line
PTV3 [20, 111]	**59.34**	29.51	**92.17**	**38.21**	**77.45**
PTV2 [20, 110]	51.09	26.78	90.88	13.97	72.74
PTV1 [20, 121]	52.27	28.84	91.48	18.45	70.30
KPConv [96]	50.81	20.35	89.31	22.85	70.73
PointNet++ [79]	43.01	**53.78**	14.31	26.84	61.60
PointNet [78]	35.42	46.15	18.16	8.31	40.47
RandLaNet [44]	7.25	8.20	11.84	0.98	9.32

Table 4.7: Benchmark results of three dimensional semantic segmentation baselines on the TS-RGB test set with RGB channels and normal vectors included. All values are reported in percentages (%).

Model	Mean IoU	Noise	Vegetation	Tower	Power Line
PTV3 [20, 111]	57.97	29.02	**92.71**	33.56	76.58
PTV2 [20, 110]	54.25	28.65	91.42	23.65	73.29
PTV1 [20, 121]	**59.27**	31.47	91.57	**34.99**	**79.06**
KPConv [96]	52.45	22.10	91.23	30.75	65.71
PointNet++ [79]	41.03	**55.32**	13.50	29.01	60.25
PointNet [78]	37.02	45.08	17.05	9.52	41.58
RandLaNet [44]	8.05	7.89	12.53	1.54	8.86

improvements. While PTV1 achieves the highest mean IoU at 59.27%, other models see little change or even a decrease compared to the RGB only configuration. This suggests that in the TS-RGB setting, geometric and colour features provide overlapping rather than complementary information.

4.2.4.5 Operational Relevance of RGB Data in Inspection Tasks

From an operational perspective, the results indicate that RGB data offer limited benefits for discriminating between vegetation and infrastructure in the TS-RGB dataset. The colour similarity between classes reduces the discriminative power of RGB channels. Consequently, the primary advantage of TS-RGB over TS40K lies in its higher point density and lack of ground points rather than the addition of colour information.

4.2.5 Analysis and Key Findings

Across all experiments on TS40K and TS-RGB, transformer based architectures consistently achieve the highest mIoU scores and the most balanced performance across classes. PTV3 and PTV2 are the strongest performers, with PTV3 excelling in tower and power line segmentation and PTV2 achieving slightly higher overall mIoU in certain configurations. Explicit noise detection improves performance by addressing the high density

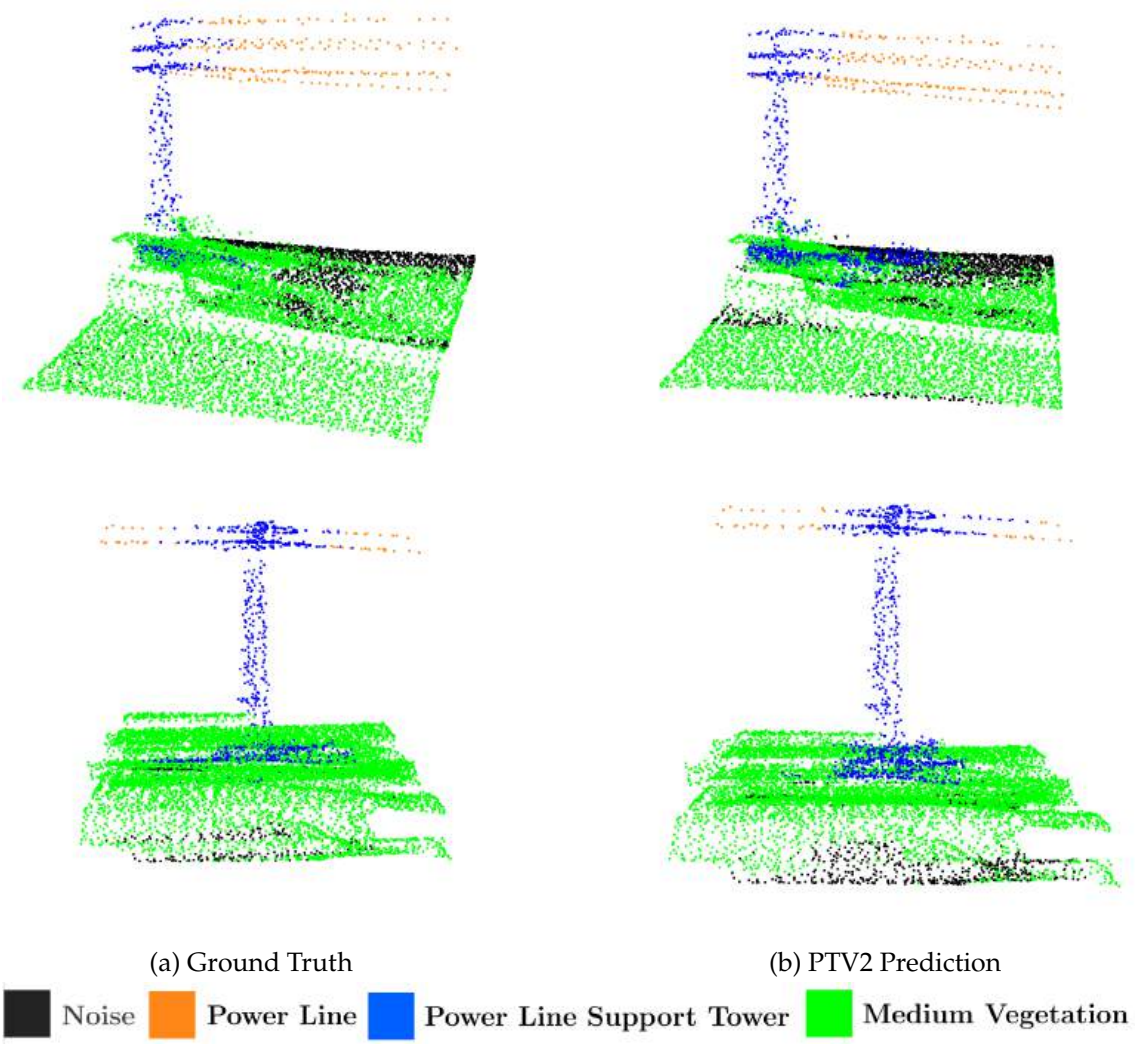


Figure 4.3: **Qualitative results of Point Transformer V2 (PTV2) [110] on the TS-RGB dataset**. We use only point coordinates as input features, without RGB or normal vectors. PTV2 achieves the highest mean IoU on TS-RGB, demonstrating strong performance in tower and power line segmentation. The model effectively distinguishes between vegetation and infrastructure, although some misclassifications occur in complex scenes.

noise in TS40K. Removing ground points during training also benefits performance on inspection elements. The inclusion of normal vectors yields only modest gains compared to noise detection, and in TS-RGB, RGB channels provide limited improvement, mostly for vegetation segmentation. Combining RGB and normal vectors does not lead to consistent benefits. From an operational standpoint, transformer based models with explicit noise detection and class weighting are the most effective choice for reliable detection of towers and power lines.

4.3 Inspection Tool for Power Grid Segmentation

The inspection tool is implemented as a functional Python application with a graphical user interface, developed to allow maintenance personnel to interact with advanced 3D semantic segmentation models without the need for programming expertise. The primary purpose of the tool is to integrate trained segmentation models into a workflow that can process LiDAR point clouds of transmission networks and produce outputs that are directly actionable for inspection and maintenance planning. Figure 4.4 shows the graphical user interface of the inspection tool, illustrating its main features and user interaction options. The interface enables users to load LiDAR datasets, execute the segmentation pipeline and export annotated point clouds. By embedding the processing logic in a user friendly interface, the tool bridges the gap between machine learning models and its use by non-experts.

The workflow of the tool follows a structured pipeline that ensures computational efficiency and compatibility with operational requirements. It consists of the following stages:

Point cloud partition: With no labels, we cannot partition the point cloud in sample types as described in TS40K. Instead, we segment the input point cloud into manageable segments based on spatial proximity. Each segment is approximately 50 meters long, which corresponds to the average distance between transmission towers in rural networks. This segmentation allows for batch processing and ensures that the system can handle large datasets even with limited GPU memory.

Preprocessing with Farthest Point Sampling: Within each segment, Farthest Point Sampling (FPS) is applied to reduce the point set to a fixed size of one hundred thousand points. Not only is this preprocessing step also part of the TS40K pipeline, but it is also crucial for ensuring that the model can operate efficiently on high density LiDAR data. FPS selects points that are well distributed across the segment, preserving the geometric structure while reducing computational load.

Prediction: The reduced segments are then processed by a trained 3D semantic segmentation model, which produces a predicted class label for each point along with a probability distribution over all classes. These probabilities are used both for assigning labels and for quantifying the model's confidence in its predictions.

Reconstruction: The labelled segments are merged to reconstruct the complete annotated point cloud. This reconstruction maintains sufficient density for reliable inspection while remaining compact enough for storage and further analysis.

Low confidence review (Optional): After reconstruction, the tool identifies points with low confidence scores based on the softmax distribution of the segmentation model. These points are flagged for manual review, allowing inspectors to focus on areas where the model's predictions are uncertain. This step is optional and can be skipped if the user prefers to review all points.

Automatic risk assessment (Optional): The final stage of the pipeline is dedicated to

evaluating vegetation encroachment risks in proximity to power grid elements. For each tower or power line point, a ball radius neighborhood is defined, with the safety distance serving as its radius. All vegetation points within this neighborhood are identified as potential risks. If any points are present, the tool then verifies whether these points are *genuine vegetation* or *artifacts of misclassification*. This verification is performed in two steps: first, a clustering technique (namely, DBSCAN) is applied to group vegetation points into clusters, even those outside the immediate neighborhood. Clusters that exceed a minimum size threshold are considered indicative of true vegetation encroachment and automatically flagged for further action, such as scheduling maintenance or issuing alerts. For clusters below the threshold, the tool leverages the model's confidence scores. High-confidence vegetation points, even if isolated, are flagged for manual review to ensure that potentially hazardous cases are not overlooked. Conversely, low-confidence points are disregarded, as they are likely to be false positives resulting from model uncertainty or sensor noise. This post-processing step is designed to balance operational efficiency and safety: it reduces the number of false alarms that would otherwise burden inspection teams, while maintaining high recall for actual hazards. The risk assessment module is fully optional and can be bypassed for workflows that prefer manual evaluation.

Tool's output: The output of the inspection tool is a reconstructed point cloud in LAS format, which includes an additional column for the predicted class labels. The output file retains the original filename with a `_segmented` suffix, saved in the same directory. The tool also provides a progress bar to indicate the status of the processing pipeline, ensuring transparency and user feedback during long-running operations.

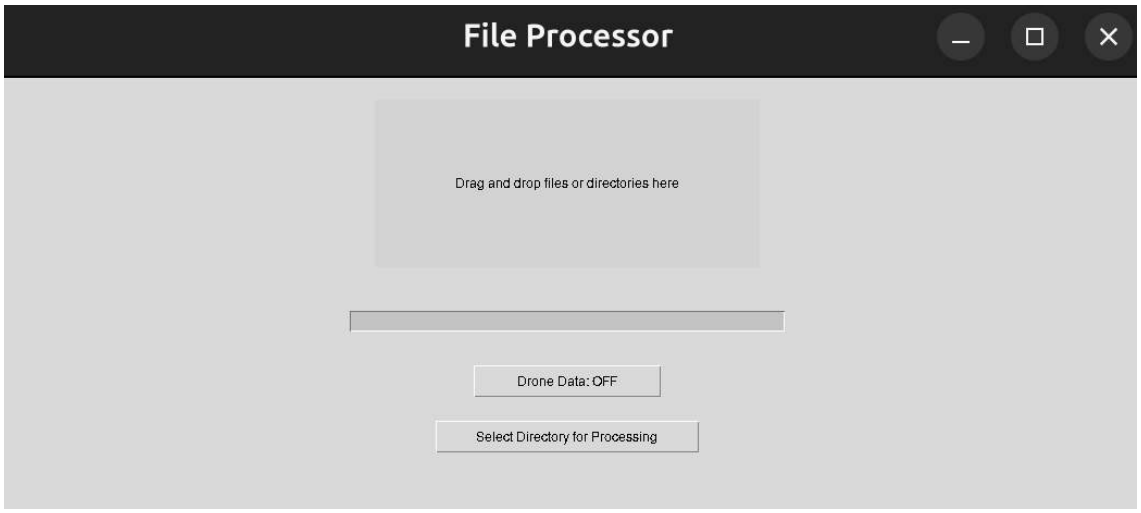


Figure 4.4: Graphical user interface of the inspection tool. Users can drop files or select a folder containing `.las` point cloud files for processing. The progress bar indicates the current status of the pipeline. Once finalized, a `.las` file is produced with the same name, only with a `_segmented` suffix added and an additional column with the segmentation results named *classification*. The "Drone Data" toggle allows users to specify whether the input corresponds to TS40K (helicopter-mounted LiDAR) or TS-RGB (UAV-captured LiDAR).

4.3.1 Performance Requirements for Power Grid Inspection

Automated inspection systems for transmission networks should be evaluated against criteria that reflect operational priorities, not only generic benchmark metrics. In the context of power grid safety, the relative costs of false negatives and false positives are highly asymmetric. A missed detection of a hazard can escalate into power outages, failures, or wildfires, with potentially severe financial, environmental, and safety consequences. In contrast, a false alarm may lead to unnecessary dispatch or on-site verification, which carries an operational cost but rarely causes critical harm. For this reason, the evaluation of segmentation models in this domain places greater emphasis on recall than on precision. The F_β score, by definition, provides a way to control this emphasis. In this work, we set $\beta = 2$, doubling the weight of recall relative to precision. This choice reflects the operational requirement to minimise false negatives even if it results in a higher incidence of false positives.

While the Intersection-over-Union (IoU) metric remains the staple for comparing state-of-the-art models in 3D semantic segmentation and is used extensively in our TS40K benchmarking, the F_2 score is more aligned with deployment evaluation because it directly penalises missed detections in critical classes. In other words, a model with slightly lower IoU but significantly higher F_2 in the tower and power line classes may be preferable for inspection operations. Table 4.8 reports the F_2 scores of our best-performing model, Point Transformer V3, trained on the TS40K dataset. The model attains 87.37% for towers and 96.05% for power lines, which comfortably meets the requirements for reliable grid inspection set by Labelec and EDP of 85% performance in power grid elements.

Table 4.8: F_2 scores for the performance of Point Transformer V3 (PTV3) on the TS40K dataset. The model shows high sensitivity in the most critical power grid classes.

Class	**TS40K F_2 Score (%)**
Noise	63.85
Ground	70.28
Low Vegetation	51.89
Medium Vegetation	71.82
Tower	87.37
Power Line	96.05

4.3.2 System Design

The deployment of a machine learning assisted power grid inspection system requires more than high benchmark performance. To be viable in the field, the system must be computationally efficient, robust to the variability of real-world conditions, and seamlessly integrated with existing utility workflows. Our design addresses each of these requirements explicitly.

Processing efficiency and scalability. LiDAR point clouds from unmanned aerial vehicle surveys are inherently high-density, often containing millions of points per kilometre.

Direct processing of such datasets can exceed the memory and runtime limits of even high-end GPUs, particularly when using transformer-based architectures. To ensure scalability, our pipeline begins by partitioning the inspection corridor into contiguous spatial segments. Each segment is further reduced to a fixed point budget through Farthest Point Sampling (FPS), which preserves the global spatial distribution of points while discarding redundancies. The inference engine is optimised for GPU execution, allowing large-scale corridors to be processed in near real time.

Robustness under varying environmental conditions. Although LiDAR data are resilient to changes in lighting and to some extent weather, they remain susceptible to artifacts such as atmospheric scattering. Our system is enhanced by an uncertainty-aware evaluation stage: each point receives a softmax confidence score from the segmentation model, and low-confidence predictions are flagged for further manual review. By concentrating human oversight on ambiguous cases, this mechanism ensures that the automated system operates autonomously for the majority of high-confidence cases while still capturing edge cases that require expert judgment.

Operational integration. For effective adoption by utility companies, inspection outputs must integrate smoothly with existing asset management workflows, which are typically built around Geographic Information System (GIS) platforms. Our system exports annotated point clouds in standard `.las` format, ensuring compatibility with industry-standard GIS tools for visualization and analysis. This enables operators to incorporate inspection results directly into their infrastructure monitoring processes, streamlining maintenance planning and decision-making. Additionally, the system is designed for scalability and efficient processing. Large-scale datasets can be handled via cloud-based infrastructure for batch processing, while edge computing solutions enable fast, on-site inference for immediate inspections.

4.3.3 Operational Costs

This analysis focuses strictly on the costs that are directly affected by the transition from fully manual inspections to machine learning assisted inspections. We therefore exclude expenses that do not change across the two regimes, such as fixed platform acquisition or routine data storage that is common to both. The goal is to isolate the drivers of cost to quantify how they evolve when inspection work shifts from manual segmentation to ML assisted.

Scope and normalization. We model three cost components that are impacted by the shift to machine learning assisted inspections. The first is labour, measured as hours per kilometre of corridor that require technician time. The second and third components are the operational consequences of false positives and false negatives. False positives consume resources because they trigger unnecessary follow up actions. False negatives carry risk because they represent missed hazards that can escalate to outages or even wildfires. To enable a uniform and interpretable formulation, we express all costs in units

of labour cost per hour C_p. The cost of a false positive is written as $C_{FP} = \alpha C_p$ and the cost of a false negative as $C_{FN} = \beta C_p$, where α and β are multipliers that express the impact of each error relative to one hour of labour.

Cost model per kilometre. For a corridor length of x kilometres, with H labour hours per kilometre, a false positives per kilometre, and b false negatives per kilometre, the total cost is

$$C \;=\; xC_pH \;+\; a\,\alpha C_p \;+\; b\,\beta C_p. \tag{4.4}$$

Dividing by C_p yields a normalized cost,

$$\tilde{C} \;=\; \frac{C}{C_p} \;=\; xH \;+\; a\,\alpha \;+\; b\,\beta, \tag{4.5}$$

which makes comparisons independent of the absolute wage rate. Machine learning assisted inspections are economically favourable when

$$\tilde{C}_M \le \tilde{C}_T \iff x(H_T - H_M) \;\ge\; \alpha(a_M - a_T) \;+\; \beta(b_M - b_T). \tag{4.6}$$

The quantities (a_M, b_M) and (a_T, b_T) are the expected false positives and false negatives per kilometre for machine learning assisted and manual inspections respectively. The values H_M and H_T are the average labour hours per kilometre for machine learning assisted and manual inspections respectively. Thus, this inequality expresses a breakeven. The left hand side is the labour time saved by automation, and the right hand side is the change in error driven costs, weighted by their operational impact. The values (a_T, b_T) are set to zero for manual inspections, assuming perfect accuracy. The values (a_M, b_M) are estimated from the model performance on a validation set, while H_M is measured from the model inference time and the uncertainty aware review mechanism. The value H_T is estimated from historical operator records of manual inspection times.

Decomposing labour under automation. The labour term for machine learning assisted operation can be written as

$$H_M \;=\; H_{\text{io}} \;+\; H_{\text{inf}} \;+\; v\,S\,h_r, \tag{4.7}$$

where H_{io} is the time required for data loading and export per kilometre, H_{inf} is the model inference time per kilometre, S is the number of analysis segments per kilometre, h_r is the average manual review time per flagged segment, and v is the fraction of segments flagged for review by the uncertainty mechanism. The uncertainty-aware review mechanism flags segments for manual inspection based on model confidence scores. The parameter v quantifies the proportion of segments requiring manual review, which is determined by the uncertainty thresholds set during deployment. Stricter thresholds decrease false negatives (b_M) but increase v, leading to more manual reviews; relaxed thresholds reduce v but may increase b_M.

With operational values. Using results from the TS40K evaluation of the F_2 optimised Point Transformer V3, we consider $H_M = 0.00026$ hours per kilometre, $a_M = 17.6$ false

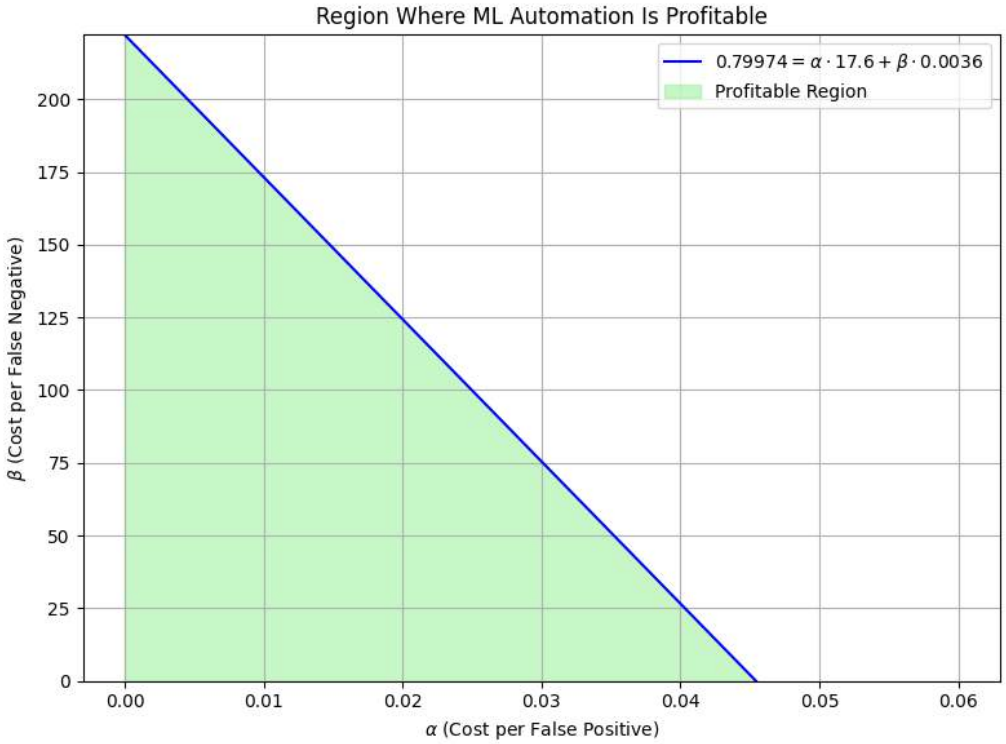


Figure 4.5: Region in the (α, β) cost space where machine learning assisted inspection is more cost effective than traditional manual inspection. The shaded area corresponds to cost configurations for which automation yields lower dimensionless operational cost $\tilde{C}$. The breakeven line is given by $0.79974 = 17.6\,\alpha + 0.0036\,\beta$ for the one kilometre case considered in this study.

positives per kilometre, and $b_M = 0.0036$ false negatives per kilometre. Based on operator records, manual inspections require on average $H_T = 0.8$ hours per kilometre. For a conservative baseline, we assume perfect manual accuracy with $a_T = 0$ and $b_T = 0$. For $x = 1$ kilometre, the inequality (4.6) becomes

$$0.79974 \;\geq\; 17.6\,\alpha \;+\; 0.0036\,\beta, \tag{4.8}$$

since $H_T - H_M = 0.8 - 0.00026 = 0.79974$. Equation (4.8) defines a region in the (α, β) space that characterises the region where machine learning assisted inspection is economically favourable. The corresponding region is illustrated in Fig. 4.5.

The coefficients in (4.8) show how much each type of error affects cost. In our case, false positives are more common, but false negatives (missed hazards) are much more expensive. If the penalty for missed hazards increases, the region where automation is cost-effective becomes smaller, so the system should focus on reducing false negatives. If the process is improved to reduce false positives, automation becomes more economically viable.

Estimating α and β from operations. Although we take a conservative approach by assuming perfect accuracy for traditional inspections ($a_T = 0$, $b_T = 0$), in practice, real values for false positives and false negatives can be retrieved from historical inspection

records. For instance, if records indicate any missed hazards, the value of β can be set to the average cost of a missed hazard relative to one hour of labour. Thus, power grid operators may assess the potential savings from machine learning assisted inspections by estimating the values of α and β based on their data. This allows them to determine whether the cost of implementing such a system is justified by the expected reduction in operational costs. On the other hand, the automatic risk assessment module can be disabled so that the system produces annotated point clouds without any risk assessment. This mode is particularly useful for generating annotated datasets for manual review, where human operators can inspect the results and make decisions based on the model's predictions. In this case, the values of α and β are not relevant, as the model is used to assist human operators rather than automate their work.

4.4 Conclusion and Future Work

This work has demonstrated the potential of 3D semantic segmentation to transform the way power grid inspections are conducted. By building on the TS40K dataset introduced in the previous chapter, we evaluated a range of state-of-the-art segmentation models under specific settings. Our benchmarking showed that transformer-based architectures, particularly Point Transformer V3, achieve both high IoU for model comparison and high F_2 scores for operational relevance, with recall exceeding 87% for towers and 96% for power lines.

We translated these benchmarking results into a deployable inspection tool for maintenance personnel, integrating GPU-accelerated segmentation, uncertainty-aware flagging, and optional automated risk assessment. A cost analysis framework, based on operational data from Labelec and EDP, quantifies the trade-offs between labour savings and error-driven costs in machine learning-assisted inspections.

While results are promising, further validation is needed. Robustness should be tested on additional datasets with varied sensors, vegetation, and infrastructure. The cost model could be refined by incorporating dynamic factors such as seasonal penalties and evolving error rates. Future directions for this work could focus on three areas: (1) field validation of the tool in live utility operations, (2) multimodal fusion of LiDAR with RGB or thermal imagery for improved detection, and (3) extending cost-benefit analysis to include predictive maintenance value. These steps will support the safe and cost-effective adoption of machine learning-assisted inspections, enhancing reliability and resilience in power systems.

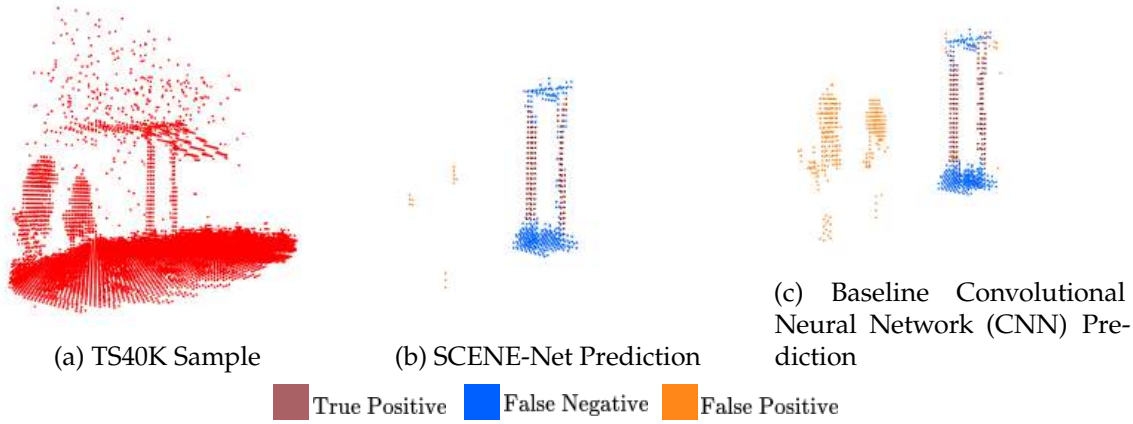
5

SCENE-NET: ADVANCING POLE SEMANTIC SEGMENTATION WITH GENEOS FOR POWER GRID INSPECTIONS

This chapter presents *SCENE-Net*, an intrinsically interpretable approach to semantic segmentation of supporting towers in power grid inspections. Our model encodes geometric inductive biases that can fully describe pole-like objects through Group Equivariant Non-Expansive Operators (GENEOs) and learns a small number of meaningful shape parameters that control their representation. The methodology builds upon the foundations established in our earlier work during the master's thesis [53], where the initial formulation of GENEO-based observers for power grid inspection was introduced. In the present chapter, we extend and refine these ideas, substantially improving performance and providing a broader experimental setting. We evaluate SCENE-Net in terms of interpretability, performance, and robustness, and contextualize our contributions within the broader field of 3D semantic segmentation. This work has been published as "*SCENE-Net: Geometric Induction for Interpretable and Low-Resource 3D Pole Detection with Group-Equivariant Non-Expansive Operators*" in the "*Computer Vision and Image Understanding*" (CVIU) journal [52]. The chapter is structured as follows: (1) we introduce the motivation and contributions of SCENE-Net, (2) we detail the methodology including GENEOs and their role as inductive biases, (3) we describe the experimental setup using the TS40K dataset, (4) we present results and analysis on interpretability, performance, and robustness, (5) we discuss the implications for industrial deployment, and (6) we conclude with a summary and future work directions.

5.1 Introduction

The TS40K dataset, introduced in Chapter 3, offers over 40,000 km of high-resolution rural electrical transmission systems for 3D scene understanding in inspection scenarios. While state-of-the-art point cloud segmentation models achieve an overall strong performance on this dataset, detecting *supporting towers* remains a crucial challenge that we must



(a) TS40K Sample (b) SCENE-Net Prediction (c) Baseline Convolutional Neural Network (CNN) Prediction

Figure 5.1: **Qualitative comparison of tower segmentation on TS40K**. (a) Input point cloud from the TS40K dataset voxelized and processed by an occupancy function. (b) SCENE-Net prediction: our intrinsically interpretable model, with only 11 geometric parameters, accurately segments the supporting tower while suppressing false positives in vegetation. (c) Baseline CNN prediction: a conventional CNN with 2190 parameters produces significant false positives in the vegetation region. The color code for semantic classes is shown below. Note: ground and power lines are mislabeled in the ground truth for this sample.

overcome in order to ensure reliable and efficient inspections. These structures exhibit large variability in design, height, and material, plus are often represented by a sparse set of points. As a result, even state-of-the-art models such as PointTransformer V3 (PTV3) [111], with more than 46 million parameters, fail to achieve consistently high Intersection-over-Union (IoU) scores for the tower class.

We argue that this difficulty stems from the absence of explicit geometric reasoning in conventional black-box models. While such networks can implicitly learn relevant structural patterns, doing so requires significant capacity and data, and still leaves gaps in generalization to rare or unusual tower configurations. In contrast, towers possess a set of recurring geometric traits that are invariant across the diversity found in TS40K: verticality, elongated cylindrical bodies, and intersections with power lines. To exploit these regularities, we introduce **SCENE-Net**, a white-box voxel-based 3D segmentation architecture that encodes *geometric inductive biases* directly into its convolutional kernels using *Group Equivariant Non-Expansive Operators* (GENEOs). Each GENEO acts as a functional observer tuned to a specific primitive relevant to the structure of towers, parameterized by a small set of meaningful attributes such as radius, height, and slope. These parameters are learned from data but retain an explicit geometric interpretation. By constraining the model to operate within a space of transformations that preserve these geometric patterns, SCENE-Net effectively detects the body of towers and reduces false positives from similar vertical structures like high vegetation with only 11 trainable parameters. Our approach contrasts with traditional black-box methods that rely on large numbers of parameters to learn complex features without explicit geometric reasoning. By leveraging GENEOs,

SCENE-Net achieves a balance between interpretability and performance, allowing for a robust tower segmentation while maintaining a low computational footprint.

In summary, our contributions are as follows:

1. A white-box architecture for 3D point clouds that utilizes geometric inductive biases as GENEOs and learns only a small number of meaningful parameters.

2. A formalization of SCENE-Net as a convex combination of equivariant observers, with a training objective and constraints that preserve interpretability.

3. An extensive evaluation on TS40K, SemanticKITTI, and ablation studies that highlights interpretability, robustness, parameter efficiency, and architectural design choices.

And we address the following research questions:

- **RQ1: What is the intrinsic meaning of the eleven learned shape parameters in SCENE-Net?** We address this by elucidating the semantic roles of each parameter and providing *post hoc* analyses of specific predictions that are independent from human interpretation (Section 5.5.1).

- **RQ2: How does the performance of SCENE-Net compare to other methods?** We answer this by benchmarking SCENE-Net against a baseline CNN and state-of-the-art approaches, including evaluation on the SemanticKITTI benchmark (Section 5.5.2).

- **RQ3: How effectively can SCENE-Net learn and represent the geometry of supporting towers?** We investigate this by showcasing the complex observer learned through the convex combination of simple 3D shapes and conducting ablation studies to analyze the impact of architectural choices (Sections 5.5.4 and 5.5.3).

