POINTCRO: CLASS-COMPETITION RELATIVE OPTIMIZATION FOR POINT CLOUD SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Current regularization methods for point cloud segmentation typically rely on surrogate objectives defined over feature geometry or output distributions. Our diagnostics show that improvements in these objectives do not consistently translate into higher segmentation accuracy, motivating explicit modeling of point-wise class competition. We observe that many prediction errors arise from imbalanced competition among the ground-truth class, the most competitive incorrect class (TopWrong), and the remaining classes. To address this issue, we propose PointCRO, a prediction-state-aware fine-tuning framework that combines reward-based output regulation with representation-level regularization. Specifically, Class-Group Relative Optimization (CGRO) formulates each point prediction as a class competition and uses competition-aware relative rewards to rebalance the ground-truth, TopWrong, and remaining classes, directly correcting unfavorable prediction states rather than optimizing generic output statistics. Reward-guided Feature-level Kullback–Leibler (RFKL) regularization transfers these reward-induced preferences to pairwise feature relations, aligning representation structure with the desired competition dynamics and stabilizing the resulting prediction correction. Experiments across multiple benchmarks demonstrate consistent improvements in semantic segmentation performance. Endpoint prediction-transition analyses further show that PointCRO better balances error correction and the preservation of correct predictions, supporting prediction-state-aware class competition as a more direct optimization target and reducing the mismatch between surrogate regularization gains and segmentation improvements.

1 INTRODUCTION

Point clouds have been widely used in various 3D perception scenarios, such as autonomous driving (Lai et al., 2023; Zhao et al., 2025) and indoor scene understanding (Yu et al., 2026b; Zhang et al., 2025a), due to their ability to capture accurate geometric structures of real-world environments. As a fundamental task in 3D scene understanding, point cloud semantic segmentation aims to assign a semantic label to each point, providing fine-grained scene perception for downstream applications. Despite remarkable progress, accurate point-level segmentation remains challenging because many errors arise from ambiguous local competition among semantically or geometrically similar categories. To improve segmentation performance, existing methods often introduce regularization strategies in either the feature space or the output space, such as enforcing intra-class compactness, enlarging inter-class separability and improving confidence calibration.

However, many regularization objectives primarily optimize surrogate properties of feature geometry or output distributions rather than directly modeling point-level prediction correction. Consequently, improvements in these proxy objectives may not consistently translate into higher segmentation accuracy, suggesting a potential mismatch between surrogate regularization and task-level error correction. To examine this behavior, we conduct diagnostic analyses on ScanNet++ (Yeshwanth et al., 2023) and ScanNetV2 (Dai et al., 2017) using VOLT-S (Yilmaz et al., 2026) as a common baseline, as summarized in Table 1. CenterLoss produces the lowest measured intra-class dispersion but decreases mIoU, whereas SupCon yields the largest feature margin with only modest segmentation gains. Similar inconsistencies are observed for output-space proxies such as ECE and TopWrong suppression. These observations motivate shifting the optimization target from generic surrogate properties toward prediction-state-aware class competition for point-level error correction.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Motivating diagnostics on the relationship between generic regularization proxies and segmentation performance. Intra-Comp., Feat. Margin, ECE, and TopWrong Supp. quantify intra-class feature compactness, G.T.–TopWrong feature separation, calibration, and dominant incorrect-class suppression, respectively; their detailed definitions are provided in Appendix A.3. Improvements in individual proxy metrics are not consistently accompanied by higher mIoU.

Dataset	Method	Feature-Space Proxies		Output-Space Proxies		Task Perf.
		Intra-Comp. ↓	Feat. Margin ↑	ECE ↓	TopWrong Supp. ↑	mIoU (%) ↑
ScanNet++	VOLT-S Baseline	0.178	0.393	0.049	0.0000	50.2
	+ CenterLoss	**0.082**	0.120	0.038	0.0027	49.5
	+ SupCon	0.258	**0.663**	0.041	0.0031	50.6
	+ CE-Margin	0.175	0.368	0.036	-0.0045	50.2
	+ Conf. Penalty	0.142	0.317	0.378	**0.0411**	50.2
	+ PointCRO (Ours)	0.180	0.411	**0.031**	0.0005	**51.2**
ScanNetV2	VOLT-S Baseline	0.185	0.398	0.028	0.0000	77.3
	+ CenterLoss	**0.092**	0.255	0.030	0.0010	77.1
	+ SupCon	0.313	**0.729**	0.030	0.0025	77.6
	+ CE-Margin	0.183	0.470	**0.018**	-0.0015	77.4
	+ Conf. Penalty	0.181	0.479	0.629	**0.0207**	77.6
	+ PointCRO (Ours)	0.258	0.715	0.031	0.0053	**78.6**

Modeling prediction-state-aware class competition explicitly poses three challenges. First, class competition varies across points and training stages, motivating prediction-state-dependent optimization signals for the ground-truth and competing incorrect classes. Second, point-level correction is inherently relative. Increasing the ground-truth (G.T.) response may be insufficient when TopWrong remains dominant, whereas uniformly suppressing non-G.T. classes can over-constrain the output distribution and distort inter-class relationships. Third, although output-space objectives influence representation learning through backpropagation, they do not explicitly specify how class-competition preferences should guide pairwise feature relations. This motivates complementing output-space relative optimization with explicit, competition-aware feature regularization.

To address these challenges, we propose PointCRO, a dual-space framework that shifts optimization from generic surrogate adjustment to prediction-state-aware point-level class competition. In the output space, Class-Group Relative Optimization (CGRO) formulates the class predictions of each point as a competition group and assigns competition-aware relative rewards according to their current prediction roles. This enables CGRO to directly rebalance the responses of the G.T., the TopWrong, and the remaining classes, thereby correcting unfavorable prediction states rather than indirectly optimizing generic output statistics. In the feature space, Reward-guided Feature-level Kullback–Leibler (RFKL) regularization transfers these reward-induced preferences to pairwise feature relations, encouraging the representation structure to support the same correction direction and providing a feature-level anchor for more stable prediction correction. By coupling output-space competition correction with reward-guided representation regularization, PointCRO explicitly aligns optimization with point-wise prediction errors and reduces the mismatch between surrogate regularization gains and actual segmentation improvements. Our contributions are summarized as follows:

- We propose PointCRO, a dual-space fine-tuning framework for prediction-state-aware point-level class competition modeling. It explicitly characterizes the competition among the ground-truth class, the most competitive incorrect class, and the remaining classes.
- We introduce CGRO in the output space and RFKL regularization in the feature space. CGRO performs state-dependent relative optimization over the class competition of each point, while RFKL propagates competition-aware preferences to pairwise feature relations, providing representation-level support for more stable prediction correction.
- Extensive experiments on standard point cloud semantic segmentation benchmarks demonstrate that PointCRO consistently improves segmentation performance over conventional regularization methods based on feature geometry or output distributions.

2 Related Work

Point Cloud Semantic Segmentation. Existing point cloud semantic segmentation methods have progressively evolved from direct point-based modeling to sparse voxel processing and attention-based contextual learning. Point-based methods (Zhang et al., 2022; 2025d; Ma et al., 2022; Yin et al., 2024; Li et al., 2024; Deng et al., 2023; Zha et al., 2024; Zou et al., 2024; Qu et al., 2025; Su et al., 2025; Wu et al., 2025a; Zha et al., 2025a; Peng et al., 2024; Zha et al., 2025b; Zeng et al., 2025) directly operate on unordered point sets and learn hierarchical representations through local neighborhood aggregation and geometric feature extraction. Sparse-convolution and point–voxel methods (Choy et al., 2019; Zhu et al., 2021; Chen et al., 2023b; Feng et al., 2024; Zhang et al., 2025b) organize irregular point clouds into computationally efficient sparse structures, enabling large-scale scene understanding through multi-scale spatial aggregation. More recently, Transformer-based (Vaswani et al., 2017; Zhao et al., 2021; Park et al., 2022; 2023; Chen et al., 2023a; Wang, 2023; Lai et al., 2023; Wu et al., 2022; 2024; 2025b; Yilmaz et al., 2026; Yang et al., 2025; Han et al., 2025; Yue et al., 2026) and Mamba-based (Gu & Dao, 2024; Han et al., 2024; Liang et al., 2024; Wang et al., 2024; Zeng et al., 2024; Zhang et al., 2025c; Li et al., 2025a;b; Yu et al., 2026a; Bahri et al., 2025) architectures have further enhanced hierarchical partitioning and long-range contextual modeling, allowing complex spatial dependencies to be captured in both indoor and outdoor scenes.

Distribution- and Feature-Level Regularization. Distribution-level methods regularize output probabilities or supervision signals through entropy minimization (EntropyMin) (Grandvalet & Bengio, 2004), confidence penalty (Conf. Penalty) (Pereyra et al., 2017), probability L2 regularization (ProbL2), and cross-entropy margin regularization (CE-Margin) (Liu et al., 2016), targeting output smoothness, confidence calibration, distributional stability, or class margins. Feature-level methods, including center loss (CenterLoss), triplet margin loss (TripletMargin) (Schroff et al., 2015), supervised contrastive learning (SupCon), and prototype contrastive learning (ProtoCon) (Li et al., 2021), impose geometric constraints to encourage intra-class compactness, inter-class separability, or prototype-based alignment. Despite their effectiveness, these proxy objectives offer limited explicit modeling of point-wise prediction states and G.T.–TopWrong competition. PointCRO organizes point-wise class predictions into competition groups and couples output-space relative optimization with reward-guided feature regularization to support targeted point-level error correction.

3 Method

Overview. Given an input point cloud $\mathbb{P} = (\boldsymbol{x}_j, y_j)_{j=1}^{N}$, where $\boldsymbol{x}_j \in \mathbb{R}^d$ denotes the j-th point, $y_j \in 1, \ldots, C$ denotes its ground-truth (G.T.) semantic label, and N is the number of points, a point cloud segmentation network $f(\cdot; \boldsymbol{\theta})$ predicts point-wise logits $\boldsymbol{z}_j \in \mathbb{R}^C$ and probabilities $\boldsymbol{p}_j = \mathrm{softmax}(\boldsymbol{z}_j)$. Let $\boldsymbol{h}_j$ denote the feature representation immediately preceding the segmentation head. PointCRO is a fine-tuning framework that augments the standard segmentation objective with two complementary components: CGRO in the output space and RFKL regularization in the feature space. CGRO models the class predictions of each point as a competition group and performs prediction-state-aware relative optimization among the G.T., TopWrong, and remaining classes. RFKL further transfers the resulting competition-aware preferences to the representation space, encouraging pre-head features to support stable point-level prediction correction.

3.1 Class-Group Relative Optimization (CGRO)

To explicitly model point-wise class competition in the output space, we introduce CGRO. For each point, CGRO treats its C class predictions as a competition group, enabling relative optimization over the ground-truth class and competing incorrect classes. Let $\mathbb{C} = 1, \ldots, C$ denote the semantic class set. To obtain a stable reference for characterizing the competition state, we maintain an epoch-level reference model. At the beginning of each epoch e, the reference parameters are synchronized as $\theta_{\mathrm{old}}^{(e)} \leftarrow \theta^{(e-1)}$ and remain fixed throughout the epoch, where $\theta^{(e-1)}$ denotes the model parameters from the preceding epoch. The corresponding reference prediction for point j is

$$\boldsymbol{p}_j^{\mathrm{old}} = \mathrm{sg}\Big[\mathrm{softmax}\left(f(\boldsymbol{x}_j; \theta_{\mathrm{old}}^{(e)})\right)\Big], \tag{1}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operation. We first characterize the local competition between the G.T. class and TopWrong class through their probability margin:

$$m_j = p^{\mathrm{old}}_{j,y_j} - p^{\mathrm{old}}_{j,c_j^-}, \quad c_j^- = \underset{c\in\mathbb{C},\, c\neq y_j}{\operatorname{argmax}}\ p^{\mathrm{old}}_{j,c}. \tag{2}$$

Here, m_j characterizes the G.T.–TopWrong competition under the reference model. A smaller margin indicates stronger competition, while $m_j < 0$ indicates that the reference model misclassifies point j. Accordingly, we define the margin-aware factor as

$$g_j = \frac{[\tau - m_j]_+}{\tau + \epsilon}, \tag{3}$$

where τ is the margin threshold, ϵ is a small constant for numerical stability, and $[\cdot]_+$ denotes the positive part. Thus, g_j assigns greater emphasis to points exhibiting stronger G.T.–TopWrong competition. We further characterize the G.T. confidence through

$$\eta_j^{\mathrm{conf}} = 1 - p^{\mathrm{old}}_{j,y_j}, \tag{4}$$

where a larger η_j^{conf} indicates lower confidence in the G.T. prediction. By jointly incorporating the confidence and competition states, the positive reward for the G.T. class is defined as

$$R^+_{j,y_j} = b\left[1 + \lambda_{\mathrm{state}}\left(\eta_j^{\mathrm{conf}} + g_j\right)\right]. \tag{5}$$

where b controls the relative scale of the G.T. reward with respect to the incorrect-class rewards, and λ_{state} controls the joint prediction-state-aware amplification induced by G.T. confidence and G.T.–TopWrong competition within the positive reward formulation.

