Supplementary Materials

Reliability-aware Conflict Calibration for Generalized Few-shot Point Cloud Segmentation

He Wang Guofeng Zhang*

State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, Zhejiang, China

*Corresponding author: Guofeng Zhang, zhangguofeng@cad.zju.edu.cn

Journal: *The Visual Computer*, CGI'2026 Special Issue

A Details of Prototype Reliability Estimation for Base–Novel Scoring

A.1 Local Prototype Construction

For each novel class c, we construct a global prototype and M local prototypes from the support foreground features. The global prototype is obtained by average pooling over all support foreground features of class c:

$$\mathbf{p}_{\mathrm{g}}^{c}=\frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{f}_i^{c},\quad \hat{\mathbf{p}}_{\mathrm{g}}^{c}=\frac{\mathbf{p}_{\mathrm{g}}^{c}}{\left\|\mathbf{p}_{\mathrm{g}}^{c}\right\|_2}. \tag{A.1}$$

To preserve finer part-level geometry, we further extract foreground patch tokens from local support regions. Let $\mathcal{T}^c=\{\mathbf{t}_i^c\}_{i=1}^{|\mathcal{T}^c|}$ denote the patch token set of class c. We use M learnable slots to aggregate these tokens into M local prototypes. The assignment weights a_{im}^c are produced by a learnable slot-to-token soft assignment module, where a_{im}^c denotes the soft assignment weight from token $\mathbf{t}_i^c$ to slot m. The m-th local prototype is computed as

$$\mathbf{p}_{\mathrm{l}}^{c,m}=\frac{\sum_{\mathbf{t}_i^c\in\mathcal{T}^c}a_{im}^{c}\,\mathbf{t}_i^{c}}{\sum_{\mathbf{t}_i^c\in\mathcal{T}^c}a_{im}^{c}},\quad \hat{\mathbf{p}}_{\mathrm{l}}^{c,m}=\frac{\mathbf{p}_{\mathrm{l}}^{c,m}}{\|\mathbf{p}_{\mathrm{l}}^{c,m}\|_2}. \tag{A.2}$$

In this way, the support branch provides one global prototype as a class-level semantic anchor and M local prototypes as reusable part-level representations. The number of slots M controls the granularity of local structural representation.

A.2 Global Prototype Reliability

For the global prototype, reliability is determined by cross-shot consistency and base ambiguity. Let $\bar{\mathbf{f}}_k^c$ denote the average foreground feature of class c in the k-th support shot, and let $\mathbf{w}_{\mathrm{b}}$ denote the frozen classifier weight of base class b. We define

$$u_{\text{cls-shot}}^{c}=\frac{1}{K(K-1)}\sum_{1\le i<j\le K}\cos\!\left(\bar{\mathbf{f}}_i^{c},\bar{\mathbf{f}}_j^{c}\right), \tag{A.3}$$

$$u_{\text{base}}^{c}=\max_{b\in\mathcal{C}_{\mathrm{b}}}\cos\!\left(\hat{\mathbf{p}}_{\mathrm{g}}^{c},\mathbf{w}_{\mathrm{b}}\right). \tag{A.4}$$

The global reliability is then computed as

$$r_{\mathrm{g}}^{c}=\sigma(\alpha_1 u_{\text{cls-shot}}^{c}-\alpha_2 u_{\text{base}}^{c}), \tag{A.5}$$

where $\sigma(\cdot)$ is the sigmoid function, α_1 and α_2 are learnable parameters. A reliable global prototype should remain stable across support shots while staying sufficiently separated from base semantics.

A.3 Local Prototype Reliability

For each local prototype $\mathbf{p}_{\mathrm{l}}^{c,m}$, reliability is determined by three factors: local compactness, cross-shot coverage, and local base ambiguity.

Local compactness. We measure whether the tokens assigned to slot m form a coherent local mode by the weighted within-slot dispersion:

$$u_{\text{comp}}^{\mathrm{l}}(c,m)=\frac{\sum_{\mathbf{t}_i^c\in\mathcal{T}^c}a_{im}^{c}\,\|\mathbf{t}_i^{c}-\mathbf{p}_{\mathrm{l}}^{c,m}\|_2^2}{\sum_{\mathbf{t}_i^c\in\mathcal{T}^c}\left(a_{im}^{c}\right)^2}. \tag{A.6}$$

A smaller value indicates that the assigned tokens are more compact and therefore more reliable.