5.2 Related Work

Explainability in 3D point clouds. Most existing work on explainability in 3D follows the paradigm established in image-based learning: train a complex black-box architecture and then apply *post hoc* techniques to generate explanations. Popular approaches include gradient-based saliency maps, local surrogate models such as LIME [81], and perturbation-based methods such as meaningful perturbations [31]. While these techniques can provide some interpretive value, they suffer from several well-known limitations. Explanations are often approximations and may not reflect the model's internal reasoning, they introduce additional computational overhead, and they remain dependent on human interpretation rather than providing a mechanistic understanding of the model's decision-making process. In the case of 3D semantic segmentation, these challenges are amplified by the lack of explainability tools in this domain and the difficulty of translating 2D techniques to spatially irregular point cloud data [57, 82]. An alternative is intrinsic interpretability,

where the architecture and parameters themselves encode domain knowledge, ensuring that each computational component has a clear semantic role. Examples include decision trees and linear models, which traditionally trade off complexity for transparency [82]. More recent methods, such as concept whitening [16] and interpretable CNNs [119], demonstrate that interpretability need not come at the expense of performance. In this work, we follow this path through SCENE-Net: a white-box architecture that leverages GENEOs to encode geometric inductive biases directly into convolutional kernels. Each GENEO is parameterized by interpretable geometric attributes, which are learned from data but retain an independent semantic meaning.

Power Line Segmentation from 3D Point Clouds. Power line inspection has traditionally relied on on-site maintenance teams and manned helicopters, which visually examine the grid using portable equipment or even the naked eye. These procedures are costly, labor-intensive, and expose workers to hazardous conditions. Consequently, process automation has become a priority for utility operators seeking to reduce costs, increase efficiency, and improve worker safety. To this end, unmanned aerial vehicles (UAVs) equipped with LiDAR sensors are now widely deployed to scan transmission corridors and produce 3D point cloud representations of the environment. Several approaches have been proposed to segment power lines directly from these data. For example, Ding et al. [24] combine simultaneous localization and mapping (SLAM) with multi-sensor data to patrol power grids, but rely on multi-view 2D raster reconstructions, which inevitably introduce projection artifacts and information loss. Guo et al. [39] project point clouds onto the xy-plane and apply clustering to segment lines, however, this approach neglects the complexity of irregular terrain and ground points, and fails to robustly handle incomplete line structures. Other methods exploit elevation statistics in combination with xy-plane projections, such as Tao et al. [94], who focus on extracting high-voltage conductors using fine-grained elevation information. While effective for delineating suspended wires, the methods above generally disregard supporting towers, which are equally important for inspection and maintenance. Supporting towers are not only themselves subject to deterioration and defect detection but also serve as reliable anchors for inferring the precise location of power lines. Furthermore, power grid safety assessment requires consideration of the broader environment, including vegetation encroachment, which cannot be addressed by methods limited to line extraction. In contrast, our approach directly leverages the full 3D scene to segment towers from the surrounding context.

Cylinder Detection in 3D Point Clouds. Detecting cylindrical structures in point clouds has long been studied in industrial and urban domains, particularly for pipelines, poles, or tree trunks. Classical methods rely on geometric fitting pipelines using RANSAC, curvature estimation, or projection-based simplifications. For example, Liu et al. [62] reduce 3D cylinder detection to circle estimation in 2D projections for structured environments, while Tran et al. [100] and Araújo et al. [2] introduce multi-stage refinement

procedures based on connectivity heuristics to improve robustness. More recently, Lu et al. [67] proposed decomposing scenes into axis-aligned slices, which facilitates efficient cylinder candidate tracking in dense scans. Although effective in controlled industrial settings, these approaches encounter significant limitations in outdoor rural inspection data. UAV LiDAR scans of power grids are typically noisy, cluttered, and highly variable in density, making assumptions of consistent geometry unreliable. Hand-tuned parameters or rigid heuristics often fail in the presence of vegetation, occlusion, or irregular terrain. Moreover, geometric fitting methods lack semantic awareness: they identify primitives purely by shape, without distinguishing between relevant (e.g., transmission towers) and irrelevant (e.g., tree trunks) structures. SCENE-Net addresses these shortcomings by encoding geometric inductive biases in the form of interpretable GENEOs that are optimized end-to-end. Rather than exhaustively fitting geometric primitives, SCENE-Net constrains its space to functions that reflect semantically meaningful patterns. Crucially, the GENEO framework supports extensions: new operators can be added to target other structural classes without redesigning the entire detection pipeline.

5.3 Methodology

5.3.1 SCENE-Net Architecture Overview

The proposed architecture builds directly on the concepts of *Group Equivariant Non-Expansive Operators* introduced in Section 2.3 and the role of *geometric inductive biases* discussed in Section 2.2. SCENE-Net embodies these principles into a voxel-based 3D semantic segmentation model tailored for supporting tower detection.

Step 1: voxelization. Let the input point cloud be $\mathcal{P} \in \mathbb{R}^{N\times(3+C)}$, where N is the number of points and $3 + C$ encodes coordinates plus optional per point features. SCENE-Net first applies a measurement function $\varphi : \mathbb{R}^3 \to \{0, 1\}$, which signals the presence of points on a regular voxel grid. Concretely, we choose binary occupancy as the admissible measurement: a voxel is active if it contains at least one point from $\mathcal{P}$ and inactive otherwise. This yields a tensor φ on a grid of size $X \times Y \times Z$.

Step 2: GENEO Layer as a geometric convolution. The measured input φ is processed by a set of m GENEOs,

$$\Gamma = \{\Gamma_i^{\vartheta_i}\}_{i=1}^{m},$$

each drawn from a parametric family with trainable shape parameters ϑ_i. Every operator acts on functions (i.e., the occupancy map φ) and is instantiated here as a 3D convolution with a kernel $g_i^{\vartheta_i}$ that encodes a specific geometric primitive (details in the next subsection). Formally, for a point $x \in \mathbb{R}^3$,

$$\Gamma_i^{\vartheta_i}(\varphi)(x) \;=\; \int_{\mathbb{R}^3} g_i^{\vartheta_i}(y)\, \varphi(x - y)\, dy, \tag{5.1}$$

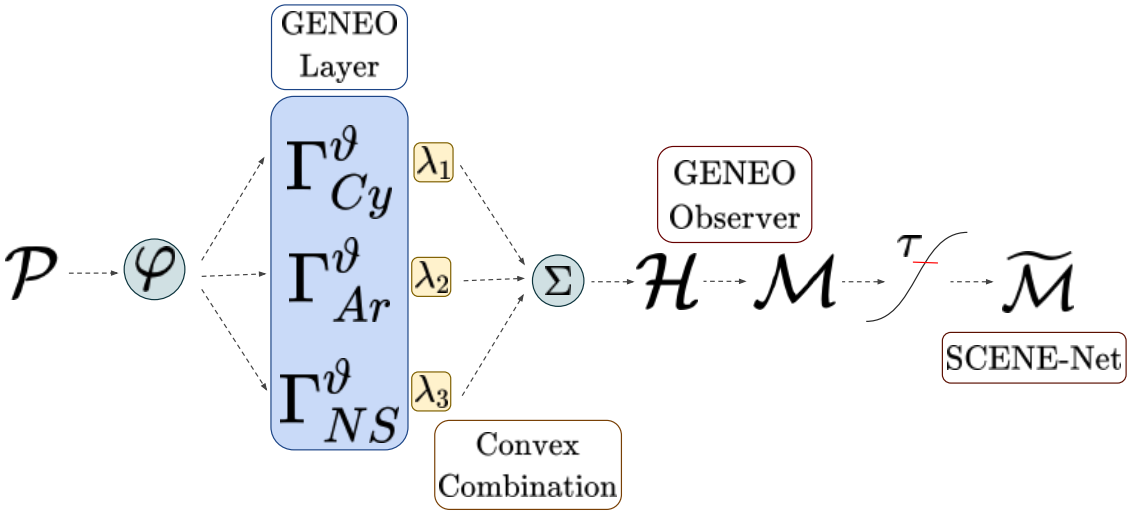


Figure 5.2: **Pipeline of SCENE-Net:** an input point cloud $\mathcal{P}$ is measured according to a function φ and voxelized. This representation is processed by a GENEO-layer, where each operator from a parametric family $\Gamma_i^{\vartheta_i}$ separately convolves the input. A GENEO observer $\mathcal{H}$ is obtained via a convex combination of the operators in the GENEO layer. The observer's response is then mapped to a probability $\mathcal{M}$ of belonging to a tower, followed by a thresholding step to produce binary voxel predictions. Thresholding is only applied after training is complete.

which is instantiated as a standard 3D correlation after sampling $g_i^{\vartheta_i}$ onto the same grid support. The kernels $g_i^{\vartheta_i}$ are designed to be equivariant to the relevant transformations of the target geometry (for example, translations and rotations around the vertical axis), and they satisfy non expansiveness as required by the GENEO framework.

Important training detail. During learning we *do not* optimize discrete kernel weights on the grid. Instead, backpropagation updates only the *shape parameters* ϑ_i that control the continuous kernels $g_i^{\vartheta_i}$. This preserves equivariance at each optimization step and keeps the parameterization compact and interpretable.

Step 3: convex observer. The responses of the m operators are aggregated by a convex combination that yields the *observer*

$$\mathcal{H}(x) \;=\; \sum_{i=1}^{m} \lambda_i \, \Gamma_i^{\vartheta_i}(\varphi)(x), \tag{5.2}$$

with non-negative coefficients λ_i also learned from data. Since convex combinations of GENEOs are GENEOs [8], $\mathcal{H}$ inherits equivariance from its components. The weights λ quantify the importance and contribution of each geometric prior to the fully discriminating towers from their environment. Intuitively, $\mathcal{H}(x)$ is a signed space whose magnitude reflects how strongly the local neighborhood at x matches the encoded tower geometry, and whose sign separates agreement from disagreement with the geometric priors. In other words, positive values of $\mathcal{H}$ indicate evidence for pole-like structures consistent with

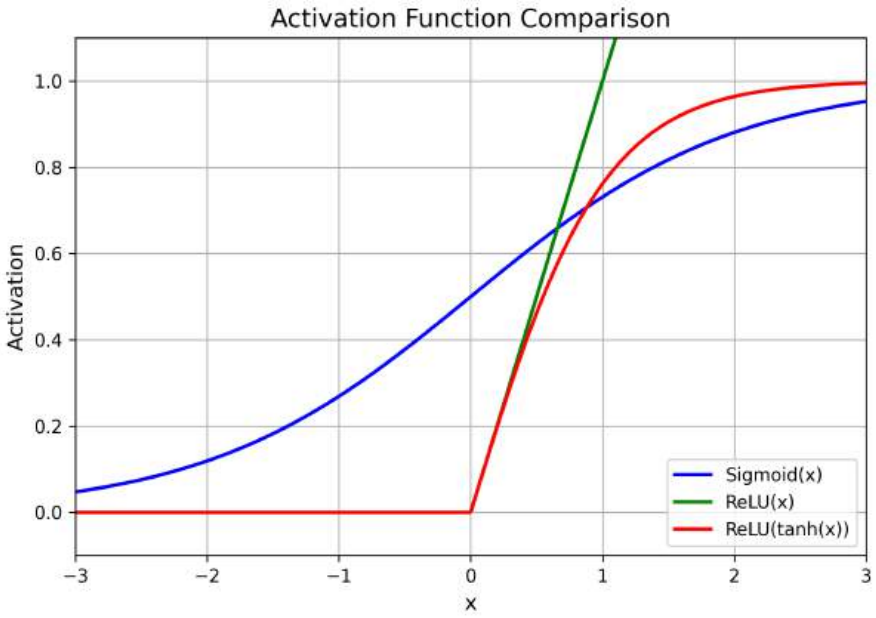


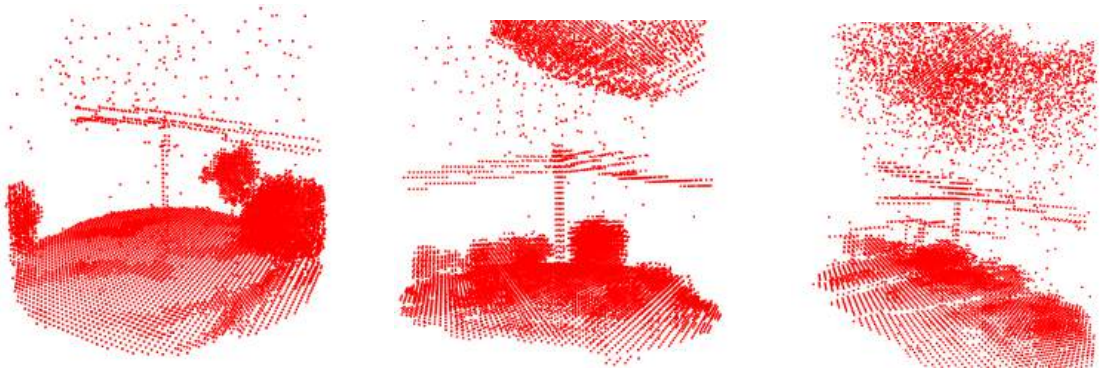
Figure 5.3: **Comparison of activation functions for probability mapping.** The tanh function normalizes the observer output to $[-1, 1]$, stabilizing the range and maintaining symmetry. The ReLU then zeroes out negative values, ensuring that only positively aligned geometric patterns contribute to the final prediction. This combination avoids assigning non-zero probabilities to negatively aligned regions and is bounded in $[0, 1]$.

the encoded priors, while negative values suppress structures that deviate from those priors, for example near spherical clumps typical of vegetation.

Step 4: probability mapping. The observer $\mathcal{H}$ in Eq. (5.2) is a convex combination of interpretable GENEOs, producing a real-valued field where positive values indicate pole-like structures and negative values indicate regions inconsistent with the encoded geometric priors. To convert this geometrically meaningful signal into a probability map $\mathcal{M}$ indicating tower presence, we first apply a tanh activation to normalize $\mathcal{H}(x)$ to $[-1, 1]$, stabilizing the range and maintaining symmetry between positive and negative activations. Next, we apply a ReLU to zero out negative components, ensuring that only positively aligned geometric patterns contribute to the final prediction. This yields a probability map bounded in $[0, 1]$:

$$\mathcal{M}(x) \;=\; \big(\tanh(\mathcal{H}(x))\big)_+, \qquad (t)_+ = \max\{0, t\}. \tag{5.3}$$

The use of a sigmoid activation is not appropriate here, as it would assign non-zero probabilities to negatively aligned regions, contradicting the semantics of $\mathcal{H}(x)$ as a geometric detector. Similarly, using ReLU alone without normalization would not guarantee bounded or calibrated outputs. Figure 5.3 provides a comparison of activation functions and illustrates why the chosen combination is best suited for this setting.



(a) (b) (c)

Figure 5.4: Input samples from TS40K illustrating the recurring geometric patterns that motivated each GENEO design: they all contain a vertical cylindrical shaft, an angled or conical section at the top, spherical clutter from vegetation, and power lines that intersect the tower in a catenary shape.

Step 5: thresholding for segmentation. A single threshold $\tau \in [0, 1]$ is selected on a validation split and applied only at inference time to convert probabilities into binary predictions:

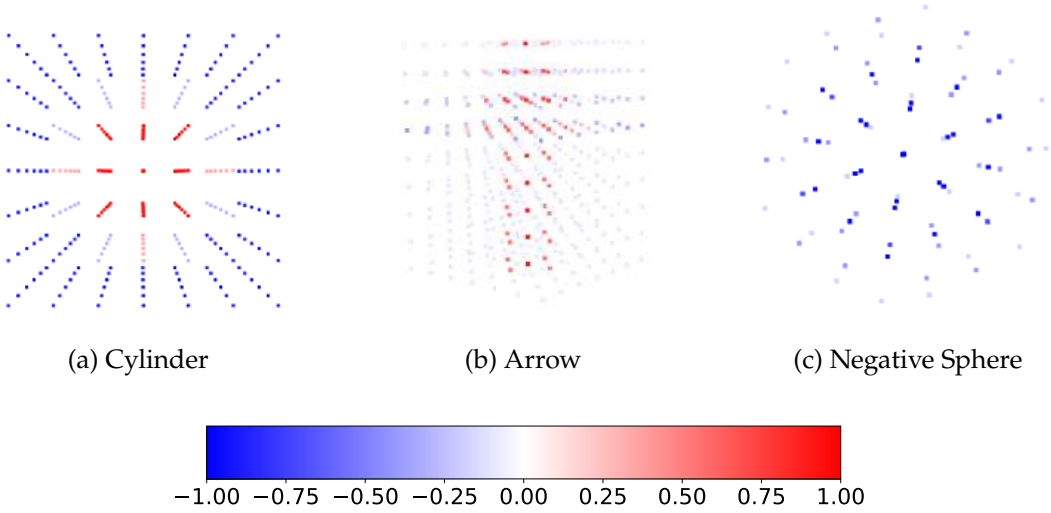
$$\widetilde{\mathcal{M}}(x, \tau) \;=\; \{\mathcal{M}(x)\} \geq \tau. \tag{5.4}$$

Thresholding is a post training decision rule that can be adapted to the desired precision recall trade off for inspection.

Discretization, resolution, and kernel support. Kernels $g_i^{\vartheta_i}$ are defined in continuous space and then sampled on the current voxel grid. This decouples the parameterization from the grid resolution and from the kernel size. In practice the number of trainable parameters is independent of the discrete kernel support chosen at inference. This property allows SCENE-Net to operate across different voxel resolutions without reparameterizing the model and to adjust kernel size to the desired voxel grid resolution while preserving the learned geometric behavior. In other words, the kernel size itself is a hyperparameter that can be tuned to boost performance while keeping the number of trainable parameters constant. This is a key advantage over traditional CNNs, where kernel size and resolution are tightly coupled to the number of parameters.

5.3.2 Encoding Geometric Inductive Biases with GENEOs

The GENEO operators used in SCENE-Net were engineered after a detailed examination of the tower geometries present in the TS40K dataset. In particular, we studied recurring structural elements in the input point clouds, focusing on their 3D shapes and symmetries. The supporting towers in TS40K display consistent geometric patterns that are preserved across their diverse structures: vertical cylindrical shafts forming the tower body, conical or angled sections at the top where power lines attach, and spherical clutter from nearby vegetation. To model these priors explicitly, we define three parametric GENEO kernels:



(a) Cylinder (b) Arrow (c) Negative Sphere

Figure 5.5: **GENEO kernels discretized on a voxel grid.** Each subfigure visualizes a different GENEO kernel used in SCENE-Net discretized in a voxel grid: (a) Cylinder, (b) Arrow, and (c) Negative Sphere. The color encodes the value of the kernel weights at each voxel, with the colorbar indicating the mapping from negative (suppressive) to positive (activating) values. The Cylinder kernel is sensitive to vertical, cylindrical structures typical of tower shafts; the Arrow kernel captures the regions where power lines attach to the tower; and the Negative Sphere kernel suppresses activations in approximately spherical regions, reducing false positives from vegetation. All kernels are parameterized continuously and then sampled on the voxel grid, ensuring resolution-agnostic behavior and interpretable geometric priors.

the *Cylinder*, *Arrow*, and *Negative Sphere*. Each sensitive to a specific geometric cue. The formulation of these kernels is tailored to maximise discriminative power for supporting towers while preserving equivariance to relevant transformation groups.

5.3.2.1 Cylinder GENEO

The **Cylinder** GENEO captures the vertical shafts that form the main body of supporting towers and is depicted in Figure 5.5a. It is equivariant with respect to the group of isometries in $\mathbb{R}^3$ that map upward-oriented vertical lines to themselves and preserve orientation, including translations in the xy-plane and rotations around the z-axis.

$$g_{\mathrm{Cy}}(x) = \exp\left(-\frac{1}{2\sigma^2}\left(\|z(x) - z(c)\|^2 - r^2\right)^2\right), \tag{5.5}$$

where z projects onto the xy-plane, c is the cylinder centre, r is the radius, and σ controls Gaussian smoothing. The parameter vector is $\vartheta_{\mathrm{Cy}} = [r, \sigma]$.

Acting on the voxelised measurement $\varphi \in \Phi$, the operator $\Gamma^{\vartheta}_{\mathrm{Cy}}$ is defined as a convolution:

$$\Gamma^{\vartheta}_{\mathrm{Cy}}(\varphi)(x) = \int_{\mathbb{R}^3} \tilde{g}_{\mathrm{Cy}}(y)\, \varphi(x - y)\, dy, \tag{5.6}$$

where $\tilde{g}_{\text{Cy}}$ is zero-sum normalised to stabilize the observer. This normalization is essential, as it leads to positive responses to occur near the cylinder's center, while negative values suppress shapes inconsistent with the model. In TS40K, this operator is essential for detecting the main tower body, as in sample (a) and (b) of Fig. 5.4, where distinct cylindrical shafts are visible. The non-expansiveness of $\Gamma^{\vartheta}_{\text{Cy}}$ is proven in Appendix A.1.

5.3.2.2 Arrow GENEO

The **Arrow** GENEO targets the conical or angled upper sections of towers where power lines attach, distinguishing them from other vertical structures such as trees. It combines a cylindrical base with a conical top, as illustrated in Fig. 5.5b.

$$g_{\text{Ar}}(x) = \begin{cases} \exp\left(-\frac{1}{2\sigma^2}\left(\|z(x) - z(c)\|^2 - r^2\right)^2\right), & v(x) < h, \\ \exp\left(-\frac{1}{2\sigma^2}\left(\|z(x) - z(c)\|^2 - \left(r_c \tan(\beta\pi)\right)^2\right)^2\right), & \text{otherwise}, \end{cases} \tag{5.7}$$

where r is the cylinder radius, r_c the cone base radius, h the height at which the cone starts, and β the cone inclination, and $v(x)$ denotes the height of point x in the reference space. The parameters are $\vartheta_{\text{Ar}} = [r, \sigma, h, r_c, \beta]$. In the TS40K sample (b) and (c) (Fig. 5.4), the conical upper section is a distinctive geometric feature with power lines intersecting it from multiple directions. Its non-expansiveness is proven in Appendix A.2.

5.3.2.3 Negative Sphere GENEO

The **Negative Sphere** GENEO is a suppressive operator designed to down-weight responses from approximately spherical structures, which are common in vegetation and can be mistaken for parts of towers if not explicitly penalized. This GENEO is illustrated in Fig. 5.5c.

$$g_{\text{NS}}(x) = -\omega \, \exp\left(-\frac{1}{2\sigma^2}\left(\|x - c\|^2 - r^2\right)^2\right), \tag{5.8}$$

with $\omega \in (0, 1]$ controlling suppression strength, and parameters $\vartheta_{\text{NS}} = [r, \sigma, \omega]$.

In TS40K samples (a) and (b) (Fig. 5.4), spherical vegetation masses surround or partially occlude towers. The Negative Sphere GENEO reduces false activations in these regions, stabilising the combined observer. Its non-expansiveness is shown in Appendix A.3.

5.3.3 Optimization and Constraints

The GENEO framework imposes a convex structure on the observer $\mathcal{H}$, which must be preserved during training. SCENE-Net's parameters consist of the combination weights $\lambda \in \mathbb{R}^m$ and the shape parameters $\vartheta \in \mathbb{R}^T$, where m is the number of GENEO kernels and T the total number of shape parameters across all operators. Both parameter sets are learned jointly to minimize a segmentation loss while respecting the convexity and non-negativity requirements of the framework.

Learning objective. We formalize the training problem as:

$$\begin{aligned} \underset{\lambda,\vartheta}{\text{minimise}} \quad & \mathbb{E}_{(X,y)\sim\mathcal{D}}\left[\mathcal{L}_{\text{seg}}(\lambda,\vartheta;X,y)\right] \\ \text{subject to} \quad & \lambda \in \Delta^{m-1}, \quad \vartheta \in \mathbb{R}_+^T, \end{aligned} \tag{5.9}$$

where $\mathcal{D}$ is the training dataset, Δ^{m-1} is the $(m-1)$-dimensional simplex, and $\mathbb{R}_+^T$ denotes the non-negative orthant for shape parameters. Non-negativity ensures that parameters retain their geometric meaning, e.g., a cylinder radius cannot be negative.

Loss function. The segmentation loss is defined as a *weighted mean squared error*:

$$\mathcal{L}_{\text{seg}}(\lambda,\vartheta;X,y) = f_w(\alpha,\epsilon;y)\cdot\left(\mathcal{M}(X)-y\right)^2, \tag{5.10}$$

where $\mathcal{M}$ is the tower probability map from Eq. (5.3), and $f_w(\alpha,\epsilon;y)$ is a weighting function to address class imbalance [90], with α controlling the weight scale and $\epsilon > 0$ avoiding zero weights.

Unlike standard binary cross-entropy (BCE), we employ mean squared error (MSE) as our loss because SCENE-Net frames segmentation as a voxel-wise regression task. In our setting, each voxel can be only partially occupied by a supporting tower, so we treat the target as a continuous-valued density rather than a strict binary label. This approach better matches the physical reality: towers are extended structures that may span several voxels and are not perfectly localized. For practical supervision, we discretize occupancy as:

$$y(v) = \begin{cases} 1.0 & \text{if voxel } v \text{ contains any tower point,} \\ 0.0 & \text{otherwise.} \end{cases} \tag{5.11}$$

Empirically, we found that MSE yields more stable training and improved final performance compared to BCE in this regression-based formulation.

Reparameterization of λ. To satisfy the simplex constraint without projecting the parameters to the simplex at each iteration, we reparameterise λ by setting:

$$\lambda_m = 1 - \sum_{i=1}^{m-1}\lambda_i, \tag{5.12}$$

reducing Eq. (5.9) to:

$$\begin{aligned} \underset{\lambda,\vartheta}{\text{minimise}} \quad & \mathbb{E}_{(X,y)\sim\mathcal{D}}\left[\mathcal{L}_{\text{seg}}(\lambda,\vartheta;X,y)\right] \\ \text{subject to} \quad & \lambda \in \mathbb{R}_+^m, \quad \vartheta \in \mathbb{R}_+^T. \end{aligned} \tag{5.13}$$

Soft regularization for non-negativity. Instead of enforcing non-negativity via hard constraints or projections, we introduce a differentiable penalty:

$$\Omega(\lambda,\vartheta) = \rho_l\sum_{i=1}^{m}h(\lambda_i) + \rho_t\sum_{i=1}^{m}\sum_{j=1}^{T_i}h(\vartheta_{ij}), \quad h(x)=\max(0,-x), \tag{5.14}$$

where ρ_l and ρ_t control the penalty strength for λ and ϑ, respectively. The final optimisation problem becomes:

$$\underset{\lambda,\vartheta}{\text{minimise}} \quad \mathbb{E}_{(X,y)\sim\mathcal{D}}\left[\mathcal{L}_{\text{seg}}(\lambda,\vartheta;X,y)\right] + \Omega(\lambda,\vartheta). \tag{5.15}$$

Why not softmax or exponential reparameterization? The convex combination in Eq. (5.2) requires non-negative weights λ, which could be enforced through standard approaches like softmax normalization. However, the softmax function creates undesirable side effects for our setting. The exponential mapping $\lambda_i = \frac{\exp(\theta_i)}{\sum_j \exp(\theta_j)}$ tends to produce highly skewed distributions where small parameter differences lead to one dominant component and near-zero contributions from others. This contradicts our goal of learning balanced combinations of geometric priors. Exponential reparameterizations $\lambda_i = \exp(\theta_i)$ avoid the normalization issue but still amplify parameter differences. Both approaches obscure the direct relationship between learned weights and their transparent meaning. Our soft penalty approach in Eq. (5.14) maintains this direct correspondence by treating λ as standard real-valued parameters while discouraging negative values through a differentiable penalty term. This preserves the interpretation where each λ_i quantifies the contribution of a specific geometric prior. Empirically, we observe convergence to non-negative values without explicit projections, while retaining the flexibility to apply additional regularization such as sparsity constraints.

Interpretability of parameters. The convexity of the observer ensures that each λ_i is a direct measure of the contribution of the i-th GENEO kernel to the final observer, while each ϑ_{ij} has a clear geometric meaning (e.g., cylinder radius, cone height, sphere suppression factor). This direct link between parameters and geometry is a key benefit of embedding inductive biases in the architecture.

5.4 Experimental Setup

5.4.1 Problem Setting: Tower Detection in TS40K

We evaluate SCENE-Net on the TS40K dataset [51], a large-scale benchmark of rural power grid environments introduced in Chapter 3. While TS40K was designed for comprehensive 3D semantic segmentation, our collaboration with EDP and Labelec defined a specific operational challenge: accurately detecting the (x, y) coordinates of supporting towers in vast rural point clouds with a lightweight model suitable for easy deployment. Identifying tower locations enables maintenance personnel to anchor the position of associated power lines and supporting infrastructure, which significantly reduces inspection times and does not require heavy 3D semantic segmentation models.

Although we adopt standard evaluation protocols from 3D semantic segmentation to ensure fair comparison with baseline and state-of-the-art methods, our main objective is to maximise *Precision* in tower detection. High Precision guarantees that predicted

tower coordinates correspond to actual supporting structures, avoiding false detections that could compromise subsequent inspection workflows. In this context, improving Precision directly translates to more reliable tower locations and safer decision-making during inspections.

5.4.2 Baselines and Evaluation Protocol

To contextualise SCENE-Net's performance, we compare it against a lightweight CNN baseline with a similar pipeline, as well as established state-of-the-art methods for 3D semantic segmentation. These include classical point-based architectures such as PointNet++ [79], convolutional approaches such as KPConv [96], and transformer-based models like Point Transformer V2 [110].

CNN Baseline. We design a convolutional neural network baseline that mirrors the architecture of SCENE-Net as closely as possible, in order to provide a fair and interpretable comparison. Both models operate on voxelized point clouds from the TS40K dataset, using only geometric occupancy information without colour or auxiliary features. The baseline CNN consists of a single 3D convolutional layer with three filters of size 9×5×5, followed by a non-linear transformation. For the CNN baseline, we apply the standard ReLU activation, whereas SCENE-Net applies ReLU(tanh(·)) to enforce bounded activations aligned with its probabilistic interpretation. The key distinction lies in the parameterization of the convolutional kernels. In the baseline CNN, kernels are unconstrained and fully learnable, resulting in a total of 2,190 trainable parameters. By contrast, SCENE-Net constrains each kernel to a parametric form grounded in geometric priors, specifically the GENEO kernels introduced in Section 5.3.2. This restriction reduces the number of trainable parameters to only 11, while preserving a direct semantic interpretation for each parameter. In principle, the unconstrained CNN has sufficient capacity to represent SCENE-Net within its parameter space, should gradient descent discover the corresponding kernel weights. This difference in kernel design also affects scalability. In CNNs, kernel size directly controls the number of parameters: increasing kernel resolution raises both parameter count and computational cost cubically. SCENE-Net, by contrast, instantiates kernel weights from continuous parametric functions, meaning its parameter count remains constant regardless of kernel size. The model is thus independent of kernel resolution, an advantage of particular relevance for large-scale 3D applications where memory efficiency and computational scalability are critical. We explore this property in detail in Section 5.5.4.

Evaluation Metrics. Performance is assessed using four metrics: First, **Precision** measures the proportion of correctly identified tower voxels among all voxels predicted as towers. In practice, this metric captures the reliability of predicted tower locations. Second, **Recall** quantifies the proportion of ground truth tower voxels that are successfully detected by the model. This metric indicates whether the model is able to consistently capture the

Table 5.1: Comparison between the CNN baseline and SCENE-Net. Both models share the same overall pipeline, but differ fundamentally in kernel definition and parameterization.