For incorrect classes, CGRO assigns negative rewards that emphasize dominant or high-confidence competing predictions. For each $c \neq y_j$, we define

$$W_{j,c} = \lambda_{\mathrm{neg}} \mathbf{1}_{\{c=c_j^-\}} \left(1 + \mathbf{1}_{\{p^{\mathrm{old}}_{j,c_j^-} > t_{\mathrm{fp}}\}}\right), \qquad c \neq y_j, \tag{6}$$

where λ_{neg} controls the suppression strength of the TopWrong class, and t_{fp} triggers an additional penalty when the dominant incorrect prediction has high confidence. Thus, the TopWrong class receives stronger suppression when its reference probability exceeds t_{fp}. When an incorrect class is simultaneously the TopWrong class and exceeds t_{fp}, the two indicators are additive, assigning stronger suppression to this dominant high-confidence competitor. The resulting negative reward is

$$R^-_{j,c} = -\,p^{\mathrm{old}}_{j,c}\left(1 + W_{j,c} g_j\right), \qquad c \neq y_j. \tag{7}$$

The class-wise reward is then defined as

$$R_{j,c} = \begin{cases} R^+_{j,y_j}, & c = y_j, \\ R^-_{j,c}, & c \neq y_j. \end{cases} \tag{8}$$

To obtain a point-wise relative learning signal that reflects intra-group class preferences, we normalize the class-wise rewards within each competition group:

$$A_{j,c} = \frac{R_{j,c} - \mu_j}{\sigma_j + \epsilon}, \quad \mu_j = \frac{1}{C}\sum_{c\in\mathbb{C}} R_{j,c}, \quad \sigma_j = \sqrt{\frac{1}{C}\sum_{c\in\mathbb{C}} (R_{j,c} - \mu_j)^2}. \tag{9}$$

Here, $A_{j,c}$ denotes the relative advantage of class c within the competition group of point j. The normalization removes point-dependent reward scales while preserving within-group class preferences under varying point-specific reward magnitudes, such that CGRO operates on relative competition rather than the absolute magnitude of raw rewards. In particular,

$$A_{j,y_j} - A_{j,c_j^-} = \frac{R_{j,y_j} - R_{j,c_j^-}}{\sigma_j + \epsilon}, \tag{10}$$

showing that the G.T.–TopWrong preference is retained after normalization. We then measure the change of each point-class probability relative to the reference prediction:

$$\rho_{j,c}(\theta) = \exp\left(\log[p_{j,c}(\theta) + \epsilon] - \log[p^{\mathrm{old}}_{j,c} + \epsilon]\right). \tag{11}$$

Since $\theta_{\text{old}}^{(e)}$ remains fixed while θ is updated, $\rho_{j,c}(\theta)$ can deviate from one within each epoch. CGRO treats the complete semantic class set of each point as a deterministic competition group and therefore applies uniform outer weighting across classes. Their prediction-state-aware optimization priorities are instead encoded by the relative advantages $A_{j,c}$, allowing the objective to model class competition without additional weighting by the reference probabilities.

$$\mathcal{L}_{\text{CGRO}} = -\frac{1}{NC}\sum_{j=1}^{N}\sum_{c\in\mathbb{C}}\min\left(\rho_{j,c}(\boldsymbol{\theta})A_{j,c}, \text{clip}\left(\rho_{j,c}(\boldsymbol{\theta}), 1-\epsilon_\rho, 1+\epsilon_\rho\right)A_{j,c}\right), \tag{12}$$

where ϵ_ρ determines the probability-ratio clipping range. The resulting relative advantages encode competition-aware preferences among the G.T., TopWrong, and remaining classes, while clipping limits excessive updates beyond the range permitted by the corresponding advantage.

3.2 Reward-guided Feature-level Kullback–Leibler (RFKL) Regularization

CGRO explicitly regulates point-wise class competition in the output space, but does not directly constrain the feature geometry supporting these prediction corrections. We therefore introduce RFKL to propagate the competition-aware preferences into the pre-head representation space to support more stable prediction correction. Based on the group-relative advantages derived by CGRO, we define the reward-guided correction-demand score of point j as

$$d_j = \text{sg}\left(A_{j,y_j} - A_{j,c_j^-}\right), \tag{13}$$

where d_j measures the reward-derived correction demand associated with the G.T.–TopWrong competition. We normalize this score over the point features used by RFKL in the current mini-batch:

$$\widetilde{d}_j = \text{clip}\left(\frac{d_j-\mu_d}{\sigma_d+\epsilon}, -d_{\max}, d_{\max}\right), \quad \mu_d = \frac{1}{N_f}\sum_{j=1}^{N_f} d_j, \quad \sigma_d = \sqrt{\frac{1}{N_f}\sum_{j=1}^{N_f}(d_j-\mu_d)^2}. \tag{14}$$

Here, N_f denotes the number of uniformly sampled point features used by RFKL in the current mini-batch, and $d_{\max}$ limits extreme normalized correction-demand scores to reduce the influence of outliers. Since softmax is shift-invariant, subtracting μ_d does not alter the unclipped relative weights and serves only to center the scores for symmetric clipping.

For each valid anchor i with $|\mathbb{P}(i)| > 0$, we construct the reward-guided reference distribution

$$q_{\text{norm}}(j \mid i) = \frac{\mathbf{1}_{\{j\in\mathbb{P}(i)\}}\exp\left(-\text{sg}\left(\widetilde{d}_j\right)\right)}{\sum_{a\in\mathbb{P}(i)}\exp\left(-\text{sg}\left(\widetilde{d}_a\right)\right)}, \tag{15}$$

where $\mathbb{P}(i) = \{j \mid y_j = y_i, j \neq i\}$ denotes the intra-class reference set for anchor i. Since the exponential weight decreases with $\widetilde{d}_j$, intra-class points with lower correction demand receive larger reference weights, thereby emphasizing reward-preferred references during feature alignment.

Next, we construct a feature-induced distribution from the pre-head point features to capture local representation relationships among points. Let $s_{i,j}$ denote the cosine similarity between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. The probability of assigning point j as a feature-space neighbor of anchor i is defined as

$$p_{\boldsymbol{\theta}}^f(j \mid i) = \frac{\exp\left(s_{i,j}/\tau_f\right)}{\sum_{a\neq i}\exp\left(s_{i,a}/\tau_f\right)}, \qquad j \neq i, \tag{16}$$

where τ_f is the feature temperature controlling the sharpness of the feature-induced neighbor distribution. RFKL aligns the feature-induced distribution with the reward-guided reference distribution to enforce reward-consistent neighborhood structure by minimizing

$$\mathcal{L}_{\text{RFKL}} = \frac{1}{|\mathbb{I}^+|}\sum_{i\in\mathbb{I}^+} D_{\text{KL}}\left(\{q_{\text{norm}}(j \mid i)\}_{j\neq i} \| \{p_{\boldsymbol{\theta}}^f(j \mid i)\}_{j\neq i}\right), \tag{17}$$

where $\mathbb{I}^+ = \{i \mid |\mathbb{P}(i)| > 0\}$ denotes the set of valid anchors with at least one intra-class reference.

Minimizing $D_{\text{KL}}(q_{\text{norm}}|p_{\boldsymbol{\theta}}^f)$ encourages the feature-induced distribution to favor reward-preferred intra-class references. Since the reference distribution is detached from gradient computation, RFKL is gradient-equivalent to a weighted supervised contrastive objective during representation optimization. Its distinction lies in deriving the reference weights from class-relative advantages, thereby incorporating output-space competition information into feature-space regularization.

Table 2: Indoor semantic segmentation mIoU (%) results on ScanNetV2, ScanNet200, and ScanNet++. Our validation results are reported as mean±std over three runs.

Method	ScanNetV2	ScanNet200	ScanNet++	
	Val	Val	Val	Test
OctFormer	75.7	32.6	–	46.0
MinkUNet	72.3	25.0	–	45.6
PTV2	75.4	30.2	–	44.5
PTV3	77.6	35.2	42.1	48.8
LSK3DNet	75.7	33.1	–	–
OACNN	76.1	–	45.8	47.1
+ PointCRO	**76.4** (±0.1)	–	**46.1** (±0.1)	–
VOLT-S	77.3	36.1	50.2	49.3
+ PointCRO	**78.6** (±0.1)	**36.4** (±0.1)	**51.2** (±0.2)	**49.8**

Table 3: Indoor semantic segmentation mIoU (%) results on S3DIS. Our validation results are reported as mean±std over three runs.

Method	Area 5	6-fold
PTV3	73.4	77.7
HPENet	72.7	78.7
PDNet-XXL	72.3	78.3
DeepLA-120	75.7	79.8
Sonata	76.0	82.3
VDG	71.5	73.2
CamPoint	83.3	–
+ PointCRO	**83.6** (±0.2)	–
PointMC	83.6	87.4
+ PointCRO	**84.1** (±0.1)	**87.8** (±0.0)

3.3 Overall Fine-tuning Objective

PointCRO augments the baseline segmentation objective with the CGRO and RFKL objectives:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \lambda_{\text{cgro}}\mathcal{L}_{\text{CGRO}} + \lambda_{\text{rfkl}}\mathcal{L}_{\text{RFKL}}, \tag{18}$$

where $\mathcal{L}_{\text{seg}}$ denotes the baseline segmentation loss, and λ_{cgro} and λ_{rfkl} control the contributions of the output-space and feature-space objectives, respectively. Together, CGRO and RFKL provide complementary prediction- and representation-level supervision for point-level error correction.

4 Experiments

Implementation Details. We evaluate PointCRO on multiple point cloud segmentation backbones to assess its effectiveness and compatibility. The model is optimized using AdamW (Loshchilov & Hutter, 2019) with cosine annealing (Loshchilov & Hutter, 2017). For fair comparison, we follow the original dataset splits and evaluation protocols without re-partitioning or joint training. At the beginning of each epoch, the reference model is synchronized with the current model and then frozen. Both models receive the same augmented input, and the updated model initializes the reference model for the next epoch. Our validation experiments are repeated three times with different random seeds, and mean±standard deviation is reported. For all fine-tuning comparisons, the baseline is initialized from the same checkpoint and optimized for the same number of epochs using only $\mathcal{L}_{\text{seg}}$. Additional implementation details are provided in the Appendix.

4.1 Main Results

Indoor semantic segmentation results. For indoor scene understanding, we evaluate PointCRO on four representative benchmarks: ScanNetV2, ScanNet200 (Rozenberszki et al., 2022), ScanNet++, and S3DIS (Armeni et al., 2016). ScanNetV2 contains 1,513 RGB-D indoor scenes and evaluates semantic segmentation over 20 classes, while ScanNet200 extends the same scans to 200 classes, introducing stronger fine-grained class confusion. ScanNet++ provides higher-fidelity indoor reconstructions, and we follow its released semantic segmentation split with 956 scenes and 100 classes. S3DIS is a benchmark for indoor scene parsing, featuring dense point cloud scans with rich semantic complexity, containing annotated point clouds from 6 areas with 13 semantic categories.

As reported in Tables 2 and 3, PointCRO improves multiple segmentation baselines across indoor benchmarks. When applied to OACNN and VOLT-S, PointCRO consistently improves the evaluated indoor settings. In particular, VOLT-S exhibits gains across ScanNetV2, ScanNet200, and ScanNet++, while OACNN improves on the benchmarks for which results are available. When applied to CamPoint and PointMC on S3DIS, PointCRO also improves the corresponding baselines under different evaluation protocols. These results indicate that the effectiveness of PointCRO is not tied

Table 4: Outdoor semantic segmentation mIoU (%) results on the validation sets of nuScenes, SemanticKITTI, and Waymo. Our validation results are reported as mean±std over three runs.

Method	nuScenes	SemanticKITTI	Waymo
SphereFormer	78.4	67.8	69.9
LSK3DNet	80.1	70.2	–
PTv2	80.2	70.3	70.6
PTv3	80.4	69.1	71.3
VOLT-S	81.1	70.3	71.2
+ PointCRO	**81.7**(±0.2)	**70.7**(±0.1)	**71.8**(±0.1)
LitePT-S	82.2	–	73.1
+ PointCRO	**82.5**(±0.1)	–	**73.3**(±0.1)

Table 5: Ablation study of PointCRO. For the RFKL-only variant, $\lambda_{\text{cgro}} = 0$, while the CGRO-derived reward and advantage are retained only for constructing the reward-guided reference distribution. UFKL denotes a uniform feature-level KL regularization without reward-guided reference weighting.

Method	mIoU ↑	mAcc ↑	OA ↑
VOLT-S Baseline	77.3	85.4	91.8
+ CGRO	77.9	85.6	92.0
+ RFKL only	78.1	86.3	92.0
+ CGRO + UFKL	78.2	86.2	92.1
+ PointCRO	**78.6**	**86.6**	**92.2**

to a specific network architecture or indoor benchmark. Its consistent improvements across different backbones, datasets, and evaluation protocols suggest that prediction-state-aware class competition provides a transferable optimization signal beyond a particular model design or data distribution. Moreover, PointCRO operates entirely at the fine-tuning stage without modifying the underlying segmentation architecture, allowing it to be readily integrated into existing point cloud segmentation models. By directly regulating point-wise competition among the ground-truth, TopWrong, and remaining classes, PointCRO provides a task-aligned optimization objective that complements conventional segmentation supervision across diverse indoor scene understanding settings.

Outdoor semantic segmentation results. To verify the generality of PointCRO, we apply it to two segmentation backbones, VOLT-S and LitePT-S, and evaluate the resulting models on three representative outdoor LiDAR semantic segmentation benchmarks. Specifically, we conduct experiments on nuScenes (Caesar et al., 2020), SemanticKITTI (Behley et al., 2019), and Waymo (Sun et al., 2020). nuScenes provides multi-sensor autonomous driving data with point-level LiDAR semantic annotations in diverse urban scenarios. SemanticKITTI extends the KITTI odometry benchmark with dense point-wise labels for LiDAR-based urban scene understanding. Waymo contains large-scale, high-quality LiDAR sequences collected from complex driving environments.

As shown in Table 4, integrating PointCRO consistently improves the corresponding baselines under the available outdoor evaluation settings. When applied to VOLT-S, PointCRO brings stable gains across all three benchmarks, indicating that the proposed competition-aware optimization remains effective beyond indoor scene understanding. When applied to LitePT-S, PointCRO further improves an already strong baseline, indicating that its benefit is complementary to strong backbone representations rather than dependent on limited model capacity. More importantly, the consistent gains from indoor RGB-D point clouds to outdoor LiDAR scenes suggest that PointCRO is not tied to a particular geometric density or scene distribution. Instead, its effectiveness appears to stem from directly regulating prediction-state-aware class competition, a mechanism that remains relevant across substantially different point cloud segmentation regimes.

4.2 Ablation Studies and Diagnostic Analysis

We conduct comprehensive ablation and diagnostic analyses on the VOLT-S baseline to examine PointCRO from four complementary perspectives. First, we evaluate the contributions of its core components. Second, we analyze the individual terms in the CGRO reward design. Third, we investigate the role of the class-wise relative optimization formulation, including group normalization and probability-ratio clipping. Finally, we examine feature-support changes associated with prediction correction to further characterize the representation-level behavior of PointCRO.

Table 5 evaluates the core components of PointCRO. CGRO improves the baseline, supporting the effectiveness of prediction-state-aware class competition in the output space. For the RFKL-only variant, we set $\lambda_{\text{cgro}} = 0$ and retain the CGRO-derived rewards and advantages solely for constructing the reward-guided reference distribution. Its improvement over the baseline shows that reward-guided feature regularization provides effective representation-level support. Combining CGRO

Table 6: Ablation study on the CGRO reward design. Each part aids performance, showing that PointCRO benefits from adaptive G.T. enhancement, margin-aware competition modeling, and wrong-class suppression.

Method	mIoU ↑	mAcc ↑	OA ↑
PointCRO	**78.6**	**86.6**	**92.2**
- Confidence-aware	78.3	86.6	92.1
- Wrong supp.	78.3	86.2	92.1
- TopWrong	78.1	86.5	91.9
- Margin-aware	77.6	85.6	91.9

Table 7: Ablation of the PointCRO objective. RWCE replaces CGRO with scalar pointwise CE reweighting using the same reference prediction-state information, while retaining RFKL to isolate differences in optimization.