Cross-shot coverage. Let $\mathcal{T}_k^c$ denote the patch token set from the k-th support shot. We compute the total assignment weight received by slot m from shot k:

$$A_{k,m}^{c}=\sum_{\mathbf{t}_i^c\in\mathcal{T}_k^c}a_{im}^{c}. \tag{A.7}$$

We then define cross-shot coverage as

$$u_{\text{cov}}^{\mathrm{l}}(c,m)=\frac{1}{K}\sum_{k=1}^{K}\mathbb{I}\!\left(A_{k,m}^{c}>\epsilon\right), \tag{A.8}$$

where ϵ is a small threshold. The cross-shot coverage term measures whether the same local pattern is repeatedly supported across different shots.

Local base ambiguity. We penalize slot-level patterns that are overly close to base classes:

$$u^{\mathrm{l}}_{\mathrm{base}}(c,m) = \max_{b\in\mathcal{C}_{\mathrm{b}}} \cos(\hat{\mathbf{p}}^{c,m}_{\mathrm{l}}, \mathbf{w}_{\mathrm{b}})\,. \tag{A.9}$$

Combining the above three factors, the local reliability is defined as

$$\begin{aligned} r^{c,m}_{\mathrm{l}} = \sigma\Big(&- \beta_1 u^{\mathrm{l}}_{\mathrm{comp}}(c,m) + \beta_2 u^{\mathrm{l}}_{\mathrm{cov}}(c,m) \\ &- \beta_3 u^{\mathrm{l}}_{\mathrm{base}}(c,m)\Big), \end{aligned} \tag{A.10}$$

where β_1, β_2, and β_3 are learnable parameters. In this way, only compact, reusable, and base-discriminative local patterns receive high reliability.

A.4 Unified Base–Novel Scoring

Unified base–novel scoring makes support-derived novel responses more comparable to base-class responses and provides conflict statistics for the subsequent calibration stage. Given the reliability-aware global and local prototypes, we compute for each query point j and each novel class c a global score and a local score:

$$s^{c}_{\mathrm{g}}(j) = r^{c}_{\mathrm{g}} \cdot \cos\big(\mathbf{f}^{\mathrm{q}}_{j}, \hat{\mathbf{p}}^{c}_{\mathrm{g}}\big)\,, \tag{A.11}$$

$$s^{c}_{\mathrm{l}}(j) = \max_{m=1,\dots,M} \big(r^{c,m}_{\mathrm{l}} \cdot \cos\big(\mathbf{f}^{\mathrm{q}}_{j}, \hat{\mathbf{p}}^{c,m}_{\mathrm{l}}\big)\big)\,. \tag{A.12}$$

The final novel score is

$$z^{c}_{\mathrm{n}}(j) = \lambda_{\mathrm{g}} s^{c}_{\mathrm{g}}(j) + \lambda_{\mathrm{l}} s^{c}_{\mathrm{l}}(j), \quad \lambda_{\mathrm{g}} + \lambda_{\mathrm{l}} = 1. \tag{A.13}$$

We then define the overall novel score and label as

$$s_{\mathrm{n}}(j) = \max_{c\in\mathcal{C}_{\mathrm{n}}} z^{c}_{\mathrm{n}}(j), \quad \hat{c}_{\mathrm{n}}(j) = \arg\max_{c\in\mathcal{C}_{\mathrm{n}}} z^{c}_{\mathrm{n}}(j). \tag{A.14}$$

Base scores are obtained from the frozen base classifier, yielding the base score $s_{\mathrm{b}}(j)$ and base label $\hat{c}_{\mathrm{b}}(j)$. By comparing frozen base-class responses and reliability-weighted novel-class responses in a unified score space, we identify the most competitive base–novel pair for each query point:

$$\hat{p}(j) = \big(\hat{c}_{\mathrm{b}}(j), \hat{c}_{\mathrm{n}}(j)\big). \tag{A.15}$$

These quantities are used in the subsequent conflict localization stage.