Property	**CNN Baseline**	**SCENE-Net**
Number of parameters	2,190	11
Layer depth	1 (single convolutional layer)	1 (GENEO-layer)
Number of kernels	3	3
Kernel size	$9 \times 5 \times 5$	$9 \times 5 \times 5$ (discretized from continuous functions)
Kernel definition	Unconstrained weights	Geometric priors (GENEOs)
Interpretability	×	✓
Resolution Agnostic	×	✓
Scalable design	×	✓

full extent of the infrastructure. However, due to the noisy nature of the supporting tower label (Section 3.6.1), where the ground beneath towers as well as the supporting structure for power lines are also annotated as towers, Recall may be affected by false positives in these areas. Third, **Intersection over Union (IoU)** evaluates the voxel-wise overlap between predictions and ground truth. IoU balances Precision and Recall, providing a single interpretable measure of segmentation quality. For the tower detection problem, IoU highlights whether the detected regions align spatially with the annotated structures. Finally, we introduce **Parameter Efficiency** to measure performance relative to model size. It is defined as

$$\text{Parameter Efficiency} = \frac{\text{Tower IoU}}{\log_{10}(\#\text{Parameters})}, \tag{5.16}$$

capturing how a model leverages its parameters to detect towers. This metric is particularly relevant for SCENE-Net, whose design philosophy prioritises compactness and interpretability.

5.5 Results and Analysis

5.5.1 RQ1: The Interpretability of SCENE-Net

The meaning of the learned shape parameters. To address the interpretability of SCENE-Net, we first analyse its eleven trainable parameters ϑ and λ after training. Each $\vartheta_i \in \vartheta$ encodes geometric attributes of a GENEO operator Γ_i, such as the radius of a cylinder or the inclination of a cone. The convex coefficients λ weigh the relative contribution of each operator in the final observer $\mathcal{H}$, exposing the balance between geometric components. For instance, in the trained model we find that the Negative Sphere GENEO Γ_{NS} receives a dominant weight of approximately 76% (see Figure 5.6). Conversely, the Arrow GENEO is assigned a smaller weight, yet its parameters encode the vertical elongation and angular intersection patterns that are characteristic of supporting towers. This division of roles across a small set of interpretable parameters illustrates the intrinsic transparency of

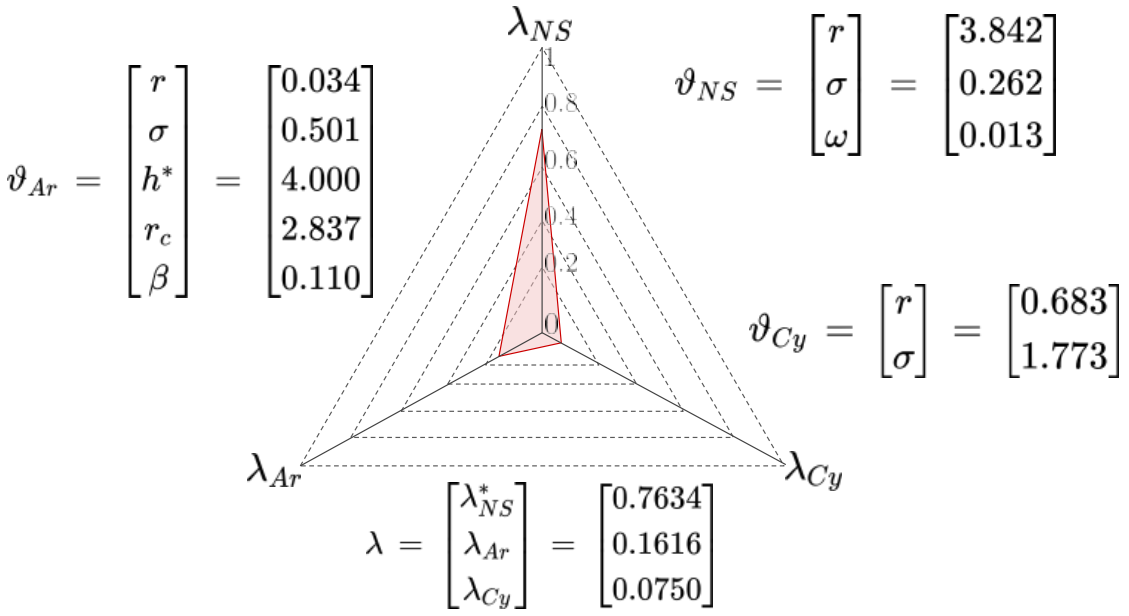


Figure 5.6: **Learned parameters of SCENE-Net.** The trainable parameters ϑ (shape parameters) and λ (convex weights) after training. Parameter h^* is not trainable, and λ^*_{NS} is defined as a function of the others, $\lambda^*_{NS} = 1 - \lambda_{Ar} - \lambda_{Cy}$.

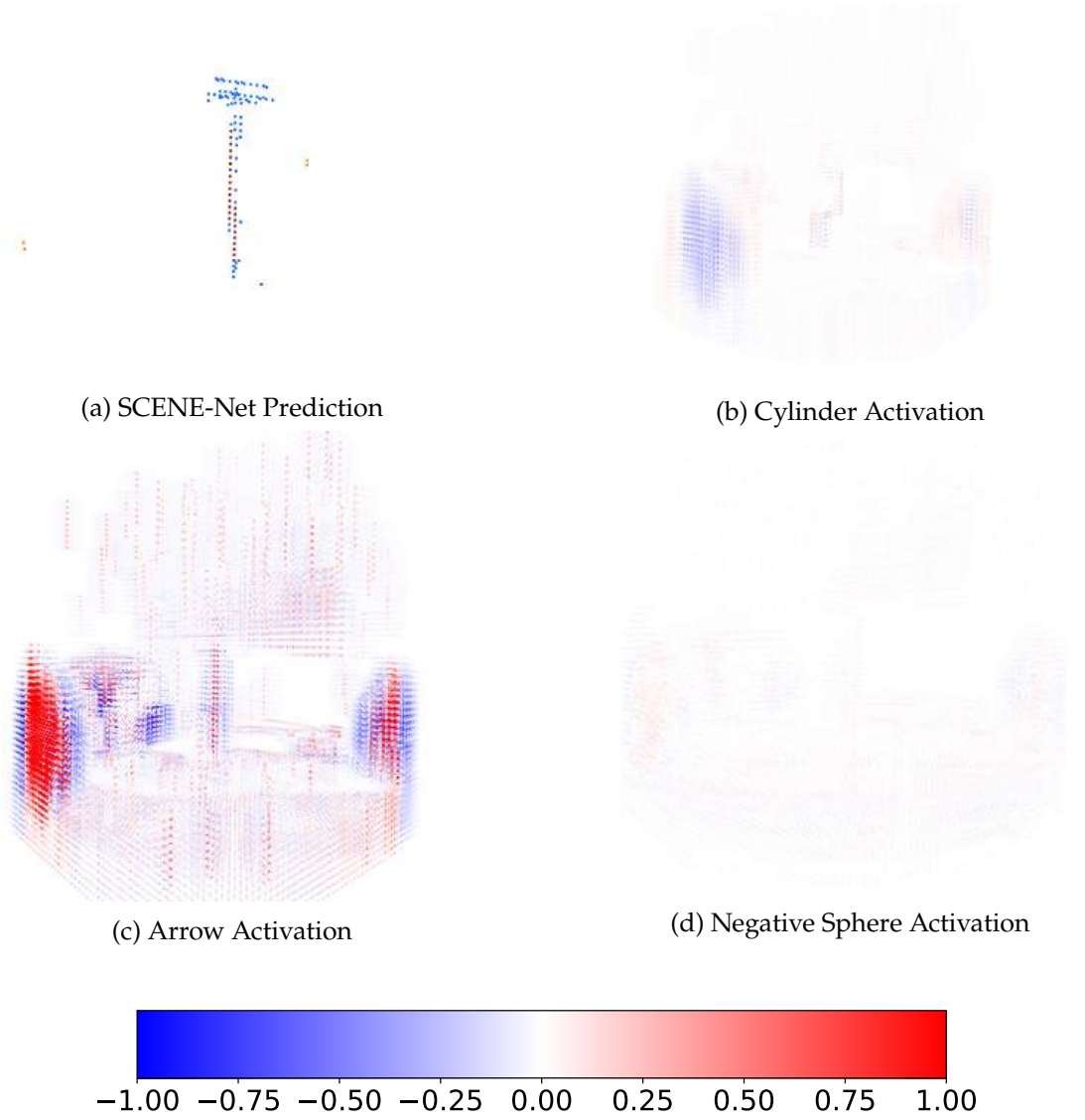
SCENE-Net: the model's behaviour can be directly traced back to the semantics of its shape parameters and their convex weights.

***Post hoc* interpretation for specific predictions.** Beyond parameter inspection, SCENE-Net's interpretability extends to specific predictions. By visualizing the activations of individual operators, we can associate their contributions with meaningful elements of the scene. As shown in Figure 5.7, the Arrow operator produces the strongest activation on tower-like regions, capturing their characteristic vertical and angular structure. Interestingly, the Cylinder provides a complementary activation, reinforcing verticality while reducing sensitivity to vegetation. The Negative Sphere presents a small negative factor ($\omega = 0.013$), thus it acts more like a stabilizing force between the other operators.

Overall, these visualizations show that SCENE-Net's behaviour can be traced directly to the interaction of its geometric operators. Rather than being a black-box, the model acts as a transparent observer, where each component contributes a distinct and interpretable role. By examining both the semantics of the learned parameters and their voxel-level activations, we converge on the same conclusion: SCENE-Net learns a compact yet expressive set of parameters with clear geometric meaning to detect supporting towers.

5.5.2 RQ2: The Performance of SCENE-Net

Baseline comparison. On TS40K, SCENE-Net achieves substantially higher Precision and Intersection over Union (IoU) than the CNN baseline, despite operating with three orders of magnitude fewer trainable parameters. Precision improves by 38%, IoU by 5%,



(a) SCENE-Net Prediction

(b) Cylinder Activation

(c) Arrow Activation

(d) Negative Sphere Activation

Figure 5.7: **Post hoc analysis of GENEO activations.** SCENE-Net predictions can be decomposed into the contributions of each GENEO. The Arrow activates strongly on tower structures, the Cylinder reinforces verticality and mitigates vegetation, and the Negative Sphere stabilizes our final observer.

while Recall decreases by 13% (Table 5.2). This trade-off reflects SCENE-Net's explicit bias: it focuses on geometry-consistent detections over broader but noisier coverage. Qualitative results (Figures 5.1 and 5.8) illustrate this effect. The CNN tends to classify most vertical structures as towers, which inflates Recall but also produces a high number of false positives. In contrast, SCENE-Net reliably segments the main body of towers and suppresses vegetation. From an application standpoint, SCENE-Net's conservative segmentation is beneficial: it reduces the risk of false positives in tower detection, which leads to more reliable inspection outcomes. Thus, despite the CNN's greater expressive power and parameter count, SCENE-Net achieves superior segmentation performance,

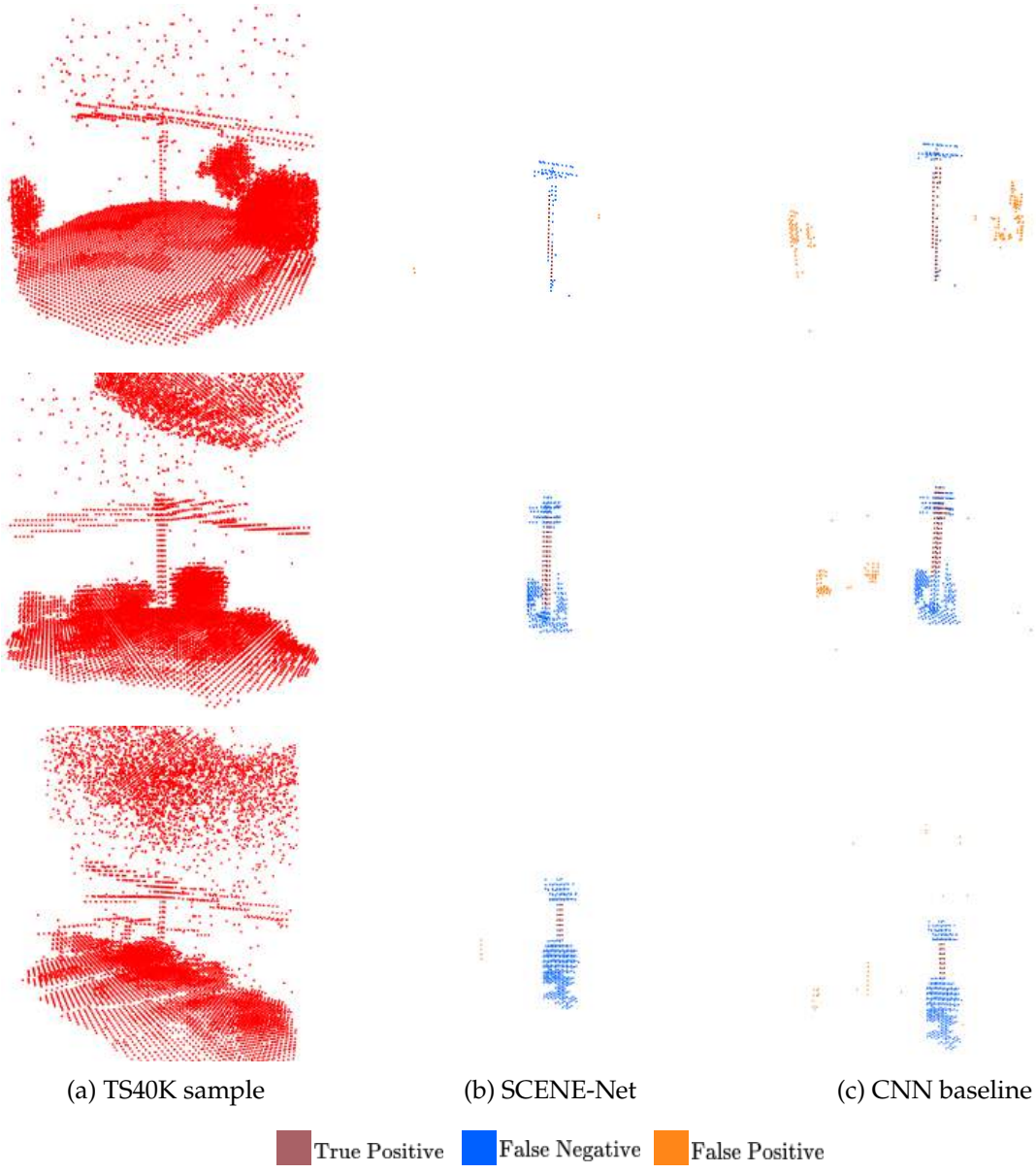
illustrating the strength of geometric inductive biases and the efficiency of our approach.

Table 5.2: 3D semantic segmentation performance on TS40K. Values are reported as mean ± standard deviation over three runs. SCENE-Net emphasizes Precision and IoU, which are most relevant for deriving accurate tower coordinates.

Method	Precision (%)	Recall (%)	IoU (%)
CNN Baseline	44 (± 7)	**26** (± 2)	53
SCENE-Net	**82** (± 8)	13 (± 5)	**58**

Comparison with state-of-the-art methods. As shown in Table 5.3, SCENE-Net achieves a parameter efficiency over an order of magnitude higher than any other model. Point-Transformer V3 yields the best absolute IoU (65.1%), but requires over four million times more parameters than SCENE-Net and with only a performance gain of 7.1%. This illustrates a key strength of our approach: SCENE-Net achieves competitive segmentation performance with a minimalist parameterization.

SCENE-Net on SemanticKITTI. To further evaluate generalization, we test SCENE-Net on SemanticKITTI [6], a challenging autonomous driving benchmark characterized by sparse, occluded point clouds with a rectangular field-of-view. Input point clouds are voxelized with a voxel size of 0.05 meters to accommodate for sparsity. As shown in Table 5.4 and the qualitative results in Figure 5.9, SCENE-Net achieves a pole IoU of 57.5% with only 11 trainable parameters. The strongest performer, TG-KD [41], attains 69.8% pole IoU but requires 2.78 million parameters and benefits from knowledge distillation from a large teacher model (Point Transformer V3 [111]). By contrast, SCENE-Net is trained from scratch without pre-training or distillation, which highlights the effectiveness of its geometric inductive priors. This comparison serves two purposes. First, it demonstrates that SCENE-Net's specialization for pole detection is effective beyond TS40K, adapting to a dataset with widely different statistics, including higher occlusion, lower density, and a richer variety of objects. Second, it highlights the efficiency of embedding geometric inductive biases: SCENE-Net delivers competitive results with five orders of magnitude fewer parameters than the strongest baselines. Although the GENEOs used here were designed with TS40K towers in mind, they remain effective in SemanticKITTI without modification. This suggests that as long as the target structures preserve consistent geometric traits, GENEO-based observers can transfer across domains. Nevertheless, additional gains could be achieved by refining the GENEO set to better capture the poles common in SemanticKITTI. Thanks to SCENE-Net's white-box design, such extensions can be made in a transparent, post hoc manner by inspecting model failures and incorporating new operators. Overall, these results illustrate that performance improvements in 3D semantic segmentation need not come from larger models. By leveraging geometric knowledge, SCENE-Net offers a lightweight and interpretable alternative, avoiding the diminishing returns of parameter scaling observed in state-of-the-art architectures.



(a) TS40K sample (b) SCENE-Net (c) CNN baseline

Figure 5.8: **Qualitative results on TS40K.** Each row shows the TS40K sample (left), SCENE-Net prediction (middle), and the CNN baseline (right). SCENE-Net focuses on tower bodies and suppresses vegetation, whereas the CNN frequently over-segments vegetation as towers. In the third row, SCENE-Net clearly identifies a second unlabeled tower, while the CNN marks both the second tower and nearby vegetation as towers.

5.5.3 RQ3: The Expressive Power of SCENE-Net as a Geometric Observer

We investigate this question by analyzing how SCENE-Net composes simple, interpretable GENEOs into a complex observer, and by conducting a study with respect to the voxel resolution and kernel size used to evaluate its performance. Additionally, we conduct an ablation study to assess the impact of different GENEO sets on performance. Together, these experiments highlight that SCENE-Net learns a semantically meaningful tower representation that is compact and resolution-agnostic.

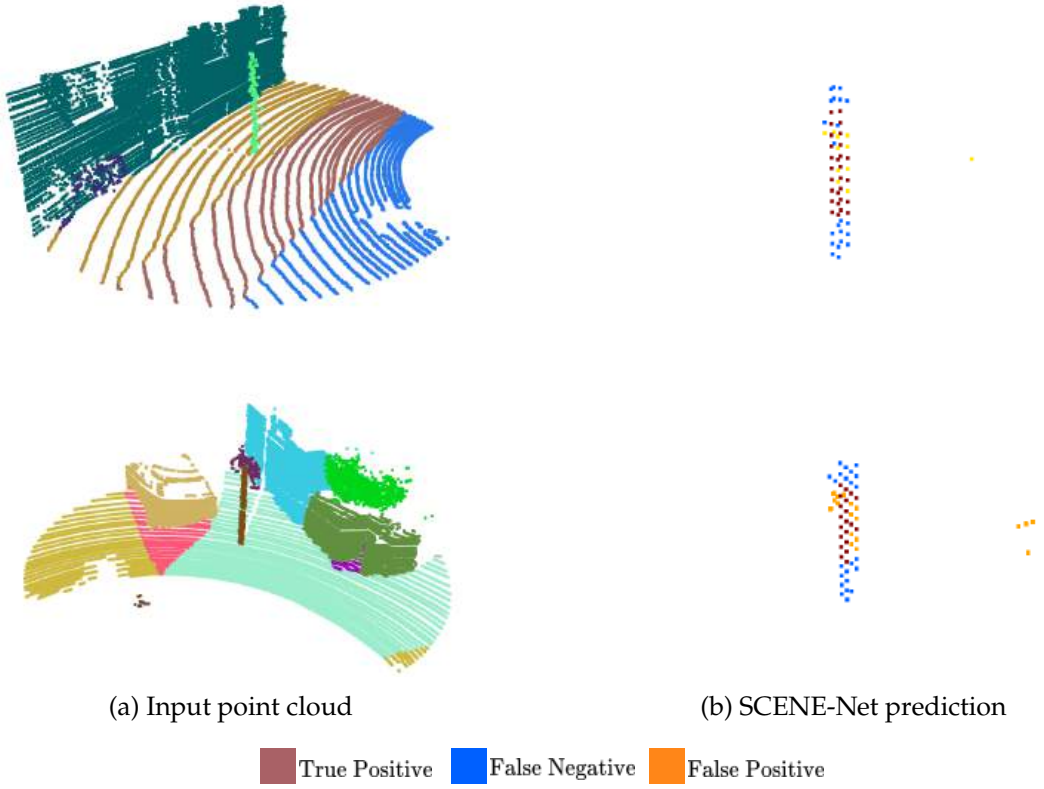
Table 5.3: Comparison of Tower IoU and parameter efficiency on TS40K. Methods are ordered by parameter efficiency. SCENE-Net achieves the highest efficiency by a wide margin.

Method	Tower IoU (%)	# Parameters (M)	Param. Efficiency
SCENE-Net (ours)	58.0	**1.1e-5**	**55.68**
PTV2 [110]	61.7	12.8	8.67
PTV3 [111]	**65.1**	46.2	8.48
PTV1 [121]	57.2	6.3	8.40
KPConv [96]	42.7	14.9	5.95
PointNet++ [79]	25.9	1.48	4.19
PointNet [78]	8.6	0.40	1.86
RandLA-Net [44]	0.0	1.24	0.00

Table 5.4: Semantic segmentation on SemanticKITTI: pole IoU, number of parameters, and parameter efficiency. SCENE-Net remains competitive despite being designed for TS40K.

Method	Pole IoU (%)	# Parameters (M)	Param. Efficiency
SCENE-Net (Ours)	57.5	**1.1e^{-5}**	**55.23**
TG-KD Student with KD [41]	**69.8**	2.78	10.83
TG-KD Student w/o KD [41]	63.8	2.78	9.90
PTV3 [111]	64.4	46.2	8.39
Cylinder3D [122]	62.4	53.0	8.08
SalsaNext [21]	54.3	6.7	7.95
JS3C-Net [114]	60.7	2.7	3.79
SparseConv [36]	57.9	2.7	3.61
SPVNAS [93]	64.3	12.5	3.46
RangeFormer [47]	66.4	24.3	3.09
RPVNet [112]	64.8	24.8	2.95
KPConv [96]	56.4	14.9	2.92
RandLA-Net [44]	44.2	1.24	2.70
UniSeg [63]	68.3	147.6	2.56
PointNet++ [79]	6.0	1.48	0.84

SCENE-Net as a complex observer built from simple shapes. SCENE-Net is a white-box model composed of interpretable geometric operators that are combined into a semantic observer. Each operator is designed to capture a fundamental geometric cue: the Cylinder encodes verticality, the Arrow emphasizes alignment with power lines, and the Negative Sphere acts as a stabilizer by suppressing spurious activations from vegetation or surrounding clutter. Through convex combination, these operators form a final observer $\mathcal{H}$ that is both expressive and compact. Figure 5.10 illustrates this process. On the left, the convex sum of the three GENEOs yields the observer $\mathcal{H}$, which activates strongly on pole-like structures while diminishing unrelated elements. On the right, $\mathcal{H}$ is transformed into a probability map $\mathcal{M}$ using the function $\mathrm{ReLU}(\tanh(\mathcal{H}(x)))$, which discards negative responses and rescales positive ones into the range $[0, 1]$. The result is a semantic density



(a) Input point cloud (b) SCENE-Net prediction

Figure 5.9: **Qualitative results on SemanticKITTI.** Left: input point clouds coloured by the original SemanticKITTI labels for easier visualization; right: SCENE-Net predictions. Despite diverse elements, sparsity, and occlusion, SCENE-Net consistently identifies pole-like structures.

estimate for tower likelihood across the scene. This transformation allows SCENE-Net to provide probabilistic, interpretable predictions directly linked to geometric priors. Although the current implementation targets pole-like towers, the modularity of SCENE-Net allows extension to other structures. Additional GENEOs could be designed to represent horizontal planes or bent forms, while preserving equivariance as long as they remain consistent with the underlying transformation groups. This highlights SCENE-Net's potential as a general framework for interpretable 3D scene understanding.

Voxel resolution and kernel size in SCENE-Net. A common limitation of voxel-based models is the dependence on grid resolution and kernel size, both of which heavily affect memory and runtime in CNNs. SCENE-Net overcomes this by defining GENEOs as continuous functions that are discretized only at inference. This means that its parameter count remains constant regardless of kernel size or voxel resolution, while its behavior generalizes across discretizations. We tested this property by applying a SCENE-Net model trained on 64^3 voxel grids with kernel size $(9, 5, 5)$ to higher-resolution grids such as 128^3, using larger kernels at inference. Figure 5.11 shows that SCENE-Net preserves

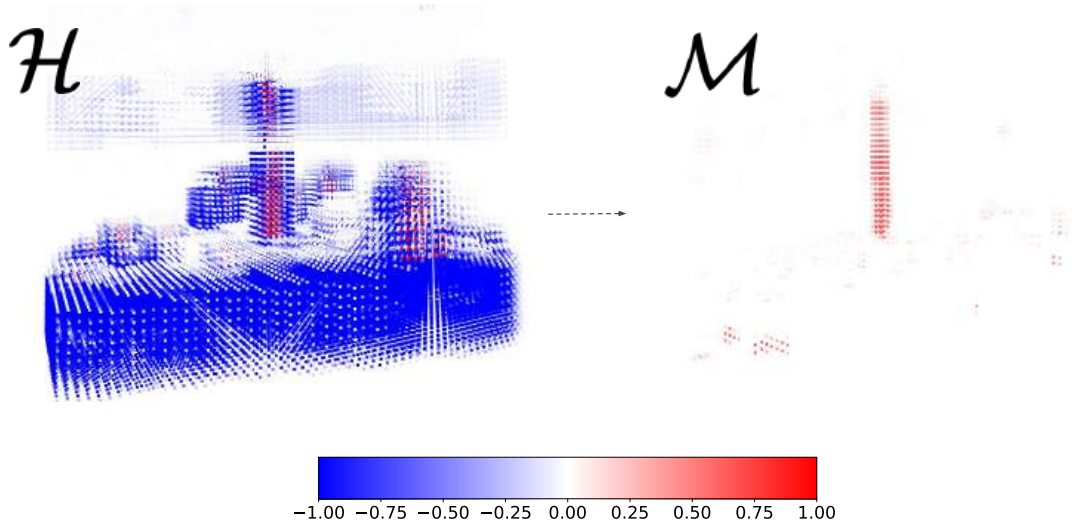


Figure 5.10: **Transformation of GENEO observers into a tower probability map.** The left side shows the convex sum of three GENEOs (Cylinder, Arrow, and Negative Sphere) forming the observer $\mathcal{H}$. The right side shows the transformation of $\mathcal{H}$ into a normalized tower probability map $\mathcal{M}$, which discards negative responses and rescales positive activations into $[0, 1]$.

tower localization at higher resolution without retraining, demonstrating its resolution-agnostic advantage enabled by its continuous formulation. To quantify this property, we evaluated the same trained model across multiple grid resolutions. Table 5.5 shows that performance peaks at the training resolution 64^3, with only modest degradation at coarser or finer grids. This reflects a natural trade-off between spatial detail and details captured by SCENE-Net across scales. In a complementary study, we tested different kernel sizes at inference. Results in Table 5.6 show that GENEO shape affects precision and recall trade-offs, with the best overall configuration being $(9, 5, 5)$. Crucially, these adjustments occur without retraining, confirming that SCENE-Net's continuous parameterization decouples kernel design from learning.

Table 5.5: Effect of voxel grid resolution on SCENE-Net performance in the TS40K validation set (inference only, model trained at 64^3 with kernel size $(5, 5, 5)$).

Grid Resolution	Precision (%)	Recall (%)
32^3	59.2	7.8
48^3	71.1	11.0
64^3	**79.7**	**11.6**
96^3	78.5	10.1
128^3	76.3	9.9

Ablation study on GENEOs. To further assess the expressive capacity of SCENE-Net, we conducted an ablation study varying the number and combination of Cylinder, Arrow, and Negative Sphere GENEOs. Each configuration was trained and evaluated on the TS40K

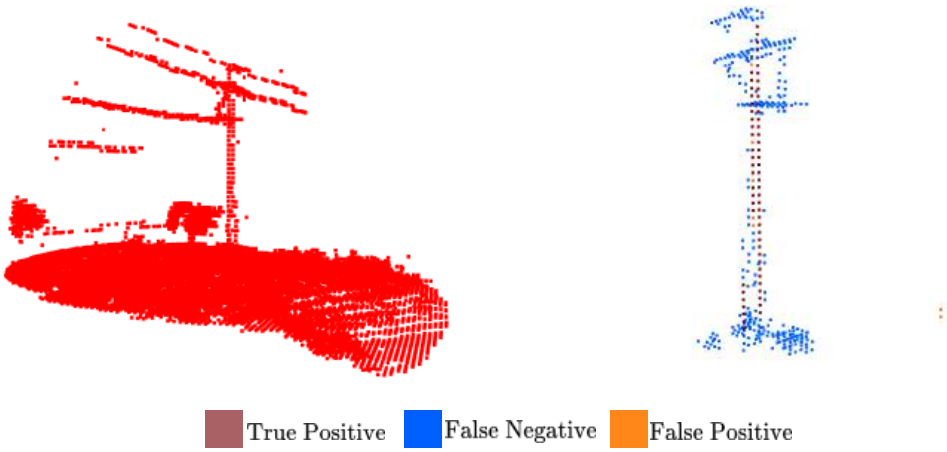


Figure 5.11: **Resolution-agnostic inference of SCENE-Net.** The model trained at 64^3 with kernel size $(9, 5, 5)$ generalizes to 128^3 using kernel $(12, 5, 5)$ without retraining, retaining accurate tower localization.

Table 5.6: Effect of GENEO kernel size on SCENE-Net performance in the TS40K validation set (inference only, model trained at 64^3 with kernel size $(5, 5, 5)$).

Kernel Size (z,x,y)	**Precision (%)**	**Recall (%)**
(3, 3, 3)	59.1	4.2
(5, 5, 5)	79.7	11.6
(7, 7, 7)	80.4	11.0
(9, 9, 9)	**82.6**	9.3
(9, 5, 5)	82.5	**11.9**
(12, 5, 5)	77.4	8.2
(5, 9, 9)	63.6	6.5

Table 5.7: Ablation study of SCENE-Net on the TS40K validation set. Columns 3–5 indicate the number of Cylinder, Arrow, and Negative Sphere GENEOs, respectively. We report Precision, Recall, and IoU to quantify the contribution of different operator combinations.

Model	# Cylinder	# Arrow	# NegSphere	Precision (%)	Recall (%)	IoU (%)
A	1	0	0	0	0	0
B	0	1	0	0	0	0
C	1	0	1	34	1	12
D	0	1	1	13	1	8
E (Ours)	**1**	**1**	**1**	**82**	**13**	**58**
F	2	2	2	56	16	53
G	3	3	3	37	22	56

validation set, with results reported in Table 5.7. The results reveal several important trends. First, models A and B, which use only a single GENEO type (Cylinder or Arrow), completely fail to detect towers. This highlights that no single primitive is sufficient on its own, for instance, cylinders capture verticality but confuse vegetation. Introducing the Negative Sphere improves results when paired with another operator (models C and D).

Table 5.8: Comparison of efficiency metrics between SCENE-Net and baselines. All models were trained on an NVIDIA RTX 4070 GPU with 8 GB RAM, for 100 epochs with a batch size of 16.

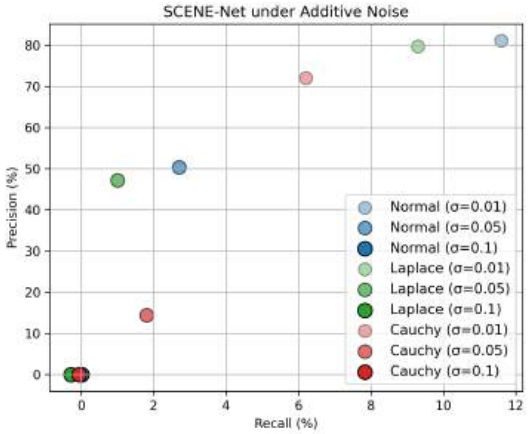
Method	Params (M)	Train time / epoch (m)	Time to converge (h)	Peak GPU mem (MB)	Inference latency (ms)
SCENE-Net (ours)	1.1×10^{-5}	2.3	**0.5**	**2128**	3.1
Baseline CNN	0.0022	**1.4**	1.1	2512	**2.9**
PointNet++	1.48	18.6	19.5	3201	12.5
KPConv	14.9	20.1	28.4	6589	20.3
Point Transformer v3	46.2	25.6	34.7	7454	35.7

These combinations partially suppress background clutter, though performance remains limited due to insufficient geometric coverage. The best performance arises from model E, the canonical SCENE-Net configuration, which balances all three GENEOs in a single convex combination. This setup achieves a strong trade-off between precision and recall, confirming that the three priors are complementary. Interestingly, adding more instances of each GENEO (models F and G) does not yield improvements. Instead, performance declines due to redundancy and dilution of the convex weights across multiple kernels. This suggests that SCENE-Net does not benefit from replicating operators, but from diverse geometric priors.