Method	mIoU ↑	mAcc ↑	OA ↑
VOLT-S Baseline	77.3	85.4	91.8
+ RWCE+RFKL	78.1	86.2	92.1
+ PointCRO w/o norm	78.1	86.6	92.0
+ PointCRO w/o clip	78.1	86.4	92.1
+ PointCRO	**78.6**	**86.6**	**92.2**

with UFKL further improves performance, while replacing UFKL with RFKL increases mIoU from 78.2% to 78.6% under the same CGRO objective, indicating the benefit of competition-aware reference weighting over uniform intra-class weighting. These results validate the complementary contributions of output-space relative optimization and reward-guided feature-space regularization.

Table 6 examines the individual components of the CGRO reward design. Removing the confidence-aware term decreases mIoU from 78.6% to 78.3%, indicating that G.T. confidence provides a complementary signal for prediction-state modeling. Removing the margin-aware term causes the largest degradation to 77.3%, highlighting the importance of explicitly modeling G.T.–TopWrong competition. Excluding TopWrong modeling also reduces mIoU to 78.1%, supporting the explicit treatment of the dominant incorrect competitor. The results show that CGRO benefits from jointly modeling G.T. confidence, local class competition, and targeted suppression of competing incorrect classes.

Table 7 evaluates the CGRO optimization formulation. With RFKL retained, replacing scalar pointwise RWCE with CGRO improves mIoU from 78.1% to 78.6%, indicating the benefit of class-wise relative optimization under the same reference-derived prediction-state information. Removing group-relative normalization or probability-ratio clipping reduces mIoU to 78.1%, supporting the contributions of both components to the complete CGRO formulation.

Table 9 analyzes feature-support changes over the corrected points of each method relative to their baseline values. PointCRO exhibits a substantially larger increase in same-class similarity than CGRO and CGRO + UFKL, while achieving a G.T.–TopWrong feature-support margin improvement comparable to CGRO + UFKL. These results associate the prediction corrections of PointCRO with stronger intra-class feature support and an improved relative separation from the dominant incorrect competitor, consistent with the intended output-to-feature correction pathway and providing complementary representation-level evidence for its correction behavior.

4.3 Comparison with Conventional Regularization

To analyze point-level correction behavior, we compare PointCRO with eight regularization strategies using the same VOLT-S baseline on ScanNetV2. Table 8 reports prediction transitions between the baseline and fine-tuned models, with points grouped by their baseline G.T.–TopWrong probability margins. This analysis examines the balance between correcting initially misclassified points and preserving initially correct predictions. Although some conventional regularizers achieve higher correction rates on initially misclassified points, they also exhibit greater degradation of initially correct predictions, particularly in low-margin buckets. For example, CE-Margin achieves the highest correction rate in the $m \leq 0$ bucket, but its greater degradation across all initially correct buckets produces a negative overall net change. PointCRO achieves the lowest degradation rates across these buckets and the largest net gain in overall point accuracy among the evaluated methods. Net change measures the difference between correction and degradation rates, both normalized by the number of evaluated points. These endpoint comparisons support an empirical advantage of PointCRO in balancing error correction with preservation of initially correct predictions under the evaluated fine-tuning setting. The full diagnostics are provided in Appendix Table 12.

Figure 1 compares PointCRO with multiple combinations of conventional regularizers under the same baseline, isolating the contribution of the proposed method from that of simply stacking exist-

Table 8: Bucket-wise point-level transition analysis on ScanNetV2. Corr. and Deg. denote conditional transition rates within their corresponding margin buckets. The overall correction and degradation rates are normalized by all evaluated points, and Net Change is computed as their difference. The margin is defined as $m = p_{\text{G.T.}} - p_{\text{TopWrong}}$. All values are reported in percentage.

Method	Corr. ↑	Deg. ↓			Net Change ↑
	$m \leq 0$	$0 < m \leq 0.1$	$0.1 < m \leq 0.3$	$m > 0.3$	Overall
+ TripletMargin	18.45	41.64	28.93	0.83	+0.15
+ SupCon	18.52	40.38	26.95	0.93	+0.10
+ ProtoCon	16.78	41.95	27.61	0.82	+0.04
+ CenterLoss	16.81	41.62	27.93	0.95	-0.08
+ ProbL2	19.44	41.06	27.24	1.02	+0.08
+ EntropyMin	17.25	44.00	30.27	0.98	-0.12
+ Conf. Penalty	19.49	41.69	29.00	0.99	+0.09
+ CE-Margin	**20.61**	41.05	30.97	1.22	-0.04
+ PointCRO	18.77	**38.52**	**23.95**	**0.59**	**+0.48**

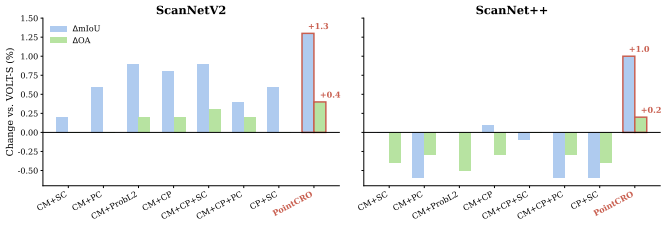


Figure 1: Comparison with combinations of conventional feature- and output-space regularizers on ScanNetV2 and ScanNet++. CM, CP, SC, PC denote CE-Margin, Confidence Penalty, SupCon, ProtoCon, respectively.

Table 9: Feature-support changes on each method's own corrected-point set. $\Delta S_{\text{G.T.}}$ and ΔM_{feat} denote the average changes in same-class similarity and the G.T.–TopWrong feature-support margin, respectively.

Method	$\Delta S_{\text{G.T.}} \uparrow$	$\Delta M_{\text{feat}} \uparrow$
CGRO	0.058	0.148
CGRO + UFKL	0.030	0.241
PointCRO	**0.366**	**0.246**

ing objectives. The evaluated combinations cover both output-space regularization, including CE-Margin, Probability L2, and Confidence Penalty, and feature-space regularization, including SupCon and ProtoCon. Across the two datasets, these conventional combinations do not yield consistent gains over the baseline and can even reduce performance in more challenging settings, indicating that combining regularizers alone is insufficient to provide reliable improvements. In contrast, PointCRO consistently achieves the highest mIoU among all evaluated methods on both datasets. This consistent advantage under the same fine-tuning protocol provides direct empirical evidence that the improvement is attributable to the proposed formulation rather than to the number or type of regularization terms used. A detailed comparison with conventional hard-example, margin-based, and feature-space regularization objectives is provided in Appendix A.6, further clarifying the conceptual and formulation differences between PointCRO and these alternatives.

5 Conclusion

This study introduces PointCRO, a class-competition relative optimization framework for point cloud semantic segmentation. By explicitly modeling the prediction-state-aware competition among the ground-truth class, the TopWrong class, and other incorrect classes, PointCRO shifts the optimization objective from generic proxy regularization toward direct point-level error correction. Specifically, CGRO regulates class competition in the output space, while RFKL transfers the reward-guided correction signal to the representation space to provide targeted feature-level support. Extensive experiments across multiple indoor and outdoor benchmarks demonstrate the effectiveness and generality of PointCRO, consistently improving different segmentation backbones and outperforming conventional feature- and output-space regularization methods.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language polishing. All AI-assisted text was carefully reviewed and revised by the authors. LLM-generated code was verified and tested for correctness by two authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Our work is designed to facilitate reproducibility. The proposed optimization scheme and its implementation details, including the optimization procedure and key training settings, are described in the main paper and supplementary material. Detailed descriptions of the datasets, data preprocessing, and experimental protocols are also provided in the main paper and supplementary material. The source code will be publicly released upon acceptance to facilitate the reproduction and further evaluation of the reported results.

REFERENCES

Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. In *CVPR*, pp. 1534–1543, 2016.

Ali Bahri, Moslem Yazdanpanah, Mehrdad Noori, Sahar Dastani, Milad Cheraghalikhani, Gustavo Adolfo Vargas Hakim, David Osowiechi, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Spectral informed mamba for robust point cloud processing. In *CVPR*, pp. 11799–11809, 2025.

J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, pp. 9297–9307, 2019.

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 *CVPR*, pp. 11618–11628, 2020.

Guangyan Chen, Meiling Wang, Yi Yang, Kai Yu, Li Yuan, and Yufeng Yue. Pointgpt: Auto-regressively generative pre-training from point clouds. In *NeurIPS*, volume 36, pp. 29667–29679, 2023a.

Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. LargeKernel3D: Scaling up kernels in 3D sparse CNNs. In *CVPR*, pp. 13488–13498, 2023b.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pp. 3070–3079, 2019.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, pp. 2432–2443, 2017.

Xin Deng, WenYu Zhang, Qing Ding, and XinMing Zhang. Pointvector: A vector representation in point cloud analysis. In *CVPR*, pp. 9455–9465, 2023.

Tuo Feng, Wenguan Wang, Fan Ma, and Yi Yang. LSK3DNet: Towards effective and efficient 3d perception with large sparse kernels. In *CVPR*, pp. 14916–14927, 2024.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *NeurIPS*, 17, 2004.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In *COLM*, 2024.

Xu Han, Yuan Tang, Zhaoxuan Wang, and Xianzhi Li. Mamba3D: Enhancing Local Features for 3D Point Cloud Analysis via State Space Model. In *ACM MM*, pp. 4995–5004, October 2024.

Zongyan Han, Mohamed El Amine Boudjoghra, Jiahua Dong, Jinhong Wang, and Rao Muhammad Anwer. All in One: Visual-Description-Guided Unified Point Cloud Segmentation. In *ICCV*, pp. 24835–24845, 2025.

Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3D recognition. In *CVPR*, pp. 17545–17555, 2023.

Chade Li, Pengju Zhang, Bo Liu, Hao Wei, and Yihong Wu. FEAST-Mamba: Feature and spatial aware mamba network with bidirectional orthogonal fusion for cross-modal point cloud segmentation. In *AAAI*, volume 39, pp. 4634–4642, 2025a.

Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

Zhe Li, Zhangyang Gao, Cheng Tan, Bocheng Ren, Laurence T. Yang, and Stan Z. Li. General Point Model Pretraining with Autoencoding and Autoregressive. In *CVPR*, pp. 20954–20964, 2024.

Zhuoyuan Li, Yubo Ai, Jiahao Lu, ChuXin Wang, Jiacheng Deng, Hanzhi Chang, Yanzhe Liang, Wenfei Yang, Shifeng Zhang, and Tianzhu Zhang. Pamba: Enhancing global interaction in point clouds via state space model. In *AAAI*, volume 39, pp. 5092–5100, 2025b.

Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. PointMamba: a simple state space model for point cloud analysis. In *NeurIPS*, 2024.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In *ICML*, pp. 507–516, 2016.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking Network Design and Local Geometry in Point Cloud: A Simple Residual MLP Framework. In *ICLR*, 2022.

Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. Fast point transformer. In *CVPR*, pp. 16949–16958, 2022.

Jinyoung Park, Sanghyeok Lee, Sihyeon Kim, Yunyang Xiong, and Hyunwoo J. Kim. Self-positioning point-based transformer for point cloud understanding. In *CVPR*, pp. 21814–21823, 2023.

Bohao Peng, Xiaoyang Wu, Li Jiang, Yukang Chen, Hengshuang Zhao, Zhuotao Tian, and Jiaya Jia. OA-CNNs: Omni-adaptive sparse CNNs for 3D semantic segmentation. In *CVPR*, pp. 21305–21315, 2024.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR*, 2017.

Wentao Qu, Jing Wang, YongShun Gong, Xiaoshui Huang, and Liang Xiao. An end-to-end robust point cloud semantic segmentation network with single-step conditional diffusion models. In *CVPR*, pp. 27325–27335, 2025.

David Rozenberszki, Or Litany, and Angela Dai. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In *ECCV*, volume 13693, pp. 125–141, 2022.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pp. 815–823, 2015.

Kunming Su, Qiuxia Wu, Panpan Cai, Xiaogang Zhu, Xuequan Lu, Zhiyong Wang, and Kun Hu. RI-MAE: Rotation-Invariant Masked AutoEncoders for Self-Supervised Point Cloud Representation Learning. In *AAAI*, volume 39, pp. 7015–7023, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pp. 2446–2454, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, volume 30, 2017.

Peng-Shuai Wang. Octformer: Octree-based transformers for 3D point clouds. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

Zicheng Wang, Zhenghao Chen, Yiming Wu, Zhen Zhao, Luping Zhou, and Dong Xu. Pointramba: A hybrid transformer-mamba framework for point cloud analysis. *arXiv preprint arXiv:2405.15463*, 2024.

Chengzhi Wu, Yuxin Wan, Hao Fu, Julius Pfrommer, Zeyun Zhong, Junwei Zheng, Jiaming Zhang, and Jürgen Beyerer. SAMBLE: Shape-Specific Point Cloud Sampling for an Optimal Trade-Off Between Local Detail and Global Uniformity. In *CVPR*, pp. 1342–1352, 2025a.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, volume 35, pp. 33330–33342, 2022.

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 *CVPR*, pp. 4840–4851, 2024.

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 *CVPR*, pp. 22193–22204, 2025b.

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. *CVM*, 11(1):83–101, 2025.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *ICCV*, pp. 12–22, 2023.

Kadir Yilmaz, Adrian Kruse, Tristan Höfer, Daan de Geus, and Bastian Leibe. Volume Transformer: Revisiting Vanilla Transformers for 3D Scene Understanding. In *ECCV*, 2026.

Xingyilang Yin, Xi Yang, Liangchen Liu, Nannan Wang, and Xinbo Gao. Point deformable network with enhanced normal embedding for point cloud analysis. In *AAAI*, volume 38, pp. 6738–6746, 2024.

Xinxing Yu, Ajian Liu, Sunyuan Qiang, Hui Ma, Liying Yang, Yuzhong Wang, Zhi Rao, and Yanyan Liang. Pointcsp: Cross-sample semantic propagation and stability preservation in self-supervised point cloud learning. In *CVPR*, pp. 10016–10026, 2026a.

Xinxing Yu, Ajian Liu, Sunyuan Qiang, Yuzhong Wang, Hui Ma, and Yanyan Liang. Pointmc: Multi-view consistent encoding and center-global feature fusion for point clouds understanding. In *AAAI*, pp. 12169–12177, 2026b.

Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. LitePT: Lighter Yet Stronger Point Transformer. In *CVPR*, pp. 24173–24184, 2026.

Kang Zeng, Hao Shi, Jiacheng Lin, Siyu Li, Jintao Cheng, Kaiwei Wang, Zhiyong Li, and Kailun Yang. Mambamos: Lidar-based 3D moving object segmentation with motion-aware state space model. In *ACMMM*, pp. 1505–1513, 2024.

