Invaria: Learning Scale and Density Invariance in Point Clouds via Next-Resolution Prediction

Chun-Peng Chang[1] **Shaoxiang Wang**[2] **Alain Pagani**[3] **Dariu Gavrila**[1] **Holger Caesar**[1]

[1]TU Delft, Cognitive Robotics [2]RPTU [3]DFKI, Augmented Vision

Abstract

Modern image encoders achieve high generalization by decoupling semantic meaning from resolution, an ability yet to be fully realized in the 3D domain. We investigate the failure of 3D point cloud encoders to achieve similar generalization and find that existing models are highly sensitive to sampling resolution and scale changes, leading to significant performance degradation. This sensitivity is a major bottleneck for real-world deployment in robotics, as it suggests models overfit to specific quantization densities and object scales rather than learning invariant semantic features. To mitigate this dependency, we propose Invaria, a point cloud encoder that achieves scale and density invariance through next-resolution prediction and receptive field calibration. While our objective is not the explicit generation of high-resolution point clouds, we find that this training objective encourages the model to learn robust, structural invariants. The resulting encoder achieves significant performance gains during resolution shifts while maintaining high efficiency through a compact model size and reduced token requirements. Specifically, on ScanNet, Invaria achieves a 56.0% higher mIoU at $3\times$ lower resolution and a 20% improvement when the objects scale is reduced by a factor of 3. These gains are achieved with a 45% smaller model size and an average reduction of 40% in input tokens. `https://birdy666.github.io/projects/invaria/`

1 Introduction

Humans possess an inherent ability to recognize objects regardless of scale and resolution. Modern image encoders [1, 2] have successfully mirrored this capability, enabling seamless domain transfer from high-quality public datasets to real-world or simulated environments without requiring retraining or fine-tuning. In 3D point cloud, this ability is equally critical for deployment such as robotics when the device generates non-uniform and low resolution point clouds that vary with sensor quality and hardware/latency constraints [3].

However, despite significant progress in 3D perception [4, 5, 6, 7, 8, 9], our research suggests that existing point cloud methods remain inherently dependent on specific sampling resolutions, which is often defined as the discretization grid size used to quantize local signals. This dependency causes performance degradation whenever the input resolution shifts from the training distribution. In this work we focus on two primary factors contributing to this sensitivity: (1) **Density**: the number of points used to represent an object, which can affect the model's ability to recognize the object [10, 11]; and (2) **Scale**: Operations within the model such as pooling and sparse convolution rely on the grid size to define their effective receptive field [5]; changing the grid size therefore also changes the object size perceived by the model [12]. This highlights another deficiency in existing models: their failure to acquire scale-invariant features, which prevents reliable object recognition when the scale deviates from the training distribution. See Figure 1 for the example.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

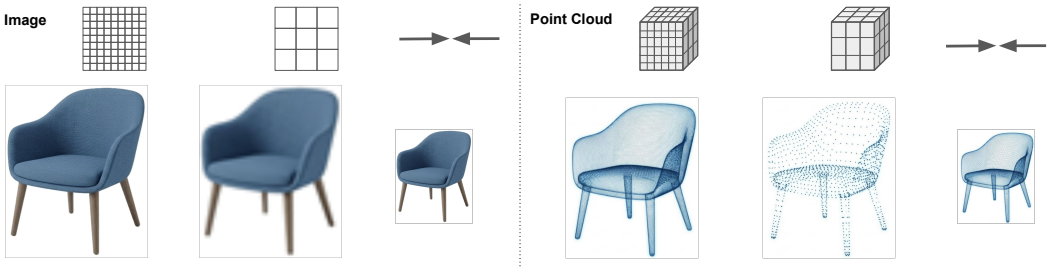


Figure 1: The generalization gap in 3D understanding. (Left) Humans and modern image encoders can seamlessly generalize a semantic concept (such as "chair") across diverse resolutions and sizes (Right). Modern 3D encoders, on the other hand do not demonstrate similar ability, as they fail to understand semantic concepts independently of the underlying point density and scale.

To address these challenges, we propose *Invaria*, a method that prevents models from depending on specific input resolutions and enhances robustness against density and scale changes. Existing encoders often struggle because they easily adopt "shortcuts" during training [13, 8]. Specifically, they learn to recognize objects based on a specific point density or a fixed receptive field defined by the grid size (scale), rather than the object's actual shape. Invaria mitigates this by utilizing a next-resolution prediction training pipeline with receptive field calibration. By predicting the semantic features of higher-resolution targets from sparser inputs in the latent space, the model learns to decouple semantic meaning from specific sampling resolutions. This training objective encourages the encoder to capture the object's fundamental structural invariants, the geometric relationships that define it regardless of resolution or scale.

Beyond robust feature representation, model efficiency, specifically in terms of parameter count and computational overhead, is a critical metric for 3D perception. Following the success of transformers [14, 15, 16, 17], state-of-the-art models [6, 7, 18, 19] have shifted toward transformer-based architectures, which are powerful yet often computationally expensive. Furthermore, as self-supervised learning (SSL) and scaling laws have proven their efficacy, recent works [8, 9, 20] have leveraged SSL and increased model and data scales to enhance performance. While effective, this progress comes at a significant cost: the reliance on massive data and model sizes leads to high computational requirements, complicating real-world deployment on resource-constrained devices.

Conversely, by training with our proposed pipeline, our method demonstrates robust performance across varying resolutions while maintaining competitive performance at the default resolution with a significantly smaller model size. Specifically, when compared to state-of-the-art models on ScanNet, our method achieves a 56.0% higher mIoU at $3\times$ lower resolution (i.e., $3\times$ the default grid size); and 20% higher mIoU when the objects scale is reduced by a factor of 3. Moreover, this resolution-invariant property provides a significant computational advantage: the model can encode high-resolution features from lower-resolution inputs. This asymmetric encoding yields an average reduction of around 40% in input tokens, which directly lowers the cost of transformer-based architectures, whose computation and memory grow with the number of input tokens. Consequently, our approach allows the network to maintain dense semantic representations while significantly lowering the computational overhead associated with the backbones.

In summary, the main contributions of this work are as follows:

- We investigate the sensitivity of existing point cloud encoders to sampling resolution and specifically focus on how the density shortcut and scale dependency can degrade the model's performance.
- We introduce **Invaria**, a point cloud encoder designed to learn scale- and density-invariant features. Specifically, we apply next-resolution prediction with receptive field calibration to encourage the model to align features in the latent space that remain robust as resolution and scale shift.
- In contrast to current methods that rely on increasing data and model scales, Invaria achieves invariant features with a more compact architecture. Moreover, by leveraging resolution-invariant representations we can further reduce input token counts by utilizing lower-resolution point clouds, a crucial advantage for transformer-based backbones.

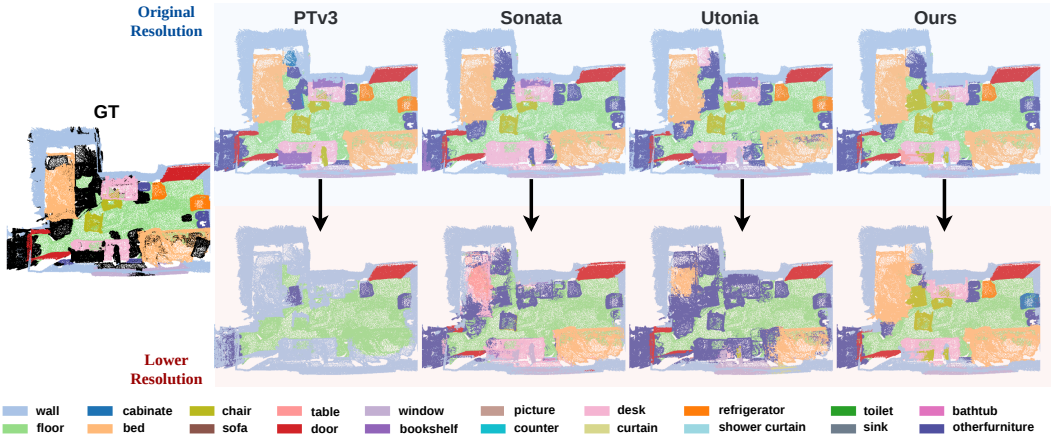


Figure 2: Comparison on ScanNet semantic segmentation. The first row displays model performance when evaluated at the same resolution. The second row illustrates the impact of shifting the resolution during inference. For clear visual comparison, results from both methods are projected on the original point cloud. The black spot in GT means no label. See Section F Figure 8 for more examples.

2 Semantic Consistency and Resolution Shift

The representation of the continuous world in digital form necessitates a discretization process that reduces continuous visual detail into a discrete set of data points. In the 2D domain, this is achieved by sampling a scene at a specific resolution with corresponding number of pixels. Modern image encoders [21, 22, 19, 23, 24] have developed strong capabilities in extracting high-level abstract features that maintain semantic consistency across these variations. Similarly, in 3D, the resolution R is usually defined by the grid size τ used for spatial quantization, which serves as a discretization threshold that quantizes local signals of the continuous scene $\mathcal{C}$ into a single representative point. This sampling process $\mathcal{S}$ forms the point cloud $\mathcal{P} = \mathcal{S}(\mathcal{C}; \tau)$. Consider two point clouds derived from $\mathcal{C}$ with grid size τ_1 and τ_2, a robust model $\mathcal{M}$ should satisfy $\mathcal{M}(\mathcal{S}(\mathcal{C}; \tau_1)) \approx \mathcal{M}(\mathcal{S}(\mathcal{C}; \tau_2))$. However, as shown in Figure 2, we observe that existing 3D point cloud encoders [18, 8, 9] often fail to achieve this semantic consistency. This suggests that models can easily take an easy shortcut and couple semantic features with a specific resolution.

2.1 What Makes the Models Sensitive to Resolution?

The Scale Dependency The first reason is their inherent scale dependency. At first glance, it may seem difficult to associate resolution with scale in objects; however, this dependency is one of the root causes of the model's sensitivity to sampling resolution. The core issue is architectural: operators such as *pooling*, *relative positional encoding*, and *sparse convolution* are tightly coupled to grid size that define the resolution, which they use to quantize coordinates and thereby define the adjacency relations between features at every stage of the model. Because the grid size determines the scale of the receptive field, altering it is functionally **equivalent to rescaling every object in the scene** from the model's point of view.

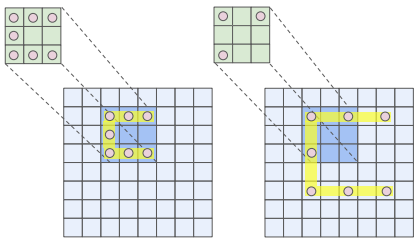


Figure 3: Impact of scale on the relative receptive field of an operator Ω, such as pooling, sparse convolution. While the kernel $\mathcal{K}$ of the operator (green cells) remains a fixed size, the actual spatial volume it aggregates change proportionally with the scale. This coupling causes the model to process inconsistent semantic volumes when the scale changes.

Moreover, humans recognize an object such as a "chair" through its structural invariants, the arrangement of seat, back, and legs, regardless of its size. This form of scale-invariant recognition is a hallmark of genuine semantic understanding, and it is essential at deployment time, where test-time object sizes frequently diverge from those seen in training or are expressed in entirely different metric units.

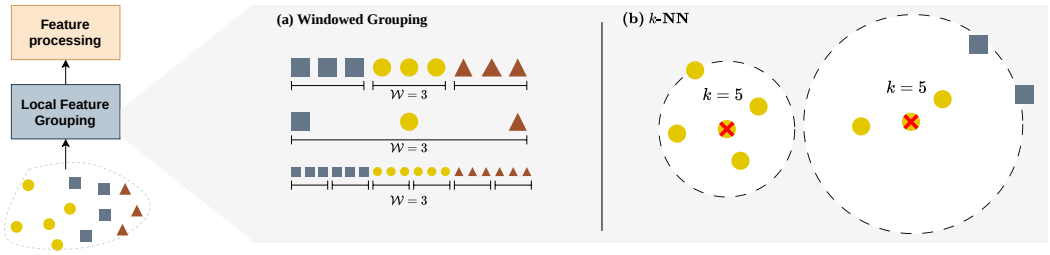


Figure 4: Impact of density on common local feature grouping methods. (a) Due to the $O(N^2)$ complexity of global attention, most transformer-based methods adopt patched-attention as a scalable alternative. This introduces a bottleneck: a fixed patch window $\mathcal{W}$ captures inconsistent semantic volumes depending on the sampling resolution. (b) In k-NN based aggregation, the spatial receptive field is inversely proportional to point density. To capture the same k neighbors in a sparse region, the model must aggregate information from a much larger physical area.