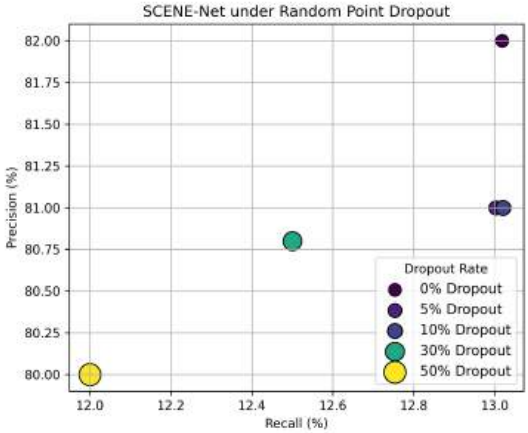
Efficiency Analysis We conducted a comprehensive assessment of SCENE-Net's computational efficiency and learning dynamics relative to key baselines, using a single NVIDIA RTX 4070 GPU (8 GB RAM). All models were trained for 100 epochs with a batch size of 16. For SCENE-Net and the CNN baseline, input point clouds were voxelized into 64^3 grids (approximately 1 MB per sample, 16 MB per batch), while other baselines processed raw point clouds of 10,000 points (about 0.12 MB per sample, 2 MB per batch). Table 5.8 summarizes the results: SCENE-Net achieves remarkable parameter efficiency, requiring only 11 trainable weights, compared to over 2,000 for the CNN baseline and millions for PointNet++, KPConv, and Point Transformer v3. This compactness leads to reduced GPU memory usage during training (peak 2.1 GB for SCENE-Net, 2.5 GB for CNN, and up to 7.4 GB for transformer models) and faster inference (3.1 ms per sample, versus 12–36 ms for state-of-the-art methods). Although SCENE-Net incurs a slightly longer per-epoch training time due to kernel recomputation (2.3 minutes vs. 1.4 minutes for CNN), it converges much faster, reaching stable performance within 5 epochs, compared to 20 for the CNN and near the end of training for larger models. These results demonstrate that SCENE-Net is highly efficient in terms of parameters, memory, and training speed, making it well-suited for deployment in resource-constrained environments and scenarios with limited data.

5.5.4 Robustness of SCENE-Net

Robustness is an essential requirement in 3D scene understanding tasks, particularly in real-world deployments where LiDAR artifacts, sparse sampling, and noisy annotations are prevalent. In this section, we assess SCENE-Net's resilience to three common sources



(a) Precision vs. Recall under additive noise. Each color represents a different noise distribution, and the marker intensity corresponds to noise scale. See Table 5.9.



(b) Precision vs. Recall under random point dropout. Color intensity reflects dropout probability. See Table 5.10.

Figure 5.12: Performance of SCENE-Net under different perturbations: additive noise (left) and random point dropout (right).

of uncertainty: (1) geometric noise in point clouds, (2) partial input due to sparsity or occlusion, and (3) label noise in ground truth annotations.

Geometric Noise. To simulate sensor noise and structural irregularities, we injected test-time additive noise into the spatial coordinates (x, y, z) of the input point clouds and evaluated the already-trained SCENE-Net. Three distributions were tested: Normal, Laplace, and Cauchy, each with increasing variance σ. Results in Table 5.9 and Figure 5.12a show that SCENE-Net maintains high precision under mild perturbations ($\sigma = 0.01$), with only modest reductions in recall. Performance degrades sharply at higher noise intensities ($\sigma = 0.1$ or Cauchy noise), where tower geometry becomes unrecoverable. In such cases, SCENE-Net tends to abstain from prediction altogether rather than producing false positives. This conservative behavior is desirable in safety-critical inspection, as it avoids spurious detections that could mislead operators.

Point Dropout and Occlusion. We further evaluated robustness under point sparsity by randomly removing points with dropout probabilities between 0% and 50%, simulating occlusion and sensor failure. As shown in Table 5.10 and Figure 5.12b, SCENE-Net remains stable even at 50% dropout, with negligible losses in precision and recall. This resilience arises from two properties of the design: (i) supporting towers account for less than 1% of all points in TS40K and are unlikely to be entirely removed, and (ii) binary voxelization treats any non-empty voxel as active, preserving the global tower structure even when local points are missing.

Table 5.9: SCENE-Net performance under different noise conditions. Coordinates are normalized to $[0, 1]$ prior to noise injection.

Noise Type	Precision (%)	Recall (%)
No noise (base model)	82.0	13.0
Normal (0, 0.01)	81.1	11.6
Normal (0, 0.05)	50.4	2.7
Normal (0, 0.1)	0.0	0.0
Laplace (0, 0.01)	79.7	9.3
Laplace (0, 0.05)	47.2	1.0
Laplace (0, 0.1)	0.0	0.0
Cauchy (0, 0.01)	72.1	6.2
Cauchy (0, 0.05)	14.4	1.8
Cauchy (0, 0.1)	0.0	0.0

Table 5.10: SCENE-Net performance under random point dropout. Each dropout rate was applied at test time.

Dropout Probability	Precision (%)	Recall (%)
0%	82.0	13.0
5%	81.0	13.0
10%	81.0	13.0
30%	80.8	12.5
50%	80.0	12.0

Label Noise. Finally, annotation noise remains a dominant challenge in supervised 3D learning. In TS40K, nearly half of the voxels labeled as "supporting tower" correspond to ground or structures suspending power lines due to inspection-oriented annotations [51]. This noise produces a mismatch between the geometric priors of towers and their annotated voxels. Despite this imperfect supervision, SCENE-Net consistently activates only on tower-like geometry and suppresses mislabeled surroundings. Figure 5.13 illustrates this effect: although the ground patch is labeled as "tower", the model highlights only the vertical structure. This explains the lower recall but consistently high precision observed in Table 5.2, and it demonstrates that SCENE-Net favors conservative but trustworthy predictions over overfitting to noisy supervision.

Overall, SCENE-Net demonstrates robustness to geometric noise, point dropout, and label noise. Its resilience is a direct consequence of embedding parametric GENEOs, which serve as intrinsic regularizers. By constraining the hypothesis space to geometrically meaningful functions, SCENE-Net avoids spurious correlations and maintains stable predictions under perturbation. This robustness is an architectural consequence of designing observers around interpretable geometric inductive biases.

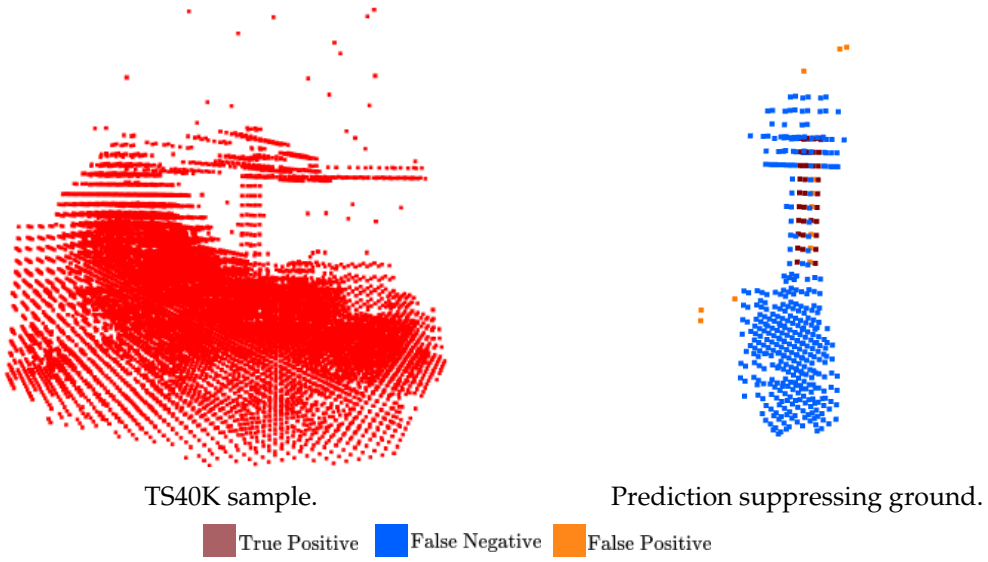


Figure 5.13: **SCENE-Net is robust to label noise.** Left: TS40K sample. Right: SCENE-Net prediction against the ground truth. Our model correctly detects only the tower structure and suppresses the mislabeled ground region.

5.6 Conclusions and Future Work

In this chapter, we introduced SCENE-Net, a white-box architecture for 3D semantic segmentation grounded on Group Equivariant Non-Expansive Operators (GENEOs). By embedding geometric inductive biases directly into its kernels, SCENE-Net demonstrates that interpretability and efficiency need not come at the expense of predictive performance. Unlike conventional black-box networks, SCENE-Net restricts its search space to geometrically meaningful operators, thereby enhancing generalization, robustness, and transparency.

Through extensive evaluation on the TS40K dataset, we showed that SCENE-Net achieves competitive segmentation performance despite operating with only 11 parameters. Compared to a CNN baseline with orders of magnitude more parameters, SCENE-Net demonstrated superior Precision and IoU. Benchmarking against state-of-the-art methods on TS40K and SemanticKITTI further revealed that SCENE-Net offers unmatched parameter efficiency, while qualitative results highlight its ability to capture pole-like objects under clutter, noise, and occlusion.

From a deployment perspective, SCENE-Net offers three main advantages. First, its low computational footprint makes it feasible for on-device inference in UAVs or edge devices, without the need for cloud computing. Second, its interpretable GENEO operators provide a level of transparency that allows engineers to audit and debug the model's behavior, a critical property in inspection pipelines. Third, its ability to transfer across datasets such as SemanticKITTI demonstrates that its domain of application is not limited to a specific dataset or environment.

Nevertheless, some limitations remain. Designing GENEOs requires manual effort

and domain knowledge, which constrains flexibility compared to purely data-driven approaches. Additionally, the current implementation is tailored to binary segmentation of tower-like structures, leaving open questions regarding scalability to multi-class tasks and more complex scenes. Finally, although interpretable, SCENE-Net may not always match the raw performance of large black-box models, which could limit its adoption in real-life applications. Looking forward, several promising research directions emerge:

- **Extending to multi-class segmentation.** Future work should explore the design of GENEO observers with diverse geometric primitives (e.g., planes, ellipsoids, or composite structures) to enable richer scene understanding beyond towers. Composing multiple observers into a multi-class pipeline could preserve interpretability while broadening applicability.
- **Building a library of GENEOs.** A standardized collection of reusable GENEO operators could serve as a geometric foundation for a wide variety of 3D tasks. Such a library would reduce the need for manual design in each application and foster reproducibility across domains.
- **Hybrid pipelines with black-box models.** SCENE-Net can act as a feature extractor, providing interpretable geometric knowledge to complement deep networks such as transformers or sparse convolutions. This hybrid approach could combine the transparency of GENEOs with the expressive power of black-box models, pushing the state of the art.
- **Industrial deployment and validation.** Evaluating SCENE-Net in real-world UAV inspection workflows, including integration with real-time processing pipelines, uncertainty estimation, and human-in-the-loop systems, is a critical next step for adoption by utility operators.

We explore the first three directions in Chapters 6 and 7. In conclusion, SCENE-Net demonstrates that embedding geometric inductive biases through GENEOs yields models that are not only efficient and robust but also interpretable and industrially relevant. While this work establishes a foundation for interpretable 3D semantic segmentation, future research promises to expand its reach, blending human-understandable geometric reasoning with the power of modern deep learning.

6

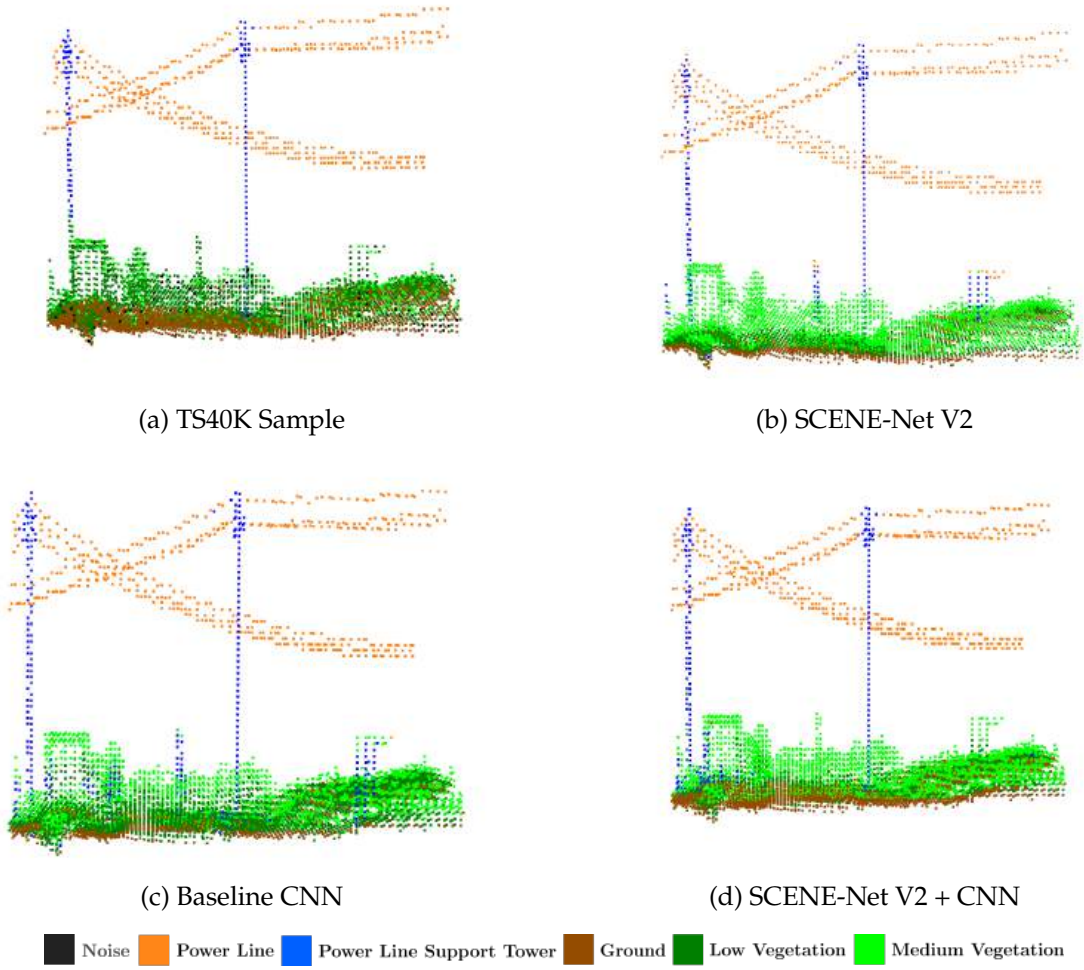
SCENE-NET V2: INTERPRETABLE MULTICLASS 3D SCENE UNDERSTANDING WITH GEOMETRIC INDUCTIVE BIASES

This chapter presents *SCENE-Net V2*, the first gray-box model for multiclass 3D semantic segmentation that leverages Group Equivariant Non-Expansive Operators (GENEOs) to encode geometric cues as interpretable inductive biases. The model builds upon the foundations introduced in *SCENE-Net* (Chapter 5), extending the methodology from pole-like object detection to a wider variety of 3D elements in the TS40K dataset and demonstrating its applicability to complex, multiclass scenarios. We introduce novel GENEO kernels that encode general geometric priors, such as the ellipsoid, enabling the model to leverage shape information more effectively across different object classes. This work has been published as "*SCENE-Net V2: Interpretable Multiclass 3D Scene Understanding with Geometric Priors*" in the *Geometry-grounded Representation Learning and Generative Modeling (GRaM) Workshop* at the 41st International Conference on ML (ICML 2024).

The chapter is structured as follows: (1) we introduce the motivation and contributions of SCENE-Net V2; (2) we detail the methodology, including the GENEO framework, novel geometric kernels, and architectural design; (3) we describe the experimental setup on the TS40K dataset and present results alongside ablation studies and interpretability analyses; (4) we conclude with a discussion of findings, limitations, and directions for future research.

6.1 Introduction

Recent progress in 3D scene understanding has largely focused on scaling model capacity and dataset size to achieve higher performance. For example, Point Transformer V3 [111] more than tripled its parameters compared to Point Transformer V2 [110], while other benchmarks such as $(AF)^2$-S3Net [17], 2DPASS [115] and UniSeg [63] rely on multi-channel fusion strategies that significantly increase computational and memory requirements. Although effective, these approaches often disregard the intrinsic geometric information of



(a) TS40K Sample

(b) SCENE-Net V2

(c) Baseline CNN

(d) SCENE-Net V2 + CNN

Figure 6.1: **3D semantic segmentation of the TS40K dataset**. For the sample in (a), SCENE-Net V2 (b) correctly detects the supporting tower while the baseline CNN (c) confuses medium vegetation with tower structures. By incorporating SCENE-Net V2 as a geometric feature extractor in the same CNN architecture (d), segmentation performance is substantially improved. This hybrid model achieves better delineation of grid components by adding only 540 interpretable parameters, highlighting the efficiency and transferability of GENEO-based priors.

point clouds, which can be essential for accurate and robust scene understanding. Furthermore, many real-world applications, including autonomous driving and infrastructure monitoring, require models that are not only accurate, but also lightweight, transparent, and ethically deployable [25, 38, 61].

In Chapter 5, SCENE-Net introduced a different paradigm: instead of relying on opaque learned kernels, we employed Group Equivariant Non-Expansive Operators (GENEOs) [8, 13] to embed geometric cues directly into the model architecture. With only 11 interpretable parameters, SCENE-Net successfully detected pole-like structures in multiple datasets, demonstrating the feasibility of white-box 3D scene understanding. However, this model remained limited in scope: its reliance on a small set of hand-crafted GENEOs restricted its application to single-class detection tasks, and the strictly white-box

formulation constrained its flexibility when scaling to complex, multiclass environments.

To address these limitations, we introduce **SCENE-Net V2**, the first gray-box model for multiclass 3D semantic segmentation. This new version enhances the original design along three key dimensions. First, we expand the set of GENEO kernels by introducing novel geometric priors (disk, cone, ellipsoid) that capture a broader range of 3D structures beyond pole-like elements. Second, we design a more expressive architecture capable of handling multiclass segmentation tasks, bridging the gap between interpretability and scalability. Finally, we demonstrate how GENEO-based operators can be combined with standard black-box classifiers to form hybrid models that inherit interpretability from the GENEOs while achieving improved performance with minimal additional parameters. Figure 6.1 illustrates these advantages in practice.

In summary, our contributions are threefold:

- We propose SCENE-Net V2, the first gray-box model for multiclass 3D semantic segmentation, combining interpretable GENEO-based operators with standard classifiers.
- We introduce novel GENEO kernels with general geometric priors, enabling the detection of diverse 3D elements.
- We study the use of SCENE-Net V2 as a geometric feature extraction module for black-box models, showing that it substantially improves performance at negligible computational cost.

6.2 Methodology

6.2.1 Architecture Overview

SCENE-Net V2 is a gray-box architecture for multiclass 3D semantic segmentation that combines an interpretable feature extractor based on GENEOs with a lightweight classifier. Compared to SCENE-Net, which employed three task specific GENEOs and a single white-box layer for pole detection, SCENE-Net V2 broadens the set of geometric inductive biases by introducing additional families of GENEO kernels and by composing them into multiple observers through convex combinations. This design preserves intrinsic geometric interpretability during feature extraction while enabling the SCENE-Net architecture to evolve to multiclass settings through a simple classifier head.

Let $\mathcal{P} \in \mathbb{R}^{N\times(3+C)}$ denote a point cloud with spatial coordinates and optional point wise channels. The first step maps $\mathcal{P}$ to a functional representation via a measurement $\varphi\colon \mathbb{R}^3 \to \{0,1\}$ that marks voxel occupancy. This functional view is similar to SCENE-Net's and is central to GENEO theory, since operators act on functions. Next, the GENEO Layer instantiates a family of parametric operators $\Gamma = \{\Gamma_j^{\vartheta_j}\}_{j=1}^m$. Each operator is implemented as a convolution on the voxelized domain, where the convolution kernel is not freely learned but is a parametric function discretized on a grid generated from interpretable shape

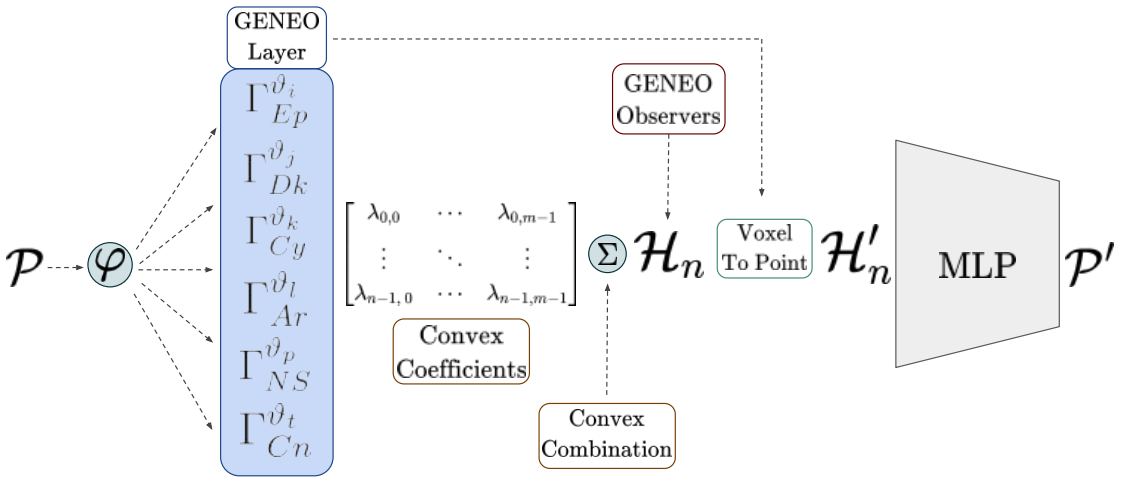


Figure 6.2: **Overview of the SCENE-Net V2 pipeline**. The input point cloud $\mathcal{P} \in \mathbb{R}^{N\times(3+C)}$ is first mapped by a measurement function $\varphi\colon \mathbb{R}^3 \to \{0,1\}$ onto a voxel grid that encodes point occupancy. A GENEO Layer with m kernels discretized from a set of parametric continuous functions $\{\Gamma_j^{\vartheta_j}\}_{j=1}^m$ that encode relevant geometric cues convoles the input to extract shape information. These responses are combined into n GENEO observers $\mathcal{H} = \{\mathcal{H}_i\}_{i=1}^n$ by a convex coefficient matrix $\Lambda \in \mathbb{R}^{n\times m}$, thereby yielding interpretable mid level features in which each observer's contribution can be traced back to specific geometric priors. A voxel to point transformation aggregates voxel features at the original point locations to form $\mathcal{H}'$, which is then fed to a compact multi layer perceptron that outputs per point semantic labels. Unlike standard CNNs, the number of trainable parameters in the GENEO feature extractor depends on the number of kernels and observers rather than on the discretization size of the kernels, which supports high resolution grids with constant memory use.

parameters ϑ_j that define a geometric prior. For instance, these parameters can include radius and Gaussian spread for a cylinder or principal radii for an ellipsoid. Optimizing ϑ_j rather than arbitrary kernel weights maintains equivariance to the chosen transformation group.

To effectively address the multiclass segmentation setting, SCENE-Net V2 introduces a set of n GENEO observers $\mathcal{H} = \{\mathcal{H}_i\}_{i=1}^n$, each formed as a convex combination of the GENEO Layer responses. This is a key difference from SCENE-Net, which used a single layer with three operators. With a coefficient matrix $\Lambda = (\lambda_{ij}) \in \mathbb{R}^{n\times m}$, the ith observer is

$$\mathcal{H}_i(x) \;=\; \sum_{j=1}^{m} \lambda_{ij}\, \Gamma_j^{\vartheta_j}(\varphi)(x)\,. \tag{6.1}$$

Because convex combinations of GENEOs are themselves GENEOs, each $\mathcal{H}_i$ inherits the equivariance and non expansiveness of its contributors. The nonnegativity and unit

sum constraints on the rows of Λ give each observer a clear decomposition into the contributions of the underlying GENEO kernels. This design allows SCENE-Net V2 to express a rich variety of patterns in the data while retaining interpretability. In contrast, SCENE-Net relied on a single observer and was focused on a specific pattern: pole detection.

The observer feature maps are next transferred from the voxel grid back to the point domain through a voxel to point transformation, yielding $\mathcal{H}'$ at the original point locations. This step aligns the geometry-informed measurements with the point-based target label for multiclass segmentation. Finally, a compact multi layer perceptron (Multi-Layer Perceptron (MLP)) classifier maps $\mathcal{H}'$ to class logits for each point, producing the final per point predictions. This final stage is a black-box component, which is why the overall model is gray-box. Nevertheless, the end to end prediction can still be analyzed by tracing the classification weights to observers and then to the contributing GENEO kernels via Λ.

Two practical properties follow from this design. First, the parameter count of the feature extractor depends on the number of kernels and observers but not on the kernel discretization size. This decouples the model size from the grid resolution and enables the use of larger 3D kernels without inflating the number of learnable parameters. Second, because the learned quantities are the shape parameters and the convex coefficients, the responses retain a direct geometric meaning. This facilitates the interpretation of the model's decisions in terms of the underlying geometric cues found in data. Finally, the design allows for efficient backpropagation through the observer and GENEO layers, ensuring that gradients are properly propagated to the shape parameters and convex coefficients. These aspects unlock multiclass scene understanding for the SCENE-Net architecture while preserving the interpretability and efficiency that characterize GENEO-based models.

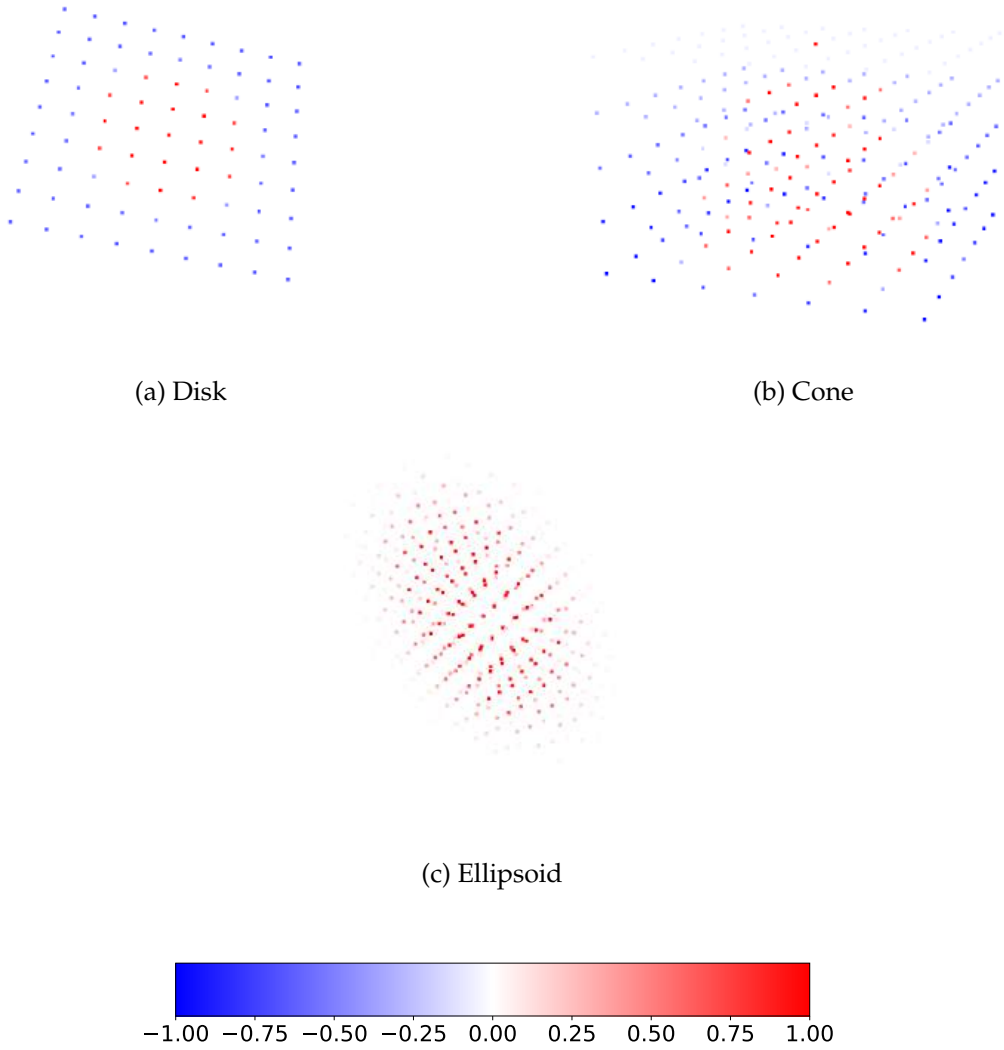
6.2.2 GENEO Kernels: Geometric Inductive Biases

SCENE-Net V2 leverages a diverse set of GENEO kernels, each tailored to capture a distinct geometric pattern commonly found in 3D scenes. These kernels are defined as parametric families of functions, with parameters that are optimized during training to fit the data while preserving geometric interpretability. Figure 6.3 shows the discretized versions of the three new kernels introduced in SCENE-Net V2, while the original three kernels from SCENE-Net can be seen in Figure 5.5.

Formally, a GENEO Γ^{ϑ} acts on $\varphi \in \Phi$, where Φ denotes the space of admissible measurements $\varphi\colon \mathbb{R}^3 \to \{0,1\}$. Given a geometric prior g, normalized to $\tilde{g}$, the operator is defined as

$$\Gamma^{\vartheta}\colon \Phi \to \Psi, \quad \psi = \Gamma^{\vartheta}(\varphi), \qquad \psi(x) = \int_{\mathbb{R}^3} \tilde{g}(y)\,\varphi(x-y)\,dy, \tag{6.2}$$

where Ψ is a new functional space representing the transformed point cloud. Positive values of $\psi(x)$ correspond to regions where the prior geometry is detected, while negative



(a) Disk

(b) Cone

(c) Ellipsoid

Figure 6.3: **Visualization of the new GENEO kernels discretized in a voxel grid**. The three kernels (Cylinder, Arrow, Negative Sphere) from SCENE-Net can be visualized in Figure 5.5 (see Chapter 5, Section 5.3.2) and are essential for describing pole-like objects and suppressing spherical artifacts. SCENE-Net V2 extends this family with three additional kernels (Disk, Cone, Ellipsoid) that generalize the range of geometric priors, enabling the model to capture planar, conical, and ellipsoidal structures commonly present in real-world 3D environments.

values penalize mismatched shapes. This construction enables each kernel to emulate a human observer searching for specific geometric configurations in the data.

6.2.2.1 Cylinder, Arrow, and Negative Sphere

The **Cylinder**, **Arrow**, and **Negative Sphere** GENEOs were first introduced in SCENE-Net (see Chapter 5, Section 5.3.2) to model power line supporting towers and suppress vegetation. For convenience, we briefly recall them here.

Cylinder GENEO. The cylinder kernel is rotationally equivariant around the z-axis and translationally equivariant in the xy plane. Its parameters $\vartheta_{Cy} = [r, \sigma]$ define the cylinder radius and the Gaussian spread:

$$g_{\mathrm{Cy}}(x) = \exp\left(-\frac{1}{2\sigma^2}\left(\|z(x) - z(c)\|^2 - r^2\right)^2\right).$$

This kernel is essential for detecting tower shafts and other cylindrical structures. Figure 5.5a illustrates this operator discretized in a voxel grid.