Ziyin Zeng, Mingyue Dong, Jian Zhou, Huan Qiu, Zhen Dong, Man Luo, and Bijun Li. DeepLA-Net: Very Deep Local Aggregation Networks for Point Cloud Analysis. In *CVPR*, pp. 1330–1341, 2025.

Yaohua Zha, Huizhen Ji, Jinmin Li, Rongsheng Li, Tao Dai, Bin Chen, Zhi Wang, and Shu-Tao Xia. Towards compact 3D representations via point feature enhancement masked autoencoders. In *AAAI*, volume 38, pp. 6962–6970, 2024.

Yaohua Zha, Tao Dai, Hang Guo, Yanzi Wang, Bin Chen, Ke Chen, and Shu-Tao Xia. Point cloud mixture-of-domain-experts model for 3D self-supervised learning. In James Kwok (ed.), *IJCAI*, pp. 2332–2340, August 2025a.

Yixin Zha, Chuxin Wang, Wenfei Yang, Tianzhu Zhang, and Feng Wu. Exploring vision semantic prompt for efficient point cloud understanding. In *ICML*, volume 267, pp. 74271–74287, 2025b.

Jianhui Zhang, Yizhi Luo, Zicheng Zhang, Xuecheng Nie, and Bonan Li. CamPoint: Boosting Point Cloud Segmentation with Virtual Camera. In *CVPR*, pp. 11822–11832, 2025a.

Qi Zhang, Jibin Peng, Zhao Huang, Wei Feng, and Di Lin. Generative Hard Example Augmentation for Semantic Point Cloud Segmentation. In *CVPR*, pp. 22205–22214, 2025b.

Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-M2AE: multi-scale masked autoencoders for hierarchical point cloud pre-training. In *NeurIPS*, volume 35, pp. 27061–27074, 2022.

Tao Zhang, Haobo Yuan, Lu Qi, Jiangning Zhang, Qianyu Zhou, Shunping Ji, Shuicheng Yan, and Xiangtai Li. Point Cloud Mamba: Point Cloud Learning via State Space Model. In *AAAI*, pp. 10121–10130, 2025c.

Xiangdong Zhang, Shaofeng Zhang, and Junchi Yan. Towards More Diverse and Challenging Pre-training for Point Cloud Learning: Self-Supervised Cross Reconstruction with Decoupled Views. In *ICCV*, pp. 28696–28706, 2025d.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H. S. Torr, and Vladlen Koltun. Point transformer. In *ICCV*, pp. 16239–16248. IEEE, 2021.

Weiguang Zhao, Rui Zhang, Qiufeng Wang, Guangliang Cheng, and Kaizhu Huang. Bfanet: Revisiting 3D semantic segmentation with boundary feature analysis. In *CVPR*, pp. 29395–29405, 2025.

Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3D convolution networks for lidar segmentation. In *CVPR*, pp. 9939–9948, 2021.

Yanmei Zou, Hongshan Yu, Zhengeng Yang, Zechuan Li, and Naveed Akhtar. Improved mlp point cloud processing with high-dimensional positional encoding. In *AAAI*, volume 38, pp. 7891–7899, 2024.

A APPENDIX

A.1 COUNTERFACTUAL SENSITIVITY ANALYSIS OF TOPWRONG COMPETITION

To examine the sensitivity of point-level prediction correction to the dominant incorrect class, we conduct a post-hoc counterfactual intervention analysis on initially misclassified baseline predictions without updating the network parameters. Given an intervention set $\mathcal{S}_j$ and a fixed total suppression budget δ, the intervened logits are defined as

$$\widetilde{z}_{j,c} = \begin{cases} z_{j,c} - \dfrac{\delta}{|\mathcal{S}_j|}, & c \in \mathcal{S}_j, \\ z_{j,c}, & c \notin \mathcal{S}_j. \end{cases} \tag{19}$$

Accordingly, all intervention strategies share the same total suppression budget, while the per-class perturbation magnitude depends on the number of selected incorrect classes.

We compare four allocation strategies: suppressing a randomly selected incorrect class, all incorrect classes, the top-three incorrect classes, or TopWrong alone. As summarized in Table 10, concentrating the fixed suppression budget on TopWrong yields the highest aggregate correction rate on both datasets across all evaluated budget levels. This result indicates that baseline prediction errors are particularly sensitive to perturbations of the dominant incorrect class, motivating its explicit treatment in PointCRO. Importantly, because the fixed budget is distributed across the selected classes, this analysis should be interpreted as a diagnostic study of prediction sensitivity under different budget-allocation strategies rather than as a controlled comparison of equal per-class perturbations. It therefore does not establish the general optimality of TopWrong-only suppression or characterize the updates induced during network training.

Table 10: Counterfactual intervention analysis on initially misclassified points. Given an intervention set $\mathcal{S}_j$, a fixed total suppression budget δ is uniformly allocated across the selected wrong classes, i.e., $\widetilde{z}_{j,c} = z_{j,c} - \delta/|\mathcal{S}_j|$ for $c \in \mathcal{S}_j$. We report the percentage of initially misclassified points corrected to the G.T. class.

Intervention	ScanNetV2			ScanNet++		
	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.5$	$\delta = 0.1$	$\delta = 0.2$	$\delta = 0.5$
No intervention	0.00	0.00	0.00	0.00	0.00	0.00
Random-1 incorrect	0.15	0.29	0.70	0.03	0.05	0.13
All incorrect classes	0.16	0.31	0.77	0.03	0.05	0.13
Top-3 incorrect classes	0.98	1.95	4.75	0.85	1.71	4.31
TopWrong only	**2.87**	**5.61**	**13.23**	**2.55**	**5.11**	**12.66**

A.2 COMPARISON BETWEEN CONVENTIONAL GRPO AND POINTCRO

As summarized in Table 11, conventional GRPO evaluates the quality of sampled responses, whereas CGRO constructs class-specific optimization scores from reference predictions. These scores define relative preferences among the semantic classes associated with each point. Specifically, CGRO assigns positive rewards to the G.T. class and negative rewards to incorrect classes, with additional penalties for TopWrong when the G.T.–TopWrong margin is insufficient. The confidence-aware and margin-aware terms adapt these raw rewards to the reference prediction state. Group normalization subtracts the mean reward across classes and divides by the standard deviation with a numerical stabilizer, producing zero-mean relative advantages while preserving class ordering. As the reference G.T. probability approaches one, the G.T. reward converges to its base value, and the incorrect-class rewards approach zero. These limiting behaviors characterize the raw reward construction and do not directly determine the magnitude of the resulting updates. In particular, this limit does not imply that the normalized advantages vanish, since they depend on the relative configuration of rewards within the class group. With rewards detached from gradient computation, CGRO guides class competition through relative advantages, while the logit gradients of the CGRO objective depend jointly on these advantages, the current and reference probabilities, and the active clipping branches.

Table 11: Comparison between conventional GRPO and the proposed PointCRO. PointCRO reformulates the group-relative principle for discriminative point cloud segmentation.

Aspect	Conventional GRPO	PointCRO / CGRO
Task type	Generative policy optimization	Discriminative semantic segmentation
Group construction	Multiple sampled responses per query	Class predictions of each point
Optimization unit	Response-level sequence probability	Point-class probability
Reward interpretation	Response-quality reward	Prediction-state-aware correction signal
Competition modeling	Relative quality among responses	G.T., TopWrong, and other wrong-class competition
Stabilization	Group-normalized advantage and clipped ratio	Class-group advantage and clipped probability ratio
Feature-space support	Usually not considered	RFKL regularizes pre-head features

A.3 Definitions of Diagnostic Proxy Metrics

Table 1 uses several feature- and output-space diagnostic metrics to examine whether improvements in commonly optimized surrogate properties are consistently associated with improvements in semantic segmentation performance. These metrics are used only for post-hoc analysis and are not included in the training objectives of PointCRO. All methods are evaluated on the same validation split, and the feature-based metrics are computed from the pre-head point representations used by the corresponding segmentation model.

Notation. For an evaluated model, let $\boldsymbol{h}_i \in \mathbb{R}^D$ denote the pre-head feature of point i, $\boldsymbol{p}_i \in \mathbb{R}^C$ denote its predicted class-probability vector, and y_i denote its ground-truth semantic label. We first ℓ_2-normalize the feature representation as

$$\bar{\boldsymbol{h}}_i = \frac{\boldsymbol{h}_i}{\|\boldsymbol{h}_i\|_2}. \tag{20}$$

Let $\mathcal{I}_c = i \mid y_i = c$ denote the set of evaluated points belonging to class c. The normalized feature prototype of class c is defined as

$$\boldsymbol{\mu}_c = \frac{\sum_{i \in \mathcal{I}_c} \bar{\boldsymbol{h}}_i}{\left\| \sum_{i \in \mathcal{I}_c} \bar{\boldsymbol{h}}_i \right\|_2}. \tag{21}$$

For each point, we define its most competitive incorrect class, denoted as TopWrong, according to the model prediction:

$$c_i^- = \underset{c \neq y_i}{\operatorname{argmax}}; p * i, c. \tag{22}$$

Thus, all feature- and output-space diagnostics are evaluated with respect to the prediction state of the model being analyzed.

Intra-class compactness (Intra-Comp.). We quantify intra-class feature compactness by measuring the cosine distance between each point feature and its corresponding ground-truth class prototype:

$$D_{\text{intra}} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \bar{\boldsymbol{h}} * i^\top \boldsymbol{\mu} * y_i\right). \tag{23}$$

A smaller D_{intra} indicates that point features are more tightly concentrated around their ground-truth class prototypes. We report D_{intra} as *Intra-Comp.* in Table 1. This metric characterizes intra-class feature concentration but does not account for the location of competing classes; consequently, a lower intra-class dispersion does not necessarily imply improved semantic discrimination.

Feature-space competition margin (Feat. Margin). To characterize feature-level competition between the ground-truth class and the dominant incorrect prediction, we compute the difference between their prototype similarities:

$$m_i^f = \bar{\boldsymbol{h}} * i^\top \boldsymbol{\mu} * y_i - \bar{\boldsymbol{h}} * i^\top \boldsymbol{\mu} * c_i^-. \tag{24}$$

The dataset-level feature margin is then

$$M_{\text{feat}} = \frac{1}{N} \sum_{i=1}^{N} m_i^f. \tag{25}$$

A larger M_{feat} indicates stronger relative feature support for the ground-truth class over the current TopWrong class. Unlike conventional inter-class distances computed independently of the prediction state, this metric explicitly evaluates the feature-space separation associated with the dominant point-wise competitor. We report M_{feat} as *Feat. Margin*.

Expected calibration error (ECE). We additionally measure the calibration of the output distribution using Expected Calibration Error. For each point, let

$$\hat{y} * i = \underset{c}{\text{argmax}}; p * i, c, \qquad q_i = \max_c; p_{i,c} \tag{26}$$

denote the predicted label and its confidence, respectively. We partition the confidence interval $[0, 1]$ into B disjoint bins $\mathcal{B} * b * b = 1^B$. For each bin, prediction accuracy and mean confidence are computed as

$$\text{acc}(\mathcal{B}_b) = \frac{1}{|\mathcal{B} * b|} \sum * i \in \mathcal{B} * b\mathbf{1} * \hat{y}_i = y_i, \tag{27}$$

and

$$\text{conf}(\mathcal{B}_b) = \frac{1}{|\mathcal{B} * b|} \sum * i \in \mathcal{B} * bq_i. \tag{28}$$

ECE is defined as

$$\text{ECE} = \sum * b = 1^B \frac{|\mathcal{B}_b|}{N} \left|\text{acc}(\mathcal{B}_b) - \text{conf}(\mathcal{B}_b)\right|. \tag{29}$$

A lower ECE indicates better agreement between prediction confidence and empirical correctness. The same binning protocol is used for all compared methods.

TopWrong suppression (TopWrong Supp.). To quantify how strongly a regularizer suppresses the dominant incorrect prediction, we first compute the average TopWrong confidence of model $\mathcal{M}$:

$$T(\mathcal{M}) = \frac{1}{N} \sum_{i=1}^{N} \max_{c \neq y_i} p_{i,c}^{\mathcal{M}}. \tag{30}$$

Let $\mathcal{M} * 0$ denote the corresponding unregularized VOLT-S baseline and $\mathcal{M}$ denote a model trained with an additional regularization objective. We define TopWrong suppression relative to the baseline as

$$S * \text{TW}(\mathcal{M}) = T(\mathcal{M} * 0) - T(\mathcal{M}). \tag{31}$$

Accordingly,

$$S * \text{TW}(\mathcal{M}_0) = 0. \tag{32}$$

A positive value indicates that the regularized model decreases the average confidence of its strongest incorrect prediction relative to the baseline, whereas a negative value indicates an increase. This baseline-relative definition explains the zero TopWrong Supp. value reported for VOLT-S in Table 1.

Computation protocol. All diagnostic metrics are computed after training with the model in evaluation mode. For each compared method, we extract the pre-head features and output probabilities from exactly the same validation samples. Class prototypes and feature-space metrics are computed independently for each trained model, whereas TopWrong Supp. is measured relative to the corresponding unregularized VOLT-S baseline. No diagnostic metric is used for model selection or directly optimized during training. Therefore, Table 1 is intended to analyze the relationship between surrogate properties and downstream segmentation accuracy rather than to define additional training objectives.

The four diagnostics capture complementary properties: Intra-Comp. measures intra-class feature concentration, Feat. Margin measures prediction-state-dependent feature separation between the G.T. and TopWrong classes, ECE measures output calibration, and TopWrong Supp. measures the reduction of dominant incorrect confidence relative to the baseline. Their disagreement with mIoU in Table 1 therefore indicates that improving an individual feature- or output-space proxy does not necessarily translate into improved point-level semantic segmentation.

A.4 Point-level competition diagnostics on ScanNetV2 and ScanNet++

Table 14 reports the aggregate segmentation performance of conventional regularizers relative to the VOLT-S baseline. To examine the corresponding changes in individual predictions, we provide bucket-wise competition diagnostics on ScanNetV2 and ScanNet++ in Tables 12 and 13, respectively. Points are grouped according to their baseline G.T.–TopWrong probability margins, and these bucket assignments are retained across the evaluated methods. The diagnostics compare baseline and fine-tuned predictions: correction records transitions from initially incorrect to correct predictions, while degradation records transitions from initially correct to incorrect predictions. Within each margin bucket, both transition rates are normalized by the number of points in that bucket, whereas the overall rates are normalized by the total number of evaluated points. Net change is computed as correction minus degradation under the corresponding normalization; the overall net change therefore equals the change in overall point accuracy on the same evaluation set. We additionally report changes in G.T. confidence, TopWrong probability suppression, and the G.T.–TopWrong margin relative to the baseline to characterize local class competition alongside prediction transitions. Transition rates are reported as percentages, while probability differences, margin differences, and net accuracy changes are expressed in percentage points. The #Points column reports the number of evaluated points in each group. For compactness, ProtoCon and Conf. Penalty denote Prototype-Contrastive and Confidence Penalty, respectively.