B Details of Conflict Localization and Selective Calibration

B.1 Point-level Conflict Identification

Based on the unified score vector $\mathbf{z}(j)$, the candidate base–novel conflict pair $\hat{p}(j)$, and the base–novel margin $m(j)$, we further define a point-level conflict score to quantify local ambiguity. Specifically, we first convert the unified scores into a posterior distribution over the joint label space:

$$p_c(j) = \frac{\exp(z_c(j))}{\sum_{c'\in\mathcal{C}_{\mathrm{b}}\cup\mathcal{C}_{\mathrm{n}}} \exp(z_{c'}(j))}, \quad c \in \mathcal{C}_{\mathrm{b}} \cup \mathcal{C}_{\mathrm{n}}. \tag{B.16}$$

The prediction entropy is then given by

$$H(j) = -\sum_{c\in\mathcal{C}_{\mathrm{b}}\cup\mathcal{C}_{\mathrm{n}}} p_c(j) \log p_c(j). \tag{B.17}$$

Based on the margin and entropy, we define the point-level conflict score as

$$\phi(j) = \exp\left(-\frac{|m(j)|}{\tau_{\mathrm{m}}}\right) \cdot \frac{H(j)}{\log(|\mathcal{C}_{\mathrm{b}}| + |\mathcal{C}_{\mathrm{n}}|)}. \tag{B.18}$$

The two factors in $\phi(j)$ characterize complementary aspects of ambiguity. The margin term reflects whether the base and novel scores are close, implying strong competition between the two classes, while the entropy term measures the uncertainty of the prediction when all base and novel classes are jointly considered. Therefore, a high conflict score indicates that the point lies in a highly ambiguous region with strong base–novel competition and high uncertainty in the joint label space. We collect points with high conflict scores into the conflict point set Ω_{conf}. These points are used only to identify candidate conflict regions; the final calibration unit is defined at the region level.

B.2 Conflict Region Construction

After obtaining Ω_{conf}, we construct a neighborhood graph on the query point cloud. Let $\mathcal{N}_k(j)$ denote the k-nearest neighbors of point j. An edge is created between two points i and j only if

(i) both points belong to Ω_{conf}, and

(ii) they share the same candidate pair, i.e., $\hat{p}(i) = \hat{p}(j)$.

This yields the filtered graph

$$\mathcal{G}_{\mathrm{conf}} = (\Omega_{\mathrm{conf}}, \mathcal{E}_{\mathrm{conf}})\,, \tag{B.19}$$

$$\begin{aligned} \mathcal{E}_{\mathrm{conf}} = \{(i,j) \mid\ & i \in \mathcal{N}_k(j),\ i,j \in \Omega_{\mathrm{conf}}, \\ & \hat{p}(i) = \hat{p}(j)\}. \end{aligned} \tag{B.20}$$

The connected components of this graph are treated as candidate conflict regions. Conflict points are used only to initialize region construction. In point clouds, ambiguous points are often sparse, fragmented, and locally unstable due to missing observations, thin structures, or noisy features. Aggregating neighboring conflict points into regions therefore provides a more stable and semantically meaningful unit for calibration. For each candidate region R_t, we determine its dominant pair through conflict-score-weighted voting:

$$p_t = \arg\max_{(c_\mathrm{b}, c_\mathrm{n})} \sum_{j \in R_t} \mathbb{I}[\hat{p}(j) = (c_\mathrm{b}, c_\mathrm{n})]\, \phi(j), \tag{B.21}$$

where $p_t = (c_\mathrm{b}^t, c_\mathrm{n}^t)$ denotes the dominant conflict base–novel class pair of R_t.

The pair consistency score γ_t, as defined in the main text, measures how strongly the points in R_t support this dominant pair. We retain only regions that satisfy both a minimum size requirement and a high pair-consistency score, and denote the retained region set by $\mathcal{R}$. The minimum size threshold n_min suppresses isolated noisy points and accidental high-entropy responses, while the pair-consistency threshold τ_pair ensures that each retained region is dominated by a single base–novel pair rather than mixed local ambiguities. Since this stage serves as local structural verification rather than category rediscovery, we set τ_pair conservatively and keep n_min small to moderate, so that fragmented but semantically consistent conflict regions can still be preserved. In practice, we use $\tau_\mathrm{pair} = 0.75$ and $n_\mathrm{min} = 12$, which provide a good balance between suppressing noisy conflict proposals and preserving localized true conflicts. We use $k = 16$ for k-nearest-neighbor-based conflict region construction. This value provides a balanced local neighborhood for region aggregation: a smaller k tends to fragment coherent conflict regions into isolated components, while a larger k may over-connect nearby structures and expand the calibrated regions. Thus, $k = 16$ offers sufficient local connectivity without excessively merging different local structures. The subsequent calibration step is applied only to these retained regions.