Arrow GENEO. The arrow kernel combines a cone and a cylinder to represent elongated vertical structures with tapered ends. Its parameters $\vartheta_{Ar} = [r, \sigma, h, r_c, \beta]$ control the dimensions and inclination, and $v(x)$ denotes the height of point x the reference space:

$$g_{\mathrm{Ar}}(x) = \begin{cases} \exp\left(-\frac{1}{2\sigma^2}(\|z(x) - z(c)\|^2 - r^2)^2\right), & v(x) < h, \\ \exp\left(-\frac{1}{2\sigma^2}(\|z(x) - z(c)\|^2 - (r_c \tan(\beta\pi))^2)^2\right), & \text{otherwise.} \end{cases}$$

Figure 5.5b depicts the arrow kernel. Even though it is generalized from the cone and cylinder kernels, it is still particularly useful for detecting power line supporting towers, as it captures both the shaft and the connecting power lines.

Negative Sphere GENEO. This kernel was originally used to suppress spherical vegetation crowns that caused false positives in SCENE-Net. Its parameters $\vartheta_{NS} = [r, \sigma, \omega]$ control radius, spread, and weighting factor:

$$g_{\mathrm{NS}}(x) = -\omega \exp\left(-\frac{1}{2\sigma^2}\left(\|x - c\|^2 - r^2\right)^2\right).$$

In SCENE-Net V2 we lift the constraint $\omega > 0$, allowing it either to suppress or enhance spherical structures depending on the data. In Figure 5.5c, we show the resulting negative sphere kernel.

6.2.2.2 Disk GENEO

The disk kernel enables the model to capture planar surfaces. It is rotationally equivariant around the z-axis, but also includes learnable rotation angles ϕ_x and ϕ_y around the x and y axes. Its parameters $\vartheta_{Dk} = [\phi_x, \phi_y, \sigma, r, h]$ define orientation, radius, and location:

$$g_{\mathrm{Dk}}(x) = \begin{cases} \exp\left(-\frac{1}{2\sigma^2}\left(\|z(R_{\phi_x,\phi_y}(x)) - z(c)\|^2 - r^2\right)^2\right), & v(x)(R_{\phi_x,\phi_y}(x)) = h, \\ 0, & \text{otherwise.} \end{cases}$$

This kernel captures ground, building facades, and other flat surfaces in multiple orientations. Figure 6.3a depicts this operator.

6.2.2.3 Cone GENEO

The cone kernel is designed for small conical shapes such as vegetation tips or architectural features. It provides rotational equivariance around the z-axis. Its parameters $\vartheta = [r, \sigma, \beta]$ govern the base radius, Gaussian spread, and inclination:

$$g_{\mathrm{Cn}}(x) = \exp\left(-\frac{1}{2\sigma^2}\left(\|z(x) - z(c)\|^2 - (r\tan(\beta\pi))^2\right)^2\right).$$

It generalizes the arrow prior by modeling standalone conical structures. Figure 6.3b illustrates the resulting cone kernel.

6.2.2.4 Ellipsoid GENEO

The ellipsoid kernel generalizes the negative sphere by allowing anisotropic scaling along the three axes. It adapts to diverse shapes such as elongated crowns or irregular objects. Its parameters $\vartheta = [a, b, c, \omega]$ define the radii and scaling factor:

$$g_{\mathrm{El}}(x) = \omega \exp\left(-\tfrac{1}{2}(x-c)^\top \Sigma^{-1}(x-c)\right), \qquad \Sigma = \mathrm{diag}(a^2, b^2, c^2).$$

This kernel is particularly useful for modeling elongated or flattened structures that spherical priors cannot capture. Figure 6.3c illustrates this operator discretized.

Together, these six GENEO kernels enable SCENE-Net V2 to model a significantly broader set of 3D structures than its predecessor, while retaining interpretability through their explicit geometric parametrization.

6.2.3 Optimization Strategy

Training SCENE-Net V2 requires preserving the mathematical properties of the GENEO framework while adapting the parameters to data. In particular, the convex structure of the observers and the non-negativity of the shape parameters must be maintained throughout optimization. This distinguishes SCENE-Net V2 from conventional convolutional networks, where kernels are freely optimized without such constraints, but also from SCENE-Net, whose simple design did not warrant additional regularization mechanisms. Formally, the learning objective is defined as

$$\begin{aligned} \min_{\Lambda,\vartheta} \quad & \mathbb{E}_{(X,y)\sim\mathcal{D}}\left[\mathcal{L}_{\mathrm{seg}}(\Lambda, \vartheta; X, y)\right] \\ \text{s.t.} \quad & \Lambda \in \Delta^{m-1\times n}, \quad \vartheta \in \mathbb{R}_+^T, \end{aligned} \tag{6.3}$$

where $\Delta^{m-1\times n}$ denotes the $(m-1)$-dimensional simplex for each of the n observers, ensuring that the convex coefficients Λ form valid convex combinations of GENEO kernels, and $\mathbb{R}_+^T$ is the non-negative orthant for most shape parameters ϑ. These constraints are essential to maintain the semantic meaning of parameters such as radii or scaling factors.

Segmentation loss. The segmentation loss $\mathcal{L}_{\text{seg}}$ is defined as a weighted cross-entropy:

$$\mathcal{L}_{\text{seg}}(\Lambda, \vartheta; X, y) = f_w(\alpha, \epsilon; y)\,\text{CE}\big(\mathcal{M}_{\Lambda,\vartheta}(X), y\big), \tag{6.4}$$

where $\mathcal{M}_{\Lambda,\vartheta}(X)$ denotes the model output, CE the cross entropy loss, and $f_w(\alpha, \epsilon; y)$ a weighting function that corrects for class imbalance as proposed by Steininger et al. [90]. The hyperparameter α controls the emphasis on minority classes, while ϵ prevents weights from vanishing. This weighting scheme is particularly relevant for TS40K, where transmission system classes are strongly underrepresented.

Reparametrization of Λ. To simplify optimization, the convexity constraint on Λ can be simplified by reparametrization, just as in SCENE-Net. Specifically, for each observer j, we set

$$\Lambda_{m,j} = 1 - \sum_{i=1}^{m-1} \Lambda_{i,j}, \tag{6.5}$$

which reduces the constraint set to $\Lambda \in \mathbb{R}_+^{m\times n}$ and $\vartheta \in \mathbb{R}_+^T$. The optimization problem then becomes

$$\begin{aligned} \min_{\Lambda,\vartheta} \quad & \mathbb{E}_{(X,y)\sim\mathcal{D}}\big[\mathcal{L}_{\text{seg}}(\Lambda, \vartheta; X, y)\big] \\ \text{s.t.} \quad & \Lambda \geq 0, \quad \vartheta \geq 0. \end{aligned} \tag{6.6}$$

Soft Regularization. To further guide the optimization, we add a regularization term composed of two components: a non-negativity penalty on appropriate GENEO parameters, which ensures that the shape parameters remain non-negative, and an Elastic Net penalty [124] on the convex coefficients Λ, which encourages the model to specialize observers in different aspects of the data. This yields the final optimization problem

$$\min_{\Lambda,\vartheta} \quad \mathbb{E}_{(X,y)\sim\mathcal{D}}\big[\mathcal{L}_{\text{seg}}(\Lambda, \vartheta; X, y)\big] + \Omega(\Lambda, \vartheta). \tag{6.7}$$

The regularizer is defined as

$$\begin{aligned} \Omega(\Lambda, \vartheta) = \rho_l \left(\sum_{j=1}^{n} \sum_{i=1}^{m} h(\Lambda_{i,j}) + \sum_{i=1}^{T} h(\vartheta_i) \right) \\ + \rho_t \left(\eta \sum_{j=1}^{n} \sum_{i=1}^{m} \|\Lambda_{i,j}\|_1 + (1-\eta) \sum_{j=1}^{n} \sum_{i=1}^{m} \|\Lambda_{i,j}\|_2^2 \right), \end{aligned} \tag{6.8}$$

where $h(x) = \max(0, -x)$ penalizes negative values, ρ_l and ρ_t control the strength of the two penalties, and $\eta \in [0, 1]$ balances sparsity (L^1) against weight decay (L^2).

This formulation promotes two desirable behaviors. First, non-negativity ensures the interpretability of parameters such as radii or inclinations. Second, the Elastic Net encourages observers to specialize by pruning redundant kernels or preventing trivial

convex combinations. As a result, SCENE-Net V2 focuses on meaningful geometric cues and adapts them effectively to the data, balancing various learned shapes with respect to different target classes. This extends the simple but rigid formulation of SCENE-Net into a more expressive optimization framework suitable for multiclass tasks.

6.2.4 Interpretability Mechanism

A central feature of SCENE-Net V2 is its mechanistic interpretability, which differs from the intrinsic interpretability of SCENE-Net due to the introduction of an MLP classifier. Unlike SCENE-Net, where parameters could be analyzed at face value to understand their direct impact on predictions, SCENE-Net V2's observer outputs are processed by a black-box MLP, preventing direct interpretation of how these features are utilized for classification. However, this design still enables mechanistic understanding through the convex combination of GENEO kernels into observers. Each observer $\mathcal{H}_i$ is expressed as a weighted sum of interpretable kernels (Equation 6.1), with coefficients λ_{ij} that quantify the relative importance of each geometric prior. The convex coefficients Λ can be inspected to identify which priors dominate a given observer, enabling users to trace class activations back to specific observers and then analyze the geometric biases that compose them. For example, in the analysis of TS40K samples, observers with large weights on the disk prior were consistently associated with the detection of power lines, while combinations involving the cylinder prior captured the body of towers. This mechanistic approach allows tracing predictions to concrete geometric evidence through a concrete path: starting at the prediction, to the observer activation, and arriving at the geometric prior composition.

This mechanism provides superior interpretability compared to post-hoc explainability methods, which attempt to justify or rationalize the decisions of black-box networks after training [61, 82]. While such methods rely on human interpretation and external approximations, SCENE-Net V2's interpretability is mechanistic and built into the architecture itself. Although the final classification is performed by a black-box MLP, the ability to systematically link predictions back to interpretable observer activations and their geometric compositions renders SCENE-Net V2 a gray-box model that maintains transparency throughout the feature extraction process.

6.3 Experiments

6.3.1 Experimental Setup

Dataset. We evaluate SCENE-Net V2 on the TS40K dataset, introduced in Chapter 3, a large-scale outdoor LiDAR dataset of electrical transmission systems across Europe in rural areas. Building upon the foundations of SCENE-Net, our approach directly addresses one of its primary limitations: the restriction to single-class pole-like object detection. While the original SCENE-Net demonstrated the viability of interpretable 3D scene understanding through GENEOs, its use was constrained to a specific object type,

Table 6.1: Key architectural differences between SCENE-Net V2 and the CNN baseline. Both models share the same overall architecture and classifier head. The difference lies entirely in the feature extraction stage: SCENE-Net V2 employs GENEO-based kernels, whereas the CNN baseline uses unconstrained convolutional kernels.

Component	**CNN Baseline**	**SCENE-Net V2**
Feature extraction	Standard 3D convolutions with random kernel weights	GENEO-based convolutions with interpretable shape parameters
Kernel dependency	Parameter count scales with kernel size	Parameter count independent of kernel size (depends on number of GENEOs and observers)
Parameter meaning	Unconstrained weights without semantic meaning	Geometric quantities such as radii, angles, and focal points
Classifier head	Multi-Layer Perceptron (shared)	Multi-Layer Perceptron (shared)
Interpretability	None (black-box)	Interpretability through convex weights and shape parameters (gray-box)

namely transmission towers and similar vertical elements. SCENE-Net V2 extends this paradigm by introducing a multiclass semantic segmentation framework.

Baselines. To assess performance, we compare SCENE-Net V2 against several representative 3D semantic segmentation methods: PointNet [78], PointNet++ [79], KPConv [96], RandLA-Net [44], Point Transformer V1 [121], Point Transformer V2 [110], and Point Transformer V3 [111]. Additionally, we implement a CNN baseline that mirrors the architecture of SCENE-Net V2 but replaces GENEO-based kernels with standard convolutional kernels initialized at random. This comparison isolates the effect of replacing unconstrained kernels with interpretable GENEO operators. In theory, the CNN kernels have the ability to instantiate the learned geometric kernels, but without the explicit interpretability provided by GENEOs and with considerably more parameters.

Implementation details. For SCENE-Net V2, GENEO parameters ϑ are initialized in positive ranges consistent with their geometric meaning (e.g., non-negative radii). Convex coefficients Λ are initialized uniformly within a normalized range to promote diversity across observers. Training is performed end-to-end using the Adam optimizer with an initial learning rate of 10^{-3}, batch size of 16, and 150 epochs. Both negativity penalties and Elastic Net regularization are applied with coefficients $\rho_l = \rho_t = 10^{-4}$ and Elastic Net balance parameter $\eta = 0.5$.

6.3.2 Results and Analysis

Overall performance. Table 6.2 reports segmentation results on the TS40K test set. Among existing baselines, Point Transformer V3 [111] achieves the highest overall accuracy with an mIoU of 68.34% using 46.2 million parameters. Point Transformer V2 [110] reaches 65.58% mIoU with 12.8 million parameters, while Point Transformer V1 [121]

Table 6.2: 3D semantic segmentation results on the TS40K test set. We report mean Intersection over Union (mIoU %), number of parameters, parameter efficiency $\frac{\text{mIoU}}{\log_{10}(\#\text{Parameters})}$, and whether the model is interpretable.

Method	mIoU (%)	#Parameters (M)	Parameter Efficiency	Interpretable?
PointNet [78]	44.58	0.40	7.96	No
PointNet++ [79]	46.90	1.48	7.60	No
KPConv [96]	57.58	14.9	8.03	No
RandLA-Net [44]	16.76	1.24	2.75	No
Point Transformer V1 [121]	62.67	6.3	9.22	No
Point Transformer V2 [110]	65.58	12.8	9.23	No
Point Transformer V3 [111]	**68.34**	46.2	8.92	No
CNN Baseline	41.69	0.26	7.69	No
SCENE-Net V2 (Ours)	45.54	**0.24**	8.46	**Yes**
SCENE-Net V2 + CNN (Ours)	**50.21**	0.26	**9.27**	**Yes**

achieves 62.67% mIoU using only 6.3 million parameters, demonstrating the best parameter efficiency among transformer-based methods. By contrast, SCENE-Net V2 reaches a mean IoU of 45.54% with only 240k parameters from which only 540 make up the GENEO-based feature extraction, the remaining parameters are used for the classifier head. This yields competitive parameter efficiency while being the only model in this comparison to offer intrinsic interpretability.

Baseline comparison. SCENE-Net V2 differs from the CNN baseline only in how their kernels are parameterized (Table 6.1). While both models share the same classifier head and architecture, the CNN baseline relies on unconstrained kernels whose number of parameters scales with kernel size. In contrast, SCENE-Net V2 employs an interpretable GENEO Layer whose parameterization is independent of kernel size. This results in fewer parameters (0.24M vs. 0.26M) and higher performance (45.54% vs. 41.69%). This is especially notable given that the CNN feature extraction process boasts of 21.4K parameters whereas the GENEO Layer only requires 540 parameters. Thus, replacing random kernels with GENEO priors provides both interpretability and better performance in the TS40K dataset.

Hybrid model. Finally, we explore the use of SCENE-Net V2 as a feature extraction module for standard black-box models. By prepending a GENEO feature extraction layer with only 540 additional interpretable parameters to the CNN baseline, mean IoU improves by 8.52%, from 41.69% to 50.21%, and parameter efficiency increases to 9.27, the highest among all tested models. Remarkably, this represents an improvement of almost 2 points in parameter efficiency (from 7.69 to 9.27) by simply adding SCENE-Net V2 as a geometric feature extractor. The hybrid model achieves the best parameter efficiency in the entire comparison, outperforming even the Point Transformer variants that require orders of magnitude more parameters. While the hybrid model's raw performance (50.21% mIoU) lags behind transformer-based methods, it remains highly competitive with KPConv (57.58% mIoU), which requires 57× more parameters (14.9M vs. 0.26M). This

Table 6.3: Performance of SCENE-Net V2 on the TS40K validation set with different kernel sizes. We report mean IoU (mIoU %) and per-class IoU (%) scores.

Kernel Size (z, x, y)	mIoU	Ground	Low Veg.	Med. Veg.	Tower	Power Line
(3, 3, 3)	28.09	52.22	9.74	25.13	19.89	33.45
(5, 5, 5)	33.15	52.78	14.13	26.15	20.76	51.94
(7, 7, 7)	36.08	61.44	13.78	28.41	**34.57**	42.19
(9, 9, 9)	37.08	59.53	11.08	28.24	20.26	66.31
(9, 5, 5)	37.71	58.17	12.34	27.09	19.75	71.22
(9, 7, 7)	32.68	57.77	12.29	27.53	22.85	42.97
(12, 12, 12)	35.29	57.30	9.21	29.34	22.98	57.62
(12, 5, 5)	**45.54**	**64.49**	**17.84**	**34.79**	21.92	**88.66**
(12, 7, 7)	32.62	53.66	12.94	28.92	17.24	50.36
(5, 9, 9)	32.04	60.36	11.45	29.24	16.28	42.86
(5, 12, 12)	33.47	55.49	14.53	28.23	16.98	52.12

demonstrates the potential of GENEO-based methods not only as standalone interpretable architectures but also as highly efficient modular components that can enhance the feature space of 3D point clouds for black-box models with minimal computational cost.

6.3.3 Ablation Studies

To better understand the design choices behind SCENE-Net V2 in the TS40K validation set, we conduct a series of ablation studies that systematically evaluate the influence of kernel size, the number of observers, the number of GENEO kernels per prior, the use of multiple GENEO layers, and the contribution of each geometric prior. These experiments serve two purposes: first, to identify configurations that maximize segmentation performance on TS40K, and second, to shed light on how geometric inductive biases affect model behavior.

Kernel size. Table 6.3 summarizes the impact of kernel discretization size on segmentation performance. Very small kernels, such as $(3, 3, 3)$, perform poorly with a mean IoU of 28.09%, as they fail to capture sufficient geometric context. Larger isotropic kernels such as $(9, 9, 9)$ improve accuracy, but overly coarse discretizations dilute shape information since negatively weighted voxels dominate over positive ones. The best results are obtained with the elongated kernel $(12, 5, 5)$, which achieves the highest mean IoU of 45.54%. It is important to note that this applies to the TS40K dataset where elongated structures are prevalent, but cannot be generalized to other datasets. This suggests that adapting kernel shapes to the specific geometric characteristics of the data is crucial for optimal performance. Conveniently, our approach lets us use the kernel size as a hyperparameter, allowing us to tailor kernel size post-hoc to maximize performance.

Number of observers. We next study the number of convex observers. As shown in Table 6.4, the best performance is obtained with 16 observers, achieving a mean IoU of 35.79%. Increasing beyond this number leads to decreased accuracy, likely due to

Table 6.4: Performance of SCENE-Net V2 on the TS40K validation set with different numbers of observers and kernel size $(7, 7, 7)$.

Observers	mIoU	Ground	Low Veg.	Med. Veg.	Tower	Power Line
8	32.87	59.30	**18.57**	26.74	15.93	**43.82**
16	**35.79**	**61.44**	12.89	**28.65**	**34.52**	41.47
32	32.17	60.53	14.12	26.95	16.74	42.51
64	28.64	57.08	10.57	26.32	11.94	37.29
128	29.39	55.76	10.92	27.16	13.58	39.53

Table 6.5: Performance of SCENE-Net V2 on the TS40K validation set with different GENEO kernel counts per geometric prior.

Kernels/prior	mIoU	Ground	Low Veg.	Med. Veg.	Tower	Power Line
4	30.73	55.45	10.73	27.86	25.39	34.21
8	**37.57**	58.87	12.76	28.01	17.98	**70.21**
16	35.79	**61.44**	**12.89**	**28.65**	**34.52**	41.47
32	31.89	59.90	11.14	28.27	20.59	39.58
64	33.37	60.95	13.75	**29.06**	21.13	41.98
128	29.05	49.24	11.28	25.57	**37.01**	22.13

redundancy and overfitting. This suggests that a moderate number of observers suffices to combine geometric priors effectively. We hypothesize that the optimal number of observers is related to the number of target classes and the geometric complexity of the dataset. It is plausible that having a number of observers comparable to or slightly exceeding the number of classes could help the model allocate specialized observers to distinct semantic categories or complex structures. However, further experiments would be needed to rigorously establish such a relationship. In practice, tuning the number of observers remains an empirical process that can be optimized to improve performance.

Number of GENEO kernels. We further investigate how the number of GENEO kernels per geometric prior influences performance. Table 6.5 shows that using 8 kernels per prior yields the highest mean IoU of 37.57%. Increasing the number of kernels beyond this point leads to diminishing returns and even performance degradation, with 128 kernels resulting in a mean IoU of only 29.05%. This suggests that adding excessive kernel redundancy does not enhance model expressivity and may, in fact, dilute the effectiveness of the observers. As the number of GENEO kernels grows, the convex weights become more dispersed, making it harder for observers to specialize and increasing the risk of spurious or irrelevant kernel contributions, even when Elastic Net regularization is applied.

Going Deeper? Multiple GENEO layers. In contrast to black-box models, which often benefit from deeper architectures, stacking multiple GENEO layers in SCENE-Net V2 does not yield performance gains. As shown in Table 6.6, introducing additional GENEO layers

Table 6.6: Performance of SCENE-Net V2 on the TS40K test set with different numbers of GENEO layers.

GENEO Layers	mIoU	Ground	Low Veg.	Med. Veg.	Tower	Power Line
(16)	**35.79**	**61.44**	**12.89**	**28.65**	**34.52**	**41.47**
(8, 16)	28.10	48.79	11.26	25.24	21.73	33.49
(8, 16, 32)	28.67	58.43	10.97	26.53	11.84	35.59
(8, 16, 32, 64)	27.95	51.83	11.87	25.79	20.36	29.91

Table 6.7: Performance of SCENE-Net V2 on the TS40K validation set with different geometric priors ablated (16 observers).

Prior Ablation	mIoU	Ground	Low Veg.	Med. Veg.	Tower	Power Line
No Cylinder	26.96	50.34	7.50	25.86	11.19	39.91
No Negative Sphere	33.03	55.63	6.17	29.36	**43.23**	30.75
No Arrow	36.83	**63.01**	**18.41**	**31.32**	20.01	51.38
No Disk	27.93	40.31	7.83	25.32	36.44	29.75
No Cone	34.59	62.04	6.42	29.17	18.04	57.26
No Ellipsoid	33.72	51.18	9.42	27.35	40.71	39.92
All GENEOs	**37.57**	58.87	12.76	28.01	17.98	**70.21**

beyond the first actually degrades segmentation accuracy. The optimal configuration uses a single GENEO layer with 16 observers, achieving a mean IoU of 35.79%. Adding further layers (e.g., with 8, 16, or 32 GENEOs) consistently reduces performance. This behavior aligns with the theoretical foundations of GENEOs: the output space Ψ after a GENEO transformation is fundamentally different from the input measurement space Φ, and subsequent GENEO layers are not inherently designed to process features in Ψ. As a result, stacking GENEO layers does not produce additional meaningful geometric features. Rather than being a limitation, this property highlights that a single GENEO layer is sufficient for extracting interpretable geometric features in 3D scene understanding.

Geometric prior ablations. Finally, we ablate individual priors to quantify their contributions. Table 6.7 shows that removing the Cylinder GENEO causes the most severe drop (mIoU 26.96%), which highlights its importance for tower detection. In contrast, the Arrow GENEO contributes least, as its role can be approximated by the Cylinder and Cone priors. Negative Sphere and Ellipsoid exhibit similar effects, consistent with their functional overlap. Overall, the full set of priors achieves the best performance, confirming their complementary roles.

The ablation studies reveal three key findings. First, kernel discretization plays a central role: elongated kernels capture geometric structures more effectively in the TS40K dataset. Second, a moderate number of observers (16) and GENEO kernels (8 per prior) achieve the best performance. Finally, not all priors contribute equally: the Cylinder GENEO is indispensable for tower detection, while others could be seen as redundant. Still the overall performance benefits from the full set of priors, confirming their relevance in model performance.

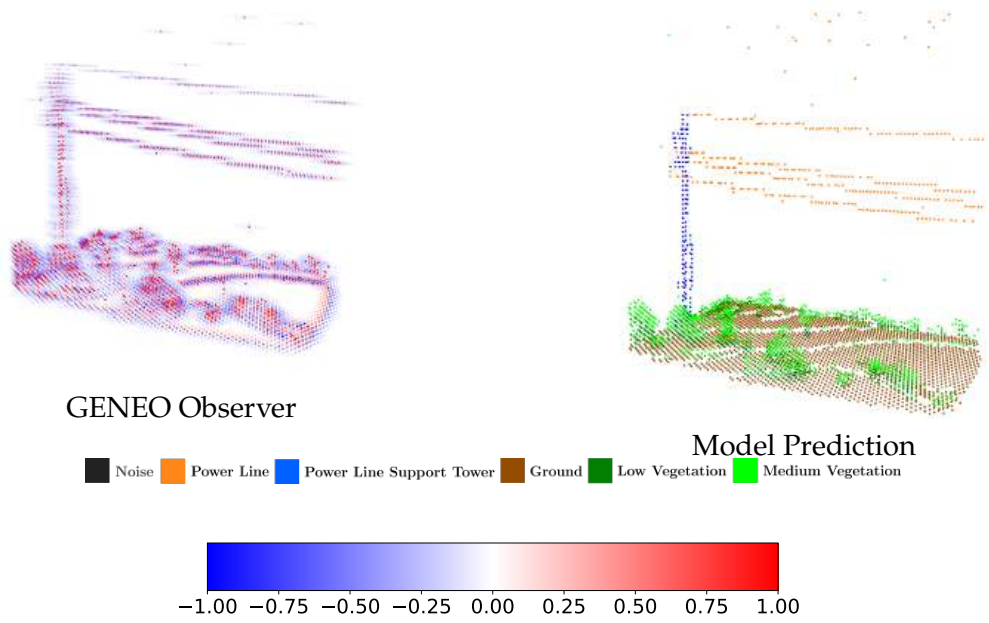


Figure 6.4: **Visualizing the inner workings of SCENE-Net V2.** Left: a GENEO observer constructed as a convex combination of geometric priors. Right: the corresponding model predictions obtained after classification. By examining convex coefficients, one can identify the dominant priors driving the observer's response. In this example, the classification of power lines is mainly aided by the geometric cues of the disk prior. Specifically, we traced back the strong activation of this observer to a disk kernel with minimal rotation and small radius parameters, indicating that the model has learned to associate horizontal planar structures with power line geometry.

6.3.4 Interpretability of SCENE-Net V2

SCENE-Net V2's mechanistic interpretability enables detailed analysis of model behavior by inspecting observer activations and their main geometric priors. Unlike black-box approaches that require explainable methods, our model's design allows direct examination of how geometric biases contribute to specific predictions. In Figure 6.4, a specific observer shows strong activation patterns that correlate with power line detection. Decomposition of this observer's convex coefficients reveals dominance of the disk prior with minimal rotation and small radius parameters, indicating the model has learned to associate horizontal planar structures with power line geometry. The classifier then leverages this observer to correctly segment power lines in the scene.

6.4 Conclusions and Future Work

In this chapter, we introduced SCENE-Net V2, the first gray-box model for multiclass 3D semantic segmentation. By leveraging Group Equivariant Non-Expansive Operators (GENEOs), SCENE-Net V2 incorporates fundamental geometric priors in its feature extraction step, bridging the gap between interpretable white-box and flexible black-box

approaches.

We addressed the key limitations of our predecessor model, SCENE-Net, by significantly expanding its scope of application from pole-like structures to multiclass scenarios with diverse 3D elements. Our approach demonstrates that GENEOs can be effectively used for multiclass classification without requiring manual curation of specific kernel sets for each individual object class, a process that would be time-consuming and difficult to compose systematically. Instead, our convex combination of complex observers from simple geometric shapes allows the model to automatically learn meaningful patterns. The addition of the black-box MLP classifier provides the model with the flexibility needed for multiclass tasks while still retaining its unique mechanistic interpretability through the GENEO-based feature extraction stage. This gray-box design enables practitioners to trace predictions back to specific GENEO kernels, offering superior transparency compared to post-hoc explanations of black-box models.

Our experimental results demonstrate that SCENE-Net V2 achieves competitive performance with the lowest parameter count among interpretable models (240k parameters, with only 540 for the GENEO feature extractor). Most significantly, incorporating SCENE-Net V2 as a geometric feature extraction module in black-box models leads to substantial performance improvements (8.52% mIoU increase in the baseline CNN) with only 540 extra parameters.

Limitations. Despite these contributions, SCENE-Net V2 has several important limitations that must be acknowledged. First, while our model achieves competitive parameter efficiency, its raw accuracy (45.54% mIoU) remains lower than state-of-the-art black-box networks such as Point Transformer V3 (68.34% mIoU). Second, the introduction of the MLP classifier results in a partial loss of interpretability compared to the fully white-box SCENE-Net. While we maintain mechanistic interpretability through the observer decomposition, the final classification with a black-box model prevents the analysis of how extracted features are utilized for decision-making.

Most critically, our evaluation is limited to the TS40K dataset rather than large-scale state-of-the-art benchmarks. This limitation stems from the memory-intensive nature of voxelizing large-scale scenes with sufficient resolution for our method. The discretization process represents our biggest computational bottleneck. Related work has demonstrated that point-based methods consistently outperform voxel-based approaches, highlighting a fundamental architectural limitation that constrains our method's scalability to large datasets.

Future Work. The limitations identified above provide clear directions for future research. Most immediately, addressing the discretization bottleneck represents a critical priority. Developing point-based variants of GENEO operators could eliminate the memory constraints associated with voxelization while potentially improving performance on large-scale benchmarks.

Beyond this, several research directions emerge from this work. First, extending GENEO-based models to additional domains such as urban scene understanding, autonomous driving datasets, and indoor environments. Second, deeper exploration of hybrid architectures that balance interpretability and scalability represents a promising avenue. Our results with the SCENE-Net V2 + CNN hybrid suggest that GENEO-based feature extractors can serve as effective modular building blocks in standard deep learning pipelines. Investigating how to integrate geometric priors into various architectures while maintaining computational efficiency could lead to higher performant models allied with interpretability. We explore this in more detail in Chapter 7, where we propose a novel framework for incorporating geometric priors into point-based networks.

In summary, SCENE-Net V2 establishes the feasibility of gray-box multiclass 3D semantic segmentation and demonstrates the value of geometric inductive biases in achieving parameter-efficient interpretable models. While computational limitations constrain its immediate applicability to the largest datasets, the foundational principles and hybrid architecture insights provide a roadmap for developing the next generation of interpretable 3D scene understanding systems.

7

GIBLy: A Lightweight Geometric Inductive Bias Layer for 3D Scene Understanding

This chapter introduces *GIBLy*, a lightweight and architecture-agnostic framework that integrates explicit geometric inductive biases into point-based neural networks for 3D scene understanding. The work builds directly upon the contributions and limitations of SCENE-Net V2 (presented in Chapter 6). While SCENE-Net V2 demonstrated the feasibility of gray-box multiclass segmentation by combining Group Equivariant Non-Expansive Operators (GENEOs) with black-box classifiers, it also exposed several challenges. Namely, the accuracy gap when compared to state-of-the-art point-based networks, the computational bottleneck caused by voxelization, and the restriction of experiments to the TS40K dataset. These limitations highlighted the need for a new approach capable of achieving scalability and performance improvements across large-scale benchmarks.

GIBLy is designed to meet this need. It re-introduces learnable parametric geometric primitives that now operate directly on raw point clouds, thereby eliminating voxelization constraints. Following SCENE-Net V2, GIBLy is lightweight, modular, and focuses on retrieving explicit geometric features for 3D backbone networks. By adding only a small number of parameters while remaining fully compatible with a wide range of architectures, including MLPs, convolutional networks, and transformers, GIBLy improves performance consistently across datasets and backbones, while preserving interpretability at the feature extraction level.