On ScanNetV2, the comparison between baseline and fine-tuned predictions shows that PointCRO achieves the highest overall net change and the lowest overall degradation among the evaluated regularizers. Although CE-Margin obtains the highest correction rate on initially misclassified points, its newly introduced errors among initially correct points outweigh its corrections, resulting in a negative overall net change. PointCRO maintains a competitive correction rate and achieves the lowest degradation rates across all initially correct margin buckets. This preservation advantage is particularly evident in the low-margin bucket, $0 < m \leq 0.1$, and the medium-margin bucket, $0.1 < m \leq 0.3$. Both groups contain correctly classified baseline points with limited G.T.–TopWrong probability separation. These endpoint results indicate that PointCRO combines correction of baseline errors with greater preservation of initially correct predictions, yielding a positive net accuracy change under the evaluated fine-tuning setting.

On ScanNet++, the endpoint prediction-transition diagnostics reveal a trend consistent with that observed on ScanNetV2. Several conventional regularizers obtain strong correction rates or favorable degradation values in individual margin buckets, but their overall net changes remain limited or negative. Such bucket-specific improvements do not necessarily translate into an overall accuracy gain, as corrections can be offset by newly introduced errors among initially correct points. Among the evaluated methods, PointCRO achieves the highest overall net change and the lowest overall degradation. Its advantage in preserving correct predictions is particularly evident in the medium-margin bucket, $0.1 < m \leq 0.3$, where it exhibits the lowest degradation rate. These points are correctly classified by the baseline but have a smaller G.T.–TopWrong probability margin than points in the high-margin bucket. These endpoint observations support a favorable balance between error correction and preservation of initially correct predictions on ScanNet++ under the evaluated protocol, complementing the corresponding findings on ScanNetV2.

A.5 Analytical Comparison with Weighted CE and Margin-based Objectives

This subsection clarifies how CGRO differs from point-wise loss reweighting and margin-based supervision. The comparison is useful because all three objectives can assign stronger updates to difficult predictions, but they do so at different levels of granularity.

In the RWCE comparison, the baseline segmentation loss and RFKL are retained, and only the CGRO output objective is replaced. The following analysis therefore focuses on the differing output terms. Let w_j denote a detached scalar weight computed from the same reference-derived prediction-state information used by CGRO. The RWCE objective for point j is defined as

$$\mathcal{L}_{\mathrm{RWCE},j} = -w_j \log p_{j,y_j}, \tag{33}$$

where w_j is a scalar point-wise weight determined by the prediction difficulty. Its logit gradient is

$$\frac{\partial \mathcal{L}_{\mathrm{RWCE},j}}{\partial z_{j,k}} = w_j \left(p_{j,k} - \mathbf{1}_{\{k=y_j\}}\right). \tag{34}$$

Table 12: Full point-level competition diagnostics across margin buckets on ScanNetV2. Bucket-wise correction and degradation rates are normalized within each bucket, whereas the overall rates are normalized by the total number of evaluated points. Except for #Points, all values are reported in percentage.

Method	Bucket	#Points	$\Delta p_{\mathrm{G.T.}}\uparrow$	TW Supp. ↑	ΔMargin ↑	Corr. ↑	Deg. ↓	Net ↑
Base.	All	35,881,054	0.00	0.00	0.00	0.00	0.00	0.00
+ TripletMargin	$m \le 0$	2,953,945	8.48	10.77	19.24	18.45	0.00	18.45
	$0 < m \le 0.1$	212,537	4.36	3.41	7.77	0.00	41.64	-41.64
	$0.1 < m \le 0.3$	467,421	2.79	1.77	4.56	0.00	28.93	-28.93
	$m > 0.3$	32,247,151	-1.10	-0.73	-1.83	0.00	0.83	-0.83
	Overall	35,881,054	-0.23	0.27	0.04	1.52	1.37	0.15
+ SupCon	$m \le 0$	2,953,945	8.76	10.19	18.95	18.52	0.00	18.52
	$0 < m \le 0.1$	212,537	5.22	3.93	9.15	0.00	40.38	-40.38
	$0.1 < m \le 0.3$	467,421	3.75	2.38	6.13	0.00	26.95	-26.95
	$m > 0.3$	32,247,151	-0.84	-0.72	-1.55	0.00	0.93	-0.93
	Overall	35,881,054	0.05	0.25	0.30	1.52	1.43	0.10
+ ProtoCon	$m \le 0$	2,953,945	6.70	9.93	16.63	16.78	0.00	16.78
	$0 < m \le 0.1$	212,537	3.20	4.71	7.91	0.00	41.95	-41.95
	$0.1 < m \le 0.3$	467,421	2.50	3.76	6.26	0.00	27.61	-27.61
	$m > 0.3$	32,247,151	-1.28	-0.64	-1.91	0.00	0.82	-0.82
	Overall	35,881,054	-0.55	0.32	-0.22	1.38	1.34	0.04
+ ProbL2	$m \le 0$	2,953,945	9.31	11.59	20.90	19.44	0.00	19.44
	$0 < m \le 0.1$	212,537	4.32	4.06	8.38	0.00	41.06	-41.06
	$0.1 < m \le 0.3$	467,421	2.84	2.42	5.26	0.00	27.24	-27.24
	$m > 0.3$	32,247,151	-2.20	-0.91	-3.11	0.00	1.02	-1.02
	Overall	35,881,054	-1.15	0.19	-0.96	1.60	1.52	0.08
+ EntropyMin	$m \le 0$	2,953,945	7.84	2.52	10.36	17.25	0.00	17.25
	$0 < m \le 0.1$	212,537	8.80	-0.60	8.21	0.00	44.00	-44.00
	$0.1 < m \le 0.3$	467,421	9.73	0.14	9.88	0.00	30.27	-30.27
	$m > 0.3$	32,247,151	5.53	0.14	5.67	0.00	0.98	-0.98
	Overall	35,881,054	5.80	0.33	6.12	1.42	1.54	-0.12
+ Conf. Penalty	$m \le 0$	2,953,945	-1.75	51.87	50.12	19.49	0.00	19.49
	$0 < m \le 0.1$	212,537	-26.56	24.70	-1.86	0.00	41.69	-41.69
	$0.1 < m \le 0.3$	467,421	-33.17	19.18	-14.00	0.00	29.00	-29.00
	$m > 0.3$	32,247,151	-62.65	-2.89	-65.54	0.00	0.99	-0.99
	Overall	35,881,054	-57.04	2.07	-54.97	1.60	1.52	0.09
+ CenterLoss	$m \le 0$	2,953,945	7.07	9.15	16.22	16.81	0.00	16.81
	$0 < m \le 0.1$	212,537	4.55	3.67	8.22	0.00	41.62	-41.62
	$0.1 < m \le 0.3$	467,421	2.95	2.12	5.07	0.00	27.93	-27.93
	$m > 0.3$	32,247,151	-1.08	-0.78	-1.86	0.00	0.95	-0.95
	Overall	35,881,054	-0.33	0.10	-0.23	1.38	1.47	-0.08
+ CE-Margin	$m \le 0$	2,953,945	11.28	11.75	23.03	**20.61**	0.00	**20.61**
	$0 < m \le 0.1$	212,537	5.19	2.12	7.31	0.00	41.05	-41.05
	$0.1 < m \le 0.3$	467,421	2.69	-0.38	2.31	0.00	30.97	-30.97
	$m > 0.3$	32,247,151	0.67	-1.26	-0.58	0.00	1.22	-1.22
	Overall	35,881,054	1.60	-0.15	1.44	**1.70**	1.74	-0.04
+ PointCRO	$m \le 0$	2,953,945	8.20	9.44	17.64	18.77	0.00	18.77
	$0 < m \le 0.1$	212,537	6.27	4.47	10.74	0.00	**38.52**	**-38.52**
	$0.1 < m \le 0.3$	467,421	5.95	3.90	9.86	0.00	**23.95**	**-23.95**
	$m > 0.3$	32,247,151	-0.31	-0.36	-0.67	0.00	**0.59**	**-0.59**
	Overall	35,881,054	0.52	0.53	1.04	1.55	**1.07**	**0.48**

Thus, RWCE incorporates prediction-state information through a scalar weight that rescales the cross-entropy logit gradient without changing its direction for each point. It retains the class competition inherent in cross-entropy, while CGRO introduces additional class-wise advantages and reference-relative probability ratios that can alter the allocation of gradients across classes.

In contrast, ignoring the probability stabilizer and the clipping boundary for the moment, the CGRO objective for point j is locally

$$\widetilde{\mathcal{L}}_{\mathrm{CGRO},j} = -\frac{1}{C}\sum_{c\in\mathbb{C}} A_{j,c}\frac{p_{j,c}}{p_{j,c}^{\mathrm{old}}}. \tag{35}$$

Table 13: Full point-level competition diagnostics across margin buckets on ScanNet++. Bucket-wise correction and degradation rates are normalized within each bucket, whereas the overall rates are normalized by the total number of evaluated points. Except for #Points, all values are reported in percentage.

Method	Bucket	#Points	$\Delta p_{\mathrm{G.T.}}\uparrow$	TW Supp. ↑	ΔMargin ↑	Corr. ↑	Deg. ↓	Net ↑
Base.	All	55,704,463	0.00	0.00	0.00	0.00	0.00	0.00
+ TripletMargin	$m \le 0$	6,554,370	4.83	6.16	10.98	**15.30**	0.00	**15.30**
	$0 < m \le 0.1$	460,479	2.39	2.29	4.68	0.00	**42.81**	**-42.81**
	$0.1 < m \le 0.3$	973,255	-0.88	-0.56	-1.44	0.00	32.69	-32.69
	$m > 0.3$	47,716,359	-0.45	-0.80	-1.25	0.00	1.15	-1.15
	Overall	55,704,463	0.19	0.04	0.24	**1.80**	1.91	-0.11
+ SupCon	$m \le 0$	6,554,370	3.74	5.77	9.51	14.22	0.00	14.22
	$0 < m \le 0.1$	460,479	0.44	1.50	1.95	0.00	43.78	-43.78
	$0.1 < m \le 0.3$	973,255	-0.22	0.86	0.64	0.00	28.71	-28.71
	$m > 0.3$	47,716,359	-0.28	-0.46	-0.74	0.00	0.91	-0.91
	Overall	55,704,463	0.20	0.31	0.51	1.67	1.65	0.03
+ ProtoCon	$m \le 0$	6,554,370	3.62	6.46	10.08	13.28	0.00	13.28
	$0 < m \le 0.1$	460,479	-0.48	2.29	1.81	0.00	46.69	-46.69
	$0.1 < m \le 0.3$	973,255	-0.50	1.95	1.45	0.00	29.48	-29.48
	$m > 0.3$	47,716,359	-0.68	-0.39	-1.06	0.00	**0.79**	**-0.79**
	Overall	55,704,463	-0.17	0.48	0.32	1.56	1.57	-0.01
+ ProbL2	$m \le 0$	6,554,370	3.27	5.53	8.80	12.95	0.00	12.95
	$0 < m \le 0.1$	460,479	0.97	1.40	2.37	0.00	45.20	-45.20
	$0.1 < m \le 0.3$	973,255	-0.40	0.09	-0.31	0.00	31.45	-31.45
	$m > 0.3$	47,716,359	-0.91	-0.59	-1.50	0.00	1.11	-1.11
	Overall	55,704,463	-0.39	0.16	-0.23	1.52	1.88	-0.35
+ EntropyMin	$m \le 0$	6,554,370	4.61	-2.38	2.23	14.82	0.00	14.82
	$0 < m \le 0.1$	460,479	7.78	-3.00	4.78	0.00	44.86	-44.86
	$0.1 < m \le 0.3$	973,255	8.02	-2.19	5.83	0.00	31.75	-31.75
	$m > 0.3$	47,716,359	6.89	-0.08	6.81	0.00	1.44	-1.44
	Overall	55,704,463	6.65	-0.42	6.24	1.74	2.16	-0.41
+ Conf. Penalty	$m \le 0$	6,554,370	-2.22	35.53	33.31	13.95	0.00	13.95
	$0 < m \le 0.1$	460,479	-18.56	16.50	-2.06	0.00	45.64	-45.64
	$0.1 < m \le 0.3$	973,255	-24.03	12.36	-11.67	0.00	34.45	-34.45
	$m > 0.3$	47,716,359	-38.31	-0.49	-38.80	0.00	1.25	-1.25
	Overall	55,704,463	-33.65	4.11	-29.54	1.64	2.05	-0.41
+ CenterLoss	$m \le 0$	6,554,370	3.51	6.58	10.09	13.66	0.00	13.66
	$0 < m \le 0.1$	460,479	-0.24	2.10	1.86	0.00	47.17	-47.17
	$0.1 < m \le 0.3$	973,255	-2.30	0.48	-1.82	0.00	31.97	-31.97
	$m > 0.3$	47,716,359	-0.52	-0.62	-1.14	0.00	0.98	-0.98
	Overall	55,704,463	-0.08	0.27	0.19	1.61	1.79	-0.18
+ CE-Margin	$m \le 0$	6,554,370	5.17	2.90	8.07	13.73	0.00	13.73
	$0 < m \le 0.1$	460,479	3.60	-0.84	2.75	0.00	45.95	-45.95
	$0.1 < m \le 0.3$	973,255	2.08	-2.00	0.08	0.00	33.58	-33.58
	$m > 0.3$	47,716,359	3.69	-0.87	2.82	0.00	1.25	-1.25
	Overall	55,704,463	3.84	-0.45	3.39	1.62	2.04	-0.42
+ PointCRO	$m \le 0$	6,554,370	4.22	4.70	8.92	13.53	0.00	13.53
	$0 < m \le 0.1$	460,479	2.08	0.34	2.41	0.00	43.01	-43.01
	$0.1 < m \le 0.3$	973,255	1.38	-0.38	1.00	0.00	**26.99**	**-26.99**
	$m > 0.3$	47,716,359	0.45	-0.58	-0.13	0.00	0.81	-0.81
	Overall	55,704,463	0.92	0.05	0.98	1.59	**1.52**	**0.07**

Let $\alpha_{j,c} = A_{j,c}/p_{j,c}^{\mathrm{old}}$. The corresponding logit gradient is

$$\frac{\partial \widetilde{\mathcal{L}}_{\mathrm{CGRO},j}}{\partial z_{j,k}} = -p_{j,k}\left(\alpha_{j,k} - \sum_{c\in\mathbb{C}} p_{j,c}\alpha_{j,c}\right). \tag{36}$$

The second term in Eq. 36 couples the update of each class to the advantages of all other classes in the same prediction group. Consequently, CGRO assigns class-specific directions to the G.T., Top-Wrong, and remaining wrong classes, rather than applying one scalar difficulty weight to the whole point. The factor $1/p_{j,c}^{\mathrm{old}}$ also makes the update relative to the epoch-level reference distribution. At a reference-matching prediction $p_j = p_j^{\mathrm{old}}$, ignoring the probability stabilizer and with clipping

Table 14: Comparison with conventional distribution-level and feature-level regularization methods on ScanNetV2 and ScanNet++. All methods are applied on top of the same VOLT-S baseline.