We can formalize this bottleneck by viewing these operations as a generalized neighborhood aggregation. Let $\mathcal{Q}_\tau$ denote a discrete spatial grid with a resolution determined by the grid size τ. For a non empty grid cell $c \in \mathcal{Q}_\tau$, its output feature $\mathcal{F}_c$ is computed as:

$$\mathcal{F}_c = \Omega\left(\left\{f_i \mid \frac{d(c,i)}{\tau} \leq \mathcal{K}\right\}\right) \tag{1}$$

where $\{f_i\}$ is the set of input features, $d(\cdot)$ is a distance metric, $\mathcal{K}$ is a fixed kernel size constant and Ω denotes a operator that maps the neighborhood set to a single feature. In this formulation, the term $d(c,i)/\tau$ represents the normalized grid distance. As illustrated in Figure 3, this creates a direct representational dependency: when the resolution τ or object's size changes, the normalized distance to neighbors changes, effectively altering the volume covered by the fixed kernel $\mathcal{K}$. Consequently, the model perceives a structural change in the object's geometry, providing an easy representational shortcut to learn features that are coupled to a specific scale.

The Density Shortcut The other critical factor affecting the performance of point cloud encoders [4, 18] is density, defined as the number of points sampled from a scene. Existing works with common local feature grouping methods, such as k-NN [6, 7, 25] and window-attention/voxel-partitioning [26, 27, 28, 29, 30, 31], are inherently coupled to point density. As shown in Figure 4, as the number of points used to represent an object changes, the local group captures a different spatial context to satisfy the fixed k requirement. Let $\mathcal{N}_k(x;\mathcal{P})$ denote the set of k neighbors of a query point $x \in \mathcal{P}$. The grouping operation $\mathcal{G}$ then constructs the local point group g_x as:

$$g_x = \mathcal{G}(\{x_j \mid x_j \in \mathcal{N}_k(x;\mathcal{P})\}) \tag{2}$$

Easy Optimization Paths vs. Invariant Features. While deep networks are theoretically expressive enough to adapt to these shifts, they typically favor the most efficient optimization path: a representation "shortcut" that couples semantic features to training-specific densities and scales [13, 8].

3 Achieving Scale and Density Invariance in Point Cloud Encoders

While existing architectures still inevitably rely on the local operators or grouping methods discussed in Section 2, we believe that through a proper training paradigm and simple architecture modification, a model can avoid taking the shortcut and instead learn representations that are invariant to both scale and density. Our proposed method is shown in Figure 5.

3.1 Next-Resolution Prediction for Resolution Invariant Features Understanding

In Invaria, we use a next-resolution prediction objective to help the model learn resolution invariant features. This approach is inspired by the success of Autoregressive (AR) models such as GPT [32, 33] and their recent advancements in the visual domain [34, 35, 36]. Unlike generative models designed for explicit point cloud generation or upsampling, our method utilizes the prediction of a higher resolution point cloud as an objective, forcing the model to learn high-level features across varying resolutions, preventing it from taking the shortcut as discussed in Section 2.

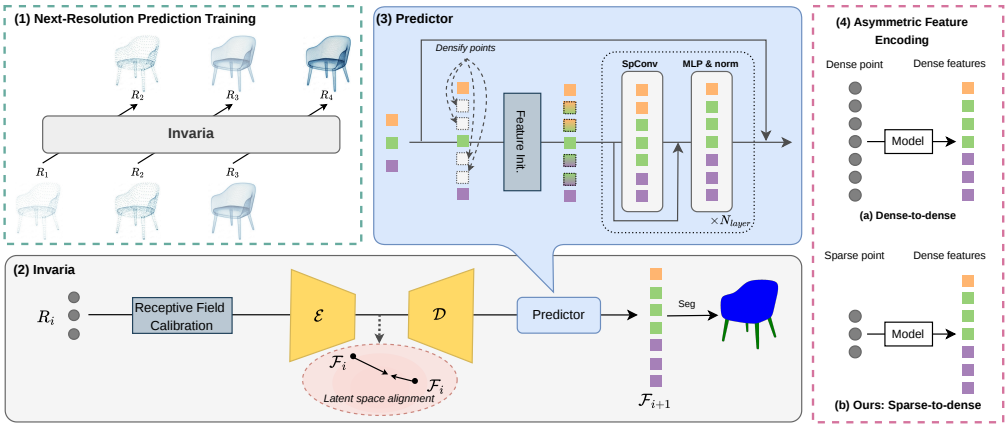


Figure 5: Invaria framework. We employ a next-resolution prediction objective to learn scale- and density-invariant features. This approach decouples semantic features from the input point cloud's resolution with a smaller model size. Leveraging these invariant representations, our model employs asymmetric encoding strategy during inference. This allows a large part of the backbone to process sparser inputs, thereby reducing computation. Unlike conventional point upsampling, which outputs raw points, our approach propagates dense semantic features directly within the latent space.

Receptive Field Calibration. To mitigate the relative receptive field shift inherent in operator Ω (illustrated in Figure 3), we introduce a receptive field calibration approach. Rather than relying on a static grid size to define feature adjacency, we define the internal grid dynamically based on the given point cloud $\mathcal{P} = \{x_j\}_{j=1}^{N}$, where N denotes any number of points sampled to simulate a lower resolution input. First, we draw a small set of N_a anchor points $\mathcal{P}_a \subset \mathcal{P}$ uniformly at random ($N_a \ll N$), and for each anchor $a_i \in \mathcal{P}_a$ we compute its nearest-neighbor distance to its closest distinct neighbor in $\mathcal{P}$ as $d_i = \min_{x \in \mathcal{P} \setminus \{a_i\}} \|a_i - x\|$. The calibrated internal grid size is then obtained by $\psi(\cdot)$, which aggregates these per-anchor distances and rescaling by a constant scaling factor α:

$$\psi(\mathcal{P}) = \alpha \cdot \rho\left(\{d_i\}_{i=1}^{N_a}\right) \tag{3}$$

where $\rho(\cdot)$ is a reduction function, see Appendix Section B for the details of $\rho(\cdot)$. Unlike existing work [20], which attempts to bridge this gap using Granularity-Prompted Rescaling, our approach presents two advantages. First, we do not require prior knowledge of the physical scale to manually normalize the scene to a predefined canonical granularity. Second, by forcing all observations into a single predefined standard scale, such methods fail to capture scale-invariant semantic features and remain sensitive to scenes in which certain objects have a size that differs from those seen during training.

Point Features Extraction. Let $\mathcal{P} = \{x_j\}_{j=1}^{N}$ represent a given point cloud, and let the input sparser point cloud $\mathcal{P}_i$ be a sampled subset such that $\mathcal{P}_i \subset \mathcal{P}$ with cardinality $|\mathcal{P}_i| = N_i$, where $N_i < N$. In this framework, a U-Net-like backbone [37] consisting of an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$ extracts latent features from the sparse input:

$$\mathcal{F}_i = \mathcal{D}(\mathcal{E}(\mathcal{P}_i)) \tag{4}$$

where $\mathcal{F}_i \in \mathbb{R}^{N_i \times C}$ represents the latent feature representation and C denotes the latent feature dimension. Our implementation of $\mathcal{E}$ and $\mathcal{D}$ follows the standard U-Net architecture.

Latent Space Alignment. Receptive field calibration adapts the internal grid to local density, but the backbone is still invoked at different input resolutions. To prevent the model from learning distinct latent spaces for different densities, we align bottleneck features across resolutions. For an input $\mathcal{P}$ presented at M resolutions $\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(M)}$, we obtain a set of bottleneck tokens $\mathcal{Z}^{(m)}$ for each resolution, where each token $z \in \mathcal{Z}^{(m)}$ is associated with a spatial location $c(z) \in \mathbb{R}^3$. For every pair of resolutions (m, l) with $m < l$, we define the spatial nearest neighbor of $z \in \mathcal{Z}^{(m)}$ as $\pi_l(z) = \arg\min_{z' \in \mathcal{Z}^{(l)}} \|c(z) - c(z')\|$. We minimize the cosine distance between matched pairs:

$$\mathcal{L}_{\text{align}} = \frac{1}{\binom{M}{2}} \sum_{m=1}^{M-1} \sum_{l=m+1}^{M} \frac{1}{|\mathcal{Z}^{(m)}|} \sum_{z \in \mathcal{Z}^{(m)}} \Big(1 - \cos\big(z, \pi_l(z)\big)\Big) \tag{5}$$

The Predictor. To synthesize higher-resolution features from the latent space, we utilize a predictor module Φ. Given the latent features $\mathcal{F}_i$ extracted from the sparse input $\mathcal{P}_i$ (as shown in Equation (4)), we densify the point set by adding intermediate points. We determine the spatial locations for these new points using ground-truth higher-resolution coordinates when accessible, or otherwise via interpolation. In either case, the corresponding features are then initialized via nearest-neighbor interpolation from the processed latent features to bridge the sampling density gap. These features are then iteratively processed through N_{layer} refinement blocks (each consisting of sparse convolution, MLP and norm operations) to produce the next higher-resolution features, where $\{\hat{f}_j\}_{j=1}^{N_{i+1}}$ denotes the set of reconstructed higher resolution point features.

$$\mathcal{F}_{i+1} = \Phi(\mathcal{F}_i) = \{\hat{f}_j\}_{j=1}^{N_{i+1}} \tag{6}$$

3.2 Asymmetric Point Feature Encoding.

The High Resolution Trade-off. It is intuitive that a lower-resolution point cloud degrades performance due to the loss of fine-grained information. Surprisingly, higher resolutions do not necessarily improve performance; in contrast, they may degrade accuracy more than lower-resolution inputs due to a reduced receptive field. As shown in Figure 6, for an operation that aggregates a fixed number of k input points, the physical receptive field shrinks significantly as the sampling density increases. For an operation taking k points at the grid size τ, the reach $\mathcal{R}$ scales as:

$$\mathcal{R} \propto \tau \cdot \sqrt[3]{k} \tag{7}$$

This indicates that at higher resolutions, the spatial receptive field for a fixed k is limited, hindering the model's ability to capture long-range structural dependencies. Therefore, we utilize an asymmetric point feature encoding strategy by processing a sparser input.

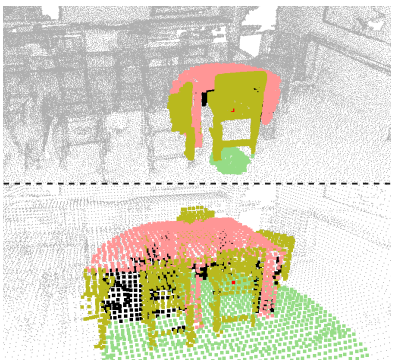


Figure 6: Under the same configuration, the higher-resolution input (top) covers a smaller receptive field.

Asymmetric Inference Strategy Having trained Invaria to learn resolution-invariant representations through the next-resolution prediction training described in Section 3.1, we can capitalize on this robustness during the inference stage by adopting an asymmetric encoding strategy that uses sparser point input to produce higher resolution features. By utilizing sparser, lower-resolution inputs for the U-Net backbone, the model maintains a larger effective receptive field and simultaneously reduces the computational FLOPs required compared to dense processing. See Section B for the lower-resolution sampling setting.

3.3 Training Objective

To ensure robust semantic understanding across varying scales, we supervise the network across M resolution levels. The model takes a point cloud at resolution m as input and is trained to predict the features of the subsequent resolution $m+1$. The total training objective combines a supervised segmentation loss, comprising Cross-Entropy (CE) and Lovász-Softmax (LS) [38], with a consistency alignment term, $\mathcal{L}_{\text{align}}$ formulated in Equation (5), where $\lambda^{(m)}$ denotes the weighting factor for the m-th resolution level, and β is a scaling constant.

$$\mathcal{L} = \sum_{m=1}^{M-1} \lambda^{(m)} \left(\mathcal{L}_{\text{CE}} + \mathcal{L}_{\text{LS}}\right) + \beta \mathcal{L}_{\text{align}} \tag{8}$$

4 Discussion and Analysis

Does performance degradation stem from insufficient geometric cues? No. As shown in Figure 7, training PTv3 [18] from scratch across varying resolutions, and evaluating each on its respective training distribution ($\tau_{train} = \tau_{eval}$), reveals that low resolution point clouds retain sufficient information for accurate recognition. Instead, models take the shortcut and learn resolution-specific features that are coupled with resolution. Moreover, Table 1, Table 2 shows that performance degrades even more when higher-resolution ($\tau = 1$cm) input is provided. We also implemented a lightweight version, *Invaria-s*; please refer to Appendix Section B for further details.