The remainder of this chapter is organized as follows. First, we provide an introduction that motivates the need for GIBLy by revisiting the limitations of SCENE-Net V2 and positioning a point-based approach as a scalable solution. Next, we present the design of the *Geometric Inductive Bias Layer*, detailing its formulation. We then describe the experiments, including implementation details, evaluations across multiple datasets and architectures, and ablation studies to assess the influence of key design choices. Finally, we conclude by summarizing the contributions of GIBLy, discussing how it addresses the shortcomings of SCENE-Net V2, and outlining promising directions for future work.

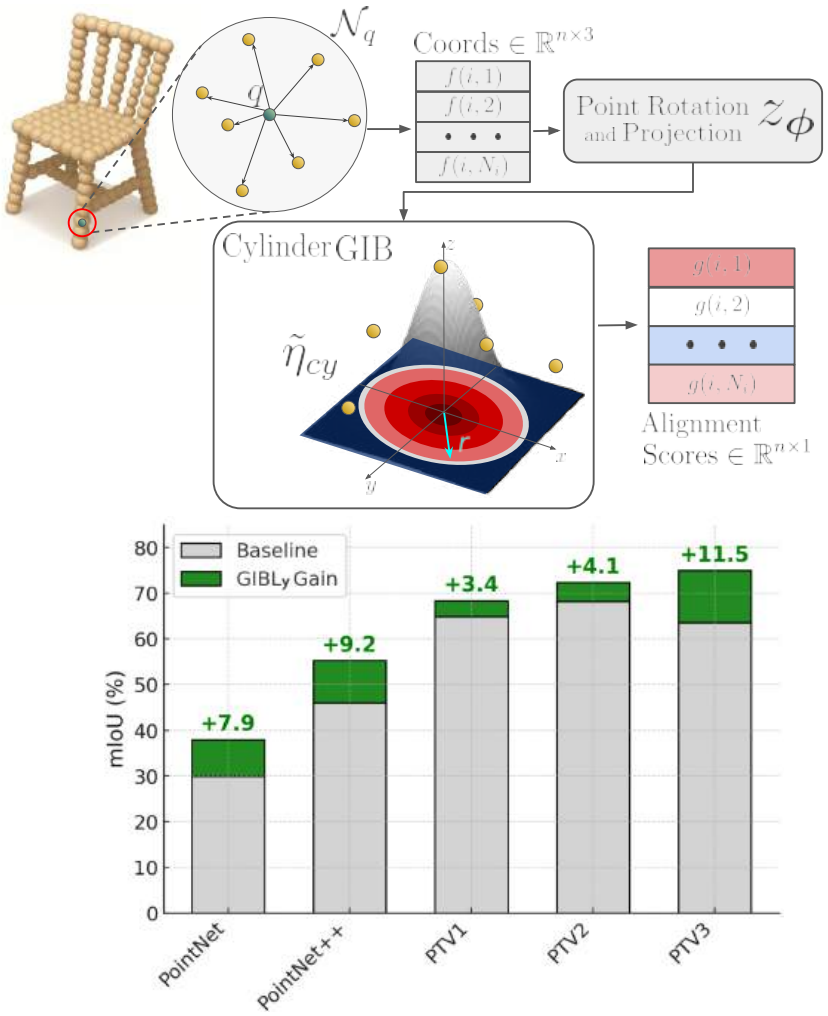


Figure 7.1: **GIBLy injects learnable geometric priors to improve 3D understanding.** *Top*: Illustration of a Cylinder geometric inductive bias (GIB) applied to a local point cloud neighborhood $\mathcal{N}_q$, such as a chair leg. The GIB is parameterized by a learnable orientation ϕ and radius r, and computes an alignment score that quantifies how well the points fit the cylindrical primitive. *Bottom*: Quantitative results on the TS40K dataset [51] show that introducing a single GIB-Layer (GIBLy) into various backbone architectures leads to substantial improvements in mean Intersection-over-Union (mIoU), with gains up to **+11.5%** for Point Transformer V3 [111], while adding only **58K** additional parameters. This demonstrates that GIBLy provides a lightweight and effective way to inject interpretable geometric knowledge into 3D Deep Learning (DL) models.

7.1 Introduction

3D data is ubiquitous in applications such as autonomous driving, augmented reality, and robotics. State-of-the-art methods in 3D scene understanding can be broadly categorized by their feature extraction strategies. Point-based approaches such as PointNet [78] and PointNet++ [79] rely on shared multilayer perceptrons to extract features directly from raw point clouds. Convolution-based methods instead impose regularity by rasterizing point clouds into volumetric grids [72, 73, 77] or by defining kernels that operate directly on

irregular point sets [58, 96, 109]. More recently, transformer-based architectures [48, 106, 110, 111, 121] have emerged, exploiting attention mechanisms [104] to achieve superior point-wise feature extraction and segmentation accuracy.

Despite their progress, these methods typically rely on learning geometric relationships implicitly from data rather than incorporating explicit geometric priors that could guide feature extraction more directly. As a result, networks often require large models, substantial training data, and significant computational resources to rediscover basic geometric structures that are common across tasks. For example, a model may need to learn from scratch what a flat surface or elongated cylinder looks like, even though these are universal concepts that could be encoded as priors. Convolutional networks succeed in image processing precisely because of built-in inductive biases such as locality and translation equivariance. In contrast, many 3D backbones, particularly MLPs and transformers, lack analogous built-in geometric biases and must learn spatial relationships entirely from data. This gap motivates new approaches to integrate explicit geometric cues into 3D models in a way that is lightweight, interpretable, and architecture-agnostic.

An example of this line of research is SCENE-Net V2, which constituted a significant advancement towards interpretable multiclass 3D semantic segmentation. By employing Group Equivariant Non-Expansive Operators (GENEOs), SCENE-Net V2 showed how geometric inductive biases can be integrated into feature extraction, thereby bridging the gap between the fully interpretable SCENE-Net and the flexibility of black-box models. However, as detailed in Chapter 6, three limitations constrain the scalability of this approach. First, despite its parameter efficiency, SCENE-Net V2 demonstrated a persistent accuracy gap relative to state-of-the-art point-based networks such as Point Transformer V3. Second, its dependence on voxelization imposed substantial computational bottlenecks, precluding evaluation on large-scale datasets such as SemanticKITTI [6]. Third, the experimental validation was largely confined to the TS40K dataset, which limits the scope of its findings and claims.

To address these challenges, we propose *GIBLy*: a lightweight geometric inductive bias layer that directly tackles the shortcomings of SCENE-Net V2 while advancing the broader goal of integrating explicit geometric priors into 3D DL. GIBLy eliminates the need for voxelization, scales efficiently to large datasets, and bridges the accuracy gap with point-based methods. By re-introducing learnable parametric primitives, such as cylinders, cones, disks, and ellipsoids, that operate directly on raw point clouds, GIBLy provides interpretable, shape-aware features for any 3D architecture. This approach adds minimal parameters yet consistently enhances performance and interpretability across MLPs, convolutional networks, and transformers.

Our main contributions are:

- We introduce GIBLy, a geometric inductive bias layer that is lightweight, interpretable, and architecture-agnostic. It provides explicit geometric cues for 3D scene understanding and can be seamlessly integrated into diverse 3D backbones.

- We demonstrate that GIBLy improves performance across a wide range of segmentation models and benchmarks with minimal additional parameters and no modifications to the base architecture, often boosting mIoU by up to 10% with only 50K additional parameters.

- We provide extensive ablation studies showing how placement strategy, neighborhood size, number of GIB instances, and primitive families influence performance and efficiency.

7.2 The Geometric Inductive Bias Layer (GIBLy)

7.2.1 Geometric Inductive Biases (GIBs)

7.2.1.1 Motivation.

A fundamental observation in 3D scene understanding is that many objects can be described as combinations of simple geometric primitives. A common table, for example, can be abstracted as a flat horizontal plane representing the tabletop supported by four vertical cylinders corresponding to its legs. Likewise, natural environments contain elements such as tree trunks or rocks that can often be approximated by cylinders and ellipsoids. Thus, we argue that these primitives provide a compact language for describing shapes that are recurrent in both man-made and natural scenes. However, despite their ubiquity, these structures exhibit variations in scale, proportion, and orientation. A table leg can be thicker or thinner, or a tree trunk can be tilted. Relying on fixed handcrafted priors is therefore too rigid for robust scene interpretation. Previous approaches that incorporated geometric priors [10, 52] were restricted to narrow application domains precisely because their priors could not adapt to diverse scenarios. Contrastingly, our work shifts perspectives: we use GIBs as a general mechanism to extract explicit geometric relationships between points in a point cloud. Rather than constraining the network to operate solely within the space defined by these primitives, we compute GIB alignment scores as additional features and concatenate them to the original point cloud representation. This enables state-of-the-art architectures, such as Point Transformer V3, to leverage geometric cues alongside learned features, improving performance and generalization across diverse 3D scene understanding tasks.

7.2.1.2 GIBs on raw point clouds.

A key design choice in GIBLy is to apply geometric inductive biases directly to raw point clouds, thereby avoiding the need for voxelization. Point clouds differ fundamentally from images since they lack a fixed grid structure: points are irregularly distributed in three-dimensional space with varying density. This irregularity makes standard convolutional kernels, which rely on a fixed grid, difficult to apply directly.

Point-based convolution methods overcome this limitation by operating on local neighborhoods of each query point. For a point x, its neighborhood $\mathcal{N}_x$ is defined by the set of nearby points within a chosen radius. By focusing on these local regions, point-based convolutions can capture geometric relationships without requiring a structured grid. Importantly, this strategy is naturally translation equivariant: the detection of a cylinder or plane, for example, is independent of where it is located in space, which is a desirable property for geometric feature extraction.

A point cloud can be formally represented as $\mathcal{P} \in \mathbb{R}^{N\times(3+C)}$, where N is the number of points and $3 + C$ denotes spatial coordinates together with any additional features such as intensity or color. A continuous convolution over $\mathcal{P}$ can be written as

$$(\mathcal{P} * g)(\mathbf{x}) = \int_{\mathbb{R}^3} g(\mathbf{y} - \mathbf{x}) f(\mathbf{y})\, d\mathbf{y}, \tag{7.1}$$

where $f : \mathbb{R}^3 \mapsto \mathbb{R}^C$ is the underlying continuous signal represented by the point cloud and g is a kernel function. In practice, this integral is approximated by summing over the neighborhood $\mathcal{N}_x$ of $\mathbf{x}$:

$$(\mathcal{P} * g)(\mathbf{x}) \approx \sum_{\mathbf{y}\in\mathcal{N}_x} g(\mathbf{y} - \mathbf{x}) f(\mathbf{y}). \tag{7.2}$$

In standard point-based convolution, the kernel g is often parameterized by a multi-layer perceptron because of its universal approximation capabilities [58, 109, 113]. However, MLP-based kernels must learn spatial relationships entirely from data, which increases the complexity of training, slows convergence, and reduces robustness to unseen configurations. Geometric inductive biases alleviate this problem by constraining g to take the form of a geometric basis function. Instead of a kernel with arbitrary parameters, we define g as

$$g(\mathbf{z}) = \eta(\mathbf{z}; \vartheta), \tag{7.3}$$

where η is a geometric function encoding a primitive shape (such as a radial basis function) and ϑ are learnable parameters controlling its characteristics (e.g., radius, orientation, or thickness). This formulation allows networks to directly encode meaningful geometric structures while retaining flexibility. Crucially, the parameters ϑ remain interpretable since they correspond to geometric quantities rather than arbitrary MLP weights.

7.2.1.3 Designing GIBs.

Many 3D objects are composites of simple parts that can be approximated by geometric primitives. For example, a cylinder with a learnable orientation and radius can describe both a leg of a table and a wheel of a car, while a plane can capture surfaces such as walls or floors. These primitives are transversal to a wide variety of datasets, appearing across both natural and man-made environments.

To embody this concept, we design a family of geometric inductive biases (GIBs) that serve as shape-aligned priors. Each GIB is implemented as a radial basis function (RBF)

defined in terms of the relative position of neighboring points. Formally, given a query point q and its local neighborhood $\mathcal{N}_q$, the position of each neighbor $x \in \mathcal{N}_q$ is defined relative to q, effectively centering the neighborhood at the query point. This relative vector is then rotated by a learnable rotation matrix $R_{\boldsymbol{\phi}}$ and projected into a canonical reference space:

$$z_{\boldsymbol{\phi}}(x) = z({R_{\boldsymbol{\phi}}}^{\top}(x - q)), \quad a_x = \eta_g(z_{\boldsymbol{\phi}}(x); \vartheta), \tag{7.4}$$

where η_g is a geometric kernel (e.g., Gaussian), $\boldsymbol{\phi} = [\phi_x, \phi_y, \phi_z]$ are learned rotation angles, and ϑ are shape parameters (e.g., radius or thickness). The output a_x represents the alignment score of the neighbor x with respect to the geometric prior centered on q.

This design confers two advantages. First, it allows each GIB to adapt flexibly to different instances of the same primitive family: a cylinder can become narrow or wide, tilted or vertical, depending on the learned parameters. Compared to SCENE-Net V2, our approach extends the use of a learnable rotation matrix to every GIB instead of just the Disk prior, enabling more diverse geometric alignments. Second, it preserves interpretability, since every parameter corresponds to an intuitive geometric attribute.

Real-world objects rarely conform to strict geometric boundaries, and fixed hard-coded shapes are therefore too restrictive. By relying on RBF-like functions, GIBs provide continuous similarity measures between local neighborhoods and the encoded geometric priors. This formulation improves on the smoothness of shape boundaries compared to our previous voxel-based approaches (both SCENE-Net and SCENE-Net V2). In the voxel-based setting, kernels were discretized and applied to quantized grids, which limited their resolution. In contrast, this approach applies continuous kernels directly to the raw point cloud, which takes full advantage of the smoothness of RBF-like functions.

In this work, we implement a set of GIBs based on fundamental geometric primitives: cylinders, cones, disks (planes), and ellipsoids, along with their hollow variants. These shapes were selected because they capture a wide range of structures commonly found in urban, indoor, and natural environments, such as poles, slopes, flat surfaces, rings, and volumetric bodies. More complex or composite structures can be represented by linear combinations of multiple GIBs, forming composite biases. Below, we introduce the full family of GIBs implemented in this work:

Cylinder GIB. The Cylinder GIB is designed to model long, tubular structures that are pervasive in both indoor and outdoor settings. Examples include the legs of furniture, utility poles, tree trunks, and signposts. The kernel is defined as a Gaussian function centered on a cylindrical axis, favoring radial symmetry around a central spine:

$$\eta_{cy}(x) = \exp\left(-\frac{\|z_{\boldsymbol{\phi}}(x)\|^2}{2r^2}\right). \tag{7.5}$$

Here, $z_{\boldsymbol{\phi}}(x)$ denotes the projection of a neighbor x into the canonical coordinate system aligned with the learned rotation matrix $R_{\boldsymbol{\phi}}$. The norm $\|z_{\boldsymbol{\phi}}(x)\|$ corresponds to the radial

distance from the cylinder's central axis. The parameter r defines the radius of the cylinder, controlling the spread of the Gaussian kernel. Neighbors located near the axis receive high alignment scores, while those further away are progressively suppressed. The learnable parameters for this GIB are therefore $\vartheta_{cy} = [r, \boldsymbol{\phi}]$. The cylinder GIB captures both vertical and arbitrarily oriented cylindrical structures (new orientations are learned by tuning $\boldsymbol{\phi}$ to the data), making it well suited to describe common man-made and natural forms.

Hollow Cylinder GIB. In many real-world sensing scenarios, cylindrical structures are not captured as solid volumes but rather as hollow surfaces. This is especially common with LiDAR and Red, Green, Blue, Depth (RGB-D) data, where only the external shell of an object is sampled. Examples include pipes, cables, and hollow rods. To address this case, we define the Hollow Cylinder GIB, which assigns high alignment scores to points that lie near a ring at a given distance from the central axis:

$$\eta_{hcy}(x) = \exp\left(-\frac{(\|z_{\boldsymbol{\phi}}(x)\| - r)^2}{2t^2}\right). \tag{7.6}$$

Unlike the solid cylinder, this function peaks at a radius r from the axis and suppresses both the interior and the exterior. The parameter t controls the shell thickness, effectively defining the tolerance band around the ring. The shape parameters of this primitive are thus extended to $\vartheta_{hcy} = [r, t, \boldsymbol{\phi}]$.

Cone GIB. The Cone GIB is designed to capture tapering structures such as treetops, roofs, or conic shapes. These forms are characterized by a gradual change in radius as height increases, which can be expressed through a Gaussian kernel whose spread adapts with elevation:

$$\eta_{cn}(x) = \exp\left(-\frac{\|z_{\boldsymbol{\phi}}(x)\|^2}{2\left(r \cdot v(x) \cdot \tan(\beta\pi)\right)^2}\right). \tag{7.7}$$

Here, $z_{\boldsymbol{\phi}}(x)$ denotes the centered and rotated coordinates of point x, and $v(x)$ refers to its vertical displacement relative to the cone apex. The denominator dynamically adjusts the scale of the kernel as a function of height and slope, allowing the receptive field to expand with distance from the apex. The parameter r sets the reference radius at the base of the cone, while the slope parameter $\beta \in [0, 0.5)$ regulates steepness. Constraining β ensures that $\tan(\beta\pi)$ is positive and finite, thereby avoiding degenerate vertical slopes. Small values of β produce steep, narrow cones, while larger values yield flatter structures. Together, r, $v(x)$, and β allow the bias to adapt its radial spread smoothly as height increases, enabling flexible modeling of tapering forms. The shape parameters for this GIB are defined as $\vartheta_{cn} = [r, v(x), \beta, \boldsymbol{\phi}]$.

Hollow Cone GIB. Similar to the Cylinder GIB, conical structures can be represented only as outer shells, as in sparse LiDAR scans of treetop canopies or hollow architectural

features. To model these, we define a hollow variant:

$$\eta_{hcn}(x) = \exp\left(-\frac{\left(\|z_\phi(x)\| - r \cdot v(x) \cdot \tan(\beta\pi)\right)^2}{2t^2}\right). \tag{7.8}$$

This kernel peaks along the cone surface, rewarding points located close to the expected radial distance at height $v(x)$. The thickness parameter t controls the tolerance around the surface, while the shape parameters extend to $\vartheta_{hcn} = [r, \beta, t, \phi]$.

Disk GIB. The Disk GIB is intended to capture flat structures such as tabletops, floors, or walls. Its kernel combines radial symmetry with a vertical gating term that constrains points to lie near a plane:

$$\eta_{dk}(x) = \exp\left(-\frac{\|z_\phi(x)\|^2}{2r^2} \cdot |w - v(x)|\right). \tag{7.9}$$

Here, $v(x)$ denotes the height of point x in the rotated frame, and w is a learnable threshold that specifies the disk's vertical position. The Gaussian term enforces a cylindrical shape, while the vertical factor attenuates points located above or below the disk plane. The shape parameters are $\vartheta_{dk} = [r, w, \phi]$.

Compared to the Disk GENEO introduced in Chapter 6 Section 6.2.2, we observe that this definition is more flexible due to the addition of the gating mechanism. This was not possible in the voxel-based version without making w a hyperparameter. Here, w is learned directly from data, allowing the disk to adapt to different heights in the scene.

Hollow Disk GIB. The hollow disk variant emphasizes rims and holes, which appear frequently in man-made settings. The kernel is defined as:

$$\eta_{hdk}(x) = \exp\left(-\frac{(\|z_\phi(x)\| - r)^2}{2t^2} \cdot |w - v(x)|\right). \tag{7.10}$$

This kernel peaks at a fixed radial distance r, rewarding points lying near the circumference while suppressing the interior and exterior. The thickness t controls tolerance around the rim, and the vertical gating ensures planarity. Parameters are $\vartheta_{hdk} = [r, t, w, \phi]$.

Ellipsoid GIB. The Ellipsoid GIB generalizes the notion of a sphere to ellipsoids of arbitrary shape and orientation. It is well suited for volumetric forms such as rocks, bushes, or human heads. Formally, it is defined as:

$$\eta_{ellip}(x) = \exp\left(-\frac{x^\top \Sigma^{-1}(\phi) x}{2}\right). \tag{7.11}$$

Here, $\Sigma^{-1}(\phi)$ is a positive semi-definite precision matrix encoding anisotropic scaling and orientation. By adjusting the eigenvalues and eigenvectors of Σ^{-1}, the ellipsoid can stretch or compress along different axes. This flexibility enables the bias to approximate a wide range of volumetric objects. The parameters are $\vartheta_{ellip} = [\Sigma, \phi]$.

Hollow Ellipsoid GIB. Many real-world volumetric objects are observed only as outer surfaces, such as cars in autonomous driving datasets. To isolate such cases, we define the Hollow Ellipsoid GIB:

$$\eta_{hellip}(x) = \exp\left(-\frac{\left(\sqrt{x^\top \Sigma^{-1}(\phi)x} - r\right)^2}{2t^2}\right). \tag{7.12}$$

The Mahalanobis distance $\sqrt{x^\top \Sigma^{-1}(\phi)x}$ defines the ellipsoidal shape, while the parameter r sets the peak distance from the center, effectively defining the shell radius, and t controls its thickness. This kernel therefore rewards points lying on the ellipsoidal surface and suppresses both the interior and exterior. The parameters are $\vartheta_{hellip} = [\Sigma, r, t, \phi]$.

7.2.2 GIB normalization

When integrating geometric inductive biases into a deep learning pipeline, it is important to ensure that the alignment scores they produce are comparable across neighborhoods and scales. Without normalization, biases may yield responses of different magnitudes depending on the density of the point cloud, the size of the neighborhood, or the scale of the primitive. This variation can lead to unstable training and weaken the interpretability of the results. To address this issue, we introduce a normalization procedure that places all GIB responses on a common scale. The idea is conceptually similar to batch normalization in convolutional networks, but adapted to the geometric nature of our operators. In batch normalization, feature maps are standardized across a mini-batch to stabilize learning. In contrast, GIB normalization is performed at the level of neighborhoods: we standardize alignment scores relative to the expected response of the primitive within the entire volume of a local neighborhood. For a given GIB instance η, its score over a neighborhood $\mathcal{N}_q$ quantifies the degree to which neighbors conform to the shape encoded by η. The normalized bias $\tilde{\eta}$ is defined so that points that align with the geometric prior yield positive values, while those that deviate yield negative values. This enhances interpretability, since the sign of the response encodes whether a region agrees with or contradicts the expected geometry. We compute the expected score of η within $\mathcal{N}_q$ using a Monte Carlo approximation of the integral over the neighborhood:

$$\Xi(x) = \int_{\mathcal{N}_q} \eta(y)\, dy \approx \sum_{y \in \mathcal{MC}} \eta(y), \tag{7.13}$$

where $\mathcal{MC} = \{y \in \mathbb{R}^3 \mid \|y\| \leq r\}$ is a Monte Carlo sample of the neighborhood with radius r. The normalized score is then obtained by subtracting this expectation:

$$\tilde{\eta}(x) = \eta(x) - \frac{\Xi(x)}{|\mathcal{MC}|}. \tag{7.14}$$

This makes it possible to combine multiple biases and to propagate their outputs through a network in a stable manner. Figure 7.2 illustrates how normalization transforms raw scores into a balanced representation.

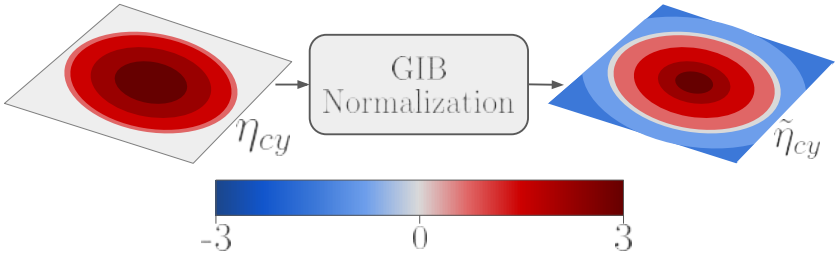


Figure 7.2: **GIB normalization.** Normalizing a Cylinder GIB ensures that neighbors aligning with the desired geometric configuration receive positive values, while those deviating receive negative values. Here, points close to the cylinder axis are emphasized, and points further away are penalized.

7.2.3 Composite Biases

Although individual GIBs provide meaningful responses to specific primitives, a single bias is rarely sufficient to capture the complexity of real objects. Most structures are combinations of simpler forms, and thus richer descriptors are needed. To address this, we define *composite biases* which combine multiple GIBs into higher-level features. These are analogous to observers in GENEO-based methods, however we do not adhere to the convexity constraint usually imposed in observers.

Let $E = \{\eta_j\}_{j=1}^{m}$ denote a set of m GIBs drawn from the families introduced earlier (cylinders, cones, disks, ellipsoids, and their hollow counterparts). These are combined linearly using a weight matrix $W \in \mathbb{R}^{n\times m}$ to form n composite biases $\gamma = \{\gamma_i\}_{i=1}^{n}$, defined as

$$\gamma_i = \sum_{\eta_j \in E} W_{ij}\, \eta_j. \tag{7.15}$$

The coefficients W_{ij} control the contribution of each primitive to the composite bias γ_i. Unlike GENEO-based methods that employ convex combinations (where weights are non-negative and sum to one), we use unconstrained linear combinations to achieve greater flexibility in the composite space. This design choice provides two key advantages. First, negative weights effectively flip the response of a GIB, enabling the suppression of specific geometric patterns while emphasizing their complement. For instance, a negative coefficient on a cylinder GIB can suppress cylindrical regions while highlighting the surrounding outer areas, which is useful for detecting hollow structures or boundaries. Second, by removing the constraint that weights sum to one, the magnitude of W_{ij} serves as a scaling factor for the responses achieved by each GIB, allowing the composite bias to amplify or attenuate individual geometric features as needed. This design preserves

interpretability while enhancing expressiveness compared to traditional convex observers. However, the lack of constraints on the weights can lead to instability during training, as the model may rely on a few dominant primitives while ignoring others. This can result in overfitting to specific shapes or noise in the input data, ultimately degrading the model's generalization capabilities.

7.2.4 Regularization and Constraints

However, unconstrained linear combinations risk introducing noise from irrelevant or weakly aligned primitives. To mitigate this, we introduce a regularization strategy that follows the same principles as SCENE-Net V2 (Chapter 6 Section 6.2.3), with the exception of the convexity constraint.

The regularizer promotes sparsity with an L_1 penalty and controls overall magnitude with an L_2 penalty on the composite weights W, encouraging the network to learn concise and meaningful composite biases rather than overfitting through redundant combinations. Additionally, we maintain the non-negativity constraint on geometric shape parameters ϑ to preserve their physical interpretability. The overall regularization term is defined as:

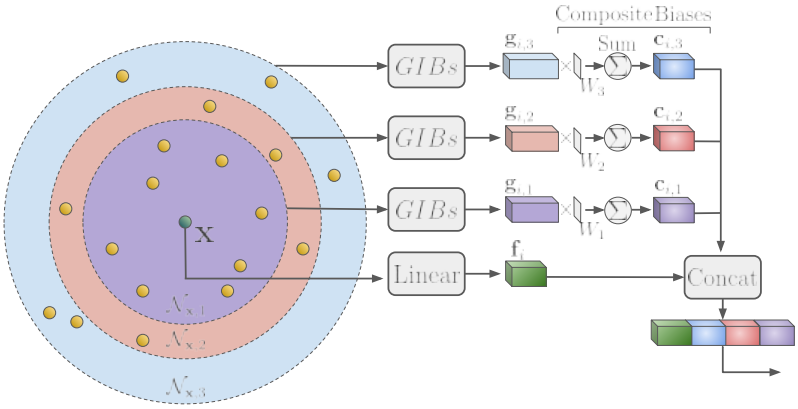
$$\Omega(W, \Sigma) = \rho_l \sum_{\vartheta \in \Sigma} \sum_{i}^{|\vartheta|} h(\vartheta_i) + \rho_t \left(\alpha \|W\|_1 + (1 - \alpha) \|W\|_2^2 \right), \tag{7.16}$$

where Σ denotes the set of shape parameters ϑ, $h(x) = \max(0, -x)$ penalizes negative values for shape parameters that must remain non-negative (such as radii), ρ_l and ρ_t control the strength of the penalties, and $\alpha \in [0, 1]$ balances sparsity (L_1) against weight decay (L_2) for the composite weights W.

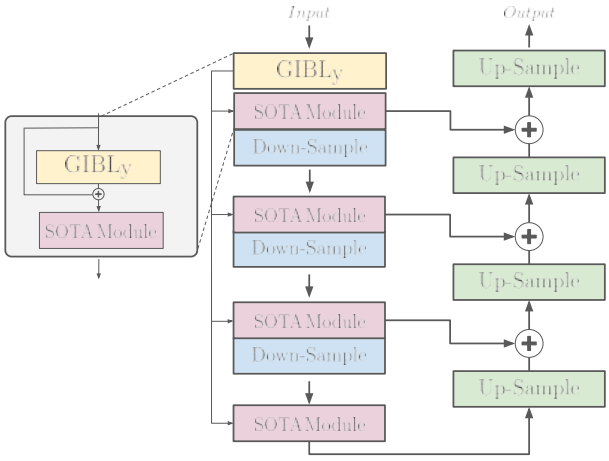
7.2.5 The GIB Layer (GIBLy)

The GIB-Layer, or GIBLy, is the architectural unit that integrates geometric inductive biases into a deep learning model. Its role is to inject explicit shape-aware features into the input of a 3D backbone, where the point cloud retains the most faithful geometric information. As illustrated in Figure 7.3 (a), the layer applies multiple GIBs to every input point across different neighborhood radii, producing alignment scores that describe how well the local geometry fits each primitive. These scores are combined into composite biases and fused with the original input features with a lightweight projection layer or an MLP. The resulting enriched representation contains both raw point attributes and explicit geometric descriptors.

We found empirically that applying GIBLy only at the input stage yields the best performance. At deeper layers, downsampling significantly reduces the number of points in the current representation, making alignment less reliable. Instead of reapplying the module at every stage, we propagate its features via skip connections to later layers of the encoder. This aligns with our intuition, geometric cues should be invariant with respect to



(a) GIBLy module



(b) Integration with 3D backbones

Figure 7.3: **Schematic of the GIBLy approach.** *(a)* The GIBLy module augments point features by evaluating a set of learnable geometric inductive biases (GIBs) at each query point. For every input point, multiple local neighborhoods are considered, and GIB alignment scores $\mathbf{g}_{i,N}$ are computed per region, where i is the point index and N is the neighborhood index. These scores are aggregated through learned weights W_N to form composite features $\mathbf{c}_{i,N}$, which encoding more complex geometric relationships. The composite features are then fused with the original point features using lightweight MLPs or projection layers $\mathbf{f}_i$, yielding an enhanced vector representation. *(b)* GIBLy is introduced at the input stage of a generic 3D backbone, where geometric detail is highest. Its outputs are propagated via skip connections to subsequent encoder stages, enabling efficient reuse of geometric information throughout the network without redundant computation. This design is compatible with a wide range of architectures, including MLPs, CNNs, and transformers.

the stage (i.e., the spatial resolution) at which they are obtained. This strategy is illustrated in Figure 7.3 (b), where GIBLy is applied once at the input and its outputs are reused in subsequent layers, ensuring that geometric information remains available throughout the network. Thus, not only is GIBLy extremely efficient in terms of parameter count, but it also reduces computational overhead by only requiring a single layer through the module.

The design is fully architecture-agnostic. Because GIBLy operates on point coordinates and outputs feature maps, it can be inserted into MLP-based backbones, convolutional networks, or transformers without structural modifications. Despite its expressiveness, the layer adds only a small number of parameters, typically on the order of tens of thousands.

Compared to SCENE-Net V2's feature extraction layer, GIBLy offers several computational advantages beyond eliminating voxelization overhead. First, unlike SCENE-Net V2, which must discretize new kernels at each forward pass as shape parameters change (requiring the entire kernel to be recomputed and imposing significant overhead), GIBLy's continuous geometric functions operate directly on point coordinates without discretization and do not need to be recomputed. Second, since our kernels are continuous functions rather than discrete weight tensors, adding multiple GIBs per primitive family incurs in a small computational cost with parallel processing. The shape parameters are computed in vectors and applied simultaneously to the point cloud, whereas SCENE-Net V2 requires additional memory to discretize each kernel and time to sweep the voxel grid. In essence, GIBLy is a more elegant solution that sidesteps the pitfalls of voxelization entirely: it attains the sharpest resolution by using continuous functions as its building blocks, it only needs to be applied once at the input stage and can be reused in subsequent layers, and is highly efficient in terms of both memory and computation.