Method	ScanNetV2		ScanNet++	
	mIoU (%) ↑	OA (%) ↑	mIoU (%) ↑	OA (%) ↑
VOLT-S Baseline	77.3	91.8	50.2	88.2
+ EntropyMin	77.3	91.6	<u>50.9</u>	87.9
+ Conf. Penalty	77.6	91.8	50.2	87.8
+ ProbL2	77.9	91.9	50.8	87.8
+ CE-Margin	77.4	91.7	50.2	87.8
+ ProtoCon	77.8	91.8	<u>50.9</u>	88.2
+ SupCon	77.6	92.0	50.6	<u>88.3</u>
+ CenterLoss	77.1	91.7	49.5	88.1
+ TripletMargin	77.8	91.9	50.2	88.1
+ OHEM	77.6	91.9	50.2	87.7
+ Focal	<u>78.0</u>	92.0	50.7	87.6
+ G.T.-TopWrong margin	77.9	<u>92.1</u>	50.3	88.1
+ Confidence Weighted Feature	77.9	91.9	50.4	87.8
+ PointCRO	**78.6**	**92.2**	**51.2**	**88.3**

inactive, Eq. (22) simplifies to

$$\frac{\partial \widetilde{\mathcal{L}}_{\mathrm{CGRO},j}}{\partial z_{j,k}} = -A_{j,k},$$

because $\sum_c A_{j,c} = 0$. Consequently, a gradient descent step of size η on the independent logits gives

$$(z'_{j,y_j} - z'_{j,c_j^-}) - (z_{j,y_j} - z_{j,c_j^-}) = \eta \left(A_{j,y_j} - A_{j,c_j^-} \right) > 0.$$

This local property characterizes how the relative reward preference promotes the G.T.–TopWrong logit margin. It does not guarantee monotonic margin improvement for individual points during shared-parameter network training, where parameter coupling, additional losses, and clipping also affect the updates. For comparison, a typical G.T.–TopWrong margin objective has the form

$$\mathcal{L}_{\mathrm{margin},j} = \left[m_0 - (z_{j,y_j} - z_{j,c_j^-}) \right]_+, \tag{37}$$

and directly modifies only the G.T./TopWrong logit difference when the margin is violated. It does not assign separate relative advantages to the remaining wrong classes, nor does it restrict the change relative to an epoch-level reference distribution.

Therefore, CGRO is not equivalent to simply reweighting cross-entropy or imposing a fixed pairwise margin. Its distinguishing elements are (i) class-wise advantages derived from the prediction-state-aware reward, (ii) coupling among all classes through group normalization and the softmax simplex, and (iii) reference-relative probability-ratio clipping. The empirical comparisons with RWCE, CE-Margin, and the "w/o clip" variant complement this analytical distinction.

A.6 Relationship to Conventional Regularization Objectives

For all competing regularizers, we tune their method-specific hyperparameters on the validation split using the same search budget adopted for PointCRO. The selected configurations are then fixed across the corresponding comparisons.

Baseline objective and comparison protocol. In all experiments, the baseline segmentation objective combines cross-entropy and Lovasz-Softmax losses:

$$\mathcal{L}_{\mathrm{seg}} = \mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{lovasz}} \mathcal{L}_{\mathrm{Lovasz}}, \tag{38}$$

where $\lambda_{\mathrm{lovasz}}$ follows the corresponding baseline configuration. The same segmentation objective, initialization, training schedule, and data processing are used for the baseline, PointCRO, and all

conventional regularization methods. Therefore, the reported differences are attributed to the additional regularization objective rather than to a change in the underlying segmentation loss. The complete fine-tuning objective of PointCRO is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \lambda_{\text{cgro}}\mathcal{L}_{\text{CGRO}} + \lambda_{\text{rfkl}}\mathcal{L}_{\text{RFKL}}. \tag{39}$$

Relation to hard-example reweighting. Focal-style objectives and hard-example mining methods modulate point-level contributions according to prediction difficulty. The RWCE baseline additionally uses the same reference-derived prediction-state information as CGRO, but aggregates it into a scalar point-wise weight. CGRO converts this information into class-wise relative advantages and applies a reference-relative objective, allowing it to modify the allocation of updates across classes beyond scalar rescaling of the cross-entropy gradient.

Relation to margin-based objectives. Margin-based objectives generally encourage separation between the ground-truth class and one negative class through a fixed or pairwise margin. CGRO instead constructs a class competition group from the epoch-level reference prediction and jointly regulates all classes through group-relative advantages. Its correction signal depends on the ground-truth confidence, the ground-truth-versus-TopWrong margin, and the probabilities of competing incorrect classes. Furthermore, the clipped probability-ratio objective constrains changes relative to a frozen reference distribution rather than directly enforcing a fixed logit margin.

Relation to Lovasz-Softmax loss. Lovasz-Softmax is an IoU-oriented surrogate that constructs a piecewise differentiable approximation to the Jaccard loss from the sorted prediction errors. Although it may assign larger effective weights to high-impact errors, it is different from focal-style confidence reweighting, which modulates point-wise cross-entropy according to the ground-truth probability. Both objectives operate at the task-loss level and do not assign reference-derived class-wise advantages to the ground-truth, TopWrong, and other incorrect classes. CGRO is complementary to the baseline CE and Lovasz-Softmax objectives because it introduces an additional prediction-state-aware class-competition signal derived from the epoch-level reference predictions.

Relation to supervised feature regularization. Standard supervised contrastive learning assigns uniform target weights to same-class positive pairs while contrasting them against other samples. RFKL retains this contrastive structure but introduces a non-uniform intra-class reference distribution:

$$q_{\text{norm}}(j \mid i) \propto \exp\Big(-\widetilde{d}_j\Big), \qquad j \in \mathbb{P}(i), \tag{40}$$

where $\widetilde{d}_j$ is obtained by batch-wise standardization and clipping of the G.T.–TopWrong class-relative advantage gap computed from epoch-level reference predictions. The target distribution is detached from gradient computation, and same-class references with smaller scores receive larger target weights. Consequently, RFKL is gradient-equivalent to a contrastive objective with non-uniform positive weights. Its distinguishing component is the construction of these weights from reference-derived class-competition information, which guides feature-level reference weighting.

Coupling between output and feature spaces. PointCRO combines CGRO and RFKL through a shared signal derived from epoch-level reference predictions. CGRO constructs class-specific rewards from the reference prediction state and converts them into group-relative advantages for output-space optimization. RFKL uses the G.T.–TopWrong advantage gap to compute reference-weighting scores, which undergo batch-wise standardization and clipping. The resulting scores determine nonuniform weights over same-class reference points, with lower scores receiving larger weights. These weights define a detached target distribution for the feature-level KL objective, which aligns feature-induced neighbor probabilities with the reward-guided reference distribution. CGRO and RFKL remain additive terms in the overall segmentation objective, while their optimization targets share the same underlying class-competition information.

Empirical comparison. Table 8 compares PointCRO with eight representative feature- and output-space regularizers under the same VOLT-S baseline. Several conventional regularizers obtain higher correction rates on initially misclassified points, but they also cause larger degradation among initially correct points. PointCRO achieves the lowest degradation rates across the positive-margin buckets and the highest overall net change, indicating a more balanced correction behavior.

Table 15: Comparison with combinations of conventional feature- and output-space regularizers on ScanNetV2 and ScanNet++. This experiment examines whether the improvement of PointCRO can be reproduced by directly stacking generic regularization terms from different spaces.

Method	ScanNetV2		ScanNet++	
	mIoU (%) ↑	OA (%) ↑	mIoU (%) ↑	OA (%) ↑
VOLT-S Baseline	77.3	91.8	50.2	<u>88.2</u>
+ CE-Margin + SupCon	77.5	91.8	50.2	87.8
+ CE-Margin + ProtoCon	77.9	91.8	49.6	87.9
+ CE-Margin + ProbL2	<u>78.2</u>	92.0	50.2	87.7
+ CE-Margin + Conf. Penalty	78.1	92.0	<u>50.3</u>	87.9
+ CE-Margin + Conf. Penalty + SupCon	<u>78.2</u>	<u>92.1</u>	50.1	<u>88.2</u>
+ CE-Margin + Conf. Penalty + ProtoCon	77.7	92.0	49.6	87.9
+ Conf. Penalty + SupCon	77.9	91.8	49.6	87.8
+ PointCRO	**78.6**	**92.2**	**51.2**	**88.3**

Table 15 evaluates direct combinations of conventional feature- and output-space regularizers. Although several combinations improve the ScanNetV2 baseline, their gains are smaller or less consistent on ScanNet++. PointCRO achieves the best result on both datasets. These results show that, among the regularizers and combinations evaluated in our experiments, the improvement of PointCRO is not reproduced by simply stacking additional conventional regularization terms.

A.7 Parameter-efficient fine-tuning with LoRA

We further evaluate PointCRO in a parameter-efficient fine-tuning setting using LoRA to examine its applicability when model adaptation is restricted to a limited set of trainable parameters. Specifically, the main segmentation backbone is frozen, and only the inserted LoRA parameters are optimized. LoRA modules are introduced into the attention query, key, value, and output projection layers, with rank $r = 8$, scaling factor $\alpha = 16$, and dropout rate $p_{\text{drop}} = 0.05$. Although the original backbone parameters remain fixed, the trainable low-rank updates can modify intermediate representations and class predictions, allowing the output-space and feature-space objectives of PointCRO to guide adaptation through the inserted modules.

Using the same LoRA fine-tuning baseline, adapter placement, and trainable parameter configuration, we compare PointCRO with conventional distribution-level and feature-level regularization methods. This comparison examines the effectiveness of the proposed competition-aware optimization signal within a common, constrained adaptation setting. As reported in Table 16, PointCRO achieves the highest mIoU among the evaluated methods while maintaining competitive OA. These results support the compatibility of PointCRO with the evaluated LoRA configuration and suggest that its optimization objectives remain useful when parameter updates are restricted to low-rank adapters while the original backbone parameters remain fixed.

A.8 PointCRO Implementation Details

The PointCRO-specific hyperparameter settings for different backbones and datasets are summarized in Table 17 and Table 18. These tables specify the fine-tuning duration, the weights of the CGRO and RFKL objectives, the parameters governing reward construction and probability-ratio clipping, and the feature temperature used by RFKL. We use a largely shared set of hyperparameters across the evaluated backbones and datasets, with selected adjustments according to dataset scale or backbone characteristics. We do not perform an exhaustive hyperparameter search for every dataset–backbone combination. The corresponding settings are reported in the tables, with identical values merged only within the same backbone family. Each merged entry applies to all dataset columns that it spans.

Unless otherwise specified, all remaining configurations, including the backbone architecture, data splits, data preprocessing and augmentation, optimizer, learning-rate scheduler, and evaluation protocol, are kept identical to those of the corresponding baseline. PointCRO therefore uses the ex-

Table 16: Comparison with conventional regularization methods under the parameter-efficient LoRA fine-tuning setting. The main segmentation backbone is frozen, and only the inserted LoRA parameters are trainable.

Method	mIoU ↑	OA ↑
Baseline	77.3	91.8
+ EntropyMin	77.1	91.7
+ Conf. Penalty	77.2	**91.8**
+ ProbL2	77.1	91.7
+ CE-Margin	77.1	91.7
+ ProtoCon	77.1	91.7
+ SupCon	77.2	**91.8**
+ CenterLoss	77.1	91.6
+ TripletMargin	77.1	91.7
+ PointCRO	**77.6**	**91.8**

Table 17: PointCRO-specific hyperparameter settings across different backbones and datasets. Identical settings are merged only within the same backbone family. All unlisted configurations are inherited from the corresponding baseline.

Hyperparameter	VOLT			OACNN		CamPoint	PointMC
	ScanNetV2	ScanNet++	ScanNet200	ScanNetV2	ScanNet++	S3DIS	S3DIS
Base reward b	20	50		20	50	20	20
CGRO weight λ_{cgro}	0.1			0.1		0.1	0.1
Margin threshold τ	0.30			0.30		0.30	0.30
Prediction-state-aware weight λ_{state}	0.50			0.50		0.50	0.50
TopWrong confidence threshold t_{fp}	0.60			0.60		0.60	0.60
Suppression strength weight λ_{neg}	0.50			0.50		0.50	0.50
RFKL weight λ_{rfkl}	1.0			1.0		1.0	1.0
Feature temperature τ_f	0.10			0.10		0.10	0.10

isting backbone and baseline training configuration, supplemented with its additional optimization objectives and the settings listed in the two tables. These tables document the default PointCRO configurations used for the evaluated indoor and outdoor benchmarks.

A.9 Computational Analysis of RFKL

RFKL transfers the point-level class-competition information modeled by CGRO from the prediction space to the representation space. Given the sampled point features $\mathbf{Z} \in \mathbb{R}^{N_f \times D}$, let $\widehat{\mathbf{Z}}$ denote their row-wise ℓ_2-normalized representations. RFKL constructs the temperature-scaled pairwise cosine similarity matrix as

$$\mathbf{S} = \frac{\widehat{\mathbf{Z}}\widehat{\mathbf{Z}}^{\top}}{\tau_f}, \tag{41}$$

where N_f denotes the number of point features uniformly sampled without replacement from the current mini-batch, D is the feature dimension, and τ_f is the feature temperature. Each entry of $\mathbf{S}$ therefore corresponds to the temperature-scaled cosine similarity between a pair of sampled point features. The feature relations are aligned with a reward-guided reference distribution that assigns larger weights to intra-class points with lower correction demand. Feature normalization requires $\mathcal{O}(N_f D)$ operations, while constructing the pairwise feature-relation matrix requires $\mathcal{O}(N_f^2 D)$ operations and $\mathcal{O}(N_f^2)$ memory. Therefore, the dominant additional time and memory complexities of RFKL are $\mathcal{O}(N_f^2 D)$ and $\mathcal{O}(N_f^2)$, respectively. The computational cost is bounded by the sampling budget N_f, which is set to 4096 or 8192 according to the dataset.