B.3 Region-level Selective Calibration

For each retained region $R_t \in \mathcal{R}$, we construct a compact region representation from the query features in that region. Specifically, we aggregate the point features by average pooling and max pooling, and concatenate the two pooled descriptors before mapping them with a lightweight MLP. This design captures both the overall regional context and the strongest local responses, while being more stable than using individual conflict points alone. Since conflict points in point clouds are often sparse and locally noisy, region-level aggregation provides a more reliable unit for calibration.

The calibration head predicts a single scalar offset Δ_t for the dominant pair $p_t = (\hat{c}_\mathrm{b}^t, \hat{c}_\mathrm{n}^t)$ of region R_t. We adopt this dominant-pair formulation because each retained region has already been filtered to preserve a single dominant base–novel confusion pattern. Under this setting, directly correcting the dominant pair is sufficient to adjust the relative score margin between the dominant base and novel classes within the region. Compared with a full multi-class update, this pair-wise design makes the correction more localized, controlled, and interpretable.

The predicted offset changes only the relative score margin between the dominant novel and base classes in R_t, without altering the scores of the other classes. Therefore, calibration acts as a local margin correction rather than a global reshaping of the score space. To supervise this update, we assign each retained region a binary sign g_t, indicating whether the local boundary should move toward the novel side or remain on the base side according to the majority label in the region. Under this formulation, $\mathcal{L}_\mathrm{sep}$ determines the direction of the boundary shift, while $\mathcal{L}_\mathrm{reg}$ constrains its magnitude. Together, they encourage local but moderate correction of ambiguous base–novel boundaries.

B.4 Base-preserving Test-time Transductive Adaptation

We next describe the practical setup of test-time transductive adaptation (TTA), focusing on how stable base anchors are selected and how the lightweight episode-specific refinement is performed. A query point is included in the stable base-anchor set only if it satisfies three conditions: it is initially predicted as a base class, it lies outside the detected conflict set, and it has high confidence within the base-class subspace. Together, these conditions identify base predictions that are both reliable and less susceptible to local base–novel ambiguity. The selected anchors therefore serve as stable reference points during test-time refinement.

During TTA, we update only the calibration-related parameters while keeping the backbone and base classifier frozen, so the refinement is confined to the region-level pair-wise calibration applied in the retained conflict regions. Accordingly, the consistency objective is imposed only on the stable base anchors, encouraging the refined model to preserve confident non-conflict base predictions while adapting the calibration module to the current episode.

In practice, the adaptation design is intentionally lightweight. For each episode, we perform only a small fixed number of refinement steps (5 by default), and update only the calibration-related parameters. To avoid unstable refinement in episodes with too few reliable base anchors, TTA is applied only when the stable base-anchor set is sufficiently large. Otherwise, the adaptation step is skipped, and the initial calibrated prediction is used directly. This design keeps the refinement stable and efficient, while preserving the base-side reliability established by the frozen classifier.

C Details of Optional Textual Prototype Augmentation

C.1 Prompt Construction

For each novel class, we first convert the dataset label into a canonical class name. This normalization is performed once for each dataset-specific label set using simple rules: lowercase, removing separators or suffixes, and converting to singular form when appropriate. For example, labels such as "trash_bin" and "office_chair" are converted to "trash bin" and "office chair", respectively. Given the canonical class name c, we construct a small set of fixed prompt templates shared across all datasets:

a photo of a {*class*}
an indoor {*class*}
a 3D scan of a {*class*}
a point cloud object of a {*class*}

We intentionally restrict the prompt set to these fixed templates. We do not use manually written long descriptions, LLM-generated attributes, scene-level text, or dataset-specific prompt engineering in experiments. This keeps the textual branch lightweight and ensures that the gain mainly comes from our conflict-aware use of semantic priors rather than aggressive prompt tuning.

For the k-th prompt $T_k(c)$, we obtain its adapted textual feature by encoding the prompt with the frozen text encoder and then mapping it into the point-feature space with the adapter, followed by normalization. Let K denote the number of prompt templates. The final textual prototype is defined as the normalized average of the K adapted prompt features:

$$\mathbf{p}_{\mathrm{t}}^{c} = \mathrm{Norm}\left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{p}_{\mathrm{t},k}^{c}\right). \tag{C.22}$$

C.2 Text-to-point Adapter

The text-to-point adapter is a lightweight residual MLP. Given a text embedding $\mathbf{e}$, we apply layer normalization and a two-layer MLP to produce a residual update, add it to a linear projection of $\mathbf{e}$, and normalize the result to obtain the adapted feature $A(\mathbf{e})$.