Table 1: Robustness to resolution shifts on ScanNet (val). We evaluate models trained at a default resolution (τ^*=2cm grid size) across varying resolutions. *SSL:* Self-Supervised Learning; *MD:* Joint Training across multiple datasets. †Results using authors' publicly released weights (*linear probing*).

Model	Params	SSL& MD	mIoU (%)				mAcc (%)				allAcc (%)			
			$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$
Sonata†[8]	124.8M	✓	59.9	72.6	62.3	34.2	70.7	83.4	73.8	43.7	84.9	89.8	85.7	72.3
Concrt-L†[9]	207.7M	✓	67.9	**78.6**	72.5	46.0	77.5	**87.4**	82.3	57.3	88.4	92.3	89.9	79.1
Utonia†[20]	157.7M	✓	73.0	76.6	71.7	47.3	81.9	85.1	80.5	56.3	**95.4**	**94.6**	89.2	78.7
Sp.Mink.[4]	39.2M	-	36.6	75.6	47.8	19.4	47.2	83.5	56.0	25.7	73.8	91.5	82.4	66.1
PTv3[18]	46.1M	-	50.5	77.6	60.4	20.5	59.3	85.0	69.0	26.0	80.6	92.0	86.0	65.9
Invaria-s	16.3M	-	73.1	73.3	72.2	70.7	80.2	80.5	79.3	78.1	89.9	89.9	89.5	88.9
Invaria	25.5M	-	**75.5**	76.2	**75.7**	**73.8**	**83.8**	84.3	**84.0**	**82.7**	91.0	91.2	**90.9**	**90.0**

Table 2: Robustness to resolution shifts on outdoor dataset (nuScenes val) (τ^*=5cm). † *linear probing*.

Model	Params	mIoU (%)				mAcc (%)			
		$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$
Sonata†[8]	124.8M	-	66.1	-	-	-	77.2	-	-
Concrt†[9]	207.7M	-	74.2	-	-	-	83.2	-	-
Utonia†[20]	157.7M	-	75.5	-	-	-	84.8	-	-
Sp.Mink.[4]	39.2M	34.1	76.7	35.4	14.5	41.4	82.0	45.7	22.6
PTv3[18]	46.1M	45.1	**80.4**	51.9	26.4	52.9	**86.2**	59.7	33.0
Invaria-s	16.3M	72.9	73.0	71.8	68.6	80.1	80.2	79.8	77.4
Invaria	25.5M	**74.3**	76.2	**75.6**	**71.9**	**82.3**	83.9	**83.4**	**80.1**

Table 3: Evaluation on model size and efficiency of transformer-based methods. † *linear probing*.

Method	Params (M)	FLOPs (G)	Mem. (GB)
Sonata†[8]	124.8	691	1.65
Concrt-L†[9]	207.7	1248	2.49
Utonia†[20]	157.7	874	1.9
PTv3[18]	46.1	352	0.59
Invaria-s	16.3	230	0.31
Invaria	25.5	267	0.54

Can Self-Supervised Training and Scaling Overcome Resolution-Specific Bias? Partially. Self-supervised learning (SSL) [39, 40] and the application of scaling laws [32] have emerged as powerful drivers for enhancing the generalizability of 3D encoders [8, 9, 20]. By pre-training on massive, unlabeled datasets, models can extract semantically rich point features. However, as shown in Figure 7, Table 1, while these methods offer improved robustness to resolution shifts compared to other baselines, this progress remains limited. Crucially, these approaches rely on massive training datasets, prolonged training times, and increased model size. This overhead hinders deployment in real-world robotic environments, where latency and hardware constraints are critical.

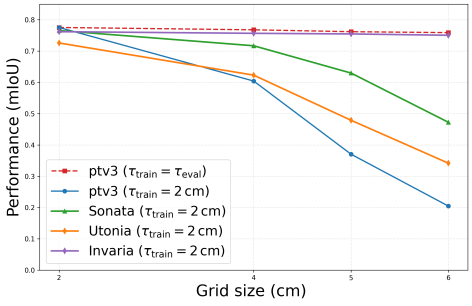


Figure 7: The performance of existing methods when evaluated on different resolutions on ScanNet. The red dashed line represents a model trained and evaluated at the same resolution, showing that even at 6cm, the point cloud retains sufficient information for high-accuracy semantic segmentation.

Model Size and Computational Overhead. As shown in Table 3, our method achieves superior performance compared to the existing works while utilizing about 45% fewer parameters and lower FLOPs. Moreover, our asymmetric feature encoding strategy allows us to reduce the input token count for our U-Net backbones by approximately 40% on average, directly contributing to the overall efficiency of the framework.

Does Resolution Augmentation Resolve Sensitivity? Limited. By testing on standard PTv3 [18] we noticed the following limitations: First, mixing resolutions induces feature interference, resulting in a 2% mIoU drop (77.6% $\rightarrow$ 75.6%) on ScanNet [41] at the native training resolution (τ^*), which suggests conflicting feature representations. Moreover, this improvement seems to be interpolative; when evaluated on denser inputs ($0.5\tau^*$) that fall outside the augmented training range, performance collapses (56.0% vs. our 75.5%). Second, this strategy is inherently unidirectional: while we can easily simulate low-resolution inputs via downsampling, we cannot easily synthesize higher-resolution features that are absent in the original dataset.

Table 4: Analysis on each category's performance on ScanNet (val). We report the relative performance drop (%) on IoU from the default training resolution to $3\times$ lower. The results suggest an even more severe performance drop for complex objects. † *linear probing*.

Method	Wall	Floor	Sofa	Chair	Toilet	Fridge	Counter	Picture	Sink
Sonata†[8]	-16.6%	-6.8%	-74.7%	-52.1%	-90.5%	-72.2%	-87.6%	-71.0%	-78.0%
Concrt-L†[9]	-13.0%	-5.8%	-16.4%	-34.5%	-96.2%	-67.1%	-61.3%	-36.6%	-72.1%
Utonia†[20]	-13.2%	-5.7%	-10.6%	-30.1%	-71.8%	-60.8%	-43.0%	-61.9%	-44.8%
Sp.Mink.[4]	-25.7%	-11.3%	-89.1%	-61.6%	-100%	-100%	-100%	-98.4%	-99.4%
PTv3[18]	-31.0%	-22.2%	-95.1%	-68.1%	-100%	-99.9%	-99.9%	-85.1%	-93.6%
Invaria-s	**-2.1%**	**-0.5%**	**-0.6%**	**-1.6%**	**-2.8%**	-8.7%	-3.8%	**-17.1%**	-4.2%
Invaria	-2.3%	-0.6%	-2.1%	-1.9%	-3.3%	**-5.8%**	**-1.5%**	-28.3%	**-0.3%**

Table 5: Scale change, where the scenes are scaled up or down. † *linear probing*.

Model	**mIoU (%)**			
	$\times 2$	$\times 1$	$\times 1/2$	$\times 1/3$
Sonata†[8]	61.6	72.6	66.9	44.8
Concrt-L†[9]	68.2	**78.6**	75.4	61.4
Utonia†[20]	74.4	76.6	69.7	42.2
Sp.Mink.[4]	33.4	75.6	42.9	14.7
PTv3[18]	47.2	77.6	23.2	9.4
Invaria-s	73.9	73.3	74.0	73.9
Invaria	**75.6**	76.2	**75.6**	**73.8**

Table 6: Robustness of different model families to resolution shifts on ScanNet (val), reported in mIoU (%). *Retention* is the mIoU at $3\tau^*$ relative to the mIoU at the default resolution τ^*.

Model	**Family**	**Neighborhood**	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$	**Retention**
PointNet[42]	Point-based	global max-pooling	-	36.7	35.8	35.1	95.8%
PointNet++[10]	Point-based	fixed-radius ball	-	50.5	48.7	40.6	80.5%
KPConv[43]	Kernel-based	radius tied to grid	-	69.7	54.8	31.5	45.3%
Sp.Mink.[4]	Sparse voxel conv.	grid kernel	36.6	75.6	47.8	19.4	25.7%
PTv2[7]	Transformer	k-NN	55.4	75.4	56.0	34.1	45.2%
PTv3[18]	Transformer	serialized patch	50.5	**77.6**	60.4	20.5	26.4%
Invaria	Transformer	serialized patch	**75.5**	76.2	**75.7**	**73.8**	**96.8%**

What kind of object classes suffer the most? While aggregate metrics such as mIoU provide a high-level overview, a per-class analysis reveals significant performance disparities across categories (Table 4). Large, structurally simple categories such as *Wall* and *Floor* exhibit high resilience across all methods, as their planar geometry remains identifiable even under $3\times$ lower resolution. However, deeper investigation into the most affected classes reveals a severe performance collapse in two distinct groups. First, complex geometries such as *Toilet*, *Sofa*, and *Sink* suffer near-total failure in baselines such as PTv3 [18] and Sp.Mink. [4], with accuracy drops reaching 90% to 100%. Second, surprisingly simple planar objects, such as *Picture* and *Fridge*, also demonstrate significant degradation. Our analysis suggests this stems from geometric ambiguity: as resolution or scale shifts, these objects lose their distinguishing features and are easily misclassified as *Wall*.

Scale and Density Analysis. We evaluate model robustness to scale changes (while maintaining the original number of points). As shown in Table 5, existing methods are remarkably brittle to scale variations; even models trained with SSL approaches on larger datasets cannot mitigate this issue. Furthermore, Appendix Table 11 examines density shifts (where the scale remains constant but the number of points is reduced), demonstrating how sparser point cloud representations lead to performance degradation.

Are Other Model Families More Robust? Beyond transformer-based encoders, we evaluate representatives of other model families under the same resolution shifts (Table 6). A natural alternative to k-NN grouping (Figure 4) is a fixed metric radius, as used in the set-abstraction layers of PointNet++ [10]. Although the physical receptive field then stays constant, PointNet++ still degrades under resolution shifts: the radius remains the same, but the number of points falling inside it changes, and so do the features built from them. KPConv [43], whose kernel radius is tied to the grid size, retains only 45.3% of its performance, consistent with the scale dependency discussed in Section 2.1. In contrast, PointNet [42], which aggregates features through global max-pooling instead of a grid-defined local neighborhood, is remarkably robust (95.8% retention), although its accuracy at the default resolution is far lower.

However, high retention can overstate robustness when the accuracy at the default resolution is low, since such models obtain most of their mIoU from large, geometrically simple classes that are inherently robust to resolution changes. For example, PointNet++ retains 98.1% and 92.4% of its IoU on *Floor* and *Wall* at $3\tau^*$, but only 46.9% and 48.1% on *Shower Curtain* and *Door*.

Table 7: Robustness to different sampling distributions on ScanNet (val), reported in mIoU (%). Each sampling scheme keeps approximately the same number of points as voxel subsampling at $3\tau^*$.

Model	Default	Uniform random	FPS	Locally varying
Sp.Mink.[4]	75.6	15.7	18.7	15.9
PTv3[18]	**77.6**	15.4	19.4	15.3
Invaria	76.2	**71.2**	**72.6**	**71.0**

Table 8: Ablation studies on ScanNet [41] val. Due to space constraints, results for (c), (f), (g), and (h) are reported using the default grid size $\tau^* = 2$cm. The final selections are indicated by green.

$\mathcal{G}(\cdot)$	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$
w	**75.5**	**76.2**	**75.7**	**73.8**
w/o	55.7	76.0	74.5	69.7

(a) **Receptive Field Calibration.** As discussed in Equation (3), without this calibration, the model's performance drops as resolution changes. (mIoU)

NRP	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$
w	**75.5**	76.2	**75.7**	**73.8**
w/o	45.4	**76.5**	21.7	9.5

(b) **Next Resolution Prediction.** Comparison between the proposed model and a baseline trained from scratch without the next-resolution prediction pretext task.