7.3 Experiments

7.3.1 Implementation Details

Baselines. To evaluate the effectiveness of our approach, we integrated GIBLy into several representative 3D learning pipelines spanning different architectural paradigms. Specifically, we considered the seminal MLP-based methods PointNet [78] and PointNet++ [79], the convolutional method KPConv [96], and three variants of the transformer-based Point-Transformer family [110, 111, 121]. These backbones were selected to cover the major classes of 3D deep learning architectures, to demonstrate the model-agnostic nature of GIBLy, and to facilitate a comparison with SCENE-Net V2. Each baseline was obtained from its official repository.

Datasets. We evaluated GIBLy on five widely adopted benchmarks for 3D semantic segmentation, covering both indoor and outdoor environments. For indoor scene understanding, we used ScanNet v2 [23] and the Stanford Large-Scale 3D Indoor Spaces dataset (S3DIS) [3]. ScanNet v2 consists of 1,201 training scenes and 312 validation scans reconstructed from RGB-D frames, with each point annotated into one of 20 semantic categories. S3DIS contains 271 rooms across six areas from three buildings, annotated with 13 semantic classes; following the standard evaluation protocol, Area 5 is withheld during training and used for testing. For outdoor benchmarks, we considered nuScenes [12], SemanticKITTI [6], and TS40K [51]. nuScenes comprises approximately 1,000 urban driving

Table 7.1: Training configuration for indoor and outdoor benchmarks. All models are trained independently under identical conditions.

Setting	**Indoor**	**Outdoor**
Epochs	300	100
Batch size	16	16
GPU	A100	A100
Point sampling	FPS (100K points)	FPS (100K points)
Validation input	Full-resolution	Full-resolution
Optimizer	AdamW [65]	AdamW [65]
Learning rate	0.0001	0.0001
Loss (weights)	Focal (0.2) [60] Tversky (0.2) [83] Lovász (0.8) [9]	Focal (0.2) [60] Tversky (0.2) [83] Lovász (0.8) [9]
Input features	xyz + normals	xyz + normals

scenes collected with a multi-sensor rig mounted on a moving vehicle. SemanticKITTI extends the raw KITTI dataset with full point-wise labels over 22 sequences, totaling around 20,000 LiDAR scans. Finally, TS40K, our dataset focusing on rural power transmission systems introduced in Chapter 3. It provides high-resolution LiDAR scans of power lines, support towers, vegetation, and surrounding terrain. Unlike urban driving datasets, TS40K emphasizes large-scale rural environments, posing challenges such as extreme class imbalance and noisy point distributions. The formerly described benchmarks are discussed in greater detail in Chapter 2, Section 2.1.6.

7.3.2 Training Setup

To ensure a fair comparison, we retrained all baseline networks and their GIBLy-augmented counterparts under identical training conditions. This step is necessary because some baselines report higher performance when trained with additional data or dataset-specific augmentations, whereas our objective is to assess the isolated effect of integrating GIBLy.

All models were trained independently on a single NVIDIA A100 GPU with a batch size of 16. Indoor datasets were trained for 300 epochs, while outdoor datasets were trained for 100 epochs. We used farthest point sampling (FPS) to select 100,000 points per scene during training, ensuring a consistent geometric coverage of the input. During validation and testing, full-resolution point clouds were used without subsampling. For optimization, we employed AdamW [65] with a base learning rate of 10^{-4}. Following prior works [77, 110, 111, 121], we included both coordinates (x, y, z) and estimated normals as input features. To mitigate class imbalance, the loss function combined Focal loss [60], Tversky loss [83], and Lovász loss [9], with respective weights 0.2, 0.2, and 0.8. The full configuration is reported in Table 7.1.

Table 7.2: Data augmentation settings used for indoor and outdoor datasets. The parameter p indicates the probability of applying each transformation.

Augmentation	**Indoor**	**Outdoor**
Random dropout (ratio 0.2, $p = 0.2$)	✓	–
Rotation z ($\theta \in [-1, 1]$, $p = 0.5$)	✓	✓
Random scale (0.9–1.0)	✓	✓
Random flip ($p = 0.5$)	✓	✓
Jitter ($\vartheta = 0.005$, clip=0.02)	✓	✓
Color jitter (std=0.05, $p = 0.95$)	✓	–
Sphere crop (max=90K points)	✓	–
Color normalization	✓	–
Coordinate normalization	✓	✓

Table 7.3: Runtime breakdown for GIBLy operations.

Operation	**Compute Time Ratio**
Neighborhood Computation	64.45%
R_ϕ Computation	12.11%
GIB Normalization	10.19%
GIB Computation	7.38%
Composite Bias Computation	2.03%
Other operations	4.84%

Data augmentation. To encourage generalization and prevent overfitting, we adopted a consistent augmentation pipeline across all experiments. This included random dropout, jittering, flips, rotations around the vertical axis, random scaling, and coordinate normalization. For indoor datasets with RGB input, additional augmentations such as color jitter and color normalization were also applied. The exact settings are reported in Table 7.2.

Compute Time Analysis. Although GIBLy introduces additional geometric computations, the overall runtime overhead is modest. Each GIB reduces to evaluating a parametric radial basis function (RBF) over the distance vectors of a given neighborhood. Once neighborhoods are constructed, applying GIBs requires only matrix multiplications and point-wise kernel evaluations, both of which are highly parallelizable on GPUs. As shown in Table 7.3, the majority of compute time is spent on neighborhood construction (~64%), a shared bottleneck across state-of-the-art point cloud methods. The actual GIB computations (including rotations, normalization, and basis function evaluations) account for less than 20% of total runtime, and composite bias formation is negligible. Importantly, this overhead occurs only once at the input stage, after which computed features are propagated through the backbone. Thus, the added cost of GIBLy is modest relative to the backbone and scales efficiently with modern hardware.

7.3.3 Evaluation

TS40K. Table 7.4 reports per-class and mean IoU on the TS40K dataset [51], which focuses on large-scale transmission systems in rural environments. This benchmark is particularly challenging due to the dominance of terrain points, the high-density noise, and the diversity in objects with low point density, such as support towers. Across the majority of the tested backbones, GIBLy provides consistent improvements in segmentation quality, which confirms the benefits of integrating geometric inductive biases into standard architectures. The largest gain is observed in PointTransformerV3 [111], where performance increases from 63.55% to 75.03% mIoU, corresponding to an improvement of +11.48. Notably, even the lightweight PointNet [78] backbone benefits substantially, improving by +7.95 mIoU. These results indicate that GIBLy can compensate for the limited representational power of older architectures, while also boosting the performance of state-of-the-art transformer-based models. A smaller gain is observed in PointNet++ [79], while KPConv [96] shows a mild regression. We attribute this behavior to the overlap between rigid kernel-based convolutions and the alignment scores introduced by GIBLy, which may result in conflicting feature extraction.

nuScenes, SemanticKITTI, ScanNet v2, and S3DIS. The broader evaluation across four widely used benchmarks is presented in Table 7.5. On S3DIS [3], GIBLy consistently improves performance across all backbones, with particularly strong gains for transformer-based models. PointTransformerV2 [110] improves by +8.68%, and PointTransformerV3 [111] by +7.52%, highlighting that GIBLy excels in structured indoor environments where geometric regularities are pronounced. SemanticKITTI [6] also benefits from the addition of GIBLy, with improvements up to +3.53% mIoU on PointTransformerV1 [121]. In ScanNet v2 [23], the benefits are most visible in transformer-based backbones, whereas PointNet and PointNet++ underperform with GIBLy. For nuScenes [12], performance varies: PointNet, PointNet++ and PointTransformerV2 improve, while KPConv, PTV1 and PTV3 regress. We interpret this as an indication that the sparsity and dynamic nature of nuScenes LiDAR sweeps may reduce the stability of geometric bias alignment.

Qualitative results. Figure 7.4 provides qualitative comparisons on the TS40K dataset. Each row shows the input point cloud, the baseline prediction from PointTransformerV3, and the prediction from its GIBLy-augmented counterpart. Baseline models frequently misclassify or miss support towers, introduce spurious labels in vegetation regions, or confuse artifacts with surrounding objects. GIBLy consistently mitigates these errors, yielding predictions that are better aligned with the underlying geometry of the scene. These results highlight the utility of explicit geometric priors: by embedding shape information directly into feature extraction, GIBLy helps networks produce more semantically coherent segmentations.

Table 7.4: Semantic segmentation results on TS40K. GIBLy improves class-wise and mean IoU across most backbones. Gains are shown in green and deficits in red.

Method		**mIoU (%)**	Noise	Ground	Low Veg.	Mid Veg.	Tower	Power Line
PointNet	base	30.01	49.36	54.52	46.00	14.23	0.00	35.28
	+ GIBLy	37.96	27.13	78.06	49.42	29.60	5.04	38.51
	Δ	+7.95	–22.23	+23.54	+3.42	+15.37	+5.04	+3.23
PointNet++	base	45.99	59.27	59.99	54.36	14.55	22.61	78.41
	+ GIBLy	55.23	65.79	72.57	29.94	59.44	18.74	84.89
	Δ	+9.24	+6.52	+12.58	–24.42	+44.89	–3.87	+6.48
KPConv	base	52.77	57.02	64.75	37.12	34.63	37.36	89.99
	+ GIBLy	46.83	53.37	61.96	35.73	31.26	12.91	85.78
	Δ	–5.94	–3.65	–2.79	–1.39	–3.37	–24.45	–4.21
PTV1	base	64.90	57.50	77.33	60.34	46.51	54.19	93.54
	+ GIBLy	68.33	63.88	79.15	68.77	50.93	52.16	95.06
	Δ	+3.43	+6.38	+1.82	+8.43	+4.42	–2.03	+1.52
PTV2	base	68.29	61.16	80.13	68.17	51.39	54.48	94.43
	+ GIBLy	72.35	67.79	82.40	73.17	54.38	60.29	96.06
	Δ	+4.06	+6.63	+2.27	+5.00	+2.99	+5.81	+1.63
PTV3	base	63.55	59.23	70.77	50.47	43.86	61.42	95.53
	+ GIBLy	75.03	68.91	82.96	73.32	55.33	72.49	97.17
	Δ	+11.48	+9.68	+12.19	+22.85	+11.47	+11.07	+1.64

7.3.4 Ablation Studies

We conducted a series of controlled experiments on the TS40K validation set to investigate the influence of core GIBLy design decisions. All ablations use PointTransformerV3 [111] as the backbone. We focus on four key aspects: the placement of the GIBLy layer, the number of neighborhood scales, the number of GIB instances per primitive, and the effect of different bias shape families.

Performance on Partial Scans. The effectiveness of GIBLy is closely tied to the availability of sufficient local geometry for its biases to align with. When the point cloud is heavily downsampled, many objects lose both fine structures and, in some cases, their overall shape. This loss directly weakens the geometric cues that GIBLy leverages, resulting in smaller improvements at very low point counts (e.g., 5,000 FPS). As shown in Table 7.6, performance gains increase steadily as more points are available, peaking near the full-resolution setting. This behavior is consistent with the design of GIBLy: richer point density preserves the local neighborhoods necessary for geometric alignment, while sparse inputs reduce the expressiveness of these cues.

GIBLy Placement. Table 7.7 reports the effect of different placement strategies for GIBLy. Applying a single GIBLy layer at the input stage achieves the strongest result, with an improvement of +10.14 mIoU over the baseline. This confirms that geometric inductive biases are most effective when applied directly to raw point coordinates, where the fidelity of geometry is highest. Adding additional layers at intermediate stages brings only

Table 7.5: Semantic segmentation results (mIoU %) and improvements (Δ) across four benchmarks. Gains are shown in green and deficits in red.

Method		nuScenes [12] mIoU	Δ	SemanticKITTI [6] mIoU	Δ	ScanNet v2 [23] mIoU	Δ	S3DIS (Area 5) [3] mIoU	Δ
PointNet [78]	base	14.56	–	16.58	–	17.20	–	19.46	–
	+ GIBLy	18.52	+3.96	19.83	+3.25	13.64	–3.56	21.10	+1.64
PointNet++ [79]	base	18.52	–	23.33	–	21.55	–	26.00	–
	+ GIBLy	21.30	+2.78	24.59	+1.26	14.78	–6.77	26.72	+0.72
KPConv [96]	base	50.74	–	45.24	–	36.95	–	30.31	–
	+ GIBLy	37.33	–13.41	47.55	+2.31	29.46	–7.49	35.71	+5.40
PTV1 [121]	base	66.29	–	50.17	–	33.16	–	47.14	–
	+ GIBLy	54.38	–11.91	53.70	+3.53	39.08	+5.92	51.14	+4.00
PTV2 [110]	base	65.40	–	54.15	–	52.05	–	56.42	–
	+ GIBLy	70.26	+4.86	55.88	+1.73	55.27	+3.22	65.10	+8.68
PTV3 [111]	base	74.20	–	55.09	–	55.64	–	60.44	–
	+ GIBLy	69.91	–4.29	58.14	+3.05	57.79	+2.15	67.96	+7.52

For fair evaluation, all models are trained under the same conditions without the use of additional training data or dataset-specific augmentation pipelines. Some methods, such as PTV3 [111], report higher performance when trained with extra data; however, here we report results under controlled and equal conditions to ensure valid comparison between each backbone and its GIBLy-augmented variant.

Table 7.6: Ablation on performance with partial scans (TS40K, PTV3 backbone).

FPS Level	PTV3 mIoU	PTV3 + GIBLy mIoU	Δ
5,000	59.12	61.07	+1.95
10,000	63.06	70.37	+7.31
100,000 (default)	63.55	75.03	+11.48
200,000	62.90	74.83	+11.93

Table 7.7: Ablation on GIBLy placement strategy. Best performance is obtained by applying GIBLy only once at the input stage.

Configuration	mIoU (%)	Δ (%)
Baseline (PTV3)	65.80	
GIBLy (input only, default)	75.94	+10.14
GIBLy (input + intermediate)	66.18	+0.38
GIBLy (every stage)	56.41	–9.39

marginal benefit, while applying GIBLy at every stage actually degrades mIoU by –9.39. We attribute this to the fact that deeper layers operate on downsampled point clouds, where the geometric detail and local neighborhood point-relationships are less reliable. In such cases, introducing explicit geometric priors may act as noise, confusing the network rather than guiding it.

Neighborhood Count. We next analyze the role of neighborhood scales in Table 7.8. Using a single radius reduces performance, showing that fixed-scale neighborhoods are insufficient to capture the multi-scale details of 3D objects. Performance improves as

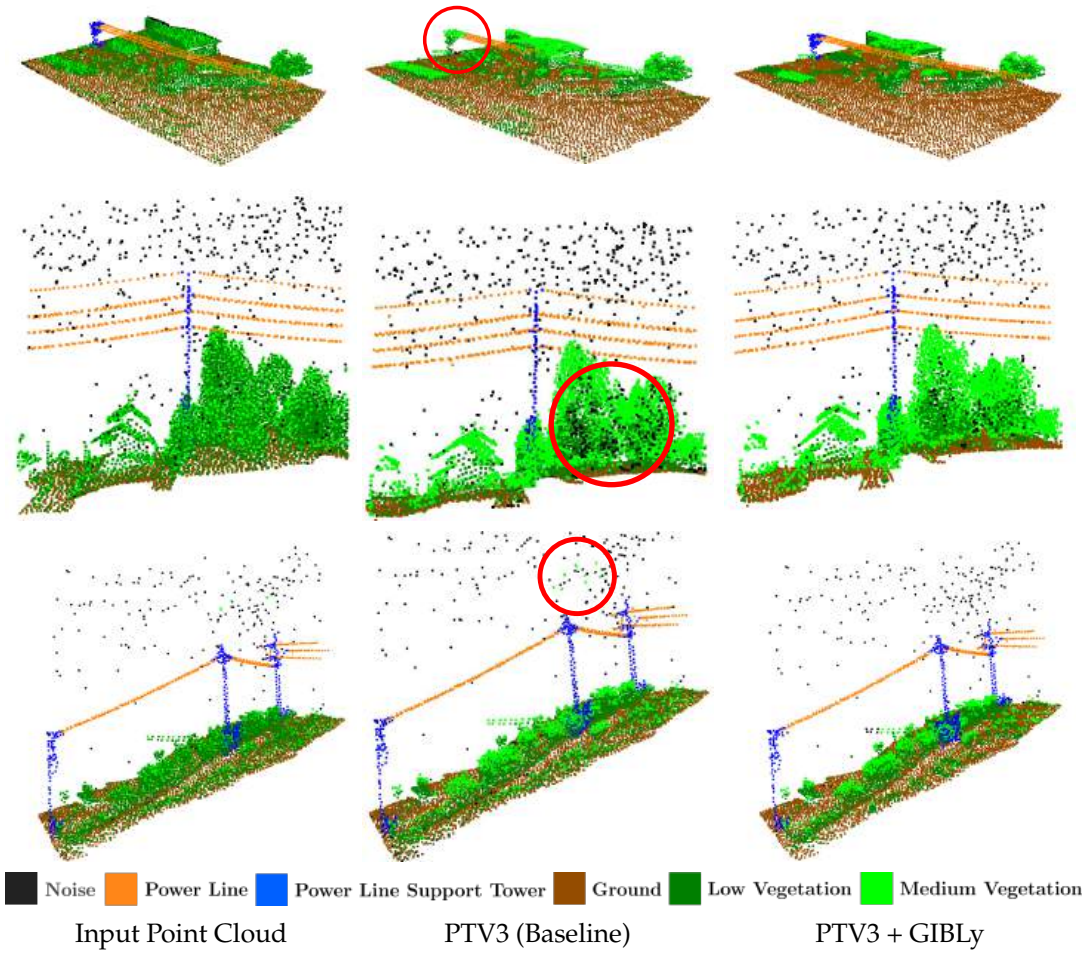


Figure 7.4: **Qualitative results on TS40K.** Each row corresponds to a different scene. From left to right: input point cloud, prediction from the baseline PointTransformerV3, and prediction from PointTransformerV3 with GIBLy. The baseline often fails to detect pylons or misclassifies vegetation, while the GIBLy-augmented model produces predictions that are more geometrically consistent and faithful to scene structure.

the number of neighborhood radii increases, with two or three scales producing the best balance between accuracy and computational cost. Beyond three radii, performance drops despite higher latency, suggesting that too many neighborhoods introduce noisy or redundant information. These results confirm that GIBLy benefits from a moderate multi-scale design that balances richness and efficiency.

Number of GIBs. Table 7.9 evaluates the effect of varying the number of GIB instances per primitive. The best performance is achieved with two GIBs per shape, which yields an improvement to 75.94% mIoU. Increasing the number beyond this produces diminishing returns, and at 16 GIBs performance degrades sharply. This can be explained by overfitting and redundancy: too many biases may overlap too much, introducing conflicting signals rather than useful diversity. Moreover, latency increases moderately with the number of

Table 7.8: Ablation on the number of neighborhood levels. Two or three levels provide the best trade-off between performance and latency.

# Neighborhoods	mIoU (%)	Latency (ms)
1	69.12	23.1
2	72.41	29.2
3 (default)	75.94	37.0
5	68.56	50.4

Table 7.9: Ablation on the number of GIB instances per primitive. A small number of diverse GIBs performs best.

GIBs per Prior	mIoU (%)	Latency (ms)
2 (default)	75.94	27.0
4	73.92	32.6
8	71.67	39.2
16	68.84	42.4

Table 7.10: Ablation on geometric bias families. Combining all GIB types yields the strongest performance.

Bias Types	mIoU (%)	Δ (%)
Radial only (cylinder, disk)	71.14	–4.80
Hollow only	71.30	–4.64
Ellipsoid only	68.42	–7.52
All GIBs (default)	75.94	

GIBs, highlighting the importance of parameter efficiency. These findings reinforce that a small, diverse set of geometric priors is preferable to an excessive collection of redundant ones. Nevertheless, optimized CUDA kernels may in the future reduce this computational cost.

Bias Shape Variants. Finally, we examine the contribution of different bias families in Table 7.10. Using the full set of GIBs achieves the highest performance. Subsets based on radial primitives (cylinders and disks) or hollow variants achieve moderately strong results but still fall short of the complete configuration. The poorest performance is observed when using ellipsoids alone, indicating that they are not sufficient to capture discriminative structures by themselves. These results highlight the value of combining complementary geometric priors.

Individual Shape Contributions. To further clarify the role of each geometric primitive, we evaluated the contribution of each GIB family independently. Table 7.11 summarizes the results. No single primitive is sufficient to capture the diversity of structures present in the data. Among individual GIBs, ellipsoids achieve the strongest performance, consistent

Table 7.11: Ablation on individual shape contributions (TS40K, PTV3 backbone).

Types of GIBs	**mIoU (%)**
All GIBs (default)	75.94
Cylinder	59.83
Ellipsoid	68.42
Disk	61.43
Cone	57.47
Hollow Cylinder	60.13
Hollow Ellipsoid	63.78
Hollow Disk	63.11
Hollow Cone	58.62

Table 7.12: Ablation on composite biases.

Number of Composites	**mIoU (%)**
0	59.60
2	66.06
4	69.33
8	72.14
16 (default)	**75.94**
64	73.75
128	72.57

with their ability to approximate a broad range of volumetric forms. The best results are obtained when combining all GIBs, this demonstrates that the diversity of geometric biases is complementary and supports the design choice to use a rich family of GIBs.

Composite Biases. In Table 7.12, we evaluate the role of composite biases. Using none severely limits performance, while increasing their number progressively improves results up to 16 composites. Beyond this point, performance saturates and slightly decreases, suggesting that a moderate number of composites provides the best balance between expressiveness and overfitting.

GIB Normalization. Table 7.13 shows the impact of normalization. Without normalization, performance drops significantly as alignment scores are dominated by neighborhood density. Standard z-score normalization partially recovers performance but produces inconsistent semantics, since well-aligned neighbors may be assigned negative values. Our proposed normalization yields the best results, confirming its importance for stable integration of GIBs.

Summary. Overall, the ablation studies confirm the central design choices of GIBLy. A single input-layer placement, a moderate number of multi-scale neighborhoods, and a small set of diverse priors produce the best results. These findings validate the guiding

Table 7.13: Ablation on GIB normalization.

Normalization Type	mIoU (%)
No Normalization	62.92
Standardization	67.84
GIB Normalization (default)	**75.94**

principle of this work: explicit, interpretable geometric biases improve feature extraction when introduced at the earliest stage of processing, provided they are applied judiciously rather than in excess.

7.4 Conclusion and Future Work

This chapter introduced GIBLy, a lightweight geometric inductive bias layer that successfully addresses the key limitations of SCENE-Net V2 while advancing the integration of explicit geometric priors into 3D deep learning. Through comprehensive evaluation across five diverse benchmarks and six different backbone architectures, we have demonstrated that GIBLy provides a practical and effective solution for enhancing 3D scene understanding.

Key contributions. GIBLy represents a significant advancement over our previous works. First, by operating directly on raw point clouds rather than voxelized representations, it eliminates the computational bottleneck that constrained SCENE-Net V2's scalability. This enables evaluation on large-scale datasets such as SemanticKITTI and nuScenes, which were previously inaccessible due to voxelization overhead. Second, GIBLy successfully bridges the accuracy gap that persisted in SCENE-Net V2, achieving performance improvements of up to +11.48% mIoU on TS40K and +8.68% on S3DIS when integrated with state-of-the-art transformer architectures. Third, the framework's architecture-agnostic design allows seamless integration into diverse 3D backbones without requiring modifications to the base networks.

Limitations. Although GIBLy demonstrates clear advantages in terms of performance and scalability, it also presents several limitations that highlight directions for future work. The current framework is built upon a predefined set of geometric primitives. While cylinders, cones, disks, and ellipsoids capture a broad range of common structures, they cannot represent the full diversity of shapes found in complex 3D environments. More specialized primitives or modular combinations of simpler components may be required to address datasets with irregular or domain-specific geometries. Another limitation concerns interpretability, compared to its predecessor SCENE-Net V2, GIBLy sacrifices part of its transparency. The reason for this is twofold. First, the model introduces a higher parameter count, approximately 58K trainable parameters. As a result, it becomes

increasingly difficult to trace how each individual geometric prior contributes to the final predictions, and a full manual analysis of all GIB responses is impractical. Second, the integration of GIBLy within deep backbones, combined with the propagation of its features throughout the encoding process, further obscures the relationship between individual priors and network outputs. Finally, we observe that performance gains are not uniform across backbones. While transformer-based models benefit strongly from the addition of GIBLy, other architectures such as convolutional networks show inconsistent trends, with improvements in some cases and regressions in others. A more detailed analysis is necessary to understand these variations and to guide the design of future architectures that can make optimal use of explicit geometric priors.

Future Work. Several promising avenues emerge for extending the capabilities of GIBLy. A first direction is the development of adaptive or deformable geometric priors that go beyond a fixed set of primitives. This would broaden the range of structures that GIBLy can represent while preserving interpretability at the level of learned parameters. A second direction involves improvements in runtime. Although GIBLy introduces only a modest overhead, the use of specialized kernels could further reduce latency and memory requirements. In particular, optimized CUDA implementations of GIB computations would be especially valuable for applications that demand real-time or near real-time inference, such as autonomous driving or aerial inspection. A third direction concerns the explicit control and selection of GIBs. Mechanisms for dynamic bias selection or dynamic composites could allow the network to automatically activate the most relevant GIBs depending on the context of the input. For example, certain biases could be suggested or prioritized when detecting objects with strong geometric regularities, such as poles or planar surfaces, while others may be suppressed in less structured regions. This selective activation would reduce redundant computation, improve efficiency, and encourage the network to make more targeted use of explicit priors.

8

Conclusions

This thesis has explored the integration of geometric inductive biases into 3D deep learning, with a particular focus on rural power grid inspections. Through a comprehensive research work spanning interpretable architectures, benchmark dataset development, and practical deployments, this work has advanced our understanding of how explicit geometric knowledge can enhance both the performance and interpretability of 3D scene understanding systems.

8.1 Summary of Contributions

Our research journey began with the observation that 3D deep learning models, despite their impressive capabilities and increasing model sizes, often lack the geometric awareness that could make them more efficient, interpretable, and robust. Unlike 2D computer vision, where models like convolutional neural networks benefit from built-in inductive biases such as translation equivariance and locality, 3D models frequently learn from scratch spatial relationships entirely from data. We believe this to be a fundamental limitation: 3D data faithfully preserves the geometric relationships between points in space and is not subject to none of the ambiguities of 2D projection. Thus, we argue that incorporating explicit geometric knowledge into 3D models is essential to improve robustness and performance.

To address these challenges, we developed a step-by-step approach to incorporate geometric inductive biases through three complementary methodologies: white-box interpretable models, gray-box hybrid approaches, and lightweight bias layers for 3D backbones. Each represents a different point along the interpretability-performance spectrum, allowing practitioners to choose the appropriate level of transparency based on their specific requirements.

SCENE-Net: Fully Interpretable 3D Segmentation. Our first contribution, SCENE-Net, demonstrated that fully interpretable models can achieve competitive performance on specialized tasks when equipped with appropriate geometric priors. By leveraging

Group Equivariant Non-Expansive Operators (GENEOs) as building blocks, SCENE-Net embedded domain knowledge directly into its architecture, making every computational step traceable and meaningful. The model achieved remarkable parameter efficiency and delivered competitive results with five orders of magnitude fewer parameters than state-of-the-art baselines. This work established that white-box approaches need not sacrifice performance when the inductive biases align well with the task structure.

SCENE-Net V2: Bridging Interpretability and Flexibility. Recognizing the limitations of purely white-box approaches in handling diverse, multiclass scenarios, SCENE-Net V2 introduced a gray-box paradigm that combines interpretable geometric feature extraction with flexible classification. By using GENEOs as feature extractors followed by traditional neural network classifiers, this approach preserved the interpretability of geometric priors while gaining the expressiveness needed for complex segmentation tasks. SCENE-Net V2 demonstrated that partial interpretability could deliver substantial benefits: it enables post-hoc analysis of geometric feature contributions while achieving improved performance on the multiclass problem in TS40K.

GIBLy: Architecture-Agnostic Geometric Inductive Bias Layer. Our final methodological contribution, GIBLy, addressed the scalability limitations of voxel-based approaches by introducing lightweight geometric inductive bias layers that operate directly on raw point clouds. GIBLy can be seamlessly integrated into any 3D backbone adding only minimal parameters while consistently improving performance. This approach eliminates computational bottlenecks that prevented evaluation on large-scale datasets and bridges the accuracy gap with state-of-the-art point-based networks. With improvements of up to +11.48% mean IoU across diverse benchmarks, GIBLy demonstrates that explicit geometric priors can enhance even the most advanced architectures.

TS40K: A Comprehensive Benchmark for Infrastructure Inspection. Beyond methodological innovations, we recognized the critical need for domain-specific evaluation frameworks. TS40K, our large-scale LiDAR dataset for rural power grid inspection, fills a significant gap in 3D CV benchmarks. Comprising over 40,000 kilometers of densely annotated transmission corridors, TS40K captures the unique challenges of infrastructure monitoring: extreme class imbalance, high-density noise, structural diversity, and realistic label noise. The dataset has enabled a comprehensive evaluation of 3D segmentation approaches and provided a foundation for developing cost-aware inspection tools.

Cost-aware Inspection Tool and its Impact. Finally, we demonstrated the practical viability of our approaches through a complete inspection pipeline that integrates CV predictions with human-in-the-loop validation and cost estimation. By benchmarking on TS40K, we showed that transformer-based models can achieve trustworthy performance, with IoU exceeding 65% for towers and 96% for power lines. This performance is further

enhanced by geometric priors, with our best model (Point Transformer V3 + GIBLy) achieving 72% IoU for towers and 97% IoU for power lines. The cost analysis framework provides utilities with quantitative tools for evaluating the trade-offs between automation benefits and error-driven costs.

8.2 Key Findings

A central finding of our work is that geometric inductive biases provide the greatest benefit when applied to data with sufficient geometric detail. Across all three methodologies, SCENE-Net, SCENE-Net V2, and GIBLy, performance improvements were most pronounced when models could exploit rich spatial relationships in high-resolution point clouds. This has important implications for sensor selection and data acquisition strategies in 3D applications.

Our three approaches confirm the existence of a transparency-performance trade-off, but also demonstrate that this trade-off need not be steep when geometric priors are well-aligned with the task. SCENE-Net achieved full interpretability with minimal performance loss on specialized tasks, SCENE-Net V2 provided partial interpretability with improved flexibility, and GIBLy sacrificed some interpretability for maximum performance gains. This spectrum allows practitioners to choose the appropriate balance based on their specific requirements for transparency, accuracy, and computational efficiency.

GIBLy's success across diverse backbone architectures and various 3D benchmarks demonstrates that geometric inductive biases provide universal benefits in 3D learning. The improvements observed across different models suggest that explicit geometric priors address fundamental limitations in how 3D models process spatial relationships, rather than compensating for specific architectural weaknesses.

Despite being developed primarily for power grid inspection, our geometric priors showed remarkable transferability across different 3D domains. SCENE-Net's GENEOs proved effective on SemanticKITTI without modification, and GIBLy demonstrated consistent improvements across indoor and outdoor benchmarks. This suggests that primary geometric structures, such as cylinders, planes, and ellipsoids, represent universal building blocks that go beyond specific application domains.

8.3 Advancements in Power Grid Inspection

Our work has established a new state-of-the-art for automated power grid inspection in 3D scene understanding. By combining a domain-specific dataset with novel methodologies, we have achieved several contributions:

Our models exceed the minimum performance thresholds established by industry experts at Energias de Portugal (EDP) and EDP's Energy R&D Laboratory (Labelec), with our best-performing system (Point Transformer V3 enhanced with GIBLy) achieving 72% IoU for towers and 97% IoU for power lines on TS40K. This result shows significant strides

since the inception of the TS40K benchmark, where the best-performing model (Point Transformer V2) achieved only 43% IoU for towers and 93% IoU for power lines.

The cost-benefit analysis developed in Chapter 4 provides quantitative evidence that automated inspection can deliver significant savings when properly integrated into existing workflows. By accounting for both labor cost reductions and error expenses, utility operators can make informed decisions about adopting this technology based on their specific operational parameters.