We further profile the baseline, the baseline equipped with CGRO, and the complete PointCRO model on ScanNetV2 under identical hardware and implementation settings. Table 19 reports training and inference measurements on ScanNetV2. The reported training time per iteration increases

Table 18: PointCRO-specific hyperparameter settings on outdoor point cloud datasets. Identical settings are merged only within the same backbone family. All unlisted configurations are inherited from the corresponding baseline.

Hyperparameter	VOLT			LitePT	
	Waymo	nuScenes	SemanticKITTI	Waymo	nuScenes
Base reward b	20	50		50	
CGRO weight λ_{cgro}	1.0			1.0	
Margin threshold τ	0.20			0.20	
Prediction-state-aware weight λ_{state}	0.50			0.50	
TopWrong confidence threshold t_{fp}	0.60			0.60	
Suppression strength weight λ_{neg}	1.0			0.50	
RFKL weight λ_{rfkl}	1.0			1.0	
Feature temperature τ_f	0.10			0.10	

Table 19: Training overhead, inference efficiency, and segmentation performance of CGRO and RFKL on ScanNetV2. Training-time overhead is calculated relative to the baseline. Neither CGRO nor RFKL modifies the inference architecture.

Method	Train Time/Iter (ms/iter) ↓	Train Overhead (%) ↓	Peak Memory (GB) ↓	Inference Throughput (samples/s) ↑	mIoU (%) ↑
Baseline	359	–	67.2	32.3	77.3
Baseline + CGRO	374	+4.18	68.1	31.7	77.9
PointCRO	460	+28.1	69.6	32.1	**78.6**

from 359 ms for the baseline to 374 ms with CGRO and 460 ms with the complete PointCRO formulation, corresponding to overheads of 4.18% and 28.1%, respectively. These measurements characterize per-iteration training costs under the evaluated configuration. Both CGRO and RFKL are used exclusively during training and preserve the inference architecture, introducing no additional inference-time parameters or theoretical FLOPs.

A.10 Feature-Support Analysis via $\Delta S_{\text{G.T.}}$ and ΔM_{feat}

Let $\mathcal{V}$ denote the set of valid evaluation points. For each point $i \in \mathcal{V}$, let $p_{i,c}^{(0)}$ be the probability assigned to class c by the baseline model. Following the notation in the main text, its baseline TopWrong class is

$$c_i^{-,(0)} = \underset{c \in \mathbb{C},\, c \neq y_i}{\operatorname{argmax}} \, p_{i,c}^{(0)}. \tag{42}$$

For each semantic class c, we select the $K = 32$ points with the largest baseline G.T.–TopWrong margins

$$m_j^{(0)} = p_{j,y_j}^{(0)} - p_{j,c_j^{-,(0)}}^{(0)} \tag{43}$$

as a fixed same-class reference set $\mathcal{R}_c$.The high-margin selection is intended to provide stable, prototypical intra-class anchors rather than to approximate the full class distribution. Accordingly, the resulting support metrics characterize feature alignment with prototypical class references. These reference identities and the baseline TopWrong labels are held fixed across all evaluated methods, preserving the reference-point membership and competing-class assignment for each query. When a query point belongs to its same-class reference set, it is excluded from the support computation to prevent trivial self-similarity from contributing to the estimated class support. The resulting self-excluded reference set is denoted by $\mathcal{R}_{y_i}^{(-i)}$. Let $\widehat{\boldsymbol{h}}_i^{(s)}$ denote the ℓ_2-normalized pre-head feature of point i at model state s, where $s = 0$ denotes the baseline and $s = (r, T)$ denotes the final checkpoint of method r. Both query and reference features are evaluated at the corresponding model state, so the support measurements capture changes in their feature relations while reference identities remain fixed. The G.T. and baseline TopWrong feature supports are computed as the average cosine

similarities between the query feature and the features in their corresponding reference sets:

$$S_{\mathrm{G.T.}}^{(s)}(i) = \frac{1}{|\mathcal{R}_{y_i}^{(-i)}|} \sum_{j \in \mathcal{R}_{y_i}^{(-i)}} \left(\widehat{\boldsymbol{h}}_i^{(s)}\right)^\top \widehat{\boldsymbol{h}}_j^{(s)}, \tag{44}$$

and

$$S_{\mathrm{TW}}^{(s)}(i) = \frac{1}{|\mathcal{R}_{c_i^{-,(0)}}|} \sum_{j \in \mathcal{R}_{c_i^{-,(0)}}} \left(\widehat{\boldsymbol{h}}_i^{(s)}\right)^\top \widehat{\boldsymbol{h}}_j^{(s)}. \tag{45}$$

The feature-support margin is

$$M_{\mathrm{feat}}^{(s)}(i) = S_{\mathrm{G.T.}}^{(s)}(i) - S_{\mathrm{TW}}^{(s)}(i). \tag{46}$$

For method r, the point-wise changes relative to the baseline are

$$\Delta S_{\mathrm{G.T.}}^{(r)}(i) = S_{\mathrm{G.T.}}^{(r,T)}(i) - S_{\mathrm{G.T.}}^{(0)}(i), \tag{47}$$

and

$$\Delta M_{\mathrm{feat}}^{(r)}(i) = M_{\mathrm{feat}}^{(r,T)}(i) - M_{\mathrm{feat}}^{(0)}(i). \tag{48}$$

We report their averages over the points that are misclassified by the baseline but correctly classified by the final checkpoint of method r:

$$\mathcal{C}_r = \left\{ i \in \mathcal{V} \mid \widehat{y}_i^{(0)} \neq y_i \wedge \widehat{y}_i^{(r,T)} = y_i \right\}. \tag{49}$$

Accordingly,

$$\Delta S_{\mathrm{G.T.}}^{(r)} = \frac{1}{|\mathcal{C}_r|} \sum_{i \in \mathcal{C}_r} \Delta S_{\mathrm{G.T.}}^{(r)}(i), \tag{50}$$

and

$$\Delta M_{\mathrm{feat}}^{(r)} = \frac{1}{|\mathcal{C}_r|} \sum_{i \in \mathcal{C}_r} \Delta M_{\mathrm{feat}}^{(r)}(i). \tag{51}$$

These metrics summarize feature-space changes accompanying prediction correction between baseline and final checkpoints. A positive $\Delta S_{\mathrm{G.T.}}$ indicates increased similarity to same-class references, while a positive ΔM_{feat} indicates improved relative support for the G.T. class over the baseline TopWrong class. Notably, $\Delta S_{\mathrm{G.T.}} > 0$ alone does not imply reduced incorrect-class similarity.

Reference identities and baseline TopWrong labels are fixed across methods, while feature vectors are evaluated at the corresponding model state. However, the corrected-point set $\mathcal{C}_r$ depends on the evaluated method, so differences in these averages may also reflect differences in subset composition. Accordingly, the statistics describe associations between prediction correction and feature changes within the respective evaluation subsets; they do not isolate the causal contribution of RFKL to correction or establish the persistence of corrected predictions throughout training.

By contrast, ΔM_{feat} jointly considers the support from the ground-truth class and the competing support from the baseline TopWrong class. It measures whether the feature representation becomes relatively more aligned with the ground-truth class than with its dominant incorrect competitor. This relative margin is more informative than $\Delta S_{\mathrm{G.T.}}$ alone, because an increase in same-class similarity does not necessarily imply that the competing wrong-class support has been reduced.

Evaluating these quantities on corrected points directly connects feature-space changes with actual prediction recovery. Moreover, fixing the reference identities and TopWrong labels according to the baseline provides a common reference frame across checkpoints, ensuring that the measured changes reflect representation evolution rather than changes in the evaluation targets.

A.11 Hyperparameter Sensitivity Analysis

We further investigate the sensitivity of PointCRO to its key hyperparameters, focusing on the weighting of its optimization objectives and the construction of class-wise rewards. First, we examine the loss weights of the two main components, λ_{CGRO} and λ_{RFKL}, which control the contributions of output-space class competition optimization and feature-space reward-guided regularization to the overall fine-tuning objective. Specifically, we vary λ_{CGRO} over $0.025, 0.05, 0.1, 0.2$ and

Table 20: Joint sensitivity analysis of λ_{CGRO} and λ_{RFKL} on ScanNetV2. Each entry reports the mIoU (%) obtained with the corresponding combination of CGRO and RFKL loss weights.

λ_{CGRO}	λ_{RFKL}			
	0.25	0.5	1.0	2.0
0.025	78.3	78.5	78.1	78.2
0.05	78.0	78.2	78.5	78.2
0.10	78.0	77.7	78.6	78.1
0.20	78.1	77.9	78.3	77.8

Table 21: Sensitivity analysis of the Base reward scale b in CGRO on ScanNetV2. All other hyperparameters are fixed to their default settings.

b	1	5	20	50	100
mIoU (%)	78.3	78.2	78.6	78.3	78.2

λ_{RFKL} over $0.25, 0.5, 1.0, 2.0$, while keeping all other training configurations unchanged. Evaluating all 4×4 combinations allows us to characterize their joint sensitivity and assess how different balances between the two auxiliary objectives affect segmentation performance under the evaluated fine-tuning configuration. The results are reported in Table 20.

We additionally analyze the sensitivity to the base G.T. reward scale b in CGRO, which scales the positive reward assigned to the ground-truth class before group-relative normalization. By adjusting the positive reward relative to the incorrect-class rewards, b influences the reward configuration from which class-wise advantages are derived. We vary b over $1, 5, 20, 50, 100$ while fixing the remaining hyperparameters to their default settings. This analysis examines how segmentation performance changes across the tested reward scales and whether the effectiveness of PointCRO depends strongly on a particular choice of b. The results are reported in Table 21.

A.12 Class-wise Analysis on ScanNetV2

To better understand how PointCRO affects different semantic classes, we provide a class-wise analysis on the ScanNet validation set in Tables 22, and 23. The results show that PointCRO improves the overall segmentation performance and brings gains to most semantic classes. This indicates that the proposed objective does not merely benefit a small subset of categories, but provides a broadly effective optimization signal for point-level semantic segmentation. A closer inspection shows that the improved categories are mainly object classes that are prone to local ambiguity or inter-class confusion, such as furniture-related and small indoor objects. These categories often share similar geometric structures or appear in spatially related contexts, making the separation between the ground-truth class and plausible incorrect classes more difficult. This behavior is consistent with the design of PointCRO, which explicitly models the local competition between the ground-truth class and the most competitive wrong class. By assigning prediction-state-aware optimization signals, PointCRO can better guide ambiguous points toward their correct semantic classes.

For the few categories with decreased IoU, Table 23 further compares the corresponding class accuracy changes. For several degraded categories, class accuracy is maintained or slightly improved while IoU decreases. This suggests that the performance drop is not necessarily caused by a reduced ability to recognize ground-truth points of these classes. Instead, it may be related to increased false-positive predictions from other categories, which affects precision and consequently reduces IoU. For categories where both IoU and class accuracy decrease, the degradation may involve both missed ground-truth points and increased confusion with neighboring or semantically related classes. Overall, the class-wise analysis suggests that PointCRO mainly improves categories with stronger prediction ambiguity, while causing only limited degradation on a small number of classes. This is aligned with the goal of prediction-state-aware class competition modeling

Table 22: Class-wise IoU comparison on the ScanNet validation set. The better result between the baseline and PointCRO is highlighted in bold.

Class	Baseline	PointCRO	ΔIoU	Class	Baseline	PointCRO	ΔIoU
Wall	87.37	**88.01**	+0.64	Picture	34.78	**38.90**	+4.12
Floor	**95.49**	95.44	-0.05	Counter	69.57	**72.31**	+2.74
Cabinet	74.64	**77.25**	+2.61	Desk	72.90	**75.82**	+2.92
Bed	**85.04**	84.09	-0.95	Curtain	82.81	**83.35**	+0.54
Chair	**92.40**	92.18	-0.22	Refrigerator	71.34	**75.35**	+4.01
Sofa	83.06	**83.81**	+0.75	Shower curtain	77.98	**78.50**	+0.52
Table	76.42	**79.61**	+3.19	Toilet	**95.57**	93.81	-1.76
Door	67.87	**68.24**	+0.37	Sink	71.70	**72.28**	+0.58
Window	71.34	**72.93**	+1.59	Bathtub	87.92	**89.32**	+1.40
Bookshelf	84.43	**84.82**	+0.39	Other furniture	62.83	**66.13**	+3.30

Table 23: Detailed analysis of categories with decreased IoU on the ScanNet validation set. ΔIoU and ΔAcc denote the changes from the baseline to PointCRO, measured in percentage points.

Class	IoU		ΔIoU	Class Acc.		ΔAcc
	Base	PointCRO		Base	PointCRO	
Floor	95.49	95.44	-0.05	98.24	98.30	+0.06
Bed	85.04	84.09	-0.95	90.86	90.54	-0.32
Chair	92.40	92.18	-0.22	95.20	95.52	+0.32
Toilet	95.57	93.81	-1.76	97.92	97.98	+0.06

A.13 Visualization Results on Multiple Datasets

Qualitative visualizations of the nuScenes, ScanNet++, and ScanNetV2 datasets are provided in Figures 2–7, with point colors indicating their corresponding semantic labels.