Focal loss	mIoU	mAcc
w	75.1	82.9
w/o	**76.2**	**84.3**

(c) **Focal Loss.** To address category imbalance, we experimented with Focal Loss; however, it does not improve the performance.

$\lambda^{(m)}$	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$
Uniform	75.2	75.9	75.7	**74.0**
Increase	**75.5**	**76.2**	**75.7**	73.8

(d) **Loss weight.** In Equation (8) we found that assigning higher weight to higher resolution increases overall performance. (mIoU)

$\mathcal{L}_{align}$	$0.5\tau^*$	τ^*	$2\tau^*$	$3\tau^*$
w	**75.5**	**76.2**	**75.7**	**73.8**
w/o	74.9	75.4	75.0	73.3

(e) **Latent space alignment.** Adding $\mathcal{L}_{align}$ helps the model perform across all tested resolution. (mIoU)

# btlnck layer	0	1	2
mIoU	74.3	**76.2**	76.0
mAcc	81.8	**84.3**	83.1

(f) **Number of bottleneck layer.** The bottleneck layer often significantly affects model size but not always provide better performance.

norm	BN	LN
mIoU	74.8	**76.2**
mAcc	82.7	**84.3**

(g) **Normalization.** Batch norm is sensitive to batch size and degrade final performance compared to layer norm.

LR Ramp %	5%	10%
mIoU	75.5	**76.2**
mAcc	82.7	**84.3**

(h) **LR Warm-up Ratio**. Smaller warm-up ratios lead to training instability and degrade final performance.

Robustness to Sampling Distribution. A real-world density change is not necessarily uniform. We therefore evaluate three sampling schemes that keep the same number of points as voxel subsampling at $3\tau^*$ (Table 7). *Uniform random* sampling keeps every point with the same probability $p = 0.115$, independent of its position, so the existing density variation of the scene is preserved and simply thinned. *Farthest point sampling* (FPS) iteratively selects the point at maximum distance from those already chosen, driving the spacing toward uniform. *Locally varying* sampling modulates p smoothly across the scene between 0.05 and 0.19 with the same mean, resulting in a $3.4\times$ density ratio within a single scene that simulates locally sparse and dense regions. Invaria remains robust under all three schemes, whereas the baselines degrade severely, showing that its robustness extends beyond global density reduction to changes in the sampling distribution and strong local density variation.

Ablations. We conducted comprehensive ablation studies to evaluate the contribution of each component to our model's performance (Table 8). As shown in Tables 8a and 8b, both our receptive field calibration and the next-resolution prediction training are critical for cultivating resolution-invariant representations. To address data imbalance, we attempted to utilize focal loss[44] in Table 8c to dynamically adjust class weights. However, while this approach initially accelerated mIoU convergence, it ultimately led to lower final performance. Analysis reveals that the model tended to over-predict challenging classes, such as windows and pictures, thereby increasing false positives and degrading overall IoU. Furthermore, Table 8d shows that assigning higher weights to higher-resolution prediction targets yields superior results, while Table 8e confirms the effectiveness of our alignment loss (Equation (5)). Finally, we investigate hyperparameter sensitivity regarding bottleneck capacity, normalization techniques, and learning rate warm-up in Tables 8f to 8h.

5 Related Work

Point Cloud Backbones and Local Neighborhoods. Point cloud backbones differ mainly in how they define the local neighborhood over which features are aggregated, and this choice determines how they respond to resolution shifts. PointNet [42] processes raw coordinates with shared MLPs and aggregates them through global max-pooling, which makes it robust to missing input points. PointNet++ [10] and PointNeXt [12] group points hierarchically within a fixed metric radius, whereas DGCNN [45] and Point Transformer v1/v2 [6, 7] rely on k-nearest neighbors, so the physical extent of the neighborhood changes with density. KPConv [43] and sparse voxel convolutions [4] tie the kernel extent directly to the grid size. More recent backbones move toward efficient structured representations: OctFormer [5] applies attention within octree-based windows, PTv3 [18] serializes points with space-filling curves into fixed-size patches, and LitePT [46] restricts attention to deep, low-resolution stages while using convolutions for early local feature extraction. In all of these designs, the neighborhood is defined by a fixed number of points, a fixed radius, or the grid size, so either its physical extent or the number of points it contains changes when the input resolution shifts (Table 6). Rather than proposing a new neighborhood definition, Invaria keeps a serialized transformer backbone and calibrates its receptive field to the input point cloud.

Scale- and Density-Invariant Features. Scale invariance has long been studied in 2D vision, where hand-crafted descriptors such as SIFT [47] and SURF [48] detect keypoints in a scale space so that the same structure is described consistently across image scales. In deep point cloud learning, PointNet++ [10] addresses non-uniform sampling density with multi-scale and multi-resolution grouping, and PointClustering [49] learns representations that are invariant to transformations of the input during unsupervised pre-training. Sonata [8] further identifies a "geometric shortcut" in self-supervised point cloud learning, where models rely on low-level spatial cues rather than high-level semantics (see Section C for a broader overview of self-supervised pre-training). Compared to 2D vision, however, dedicated scale- and density-invariant representations for 3D point clouds remain scarce, and existing approaches mostly focus on invariant local descriptors or on invariance to transformations of the input. In contrast, Invaria targets scale and density robustness at the level of the backbone representation, through next-resolution prediction and receptive field calibration.

Next-Resolution Prediction The success of Large Language Models in next-token prediction [32, 33] has inspired a shift toward generative paradigms in visual and 3D domains. Recent frameworks have adopted next-resolution prediction. This approach, popularized by the Visual Autoregressive (VAR) modeling strategy [34], treats generation as a progressive refinement from coarse global structures to fine-grained geometric details. This transition to multi-scale tokenization is exemplified by Infinity [35], which introduces bitwise autoregressive modeling to scale high-resolution synthesis by predicting hierarchical bit-representations. In the 3D domain, PointNSP [50] and SAR3D [51] adapt this philosophy by re-framing 3D generation as a sequence of nested-scale predictions. While these methods use coarse-to-fine prediction for generation, Invaria uses next-resolution prediction as a representation-learning objective: by predicting higher-resolution features from a lower-resolution input, the encoder is encouraged to learn semantic features that remain consistent across resolutions.

6 Limitations

Invaria still struggles with simple planar objects such as doors and windows, which are easily confused with walls, and is affected by the class imbalance of existing 3D datasets. Moreover, on ScanNet and nuScenes, its accuracy drops sharply below roughly 5–6% of the original points, where the geometry itself is largely lost. See Sections D and E for details.

7 Conclusion

We introduce Invaria, a 3D point cloud encoder that achieves scale and density invariance through next-resolution prediction and receptive field calibration. Our experiments demonstrate that while lower resolutions predictably degrade performance, higher-resolution inputs can paradoxically reduce accuracy as well. By extracting invariant features from sparser inputs, Invaria's asymmetric encoding strategy delivers robust performance across resolution shifts while significantly reducing computational overhead and model size. While the model currently excels at learning scale- and density-invariant features, exciting avenues for future work include better disambiguating simple planar geometries (such as differentiating doors or pictures from walls) and leveraging recent advances in generative modeling to address the severe class imbalances found in standard 3D datasets.

Acknowledgments and Disclosure of Funding

The Cynergy4MIE project is supported by the Chips Joint Undertaking and its members, including the top-up funding by National Authorities under Grant Agreement No 101140226. This work has also been partially funded by the German Research Foundation (DFG) project ARROW (GA Nr 564809505) and by the EU project SUBLIMITY (GA Nr 101294363). The authors declare no competing interests.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11713–11723, 2020.

[4] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3075–3084, 2019.

[5] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)*, 42(4):1–11, 2023.

[6] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[7] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022.

[8] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22193–22204, 2025.

[9] Yujia Zhang, Xiaoyang Wu, Yixing Lao, Chengyao Wang, Zhuotao Tian, Naiyan Wang, and Hengshuang Zhao. Concerto: Joint 2d-3d self-supervised learning emerges spatial representations. In *NeurIPS*, 2025.

[10] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[11] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7421–7430, 2019.

[12] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in neural information processing systems*, 35:23192–23204, 2022.

[13] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14496–14506, 2023.

[16] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[17] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[18] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[20] Yujia Zhang, Xiaoyang Wu, Yunhan Yang, Xianzhe Fan, Han Li, Yuechen Zhang, Zehao Huang, Naiyan Wang, and Hengshuang Zhao. Utonia: Toward one encoder for all point clouds. In *ICML*, 2026.

[21] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[23] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[24] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[25] Zetong Yang, Li Jiang, Yanan Sun, Bernt Schiele, and Jiaya Jia. A unified query-based paradigm for point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8541–8551, 2022.

[26] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8458–8468, 2022.

[27] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16949–16958, 2022.

[28] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *European Conference on Computer Vision*, pages 426–442. Springer, 2022.

[29] Cheng Zhang, Haocheng Wan, Xinyi Shen, and Zizhao Wu. Patchformer: An efficient point transformer with patch attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11799–11808, 2022.

[30] Zhijian Liu, Xinyu Yang, Haotian Tang, Shang Yang, and Song Han. Flatformer: Flattened window attention for efficient point cloud transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1200–1211, 2023.

[31] Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *Computational Visual Media*, 11(1):83–101, 2025.

[32] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[33] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI Blog*, 2018.

[34] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[35] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15733–15744, 2025.

[36] Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. *arXiv preprint arXiv:2511.04675*, 2025.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[38] Maxim Berman, Amal Rannen Triki, and Matthew 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*, pages 4413–4421, 2018.

[39] Yatian Pang, Eng Hock Francis Tay, Li Yuan, and Zhenghua Chen. Masked autoencoders for 3d point cloud self-supervised learning. *World Scientific Annual Review of Artificial Intelligence*, 1:2440001, 2023.

[40] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[41] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[42] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[43] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

[44] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[45] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[46] Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. LitePT: Lighter Yet Stronger Point Transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

[47] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[48] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[49] Fuchen Long, Ting Yao, Zhaofan Qiu, Lusong Li, and Tao Mei. Pointclustering: Unsupervised point cloud pre-training using transformation invariance in clustering. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21824–21834. IEEE, 2023.

[50] Ziqiao Meng, Qichao Wang, Zhiyang Dou, Zixing Song, Zhipeng Zhou, Irwin King, and Peilin Zhao. Pointnsp: Autoregressive 3d point cloud generation with next-scale level-of-detail prediction. *arXiv preprint arXiv:2503.08594*, 2025.

[51] Yongwei Chen, Yushi Lan, Shangchen Zhou, Tengfei Wang, and Xingang Pan. Sar3d: Autoregressive 3d object generation and understanding via multi-scale 3d vqvae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28371–28382, 2025.

[52] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[53] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[54] Nina Varney, Vijayan K Asari, and Quinn Graehling. Dales: A large-scale aerial lidar data set for semantic segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 717–726. IEEE, 2020.

[55] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[56] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1):857–876, 2021.

[57] Changyu Zeng, Wei Wang, Anh Nguyen, Jimin Xiao, and Yutao Yue. Self-supervised learning for point cloud data: A survey. *Expert Systems with Applications*, 237:121354, 2024.

[58] Mohamed Abdelsamad, Michael Ulrich, Claudius Gläser, and Abhinav Valada. Multi-scale neighborhood occupancy masked autoencoder for self-supervised learning in lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22234–22243, 2025.

[59] Chun-Peng Chang, Shaoxiang Wang, Alain Pagani, and Didier Stricker. Mikasa: Multi-key-anchor & scene-aware transformer for 3d visual grounding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14131–14140, 2024.

[60] Chun-Peng Chang, Alain Pagani, and Didier Stricker. 3d spatial understanding in mllms: Disambiguation and evaluation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13537–13544. IEEE, 2025.

[61] Chun-Peng Chang, Chen-Yu Wang, Julian Schmidt, Holger Caesar, and Alain Pagani. Seeing clearly, forgetting deeply: Revisiting fine-tuned video generators for driving simulation. *arXiv preprint arXiv:2508.16512*, 2025.

[62] Chun-Peng Chang, Chen-Yu Wang, Holger Caesar, and Alain Pagani. Probing the reliability of driving vlms: From inconsistent responses to grounded temporal reasoning. *arXiv preprint arXiv:2603.09512*, 2026.