TS40K has provided the research community with the first large-scale, publicly available benchmark for rural power grid inspection. The dataset's has many realistic challenges that stem from using real world 3D data: label noise, extreme class imbalance, high-density noise and structural diversity. These challenges reflect the conditions models will encounter in real-world deployment compared to existing 3D benchmarks.

8.4 Limitations and Future Directions

This work has taken meaningful steps toward more interpretable and effective 3D scene understanding, but several limitations remain and point to future research directions:

One limitation concerns the range of geometric inductive biases employed. The current approach relies on a predefined set of simple geometric primitives, which restricts its ability to represent more complex or irregular structures. Future work could explore adaptive or learned geometric representations that capture richer spatial patterns or automatically identify useful primitives for specific domains. Building a modular library of geometric components that can be selected and combined as needed would also improve the flexibility of the framework.

Another challenge relates to interpretability. As geometric priors are integrated into deeper and more complex architectures, understanding how explicit geometric knowledge interacts with learned representations becomes less straightforward. Developing tools to analyze or visualize this interaction would help preserve interpretability as models become more sophisticated.

A final limitation involves the scope of validation. Although the proposed methods perform consistently across several benchmarks, further evaluation on different sensors, environments, and infrastructure types would help establish their robustness and broader applicability in real-world settings.

Bibliography

[1] D. Alexiadis, D. Zarpalas, and P. Daras. "Fast and Smooth 3D Reconstruction Using Multiple RGB-Depth Sensors". In: *2014 IEEE Visual Communications and Image Processing Conference*. IEEE. 2014, pp. 173–176 (cit. on p. 7).

[2] A. M. Araújo and M. M. Oliveira. "Connectivity-based cylinder detection in unorganized point clouds". In: *Pattern Recognition* 100 (2020), p. 107161 (cit. on p. 80).

[3] I. Armeni et al. "3D Semantic Parsing of Large-Scale Indoor Spaces". In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1534–1543 (cit. on pp. 7, 14, 36, 134, 137, 139).

[4] L. Arreola et al. "Improvement in the UAV position estimation with low-cost GPS, INS and vision-based system: Application to a quadrotor UAV". In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2018, pp. 1248–1254 (cit. on p. 35).

[5] J. Baxter. "a Model of Inductive Bias Learning". In: *Journal of Artificial Intelligence Research* 12 (2000), pp. 149–198 (cit. on p. 16).

[6] J. Behley et al. "SemanticKITTI: a Dataset for Semantic Scene Understanding of LiDAR Sequences". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9297–9307 (cit. on pp. 1, 7, 15, 36, 43, 93, 124, 134, 137, 139).

[7] S. A. Bello et al. "Deep Learning on 3D Point Clouds". In: *Remote Sensing* 12.11 (2020), p. 1729 (cit. on pp. 8–10).

[8] M. G. Bergomi et al. "Towards a Topological–Geometrical Theory of Group Equivariant Non-Expansive Operators for Data Analysis and Machine Learning". In: *Nature Machine Intelligence* 1.9 (2019), pp. 423–433 (cit. on pp. 5, 21, 22, 24–26, 30–32, 82, 105).

[9] M. Berman, A. R. Triki, and M. B. Blaschko. "The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4413–4421 (cit. on p. 135).

[10] G. Bocchi et al. "A new paradigm for Artificial Intelligence based on Group Equivariant Non-Expansive Operators (GENEOs) applied to protein pocket detection". In: *Proceedings of the Statistics and Data Science Conference*. Pavia University Press. 2023, pp. 152–157 (cit. on pp. 32, 33, 125).

[11] M. M. Bronstein et al. "Geometric deep learning: Grids, groups, graphs, geodesics, and gauges". In: *arXiv preprint arXiv:2104.13478* (2021) (cit. on p. 21).

[12] H. Caesar et al. "nuScenes: a Multimodal Dataset for Autonomous Driving". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11621–11631 (cit. on pp. 1, 15, 36, 134, 137, 139).

[13] P. Cascarano et al. "on The Geometric and Riemannian Structure of The Spaces of Group Equivariant Non-Expansive Operators". In: *Arxiv Preprint Arxiv:2103.02543* (2021) (cit. on pp. 5, 22, 27, 30, 105).

[14] A. X. Chang et al. "ShapeNet: an Information-Rich 3D Model Repository". In: *Arxiv Preprint Arxiv:1512.03012* (2015) (cit. on pp. 15, 36).

[15] Y. Chen et al. "VoxelNext: Fully Sparse VoxelNet for 3D Object Detection and Tracking". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 21674–21683 (cit. on p. 2).

[16] Z. Chen, Y. Bei, and C. Rudin. "Concept whitening for interpretable image recognition". In: *Nature Machine Intelligence* 2.12 (2020), pp. 772–782 (cit. on p. 80).

[17] R. Cheng et al. "AF2-s3net: Attentive Feature Fusion With Adaptive Feature Selection for Sparse Semantic Segmentation Network". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 12547–12556 (cit. on pp. 14, 104).

[18] C. Choy, J. Gwak, and S. Savarese. "4d Spatio-Temporal Convnets: Minkowski Convolutional Neural Networks". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3075–3084 (cit. on p. 9).

[19] T. Cohen and M. Welling. "Group Equivariant Convolutional Networks". In: *International Conference on Machine Learning*. PMLR. 2016, pp. 2990–2999 (cit. on pp. 20, 21).

[20] P. Contributors. "Pointcept: A codebase for point cloud perception research". In: *Github. Available online: https://github. com/Pointcept/Pointcept (accessed on 31 March 2025)* (2023) (cit. on pp. 49, 55, 62–64, 67, 68).

[21] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy. "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds". In: *International Symposium on Visual Computing*. Springer. 2020, pp. 207–222 (cit. on p. 95).

[22] G. Curnis, S. Fontana, and D. G. Sorrenti. "Gtasynth: 3D Synthetic Data of Outdoor Non-Urban Environments." In: *Data in Brief* 43 (2022), p. 108412 (cit. on pp. 36, 37).

[23] A. Dai et al. "Scannet: Richly-Annotated 3D Reconstructions of Indoor Scenes". In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5828–5839 (cit. on pp. 1, 7, 14, 15, 36, 43, 134, 137, 139).

[24] L. Ding, J. Wang, and Y. Wu. "Electric power line patrol operation based on vision and laser SLAM fusion perception". In: *2021 IEEE 4th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*. IEEE. 2021, pp. 125–129 (cit. on p. 80).

[25] F. Doshi-Velez and B. Kim. "Towards a rigorous science of interpretable machine learning". In: *arXiv preprint arXiv:1702.08608* (2017) (cit. on p. 105).

[26] Y. Feng et al. "Gvcnn: Group-View Convolutional Neural Networks for 3D Shape Recognition". In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 264–272 (cit. on pp. 2, 8).

[27] L. Ferrari et al. "A topological model for partial equivariance in deep learning and data analysis". In: *Frontiers in artificial intelligence* 6 (2023), p. 1272619 (cit. on p. 26).

[28] M. Finzi, M. Welling, and A. G. Wilson. "A practical method for constructing equivariant multilayer perceptrons for arbitrary matrix groups". In: *International conference on machine learning*. PMLR. 2021, pp. 3318–3328 (cit. on p. 21).

[29] M. Finzi et al. "Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data". In: *International conference on machine learning*. PMLR. 2020, pp. 3165–3176 (cit. on p. 20).

[30] S. Folga et al. "National electricity emergency response capabilities". In: *Office Energy Policy Syst. Anal., US Dept. Energy, Washington, DC, USA, Rep* (2016) (cit. on p. 34).

[31] R. C. Fong and A. Vedaldi. "Interpretable explanations of black boxes by meaningful perturbation". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 3429–3437 (cit. on p. 79).

[32] P. Frosini. "G-invariant persistent homology". In: *Mathematical Methods in the Applied Sciences* 38.6 (2015), pp. 1190–1199 (cit. on p. 24).

[33] F. Fuchs et al. "Se (3)-Transformers: 3D Roto-Translation Equivariant Attention Networks". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1970–1981 (cit. on pp. 20, 21).

[34] R. Geirhos et al. "Shortcut learning in deep neural networks". In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673 (cit. on p. 21).

[35] A. Goyal and Y. Bengio. "Inductive Biases for Deep Learning of Higher-Level Cognition". In: *Proceedings of The Royal Society a* 478.2266 (2022), p. 20210068 (cit. on pp. 16, 21).

[36] B. Graham, M. Engelcke, and L. Van Der Maaten. "3D semantic segmentation with submanifold sparse convolutional networks". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2018, pp. 9224–9232 (cit. on p. 95).

[37] F. Groh, P. Wieschollek, and H. P. Lensch. "Flex-Convolution: Million-Scale Point-Cloud Learning Beyond Grid-Worlds". In: *Asian Conference on Computer Vision*. Springer. 2018, pp. 105–122 (cit. on p. 18).

[38] R. Guidotti et al. "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42 (cit. on p. 105).

[39] T. Guo et al. "Research on point cloud power line segmentation and fitting algorithm". In: *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. Vol. 1. IEEE. 2019, pp. 2404–2409 (cit. on p. 80).

[40] Y. Guo et al. "Deep Learning for 3D Point Clouds: a Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.12 (2020), pp. 4338–4364 (cit. on pp. 9, 10).

[41] L. T. Hai et al. "Topology-Guided Knowledge Distillation for Efficient Point Cloud Processing". In: *arXiv preprint arXiv:2505.08101* (2025) (cit. on pp. 93, 95).

[42] K. He et al. *Deep Residual Learning for Image Recognition. Corr Abs/1512.03385 (2015)*. 2015 (cit. on p. 8).

[43] P. Hermosilla et al. "Monte Carlo Convolution for Learning on Non-Uniformly Sampled Point Clouds". In: *ACM Transactions on Graphics (TOG)* 37.6 (2018), pp. 1–12 (cit. on pp. 11, 18).

[44] Q. Hu et al. "Randla-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11108–11117 (cit. on pp. 2, 49, 55, 62–64, 67, 68, 95, 114, 115).

[45] B. Jalil et al. "Fault detection in power equipment via an unmanned aerial system using multi modal data". In: *Sensors* 19.13 (2019), p. 3014 (cit. on p. 36).

[46] R. Jenssen, D. Roverso, et al. "Intelligent monitoring and inspection of power line components powered by UAVs and deep learning". In: *IEEE Power and energy technology systems journal* 6.1 (2019), pp. 11–21 (cit. on p. 35).

[47] L. Kong et al. "Rethinking Range View Representation for LiDAR Segmentation". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2023, pp. 228–240 (cit. on pp. 2, 95).

[48] X. Lai et al. "Stratified Transformer for 3D Point Cloud Segmentation". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 8500–8509 (cit. on pp. 2, 124).

[49] X. Lai et al. "Spherical Transformer for LiDAR-Based 3D Recognition". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 17545–17555 (cit. on p. 2).

[50] A. H. Lang et al. "PointPillars: Fast Encoders for Object Detection from Point Clouds". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 12697–12705 (cit. on pp. 50, 52).

[51] D. Lavado et al. "Learning under Noisy Labels Spurious Points and Diverse Structures: TS40K a 3D Point Cloud Dataset of Rural Terrain and Electrical Transmission Systems". In: *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*. 2025-02, pp. 7326–7336 (cit. on pp. 38, 88, 101, 123, 134, 137).

[52] D. Lavado et al. "SCENE-Net: Geometric Induction for Interpretable and Low-Resource 3D Pole Detection with Group-Equivariant Non-Expansive Operators". In: *Computer Vision and Image Understanding* (2025) (cit. on pp. 77, 125).

[53] D. R. M. M. Lavado. "Detection of Power Line Supporting Towers Via Interpretable Semantic Segmentation of 3d Point Clouds". MA thesis. Universidade NOVA de Lisboa (Portugal), 2022 (cit. on p. 77).

[54] F. J. Lawin et al. "Deep Projective 3D Semantic Segmentation". In: *Computer Analysis of Images and Patterns: 17th International Conference, Caip 2017, Ystad, Sweden, August 22-24, 2017, Proceedings, Part I 17*. Springer. 2017, pp. 95–107 (cit. on pp. 2, 8).

[55] H. Lei, N. Akhtar, and A. Mian. "Octree Guided CNN With Spherical Kernels for 3D Point Clouds". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9631–9640 (cit. on pp. 10, 18).

[56] H. Lei, N. Akhtar, and A. Mian. "Spherical Kernel for Efficient Graph Convolution on 3D Point Clouds". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.10 (2020), pp. 3664–3680 (cit. on pp. 11, 18).

[57] M. Y. Levi and G. Gilboa. "Fast and simple explainability for point cloud networks". In: *arXiv preprint arXiv:2403.07706* (2024) (cit. on p. 79).

[58] Y. Li et al. "Pointcnn: Convolution on X-Transformed Points". In: *Advances in Neural Information Processing Systems* 31 (2018) (cit. on pp. 2, 11, 18, 124, 126).

[59] Y. Li and J. Ibanez-Guzman. "LiDAR for Autonomous Driving: The Principles, Challenges, and Trends for Automotive LiDAR and Perception Systems". In: *IEEE Signal Processing Magazine* 37.4 (2020), pp. 50–61 (cit. on p. 7).

[60] T.-Y. Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988 (cit. on p. 135).

[61] Z. C. Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57 (cit. on pp. 105, 113).

[62] Y.-J. Liu et al. "Cylinder detection in large-scale point cloud of pipeline plant". In: *IEEE transactions on visualization and computer graphics* 19.10 (2013), pp. 1700–1707 (cit. on p. 80).

[63] Y. Liu et al. "Uniseg: a Unified Multi-Modal LiDAR Segmentation Network and The Openpcseg Codebase". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2023, pp. 21662–21673 (cit. on pp. 14, 95, 104).

[64] Z. Liu et al. "Point-Voxel CNN for Efficient 3D Deep Learning". In: *Advances in Neural Information Processing Systems* 32 (2019) (cit. on p. 14).

[65] I. Loshchilov. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017) (cit. on p. 135).

[66] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[67] Z. Lu et al. "Slicing-tracking-detection: Simultaneous multi-cylinder detection from large-scale and complex point clouds". In: *IEEE Transactions on Visualization and Computer Graphics* 28.12 (2021), pp. 4172–4185 (cit. on p. 81).

[68] Y. Lyu, X. Huang, and Z. Zhang. "Learning to Segment 3D Point Clouds in 2d Image Space". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12255–12264 (cit. on p. 8).

[69] X. Ma et al. "Rethinking network design and local geometry in point cloud: A simple residual MLP framework". In: *arXiv preprint arXiv:2202.07123* (2022) (cit. on p. 11).

[70] I. Maduako et al. "Deep learning for component fault detection in electricity transmission lines". In: *Journal of Big Data* 9.1 (2022), p. 81 (cit. on p. 36).

[71] S. Marconi et al. "a Data Science Challenge for Converting Airborne Remote Sensing Data Into Ecological Information". In: *PeerJ* 6 (2019), e5843 (cit. on pp. 36, 37).

[72] D. Maturana and S. Scherer. "Voxnet: a 3D Convolutional Neural Network for Real-Time Object Recognition". In: *2015 IEEE/Rsj International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 922–928 (cit. on pp. 2, 9, 123).

[73] H.-Y. Meng et al. "Vv-Net: Voxel Vae Net With Group Convolutions for Point Cloud Segmentation". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2019, pp. 8500–8508 (cit. on p. 123).

[74] K. Mo et al. "Partnet: a Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 909–918 (cit. on pp. 15, 36).

[75] J. W. Muhs and M. Parvania. “Stochastic spatio-temporal hurricane impact analysis for power grid resilience studies”. In: *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE. 2019, pp. 1–5 (cit. on p. 34).

[76] C. Park et al. “Fast Point Transformer”. In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 16949–16958 (cit. on p. 2).

[77] B. Peng et al. “Oa-cnns: Omni-adaptive sparse cnns for 3d semantic segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 21305–21315 (cit. on pp. 123, 135).

[78] C. R. Qi et al. “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation”. In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 652–660 (cit. on pp. 1, 2, 10, 49, 55, 62–64, 67, 68, 95, 114, 115, 123, 134, 137, 139).

[79] C. R. Qi et al. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: *Advances in Neural Information Processing Systems* 30 (2017) (cit. on pp. 2, 10, 49, 55, 62–64, 67, 68, 89, 95, 114, 115, 123, 134, 137, 139).

[80] G. Qian et al. “Pointnext: Revisiting PointNet++ With Improved Training and Scaling Strategies”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 23192–23204 (cit. on pp. 2, 10).

[81] M. T. Ribeiro, S. Singh, and C. Guestrin. “" Why should i trust you?" Explaining the predictions of any classifier”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144 (cit. on p. 79).

[82] C. Rudin. “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead”. In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215 (cit. on pp. 79, 80, 113).

[83] S. S. M. Salehi, D. Erdogmus, and A. Gholipour. “Tversky loss function for image segmentation using 3D fully convolutional deep networks”. In: *International workshop on machine learning in medical imaging*. Springer. 2017, pp. 379–387 (cit. on p. 135).

[84] V. G. Satorras, E. Hoogeboom, and M. Welling. “E (N) Equivariant Graph Neural Networks”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9323–9332 (cit. on pp. 20, 21).

[85] S. Shi, X. Wang, and H. Li. “Pointrcnn: 3D Object Proposal Generation and Detection from Point Cloud”. In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 770–779 (cit. on pp. 50, 52).

[86] S. Shi et al. “from Points to Parts: 3D Object Detection from Point Cloud With Part-Aware and Part-Aggregation Network”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.8 (2020), pp. 2647–2664 (cit. on pp. 50, 52).

[87] S. Shi et al. "Pv-Rcnn: Point-Voxel Feature Set Abstraction for 3D Object Detection". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10529–10538 (cit. on pp. 50, 52).

[88] M. Simonovsky and N. Komodakis. "Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs". In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3693–3702 (cit. on p. 18).

[89] H. Song, W. Choi, and H. Kim. "Robust Vision-Based Relative-Localization Approach Using an RGB-Depth Camera and LiDAR Sensor Fusion". In: *IEEE Transactions on Industrial Electronics* 63.6 (2016), pp. 3725–3736 (cit. on p. 7).

[90] M. Steininger et al. "Density-based weighting for imbalanced regression". In: *Machine Learning* 110.8 (2021), pp. 2187–2211 (cit. on pp. 87, 112).

[91] H. Su et al. "Multi-View Convolutional Neural Networks for 3D Shape Recognition". In: *Proceedings of The IEEE International Conference on Computer Vision*. 2015, pp. 945–953 (cit. on pp. 2, 8).

[92] P. Sun et al. "Scalability in Perception for Autonomous Driving: Waymo Open Dataset". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2446–2454 (cit. on pp. 15, 43).

[93] H. Tang et al. "Searching efficient 3d architectures with sparse point-voxel convolution". In: *European conference on computer vision*. Springer. 2020, pp. 685–702 (cit. on p. 95).

[94] G. Tao et al. "Study on segmentation algorithm with missing point cloud in power line". In: *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE. 2019, pp. 1895–1899 (cit. on p. 80).

[95] O. D. Team. *OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds*. `https://github.com/open-mmlab/OpenPCDet`. 2020 (cit. on p. 50).

[96] H. Thomas et al. "Kpconv: Flexible and Deformable Convolution for Point Clouds". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2019, pp. 6411–6420 (cit. on pp. 2, 10, 11, 18, 49, 55, 62–64, 67, 68, 89, 95, 114, 115, 124, 134, 137, 139).

[97] N. Thomas et al. "Tensor Field Networks: Rotation-and Translation-Equivariant Neural Networks for 3D Point Clouds". In: *Arxiv Preprint Arxiv:1802.08219* (2018) (cit. on p. 20).

[98] J. Toth and A. Gilpin-Jackson. "Smart view for a smart grid—Unmanned Aerial Vehicles for transmission lines". In: *2010 1st international conference on applied robotics for the power industry*. IEEE. 2010, pp. 1–6 (cit. on p. 35).

[99] J. Toth, N. Pouliot, and S. Montambault. "Field experiences using LineScout Technology on large BC transmission crossings". In: *2010 1st International Conference on Applied Robotics for the Power Industry*. IEEE. 2010, pp. 1–6 (cit. on p. 35).

[100] T.-T. Tran, V.-T. Cao, and D. Laurendeau. "Extraction of cylinders and estimation of their parameters from point clouds". In: *Computers & Graphics* 46 (2015), pp. 345–357 (cit. on p. 80).

[101] J. Trochta et al. "3D forest: an Application for Descriptions of Three-Dimensional forest Structures Using Terrestrial LiDAR". In: *PLOS ONE* 12.5 (2017), e0176871 (cit. on pp. 36, 37, 39).

[102] M. A. Uy et al. "Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 1588–1597 (cit. on p. 11).

[103] N. Varney, V. K. Asari, and Q. Graehling. "DALES: a Large-Scale Aerial LiDAR Data Set for Semantic Segmentation". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 186–187 (cit. on pp. 36, 37, 39).

[104] a. Vaswani. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems* (2017) (cit. on pp. 2, 12, 124).

[105] L. Wang et al. "Graph Attention Convolution for Point Cloud Semantic Segmentation". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10296–10305 (cit. on p. 18).

[106] P.-S. Wang. "Octformer: Octree-Based Transformers for 3D Point Clouds". In: *ACM Transactions on Graphics (TOG)* 42.4 (2023), pp. 1–11 (cit. on pp. 2, 13, 124).

[107] P.-S. Wang et al. "O-CNN: Octree-Based Convolutional Neural Networks for 3D Shape Analysis". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–11 (cit. on p. 10).

[108] S. Wang et al. "Deep Parametric Continuous Convolutional Neural Networks". In: *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2589–2597 (cit. on p. 18).

[109] W. Wu, Z. Qi, and L. Fuxin. "Pointconv: Deep Convolutional Networks on 3D Point Clouds". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9621–9630 (cit. on pp. 11, 18, 124, 126).

[110] X. Wu et al. "Point transformer V2: Grouped Vector Attention and Partition-based Pooling". In: *NeurIPS*. 2022 (cit. on pp. 1, 2, 13, 49, 55, 62–64, 67–69, 89, 95, 104, 114, 115, 124, 134, 135, 137, 139).

[111] X. Wu et al. "Point Transformer V3: Simpler Faster Stronger". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 4840–4851 (cit. on pp. 1, 2, 13, 62–64, 67, 68, 78, 93, 95, 104, 114, 115, 123, 124, 134, 135, 137–139).

[112] J. Xu et al. "Rpvnet: A deep and efficient range-point-voxel fusion network for LiDAR point cloud segmentation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16024–16033 (cit. on p. 95).

[113] Y. Xu et al. "Spidercnn: Deep learning on point sets with parameterized convolutional filters". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 87–102 (cit. on pp. 18, 19, 126).

[114] X. Yan et al. "Sparse single sweep LiDAR point cloud segmentation via learning contextual shape priors from scene completion". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 4. 2021, pp. 3101–3109 (cit. on p. 95).

[115] X. Yan et al. "2DPASS: 2d Priors Assisted Semantic Segmentation on LiDAR Point Clouds". In: *European Conference on Computer Vision*. Springer. 2022, pp. 677–695 (cit. on pp. 14, 104).

[116] Y. Yan, Y. Mao, and B. Li. "Second: Sparsely embedded convolutional detection". In: *Sensors* 18.10 (2018), p. 3337 (cit. on pp. 50, 52).

[117] Z. Yang and L. Wang. "Learning Relationships for Multi-View 3D Object Recognition". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2019, pp. 7505–7514 (cit. on p. 8).

[118] C. Zamuda et al. "US energy sector vulnerabilities to climate change and extreme weather". In: *US Department of Energy, https://energy. gov/sites/prod/files/2013/07/f2/20130716-Energy% 20Sector% 20Vulnerabilities% 20Report.pdf* (2013) (cit. on p. 34).

[119] Q. Zhang, Y. N. Wu, and S.-C. Zhu. "Interpretable Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018-06 (cit. on p. 80).

[120] Y. Zhang et al. "PolarNet: an Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation". In: *Proceedings of The IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9601–9610 (cit. on p. 2).

[121] H. Zhao et al. "Point Transformer". In: *Proceedings of The IEEE/CVF International Conference on Computer Vision*. 2021, pp. 16259–16268 (cit. on pp. 1, 2, 10, 12, 49, 55, 62–64, 67, 68, 95, 114, 115, 124, 134, 135, 137, 139).

[122] X. Zhu et al. "Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation". In: *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*. 2021, pp. 9939–9948 (cit. on p. 95).

[123] X. Zhu et al. "Cylindrical and Asymmetrical 3D Convolution Networks for LiDAR-Based Perception". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (2021), pp. 6807–6822 (cit. on p. 18).

[124] H. Zou and T. Hastie. "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 67.2 (2005), pp. 301–320 (cit. on p. 112).

A

GENEO Non-Expansiveness Proofs

A.1 Cylinder GENEO Non-expansiveness Proof

Theorem 1. *The operator Γ^{ϑ}_{Cy} is non-expansive.*

Proof. Take the definition of our cylinder GENEO as

$$\int_0^h \int_{\mathbb{R}^2} g_{Cy}(x)dxdy = h \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2-r^2)^2} d\tilde{x} \tag{A.1}$$

where $\tilde{x}$ and $\tilde{c}$ are the projections on the first two coordinates. We start by a variable transformation $z = \tilde{x} - \tilde{c}$ yielding, as the inner integral

$$X = \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2-r^2)^2} dx = \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz. \tag{A.2}$$

We now consider the fact that there is a positive scalar $t^\star > 0$, such that for all $t > t^\star$, $t^2 - r^2 > t$. Thus, we break the integral in (A.2) in two terms, one integrating over the disk centered in zero with radius $t^\star$, $B(0, t^\star)$, and another integrating over the complementary region, outside the ball, $\bar{B}(0, t^\star)$

$$X = \underbrace{\int_{B(0,t^\star)} e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz}_{X_0} + \int_{\bar{B}(0,t^\star)} e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz.$$

The first integral, X_0, exists and is bounded by the area of the ball A_B multiplied by the maximum C_0 defined as

$$C_0 = \max_{B(0,t^\star)} e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2}.$$

Here, by the Weierstrass extreme value theorem, C_0 is a finite number because the function to be maximized is a continuous function over a compact set. The second integral can also be bounded. According to the definition of $t^\star$,

$$\|y\| > t^\star \implies \|y\|^2 - r^2 \geq \|y\| \implies e^{-\frac{(\|y\|^2-r^2)^2}{2\sigma^2}} \leq e^{-\frac{\|y\|^2}{2\sigma^2}}.$$

the first inequality entails the second because $\|y\| \geq 0$.

The integral $C_1 = \int_{\bar{B}(0,t^\star)} e^{-\frac{\|y\|^2}{2\sigma^2}} dy$ can be easily computed. This entails that the overall integral is bounded by $C = h(A_B C_0 + C_1)$. The constant C can be used to normalize $g_{Cy}(x)$, making it non-expansive. □

A.2 Arrow GENEO Non-expansiveness Proof

Take the definition of the Arrow GENEO as:

$$\begin{aligned} &\int_0^h \int_{\mathbb{R}^2} g_{Ar}(x) dx dz \\ &= h_c \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2 - r^2)^2} d\tilde{x} \; + \\ &\quad (h - h_c) \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2 - r_h(x))^2} d\tilde{x} \end{aligned} \tag{A.3}$$

where $\tilde{x}$ and $\tilde{c}$ are the projections on the first two coordinates, and $r_h(x) = (h - \pi_3(x))r_c \tan(\beta\pi)$ defines a specific radius for a 3D point at a certain height. In A.1, we have proven that the first integral

$$h_c \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2 - r^2)^2} d\tilde{x}$$

is bounded by a constant $C = h_c(A_B C_0 + C_1)$, where h_c is the height of the cylinder, and $A_B C_0$ and C_1 define bounds when integrating inside and outside the disk $B(0, t^\star)$.

Since, $r_h(x) : \mathbb{R}^3 \to \mathbb{R}$ produces a radius for a given 3D point, the cone defined by the second integral can be encased in a cylinder with radius

$$r_{\max} = \max_{x \in \mathbb{R}^3} (h - \pi_3(x)) r_c \tan(\beta\pi).$$

Thus, we can follow the same rationale as in A.1 to prove that the second integral is also bounded. Specifically

$$(h - h_c) \int_{\mathbb{R}^2} e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2 - r_{\max})^2} d\tilde{x} \leq (h - h_c)(A_B C_2 + C_3), \tag{A.4}$$

where

$$C_2 = \max_{B(0,t^\star)} e^{-\frac{1}{2\sigma^2}(\|z\|^2 - r_{\max}^2)^2}.$$

and

$$C_3 = \int_{\bar{B}(0,t^\star)} e^{-\frac{\|y\|^2}{2\sigma^2}} dy$$

This entails that the overall integral is bounded by $C_{Ar} = C + (h - h_c)(A_B C_2 + C_3)$. The constant C_{Ar} can be used to normalize $g_{Ar}(x)$, making it non-expansive.

A.3 Negative Sphere GENEO Non-expansiveness Proof

Take the definition of the Negative Sphere GENEO as:

$$\begin{aligned} &\int_{\mathbb{R}^3} g_{NS}(x)dx \\ &= \int_{\mathbb{R}^3} -\omega \; e^{\frac{-1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2-r^2)^2} \end{aligned} \tag{A.5}$$

where $\tilde{x}$ and $\tilde{c}$ are the projections on the first two coordinates and $\omega \in (0, 1]$ is a negative factor. Since we are proving Non-expansiveness over the L_1 norm, specifically

$$\int_{\mathbb{R}^3} |g_{NS}(x)|dx < \infty, \tag{A.6}$$

the negative sign of the Negative Sphere can be disregarded. We start by a variable transformation $z = \tilde{x} - \tilde{c}$ yielding, as the inner integral

$$X = \int_{\mathbb{R}^3} \omega e^{-\frac{1}{2\sigma^2}(\|\tilde{x}-\tilde{c}\|^2-r^2)^2} dx = \int_{\mathbb{R}^3} \omega e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz. \tag{A.7}$$

We now consider the fact that there is a positive scalar $t^\star > 0$, such that for all $t > t^\star$, $t^2 - r^2 > t$. Thus, we break the integral in (A.7) in two terms, one integrating over the ball centered in zero with radius $t^\star$, $B(0, t^\star)$, and another integrating over the complementary region, outside the ball, $\bar{B}(0, t^\star)$

$$X = \int_{B(0,t^\star)} -\omega e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz + \int_{\bar{B}(0,t^\star)} \omega e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz.$$

The first integral

$$X_0 = \int_{B(0,t^\star)} \omega e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2} dz$$

exists and is bounded by the volume of the ball V_B multiplied by the maximum C_0 defined as

$$C_0 = \max_{B(0,t^\star)} \omega e^{-\frac{1}{2\sigma^2}(\|z\|^2-r^2)^2}.$$

Here, by the Weierstrass extreme value theorem, C_0 is a finite number because the function to be maximized is a continuous function over a compact set. The second integral can also be bounded. According to the definition of $t^\star$,

$$\|y\| > t^\star \implies \|y\|^2 - r^2 \geq \|y\| \tag{A.8}$$

$$\implies (\|y\|^2 - r^2)^2 \geq \|y\|^2 \tag{A.9}$$

$$\implies -(\|y\|^2 - r^2)^2 \leq -\|y\|^2 \tag{A.10}$$

$$\implies e^{-\frac{(\|y\|^2-r^2)^2}{2\sigma^2}} \leq e^{-\frac{\|y\|^2}{2\sigma^2}} \tag{A.11}$$

$$\implies \omega e^{-\frac{(\|y\|^2-r^2)^2}{2\sigma^2}} \leq \omega e^{-\frac{\|y\|^2}{2\sigma^2}}. \tag{A.12}$$

the first inequality entails the second because $\|y\|$ is a non-negative number. The integral

$$C_1 = \int_{\bar{B}(0,t^\star)} \omega e^{-\frac{\|y\|^2}{2\sigma^2}} dy$$

can be easily computed.

The overall integral is bounded by $C = V_B C_0 + C_1$. The constant C can be used to normalize $g_{NS}(x)$, making it non-expansive.