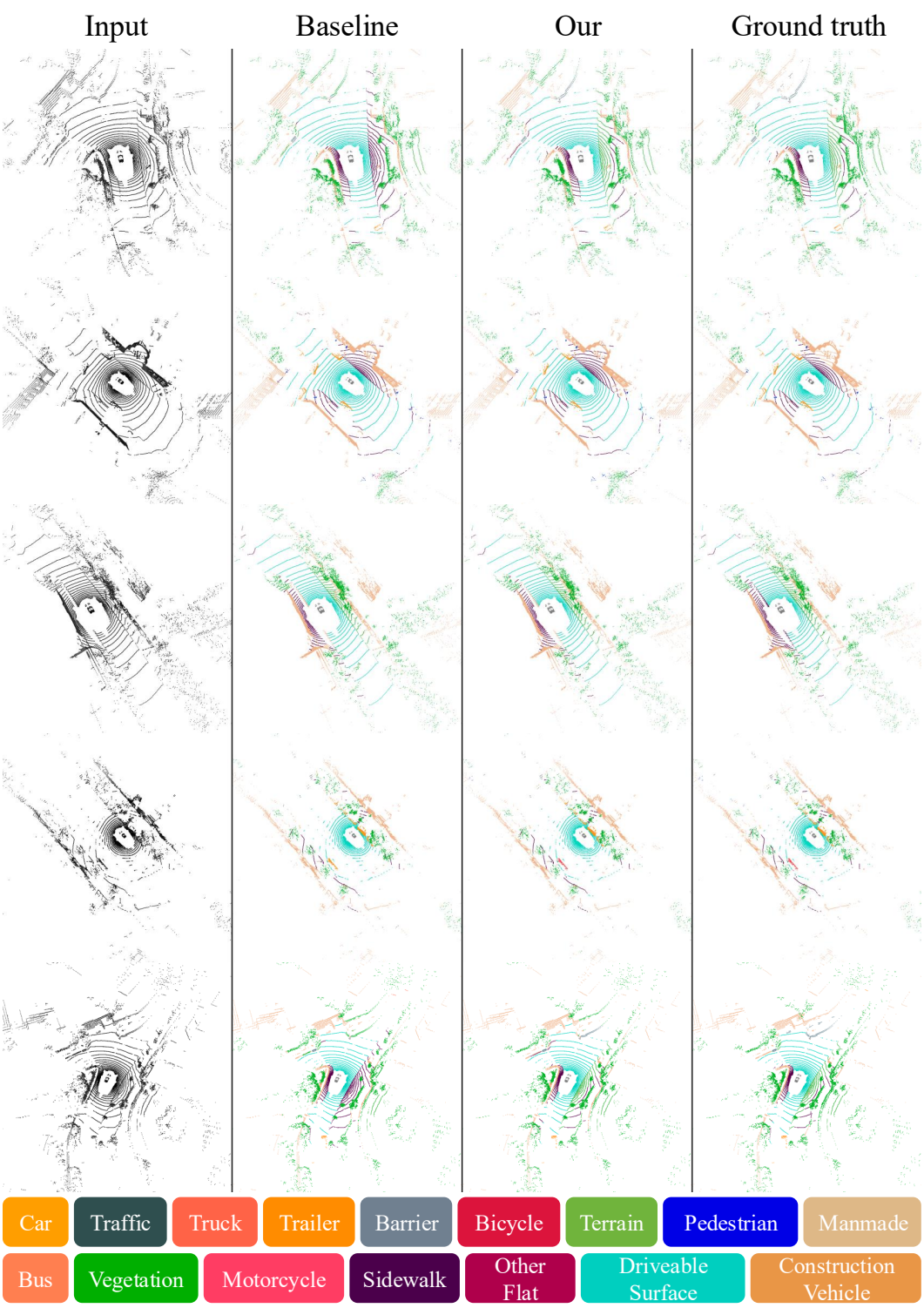


Figure 2: Qualitative comparison of segmentation results on the nuScenes dataset, showing input point clouds, ground truth, the baseline, and predicted results from baseline + PointCRO.

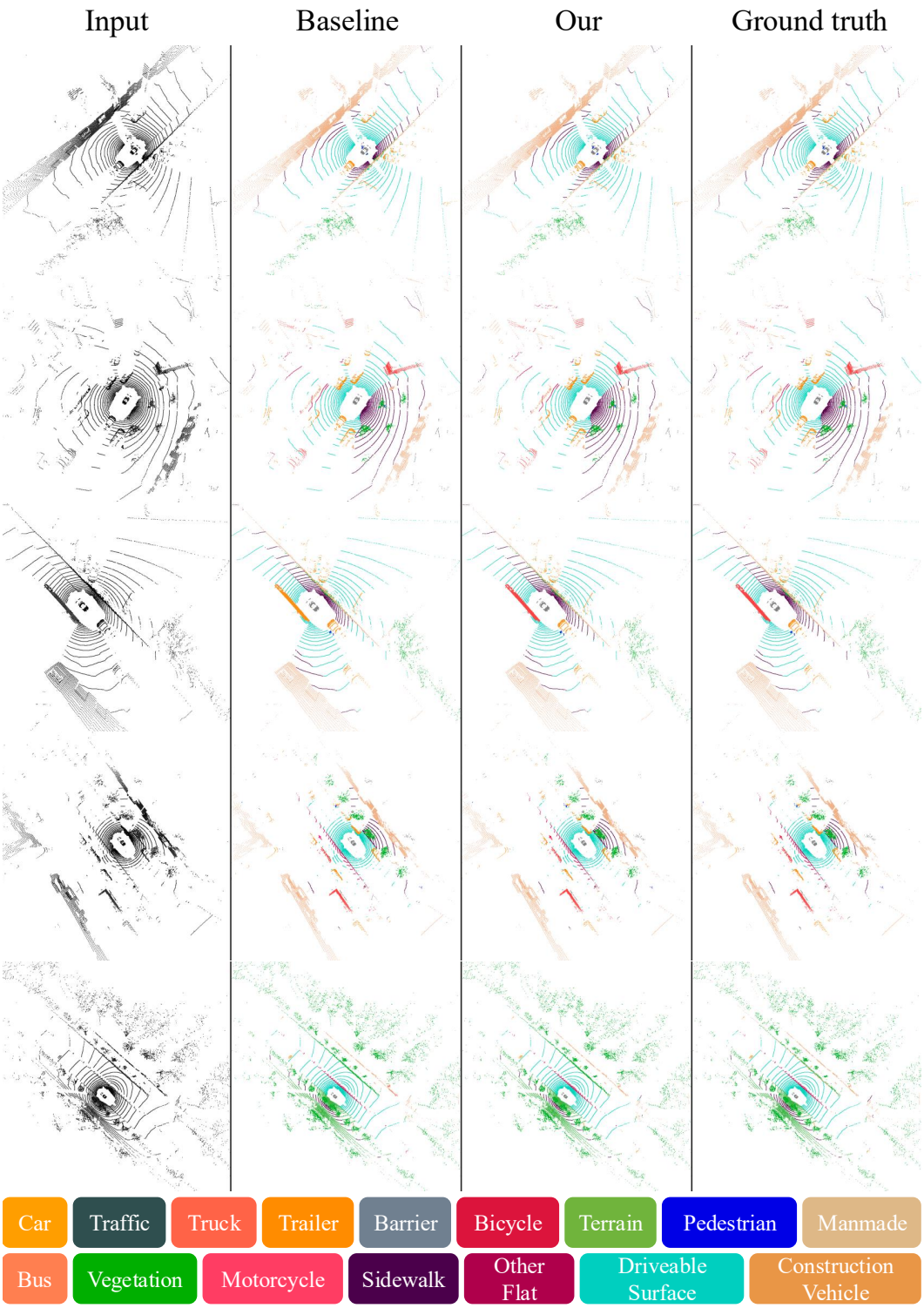


Figure 3: Qualitative comparison of segmentation results on the nuScenes dataset, showing input point clouds, ground truth, the baseline, and predicted results from baseline + PointCRO.

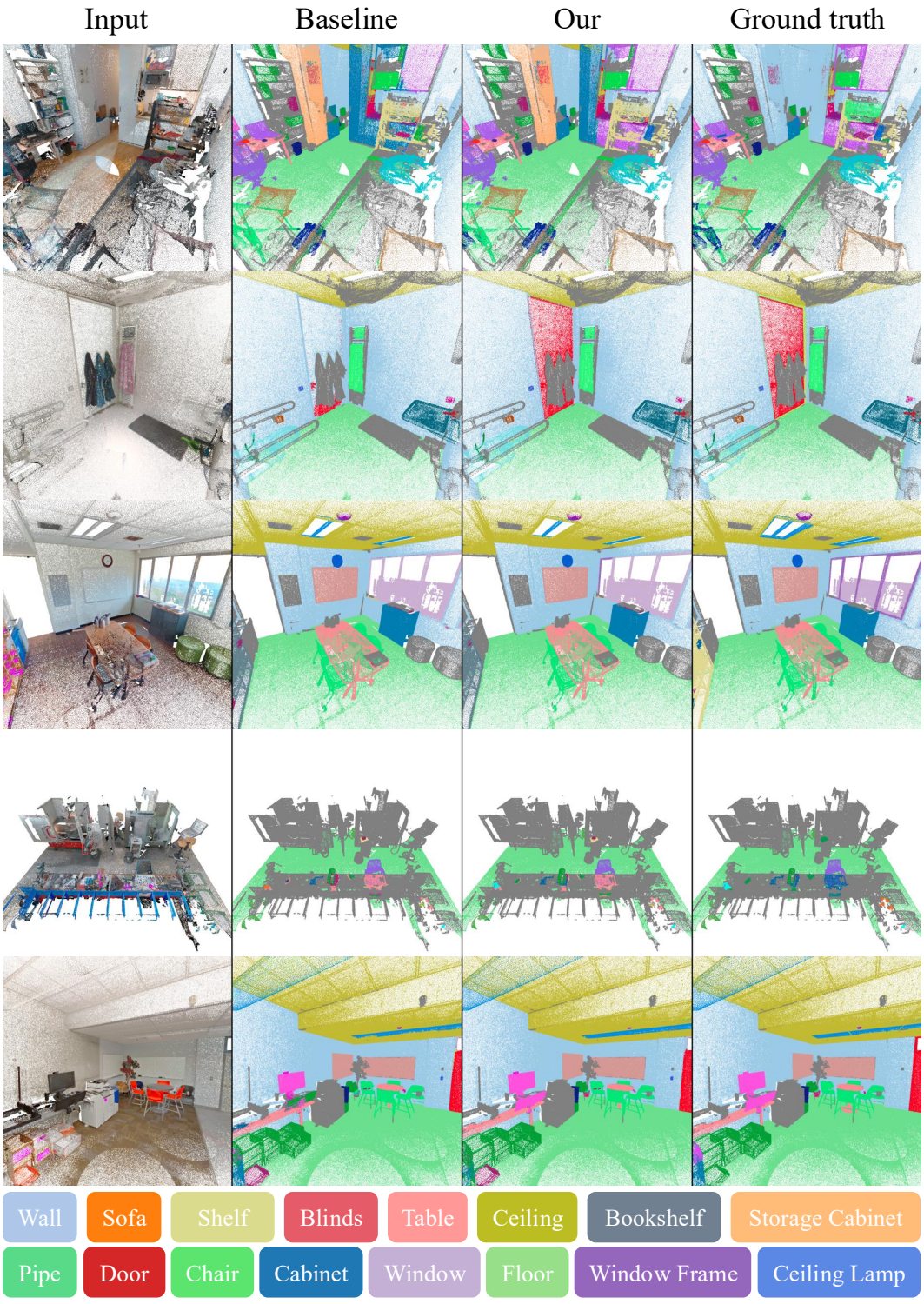


Figure 4: Qualitative comparison of segmentation results on the ScanNet++ dataset. Our method effectively preserves fine-grained boundary details and semantic consistency, significantly outperforming the baseline in challenging scenarios.



Figure 5: Qualitative comparison of segmentation results on the ScanNet++ dataset. Our method effectively preserves fine-grained boundary details and semantic consistency, significantly outperforming the baseline in challenging scenarios.



Figure 6: Qualitative segmentation results on ScanNetV2. Compared with the baseline, Our method yields more accurate predictions in challenging regions, preserving finer boundary structures and improving semantic consistency.

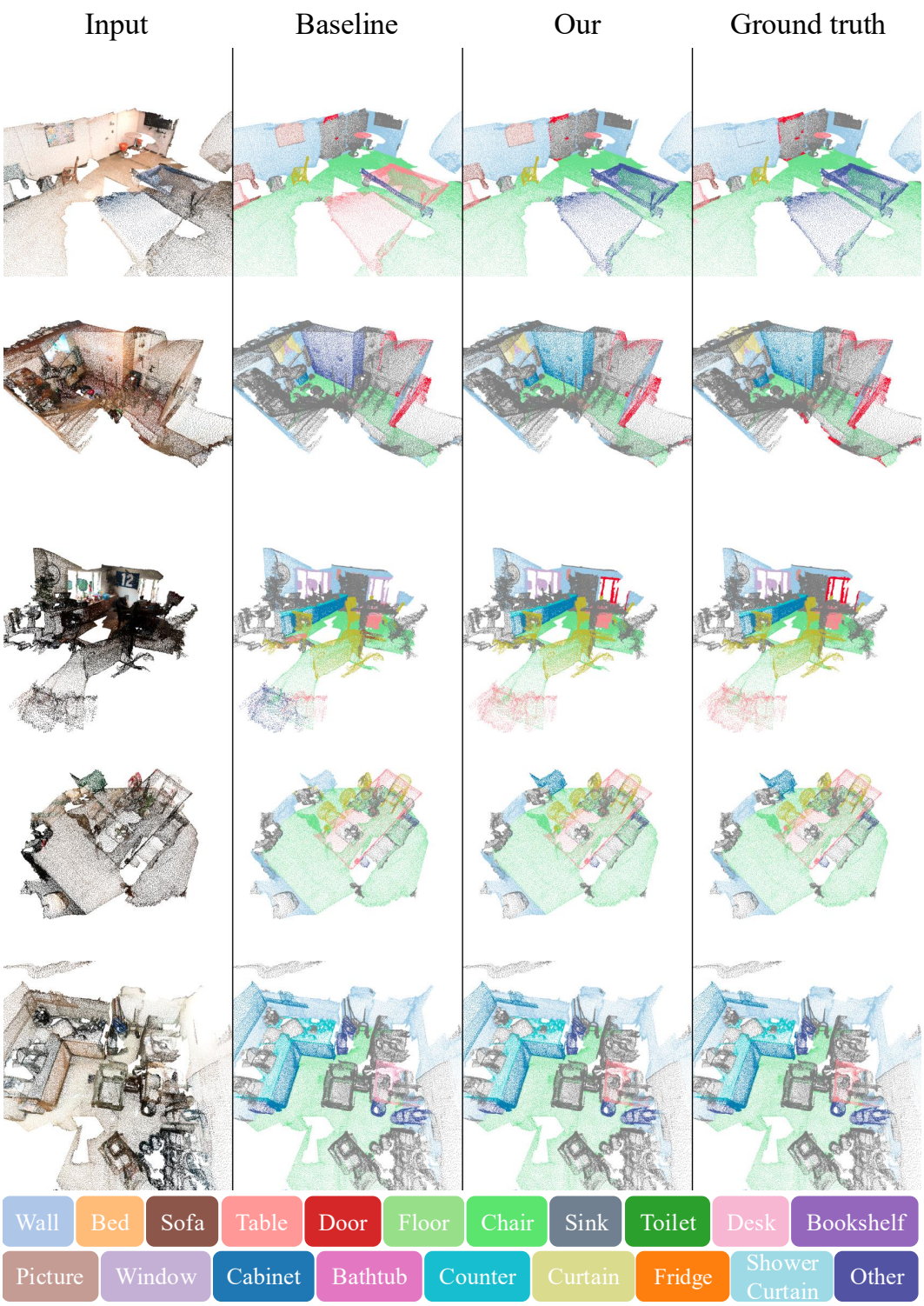


Figure 7: Qualitative segmentation results on ScanNetV2. Compared with the baseline, Our method yields more accurate predictions in challenging regions, preserving finer boundary structures and improving semantic consistency.