Appendix

This appendix provides supplementary information to support the findings presented in the main paper. Specifically, we detail our experimental environment and data usage in Section A. Section B provides comprehensive architectural and hyperparameter configurations for Invaria. Section C reviews self-supervised pre-training for point clouds. Further discussion regarding the limitations of our method is included in Section D, while Section E presents additional empirical evaluations, including per-category performance breakdowns and robustness analyses. Finally, Section F offers supplementary qualitative visualizations to further illustrate the performance of our approach compared to existing state-of-the-art baselines.

A Environment and Data Usage

Software Environment and Hardware Resources

- **OS:** Ubuntu 24.04.3 LTS (Noble Numbat)
- **Python version:** 3.10.19
- **PyTorch version:** 2.9.1+cu128 (built against CUDA 12.8)
- **Vision library:** torchvision 0.24.1
- **CUDA / cuDNN version:** 12.8 / 9.10.2
- **GPU:** NVIDIA A100 80GB $\times$ 2 for training; NVIDIA A100 80GB $\times$ 1 for evaluation.

Data Licenses In our experiments, we utilize the publicly available datasets nuScenes [52], Waymo [53], DALES [54], and ScanNet [41]. The nuScenes dataset is used under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. The Waymo Open Dataset is used under the Waymo Dataset License Agreement for Non-Commercial Use. DALES is derived from aerial LiDAR data provided by the City of Surrey under the Open Government License – City of Surrey. The ScanNet dataset is provided under a custom Terms of Use from the Stanford, Princeton, and TUM organizers, which restricts usage to non-commercial research and educational purposes. We have strictly adhered to the licensing agreements and terms of use for all datasets.

B Model Details

The proposed Invaria architecture consists of three primary modules: an encoder, a decoder, and a predictor. The encoder and decoder adopt a U-Net architecture [37], building upon the implementation framework introduced by PTv3 [18]. The predictor module integrate sparse convolutions and Multi-Layer Perceptrons, interspersed with normalization layers and skip connections to maintain feature consistency. We optimize the network parameters using the AdamW optimizer, employing a OneCycleLR learning rate scheduler to ensure stable convergence and efficient training.

Parameter Details. The encoder backbone is structured with five stages, using depths (2, 2, 2, 2, 1), channel widths (32, 64, 128, 256, 512), and attention heads (2, 4, 8, 16, 32). The decoder comprises four stages with depths (2, 2, 2, 2), channels (64, 64, 128, 256), and head configurations (4, 4, 8, 16), with a backbone output dimension of 64. Optimization is performed using AdamW with a base learning rate of 0.006 and a weight decay of 0.05. We employ the OneCycleLR scheduler with a peak learning rate of 0.006, an initial warm-up period of 10% ($pct_start = 0.1$), and a cosine annealing strategy to ensure stable convergence. For the scaling factor α: We use $\alpha = 1.0$ for ScanNet and $\alpha = 0.8$ for nuScenes, Waymo, and DALES. Receptive field calibration is not sensitive to this choice: varying α within the range of 0.5–2 changes the mIoU by less than 0.5.

Invaria-s. Building upon the PTv3 backbone, LitePT [46] introduces a lightweight architecture optimized for computational efficiency without significantly compromising discriminative power. We integrate this design into our framework to develop Invaria-s, a reduced-capacity variant of our model. Invaria-s features a notable reduction in architectural footprint, with 16.3M parameters compared to the 25.5M in the baseline Invaria, alongside a $\sim$7% reduction in FLOPs. While the results demonstrate

Algorithm 1 Receptive Field Calibration

Require: Point cloud $\mathcal{P} = \{x_j\}_{j=1}^{n_i}$; anchor count N_a; scale factor α; reduction ρ
Ensure: Calibrated grid size $g = \psi(\mathcal{P})$
$\mathcal{P}_a \leftarrow$ uniformly random subset of $\mathcal{P}$ with $|\mathcal{P}_a| = N_a$
for each $a \in \mathcal{P}_a$ **do**
    $d_a \leftarrow \min_{x \in \mathcal{P} \setminus \{a\}} \|a - x\|$ ▷ nearest-neighbor distance
end for
$g \leftarrow \alpha \cdot \rho(\{d_a\}_{a \in \mathcal{P}_a})$
return g

promising enhancements in resolution-invariant evaluation, we observe a slight performance trade-off, as shown in Table 1, with a marginal drop of approximately 3–4% in per-resolution accuracy. Nevertheless, this variant underscores the flexibility of our framework, providing a viable balance between computational efficiency and robustness for resource-constrained deployments.

Lower-Resolution Sampling. During training, we simulate lower-resolution inputs by stochastically subsampling the original point cloud $\mathcal{P}$. Given an input with N points, we define a set of target cardinalities $\mathcal{S} = \{n_0 \cdot h^i \mid i = 0, 1, \ldots, k\}$, where n_0 is the base number of points and h is the geometric scaling factor. For our experiments on ScanNet, we set $n_0 = 4096$ and $h = 4$, yielding $\mathcal{S} = \{4096, 4096 \cdot 4, 4096 \cdot 4^2\}$. We use the same $n_0 = 4096$ for all datasets. This training strategy forces the model to learn representations that are invariant to the underlying sampling density. At inference, we evaluate the model using subset cardinalities $n_i \in \mathcal{S}$, which are scaled relative to the input scene size N while strictly maintaining the constraint $n_i < N$.

Receptive Field Calibration In Section 3 we describe our Receptive Field Calibration; the detailed procedure is summarized in Algorithm 1. The most intuitive choice for the reduction $\rho(\cdot)$ is the minimum, $\rho_{\min}(\{d_a\}) = \min_a d_a$, since by construction every pair of points must be separable by the resulting grid. In practice, however, $\rho_{\min}$ tends to severely underestimate the desired cell size. The reason is geometric: anchor distances are not measured between cell centers but between arbitrary point locations, and since every interior cell has 26 neighbors in $\mathbb{R}^3$, two points sitting near a shared face, edge, or corner can be arbitrarily close while still belonging to distinct cells. Therefore, our empirical findings suggest that ρ_{mean} is a better and more stable choice.

Computational Complexity of Receptive Field Calibration. A full pairwise nearest-neighbor search over N points costs $O(N^2)$. Instead, Algorithm 1 draws a fixed number of anchors N_a (we use $N_a = 100$), independent of N, each requiring a single $O(N)$ scan, which yields a total cost of $O(N_a N) = O(N)$. The calibration is computed once per scene from the input coordinates alone and is therefore independent of the depth and width of the backbone. On an NVIDIA A100 GPU, it adds approximately 2% to the end-to-end inference latency on ScanNet and nuScenes.

C Additional Related Work

Self-supervised Pre-training and Representation Learning for Point Cloud Beyond architectural refinements, 3D point cloud understanding has shifted toward Self-Supervised Learning (SSL) [55, 56, 57, 58] to extract semantically rich representations from unlabeled data. Sonata [8] mitigates "geometric shortcuts", where models prioritized low-level spatial patterns over high-level concepts, via a multi-scale self-distillation framework that obscures spatial details to force the learning of robust semantic features. To further enrich these representations, Concerto [9] leverages multi-sensory synergy through 2D-3D cross-modal joint embedding. Building on these foundational trends, Utonia [20] further employs a unified point transformer trained across heterogeneous domains, ranging from indoor to outdoor.

D Limitation and Discussion

Ambiguity in Planar Objects. While mIoU provides a high-level summary of model performance, it often masks significant disparities between semantic categories. A granular per-class analysis

Table 9: Robustness to resolution shifts on outdoor LiDAR datasets, reported in mIoU (%). *Ret.*: mIoU at $3\tau^*$ relative to the mIoU at the default resolution τ^*.

Model	**nuScenes** (32-beam)				**DALES** (aerial)				**Waymo** (64-beam)			
	τ^*	$2\tau^*$	$3\tau^*$	Ret.	τ^*	$2\tau^*$	$3\tau^*$	Ret.	τ^*	$2\tau^*$	$3\tau^*$	Ret.
Sp.Mink.[4]	76.7	35.4	14.5	18.9%	80.2	53.2	32.9	41.0%	69.8	44.5	23.7	34.0%
PTv3[18]	**80.4**	51.9	26.4	32.9%	**81.7**	59.6	37.2	45.5%	**71.0**	54.0	31.7	44.6%
Invaria	76.2	**75.6**	**71.9**	**94.4%**	80.2	**80.1**	**76.3**	**95.2%**	70.6	**69.7**	**66.0**	**93.5%**

reveals that objects with simple, planar geometries, such as door and window, pose a distinct challenge [59, 60], as the model frequently conflates these structures with walls and floors due to their shared, near-2D geometric profile. Notably, the floor category exhibits notable resilience. This robustness is mainly attributable to the vertical Z-coordinate, which provides a strong spatial prior that facilitates disambiguation, a structural cue absent in other simple planar objects.

Class Imbalance. As illustrated in Table 12, existing 3D benchmarks, including ScanNet [41], nuScenes [52], and Waymo [53], suffer from inherent class imbalance. This imbalance also affects robustness to resolution shifts: *Wall* and *Floor* together account for over 55% of the labeled points in ScanNet, whereas fragile classes such as *Door* (4.5%) and *Shower Curtain* (0.4%) are far less frequent, which limits the geometric cues the model can learn for them. While prior literature often addresses this bias by reweighting the loss function based on class frequency (e.g., via Focal Loss [44]), our ablation study (Table 8c) demonstrates that this approach yields no performance gains for our framework. Observations of the training dynamics reveal a competitive interplay between classes, where improvements in one category appear to necessitate trade-offs in others, suggesting conflicting optimization objectives [61, 62]. Leveraging recent advancements in generative modeling, future research could potentially mitigate these persistent distribution biases by augmenting existing datasets with high-fidelity, synthetic samples of underrepresented classes.

E Additional Results

Sensitivity to Density Variations. As discussed in Section 2.1, point cloud encoders often exploit density as a "shortcut" when learning semantic features. To quantify the robustness of existing methods to such variations, we evaluate their performance under varying sampling densities. Theoretically, increasing the grid size τ by a factor of k reduces the number of points N proportional to $1/k^3$; consequently, a 2- to 3-fold increase in τ should theoretically result in an 8- to 27-fold reduction in density. Here we empirically evaluate model robustness at 10x and 20x reduced sampling rates.

The detailed results are provided in Table 11, showing our proposed Invaria can stay more robust when facing sparser input compared to other existing works.

Detailed Per-Category Analysis. We provide a comprehensive performance comparison across categories in Table 13 (resolution shifts on ScanNet), Table 14 (resolution shifts on nuScenes), and Table 15 (scale changes on ScanNet). As noted in Section D, while planar objects typically exhibit lower baseline performance, complex geometries demonstrate a more significant performance degradation as resolution or scale varies across existing state-of-the-art methods.

Additional Outdoor Datasets. To verify that our findings are not specific to ScanNet, whose point clouds are derived from reconstructed meshes, we additionally evaluate on DALES [54], an aerial LiDAR dataset, alongside the outdoor driving datasets nuScenes [52] and Waymo [53]. Unlike ScanNet, which is captured indoors with a handheld RGB-D sensor, DALES covers large-scale terrain and infrastructure scanned from an airborne, top-down viewpoint, which also distinguishes it from the ground-level driving scenes of nuScenes and Waymo. All three outdoor datasets consist of raw LiDAR measurements and therefore naturally contain real sensor artifacts such as occlusion, range-dependent noise, density fall-off, and missing returns. As shown in Table 9, the baselines lose most of their accuracy under a $3\times$ resolution shift on all three datasets, whereas Invaria consistently retains over 93% of its performance. The failure mode we identify is therefore not limited to ScanNet or to a single sensor configuration.

Table 10: Per-class IoU (%) on outdoor LiDAR datasets. Each cell reports the IoU at the default resolution $\tau^* \rightarrow$ the IoU at $3\tau^*$. Classes are ordered from spatially extended (left) to compact (right).

nuScenes	driveable surface	sidewalk	vegetation	manmade	truck	car	bicycle	pedestrian	traffic cone	mIoU
Sp.Mink.[4]	96.7→55.9	75.6→19.1	86.1→51.1	88.8→51.8	86.5→0.8	86.9→0.2	37.1→0.0	80.1→1.1	64.0→11.9	76.7→14.5
PTv3[18]	97.1→84.7	77.1→38.6	89.5→73.8	90.9→66.0	84.1→16.6	92.3→13.8	50.7→0.0	82.9→2.0	71.0→18.5	80.4→26.4
Invaria	96.7→96.1	75.6→71.6	87.7→86.4	89.3→88.0	83.8→79.2	91.8→89.0	26.9→20.2	77.6→68.3	65.2→56.4	76.2→71.9
DALES	ground	buildings	vegetation	power lines	fences	poles	cars	trucks		mIoU
Sp.Mink.[4]	97.1→87.6	96.7→57.0	93.9→65.7	92.2→17.7	63.2→14.0	69.4→20.7	85.0→0.6	43.8→0.01		80.2→32.9
PTv3[18]	97.1→92.4	97.1→67.1	93.7→69.3	94.7→26.0	66.2→26.4	74.8→16.2	85.6→0.02	44.4→0.00		81.7→37.2
Invaria	96.5→96.0	96.8→96.1	92.9→91.8	89.1→85.5	63.9→54.5	73.5→69.5	84.7→77.8	44.3→39.6		80.2→76.3
Waymo	road	sidewalk	vegetation	building	truck	car	pole	pedestrian	construction cone	mIoU
Sp.Mink.[4]	93.1→78.6	77.7→32.1	91.1→68.3	95.3→66.9	60.9→7.5	94.6→19.2	74.8→34.1	89.4→2.2	70.8→28.5	69.8→23.7
PTv3[18]	93.4→85.3	77.7→43.3	91.6→80.8	95.8→73.5	63.8→14.5	94.5→29.7	77.3→40.9	90.1→3.4	72.6→37.7	71.0→31.7
Invaria	93.3→92.8	77.6→76.0	91.4→90.6	95.7→94.8	64.2→46.1	94.8→95.1	77.1→73.8	89.8→87.8	72.2→73.5	70.6→66.0

Table 11: The density change, evaluating on sparser input point cloud.

Model	**mIoU (%)**		**mAcc (%)**		**allAcc (%)**	
	5%	10%	5%	10%	5%	10%
Sonata	36.6	60.4	45.0	69.8	65.4	82.1
Concrt-L	39.9	**65.4**	49.7	**73.8**	67.1	84.5
Utonia	41.8	63.9	50.8	72.5	68.2	83.5
Sp.Mink.	13.5	30.8	20.3	37.3	60.8	74.0
PTv3	24.8	61.7	30.0	69.4	68.9	**85.5**
Invaria-s	43.4	61.9	53.0	69.9	68.3	82.2
Invaria	**48.2**	64.0	**56.4**	72.7	**71.2**	83.1

Table 12: The distribution of each categories in ScanNet [41], Showing imbalance in different classes.

Class	Count	% Valid	Cum %	Scenes	% Scenes
Wall	10,936,492	30.48%	30.48%	305	97.8%
Floor	8,983,296	25.04%	55.52%	312	100.0%
Chair	3,194,103	8.90%	64.42%	220	70.5%
Other Furniture	1,637,177	4.56%	68.98%	240	76.9%
Door	1,600,191	4.46%	73.44%	219	70.2%
Table	1,478,887	4.12%	77.56%	178	57.1%
Cabinet	1,399,895	3.90%	81.46%	139	44.6%
Window	1,385,131	3.86%	85.32%	166	53.2%
Bookshelf	1,125,319	3.14%	88.46%	46	14.7%
Bed	915,295	2.55%	91.01%	61	19.6%
Sofa	789,769	2.20%	93.21%	69	22.1%
Curtain	657,363	1.83%	95.04%	48	15.4%
Desk	655,163	1.83%	96.87%	84	26.9%
Refrigerator	223,258	0.62%	97.49%	46	14.7%
Counter	221,560	0.62%	98.11%	47	15.1%
Picture	207,442	0.58%	98.69%	90	28.8%
Shower Curtain	158,208	0.44%	99.13%	27	8.7%
Toilet	113,661	0.32%	99.45%	51	16.3%
Sink	102,489	0.29%	99.73%	80	25.6%
Bathtub	96,355	0.27%	100.00%	31	9.9%

Per-Class Analysis on Outdoor Datasets. Table 10 reports the per-class IoU on the outdoor datasets. As on ScanNet, the sensitivity of a class follows its geometry. For PTv3, large planar surfaces are the most robust (87–95% retention on driveable surface, ground, and road), volumetric classes such as vegetation, buildings, and manmade structures lie in the middle (69–88%), and thin structures are fragile (e.g., 22%, 27%, and 40% on poles, power lines, and fences in DALES). However, the degradation is far more severe outdoors, where compact classes collapse almost entirely: on DALES, PTv3 drops from 85.6 to 0.02 IoU on cars and from 44.4 to 0.00 on trucks, and on nuScenes from 50.7 to 0.0 on bicycles and from 82.9 to 2.0 on pedestrians. In contrast, Invaria remains robust on these classes, e.g., 84.7 $\rightarrow$ 77.8 on DALES cars and 89.8 $\rightarrow$ 87.8 on Waymo pedestrians.

Stress Test under Stronger Resolution Shifts. To reveal failure cases, we further reduce the input resolution beyond the range covered in Table 1. The degradation is not uniform: it remains gradual from 100% down to about 11% of the original point count, but on ScanNet and nuScenes, accuracy drops sharply once the input falls below roughly 5–6% of the original points. We consider this a different failure mode from the one targeted in this work. The degradation addressed by Invaria is recoverable because the geometry is still present in the input, whereas below roughly 5–6% of the original points, the geometry itself is largely lost.

Table 13: Detailed analysis of each category in ScanNet as the resolution changes. ($\tau^* = 2$cm)

Grid Size	Method	mIoU	mAcc	allAcc	Wall		Floor		Cab.		Bed		Chair		Sofa		Tabl		Door		Wind		Bksh		Pict		Cntr		Desk		Curt		Frid		Shw.		Toil		Sink		Bath		Oth.	
		(%)	(%)	(%)	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
	Sonata	59.9	70.7	84.9	82.1	94.9	94.1	97.9	51.7	67.5	62.4	81.6	76.9	80.3	56.8	82.3	57.1	74.3	52.7	63.5	58.5	69.5	65.0	75.4	37.0	44.8	48.3	64.5	35.3	42.6	78.0	88.6	37.5	46.3	69.3	74.4	66.9	68.6	50.6	57.4	76.1	85.1	42.3	55.3
	Concrt-L	67.8	77.5	88.4	85.0	95.0	94.8	97.6	63.1	79.2	72.0	89.2	80.6	84.6	62.9	85.6	65.2	78.5	60.6	68.2	71.7	82.9	77.0	83.7	35.0	41.1	57.8	71.7	50.5	58.5	75.7	90.5	63.3	70.5	44.5	47.0	86.3	88.4	65.7	73.2	84.9	90.0	58.8	73.7
	Utonia	73.0	81.9	95.4	84.8	95.4	95.2	97.6	67.4	77.3	78.2	88.5	83.4	86.2	65.0	89.9	72.4	82.1	69.5	77.3	70.2	78.5	79.5	87.0	37.5	46.5	62.0	79.0	62.3	74.6	79.9	88.7	72.3	78.5	71.6	76.3	91.9	93.7	70.0	77.3	84.8	89.4	61.1	74.0
$0.5\tau^*$	Sp.Mink.	36.6	47.2	73.8	73.0	95.7	92.5	98.4	31.8	52.2	41.9	81.3	37.9	39.1	24.1	44.0	41.0	60.0	25.9	30.1	39.9	49.9	42.4	47.6	19.8	24.4	22.4	28.1	16.6	20.7	52.2	70.8	9.3	10.8	35.3	34.1	28.9	30.4	26.0	26.9	48.6	63.5	22.0	35.0
	PTv3	50.4	59.3	80.6	76.4	97.0	90.0	98.9	45.7	65.8	59.4	80.2	70.5	73.2	39.9	56.3	48.6	71.4	40.1	45.2	48.2	56.6	59.2	62.1	22.9	24.7	25.8	28.1	25.5	27.1	61.3	75.4	33.2	35.4	46.7	49.1	66.2	67.3	49.0	54.2	69.8	81.6	30.7	35.7
	Invaria	75.5	83.8	91.0	86.4	94.9	95.6	98.4	70.0	81.6	83.4	88.8	91.6	95.5	84.0	93.8	74.2	81.8	67.5	76.9	68.0	81.4	82.7	92.0	37.8	43.2	65.1	75.7	65.6	89.7	78.0	86.1	74.7	82.0	71.4	77.3	95.4	97.7	69.0	77.7	89.1	94.0	61.2	67.1
	Sonata	72.6	83.4	89.8	85.7	93.0	95.1	97.8	63.2	75.8	78.5	84.8	88.0	93.2	79.3	91.0	71.5	81.7	67.7	81.1	68.2	84.4	77.2	92.4	37.6	49.3	63.7	80.3	62.7	86.0	81.5	87.8	57.0	74.8	73.2	80.2	91.2	96.9	67.7	80.8	85.8	94.1	55.0	62.2
	Concrt-L	78.6	87.4	92.3	87.4	93.0	95.5	97.7	72.3	82.7	82.3	89.9	90.7	95.5	81.8	90.7	78.5	86.8	77.5	91.7	79.1	89.4	86.7	94.2	37.6	53.7	68.8	85.6	68.6	80.5	86.0	91.1	79.8	89.9	74.9	80.3	94.0	96.6	72.0	82.8	86.9	94.3	71.9	81.8
	Utonia	76.6	85.1	94.6	85.8	94.6	95.4	97.9	72.0	82.0	81.0	87.1	88.8	91.8	73.8	91.0	77.0	83.8	75.1	85.4	73.1	82.8	82.7	92.4	39.0	50.5	65.2	86.3	67.9	79.0	83.3	88.0	80.0	86.6	75.1	80.7	94.0	96.0	70.2	79.5	85.3	90.4	66.6	75.8
τ^*	Sp.Mink.	75.6	83.5	91.5	87.2	95.7	95.9	98.2	69.9	80.6	84.6	89.9	92.4	95.7	84.7	93.3	76.2	84.9	70.0	80.2	68.4	81.5	81.7	92.2	37.8	44.2	66.0	80.9	68.1	86.1	77.9	86.4	65.9	72.8	71.8	76.4	91.0	96.6	67.5	72.1	88.8	91.4	65.6	70.1
	PTv3	77.5	85.0	92.0	87.2	95.7	95.7	98.3	71.2	83.1	83.4	87.8	92.9	96.4	86.3	93.6	79.0	86.6	74.9	84.4	74.0	84.1	83.0	91.6	39.8	48.5	71.7	81.7	73.9	89.0	79.1	87.5	71.3	75.7	72.2	77.3	93.4	97.9	71.0	81.5	88.2	91.4	63.0	66.7
	Invaria	76.2	84.3	91.2	86.6	94.9	95.7	98.4	71.2	82.0	83.6	89.0	91.9	95.7	84.6	93.9	74.2	82.4	69.1	78.7	69.9	82.6	83.1	92.6	38.4	44.0	65.5	76.6	65.4	89.3	79.8	86.7	75.5	83.2	74.2	80.0	95.7	97.8	69.8	78.3	89.6	94.2	60.1	66.1
	Sonata	62.3	73.8	85.7	81.0	90.6	93.3	98.5	53.9	67.7	72.2	81.7	79.3	88.3	69.6	75.4	64.0	72.3	55.8	69.2	61.6	80.0	70.5	90.9	25.7	32.5	52.9	58.6	51.9	82.5	75.5	84.6	42.5	56.5	69.6	79.7	65.0	70.0	50.8	67.9	71.3	78.3	39.7	49.9
	Concrt-L	72.5	82.3	89.9	84.0	90.4	94.0	98.1	67.5	79.5	81.7	88.3	86.0	91.9	83.6	87.4	73.0	79.3	67.7	85.4	73.7	89.0	81.6	95.3	33.6	50.4	63.9	75.3	63.5	80.5	83.4	91.2	68.2	79.3	73.0	80.0	66.9	68.0	62.5	75.5	80.5	86.2	62.5	74.9
	Utonia	71.7	80.5	89.2	82.8	93.6	93.9	98.2	66.5	79.9	79.6	83.9	86.1	90.8	79.7	88.9	71.7	76.3	68.7	80.1	68.0	81.5	76.4	83.4	28.8	35.7	63.3	79.7	65.1	78.1	77.1	84.4	64.7	74.2	72.3	80.5	87.7	91.4	63.7	74.7	80.4	86.3	57.4	68.1
$2\tau^*$	Sp.Mink.	47.8	56.0	82.4	78.0	94.9	93.9	98.3	45.5	61.4	57.5	59.7	75.8	84.9	53.5	60.1	60.1	76.3	39.5	49.7	56.9	73.6	69.2	79.8	16.4	17.5	44.5	52.7	44.2	57.8	66.3	80.5	25.2	25.8	37.2	38.7	8.8	9.2	28.7	31.3	25.2	26.8	29.2	41.5
	PTv3	60.4	69.0	86.0	80.6	95.5	92.8	98.7	59.4	71.8	67.5	71.0	80.5	88.1	59.1	69.5	67.9	79.0	52.3	60.0	62.4	77.4	77.6	86.4	28.8	33.4	59.1	66.2	45.4	57.6	72.8	82.8	38.4	40.2	50.9	54.3	54.6	54.9	55.3	66.1	60.8	77.1	42.0	50.6
	Invaria	75.7	84.0	90.9	85.9	94.3	95.3	98.5	68.9	81.6	83.4	86.8	91.5	95.2	84.0	95.0	74.3	82.3	65.4	75.9	68.3	83.0	83.2	93.0	34.7	40.3	66.4	78.6	65.5	88.5	79.5	86.2	74.8	81.1	76.4	83.0	95.3	97.2	69.8	78.7	89.0	94.1	61.8	66.8
	Sonata	34.2	43.6	72.3	71.5	92.0	88.6	98.2	28.2	40.7	43.8	52.4	42.2	49.3	20.1	21.3	40.8	54.6	23.1	28.3	46.6	62.2	45.6	75.3	10.9	12.0	7.9	8.6	24.0	31.9	60.0	69.7	15.9	18.6	57.0	71.9	8.7	10.6	14.9	22.5	23.0	29.6	10.9	23.1
	Concrt-L	46.0	57.3	79.1	76.1	87.7	89.9	98.0	39.6	63.9	67.2	74.1	59.4	65.3	68.4	73.5	52.7	64.5	46.2	62.5	58.7	83.4	46.8	69.8	23.8	42.8	26.6	33.7	35.8	51.1	70.1	83.7	26.3	28.7	57.6	62.5	3.5	3.6	20.1	30.9	28.3	30.3	22.8	35.6
	Utonia	47.2	56.3	78.7	74.5	93.5	90.0	98.1	44.2	58.1	63.9	65.1	62.0	66.6	66.0	76.4	53.4	64.3	41.7	49.1	53.5	71.7	19.2	20.2	14.8	17.4	37.2	49.1	35.0	40.1	63.9	76.0	31.3	33.5	54.9	61.0	26.5	27.3	38.8	46.5	47.1	57.6	26.8	54.4
$3\tau^*$	Sp.Mink.	19.4	25.7	66.1	64.8	95.4	85.0	95.5	20.9	31.7	7.4	7.5	35.5	40.1	9.2	9.3	37.5	56.0	8.8	10.7	33.0	52.0	23.8	26.6	0.6	0.6	0.0	0.0	5.6	7.2	45.9	56.3	0.0	0.0	0.2	0.2	0.0	0.0	0.4	0.4	0.0	0.0	9.8	23.7
	PTv3	20.5	26.0	65.9	60.2	96.4	74.4	99.0	13.0	15.0	11.9	12.0	29.6	34.0	4.2	4.2	35.7	43.5	19.6	22.1	41.1	62.3	26.8	27.7	5.9	7.9	0.1	0.1	3.1	3.2	47.7	51.3	0.1	0.1	12.4	12.8	0.0	0.0	4.6	4.8	10.6	11.8	8.4	11.1
	Invaria	73.8	82.7	90.0	84.5	94.0	95.1	98.5	67.9	80.9	82.0	86.8	90.1	94.2	82.8	95.3	73.7	81.3	59.7	70.3	65.8	80.2	80.5	92.5	27.5	31.2	64.5	77.8	65.0	86.6	79.3	85.4	71.1	78.6	76.8	83.4	92.5	96.5	69.6	81.0	88.3	95.6	58.5	63.4

Table 14: Detailed analysis of each category in nuScenes as the resolution changes ($\tau^* = 5$ cm)

Grid Size	Method	mIoU	mAcc	allAcc	Barrier		Bicycle		Bus		Car		Const.		Motor.		Ped.		Cone		Trailer		Truck		Drive.		Flat		Side.		Terr.		Man.		Veg.	
		(%)	(%)	(%)	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
$0.5\tau^*$	Sp.Mink.	34.1	41.4	85.7	18.3	18.8	1.4	1.4	21.3	46.2	40.5	41.2	9.6	21.4	1.2	1.2	9.2	9.5	11.7	11.9	3.2	7.8	19.9	34.6	93.1	98.4	34.1	37.2	58.4	66.0	68.1	85.2	75.5	89.9	79.8	92.0
	PTv3	45.1	52.9	88.7	36.7	38.9	8.1	8.5	39.8	69.0	52.7	53.7	18.2	20.7	29.4	29.7	20.5	20.9	31.3	31.8	12.2	30.6	26.0	46.2	94.2	98.9	48.6	51.8	64.5	71.6	72.4	89.4	81.5	91.1	85.0	94.2
	Invaria	74.3	82.2	93.6	77.0	85.3	25.5	64.8	90.5	91.8	90.7	93.9	44.1	49.4	82.8	84.3	75.7	79.3	63.7	69.1	63.3	73.5	80.7	87.1	96.5	98.5	72.5	80.3	75.0	84.7	75.2	85.4	88.4	94.3	86.7	94.4
τ^*	Sp.Mink.	76.6	82.0	93.7	77.6	85.8	37.1	40.7	91.7	93.1	86.9	89.0	52.9	57.1	84.6	86.8	80.1	83.7	64.0	69.3	69.4	74.6	86.5	91.8	96.7	98.5	73.0	81.9	75.6	85.3	75.2	85.0	88.8	94.9	86.1	94.1
	PTv3	80.3	86.2	94.6	80.1	89.1	50.7	57.4	94.8	95.4	92.3	94.4	62.5	66.5	87.7	89.0	82.9	87.6	71.0	78.0	71.4	79.1	84.1	93.9	97.1	98.6	77.1	86.3	77.1	86.2	76.4	86.6	90.9	95.8	89.5	94.6
	Invaria	76.2	83.9	94.0	78.2	86.7	26.9	66.5	92.3	93.0	91.8	95.0	49.9	54.9	83.7	85.0	77.6	81.3	65.2	71.8	69.3	77.0	83.8	89.3	96.7	98.4	75.2	83.6	75.6	85.2	75.5	85.8	89.3	94.9	87.7	94.3
$2\tau^*$	Sp.Mink.	35.4	45.7	85.1	23.0	28.2	0.3	0.3	0.3	0.3	43.9	47.5	7.7	52.4	4.6	7.4	12.6	17.8	29.0	55.2	6.6	10.4	24.6	30.6	92.6	94.6	45.9	51.7	56.8	73.5	61.8	78.4	74.7	95.7	81.8	87.2
	PTv3	51.8	59.7	63.0	51.1	63.0	11.7	11.9	10.9	10.9	72.9	92.4	31.9	48.0	36.5	37.4	27.2	28.5	49.6	67.7	21.2	22.0	48.9	53.5	95.6	97.7	66.2	74.8	67.1	82.7	67.6	77.4	85.1	95.4	86.2	91.8
	Invaria	75.6	83.4	94.0	76.8	83.9	25.9	64.5	91.7	92.5	90.8	94.6	50.6	54.6	84.1	85.4	76.5	80.7	63.8	71.7	66.6	75.2	83.5	89.0	96.7	98.3	74.6	84.1	75.2	85.1	74.9	85.8	89.4	95.3	88.2	94.1
$3\tau^*$	Sp.Mink.	14.5	22.6	58.8	8.7	9.6	0.0	0.0	0.0	0.0	0.2	0.2	3.4	18.1	0.0	0.0	1.1	1.5	11.9	24.7	0.7	0.8	0.8	0.8	55.9	58.6	2.6	2.6	19.1	29.3	24.8	65.3	51.8	97.5	51.1	53.1
	PTv3	26.4	33.0	78.2	25.4	32.2	0.0	0.0	0.0	0.0	13.8	16.3	4.2	4.7	0.0	0.0	2.0	2.1	18.5	35.8	2.5	2.8	16.6	17.8	84.7	94.9	31.7	38.5	38.6	59.2	44.7	50.1	66.0	94.9	73.8	78.4
	Invaria	71.9	80.1	93.0	73.9	82.9	20.2	53.1	89.8	90.5	89.0	94.4	48.8	52.4	77.1	79.9	68.3	72.6	56.4	66.2	61.3	68.6	79.2	84.7	96.1	98.1	72.0	83.3	71.6	82.6	72.1	84.2	88.0	94.5	86.4	92.8

Table 15: Detailed analysis of each category in ScanNet as the scale changes. ($\tau^* = 2$cm)

Scale	Method	mIoU	mAcc	allAcc	Wall		Floor		Cab.		Bed		Chair		Sofa		Tabl		Door		Wind		Bksh		Pict		Cntr		Desk		Curt		Frid		Shw.		Toil		Sink		Bath		Oth.	
		(%)	(%)	(%)	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc	IoU	Acc
	Sonata	57.9	69.0	84.1	81.2	94.9	94.4	97.9	49.3	67.9	61.6	80.4	75.5	78.3	56.2	80.5	56.0	74.4	50.0	59.2	59.4	72.3	64.2	75.4	37.2	47.6	46.1	66.5	30.0	35.6	75.5	85.1	31.9	39.0	66.8	71.9	59.8	61.2	49.6	58.0	74.4	82.8	39.3	51.2
	Concrt-L	65.9	75.5	87.8	84.1	95.4	94.7	97.8	61.7	80.0	72.1	87.0	79.3	82.8	62.0	82.8	63.9	77.1	56.5	62.4	71.3	84.0	76.4	83.8	32.5	37.4	57.7	74.6	49.4	57.9	73.7	88.9	60.8	67.2	41.0	42.9	81.1	82.8	61.1	68.1	82.8	86.5	56.8	69.9
	Utonia	59.0	69.2	83.6	77.5	95.5	94.0	97.0	52.3	66.8	67.4	82.8	73.3	76.3	57.9	83.9	53.1	68.8	48.9	54.6	57.3	69.9	62.6	68.2	30.8	36.6	51.7	74.1	35.5	39.6	66.7	83.9	52.4	55.3	59.9	64.0	71.9	72.5	54.6	64.3	70.3	73.3	42.0	56.3
$\times 2$	Sp.Mink.	35.7	46.3	73.2	72.8	95.4	92.4	98.3	30.6	50.8	40.9	81.0	36.0	37.1	23.0	42.6	40.2	59.1	25.8	30.4	39.0	49.1	40.6	45.5	19.3	24.1	19.1	24.2	16.2	20.4	54.7	70.5	9.3	11.0	32.3	33.9	28.5	30.2	23.7	24.5	47.8	63.6	21.1	34.4
	PTv3	49.5	58.4	80.1	76.2	96.7	89.7	98.9	44.8	64.9	58.9	80.1	69.2	72.1	38.9	54.8	46.9	69.6	39.2	44.8	47.4	55.9	58.0	61.1	23.1	25.2	24.5	26.8	24.2	26.0	60.5	75.3	34.0	36.6	46.5	49.5	61.4	62.7	47.6	52.4	68.9	80.9	29.3	34.6
	Invaria	75.6	83.8	91.1	86.5	95.0	95.6	98.4	70.8	81.8	83.5	88.9	91.5	95.5	84.2	94.1	74.2	82.0	68.1	77.5	68.3	81.4	82.6	92.2	37.8	43.0	65.3	75.6	65.7	89.6	79.4	86.1	74.5	81.7	76.4	83.0	95.3	97.2	69.7	78.7	88.9	94.1	60.6	66.5
	Sonata	72.6	83.4	89.8	85.7	93.0	95.1	97.8	63.2	75.8	78.5	84.8	88.0	93.2	79.3	91.0	71.5	81.7	67.7	81.1	68.2	84.4	77.2	92.4	37.6	49.3	63.7	80.3	62.7	86.0	81.5	87.8	57.0	74.8	73.2	80.2	91.2	96.9	67.7	80.8	85.8	94.1	55.0	62.2
	Concrt-L	78.6	87.4	92.3	87.4	93.0	95.5	97.7	72.3	82.7	82.3	89.9	90.7	95.5	81.8	90.7	78.5	86.8	77.5	91.7	79.1	89.4	86.7	94.2	37.6	53.7	68.8	85.6	68.6	80.5	86.0	91.1	79.8	89.9	74.9	80.3	94.0	96.6	72.0	82.8	86.9	94.3	71.9	81.8
	Utonia	76.6	85.1	94.6	85.8	94.6	95.4	97.9	72.0	82.0	81.0	87.1	88.8	91.8	73.8	91.0	77.0	83.8	75.1	85.4	73.1	82.8	82.7	92.4	39.0	50.5	65.2	86.3	67.9	79.0	83.3	88.0	80.0	86.6	75.1	80.7	94.0	96.0	70.2	79.5	85.3	90.4	66.6	75.8
$\times 1$	Sp.Mink.	75.6	83.5	91.5	87.2	95.7	95.9	98.2	69.9	80.6	84.6	89.9	92.4	95.7	84.7	93.3	76.2	84.9	70.0	80.2	68.4	81.5	81.7	92.2	37.8	44.2	66.0	80.9	68.1	86.1	77.9	86.4	65.9	72.8	71.8	76.4	91.0	96.6	67.5	72.1	88.8	91.4	65.6	70.1
	PTv3	77.5	85.0	92.0	87.2	95.7	95.7	98.3	71.2	83.1	83.4	87.8	92.9	96.4	86.3	93.6	79.0	86.6	74.9	84.4	74.0	84.1	83.0	91.6	39.8	48.5	71.7	81.7	73.9	89.0	79.1	87.5	71.3	75.7	72.2	77.3	93.4	97.9	71.0	81.5	88.2	91.4	63.0	66.7
	Invaria	76.2	84.3	91.2	86.6	94.9	95.7	98.4	71.2	82.0	83.6	89.0	91.9	95.7	84.6	93.9	74.2	82.4	69.1	78.7	69.9	82.6	83.1	92.6	38.4	44.0	65.5	76.6	65.4	89.3	79.8	86.7	75.5	83.2	74.2	80.0	95.7	97.8	69.8	78.3	89.6	94.2	60.1	66.1
	Sonata	60.7	72.4	85.1	80.2	90.0	93.2	98.2	52.2	64.8	71.0	82.5	78.4	87.9	68.4	75.0	62.7	71.6	54.0	68.4	59.9	77.0	69.8	89.5	21.8	26.4	48.5	53.7	50.2	82.0	75.1	84.9	41.6	56.8	67.2	76.2	63.5	70.1	47.3	64.3	68.4	78.1	39.5	51.1
	Concrt-L	71.5	81.9	89.5	83.3	90.0	93.9	97.9	65.9	77.7	80.6	89.2	85.3	91.7	82.4	86.8	72.5	79.2	66.5	83.5	72.6	87.3	81.4	95.6	31.8	49.4	60.2	71.4	62.4	80.2	82.5	90.0	66.0	80.0	72.6	80.3	68.1	69.8	59.9	74.3	79.4	87.6	62.1	75.9
	Utonia	71.8	82.5	89.4	83.2	91.5	94.0	98.0	67.8	80.1	81.6	87.8	85.2	91.1	79.2	89.2	70.8	75.8	67.8	81.2	70.0	84.6	81.2	93.6	31.1	41.9	61.0	74.9	63.9	81.5	77.7	86.3	63.3	80.0	72.9	82.5	85.5	93.1	59.6	76.3	79.5	90.4	60.4	71.2
$\times 1/2$	Sp.Mink.	46.9	55.4	82.1	78.0	95.0	93.8	98.3	44.2	60.9	56.5	59.1	74.8	84.1	53.8	61.0	59.5	76.3	38.2	48.0	55.3	71.5	69.0	80.6	16.1	17.4	45.0	54.0	41.0	53.5	64.6	79.8	24.3	25.2	33.8	35.2	10.2	10.6	28.3	31.1	23.1	25.0	28.7	40.8
	PTv3	59.4	68.3	85.7	80.4	95.3	92.8	98.7	58.2	71.1	67.0	70.8	80.1	88.0	56.8	66.7	67.5	79.5	52.1	60.0	61.7	76.8	77.6	86.3	27.3	31.8	57.2	64.5	44.2	56.4	72.0	82.3	39.2	40.8	49.4	52.9	51.0	51.3	53.4	65.7	58.5	77.2	41.0	49.7
	Invaria	75.6	84.0	90.8	85.9	94.3	95.3	98.5	68.9	81.5	83.3	86.7	91.5	95.2	83.9	94.9	74.2	82.2	65.4	75.9	68.1	82.9	83.3	93.0	34.5	40.1	66.1	78.3	65.4	88.5	79.4	86.1	75.4	81.7	76.4	83.0	95.3	97.2	69.7	78.7	88.9	94.1	61.8	66.8
	Sonata	32.1	41.6	70.9	70.8	91.7	87.0	95.7	24.9	34.3	39.0	51.0	41.4	48.8	16.5	17.5	39.3	52.4	21.8	27.2	45.2	59.8	48.1	72.9	4.8	4.9	4.7	9.3	19.4	29.6	60.1	68.7	15.1	19.1	52.0	64.2	8.4	10.5	11.6	18.1	20.4	29.0	12.1	27.3
	Concrt-L	43.3	55.3	77.1	74.6	86.4	88.9	97.2	36.7	59.0	62.7	74.4	55.6	62.2	57.4	61.7	49.6	62.9	39.6	54.1	56.9	80.1	46.3	69.4	21.1	40.7	17.3	25.4	30.4	52.5	69.3	80.5	25.8	28.3	59.1	64.5	4.2	4.3	18.0	29.9	30.9	36.6	21.4	35.3
	Utonia	53.7	65.7	81.1	76.7	88.8	90.4	97.2	47.2	64.5	73.8	79.2	63.4	71.0	69.3	81.1	53.5	63.3	47.4	61.9	57.8	78.3	53.9	65.1	16.5	22.6	30.1	33.8	41.6	54.9	69.2	84.9	45.4	59.5	67.4	76.1	34.1	36.4	38.4	60.6	66.1	79.2	32.2	54.8
$\times 1/3$	Sp.Mink.	19.1	25.4	65.6	64.7	95.0	84.6	95.3	20.0	31.1	7.0	7.1	33.5	38.3	8.7	8.8	35.8	54.1	8.8	11.0	32.4	51.4	22.5	25.8	0.8	0.8	0.0	0.0	5.6	7.2	45.6	56.0	0.0	0.0	1.0	1.0	0.0	0.0	0.7	0.7	0.0	0.0	9.6	23.3
	PTv3	20.1	25.6	65.4	60.0	96.1	74.0	98.9	12.8	14.8	10.1	10.2	28.4	32.5	2.9	2.9	35.0	42.9	18.6	21.2	40.1	60.8	24.4	25.5	5.5	7.3	0.0	0.0	2.3	2.4	49.9	53.8	0.0	0.0	12.8	13.1	0.0	0.0	4.7	5.0	11.2	12.1	8.6	11.5
	Invaria	73.8	82.7	90.0	84.5	94.0	95.1	98.5	67.8	80.8	82.1	86.8	90.2	94.2	82.7	95.4	73.6	81.3	59.8	70.4	65.9	80.3	80.5	92.5	27.6	31.3	64.8	77.9	64.9	86.6	79.4	85.5	71.2	78.8	76.8	83.5	92.7	96.5	69.6	81.1	88.7	95.5	58.4	63.3

F More Visualization

In Figure 8, we provide additional qualitative comparisons between our method and existing approaches, including Sp.Mink. [4], PTv3 [18], and Utonia [20].

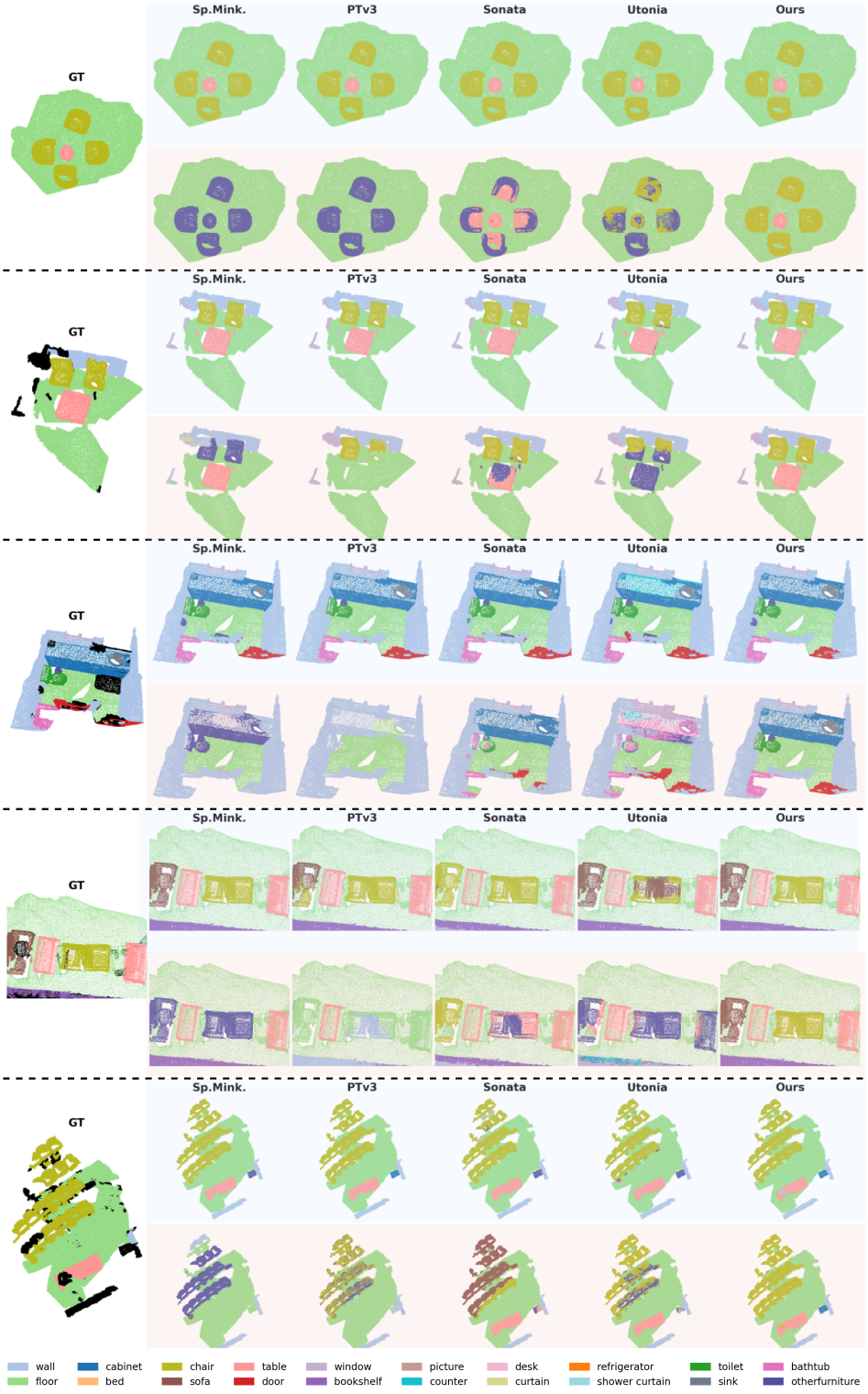


Figure 8: Comparison on ScanNet semantic segmentation. The first row displays model performance when evaluated at the same resolution. The second row illustrates the impact of shifting the resolution during inference. For clear visual comparison, results from both are projected on the original point cloud. The black spot in GT means no label.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the contributions and reflect our results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitation in Section D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The results of the paper are mainly supported empirically.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided comprehensive and sufficient information for our method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We release the code and model weights.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provided the necessary details to run our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the significant computational resources required for training and evaluation, we report results from a single representative run.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided necessary details on the compute resources we used in our experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work has no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credited and cited licenses of existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the code with documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The LLM is used only for writing, editing, or formatting purposes.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.