This design is adopted for three reasons. First, since the textual input is defined at the class level rather than the token level, a heavy cross-attention module is unnecessary. Second, the residual connection preserves the original semantic structure of the text embedding while learning only the adjustment needed for alignment with the point-feature space. Third, the lightweight formulation introduces minimal parameter overhead and allows the branch to serve as a semantic alignment module rather than an additional multimodal backbone.

C.3 Adapter Training

The adapter is trained only during the base training stage, where base classes provide stable supervision for class-level semantic alignment. The adapter maps text embeddings into the point-feature space used by the segmentation model.

To this end, we optimize the adapter with three complementary terms. Let $\mathbf{p}_{\mathrm{g}}^{b}$ denote the corresponding support-derived global prototype, and let $\mathbf{w}_{\mathrm{b}}$ denote the base-class classifier weight of class b, obtained during base training. These three terms play different but complementary roles: $\mathcal{L}_{\mathrm{cls}}$ aligns text with the classifier space, $\mathcal{L}_{\mathrm{proto}}$ aligns text with support-derived visual information, and $\mathcal{L}_{\mathrm{nce}}$ improves class discrimination. During few-shot episodic training and test-time transductive adaptation, the text encoder and adapter are frozen, and these alignment losses are no longer used, so that the textual branch serves as a stable semantic prior rather than adapting to sparse episodes. The three terms are defined as

$$\mathcal{L}_{\mathrm{cls}} = \frac{1}{|\mathcal{C}_{\mathrm{b}}|}\sum_{b\in\mathcal{C}_{\mathrm{b}}}\left(1-\cos\left(\mathbf{p}_{\mathrm{t}}^{b},\mathbf{w}_{\mathrm{b}}\right)\right), \tag{C.23}$$

$$\mathcal{L}_{\mathrm{proto}} = \frac{1}{|\mathcal{C}_{\mathrm{b}}|}\sum_{b\in\mathcal{C}_{\mathrm{b}}}\left(1-\cos\left(\mathbf{p}_{\mathrm{t}}^{b},\mathbf{p}_{\mathrm{g}}^{b}\right)\right), \tag{C.24}$$

$$\mathcal{L}_{\mathrm{nce}} = -\frac{1}{|\mathcal{C}_{\mathrm{b}}|}\sum_{b\in\mathcal{C}_{\mathrm{b}}}\left[\frac{\cos\left(\mathbf{p}_{\mathrm{t}}^{b},\mathbf{w}_{\mathrm{b}}\right)}{\tau} - \log\sum_{b'\in\mathcal{C}_{\mathrm{b}}}\exp\left(\frac{\cos\left(\mathbf{p}_{\mathrm{t}}^{b},\mathbf{w}_{\mathrm{b}'}\right)}{\tau}\right)\right]. \tag{C.25}$$

The overall alignment objective is

$$\mathcal{L}_{\mathrm{align}} = \mathcal{L}_{\mathrm{cls}} + \lambda_1\mathcal{L}_{\mathrm{proto}} + \lambda_2\mathcal{L}_{\mathrm{nce}}. \tag{C.26}$$

C.4 Reliability Terms

Textual priors can vary in reliability across episodes: a textual prototype may be unstable across prompts, inconsistent with the support representations, or too close to a base category. We therefore estimate textual reliability using three interpretable terms: prompt consistency, support agreement, and base ambiguity.

The prompt consistency term measures whether different prompts of the same class produce stable adapted textual features. Larger variation across prompt instantiations indicates lower prompt-level stability and lower textual reliability. It is defined as

$$u_{\mathrm{pc}}^{c} = \frac{2}{K(K-1)} \sum_{1 \le k < l \le K} \frac{1 + \cos\left(\mathbf{p}_{\mathrm{t},k}^{c},\, \mathbf{p}_{\mathrm{t},l}^{c}\right)}{2}. \tag{C.27}$$

The support agreement term measures how well the textual prototype agrees with the support-derived local and global prototypes. A reliable textual prior should remain compatible with the visual evidence available in the current episode. We define the support agreement term as

$$\begin{aligned} u_{\mathrm{sa}}^{c} = {} & \alpha r_{\mathrm{g}}^{c} \cdot \frac{1 + \cos\left(\mathbf{p}_{\mathrm{t}}^{c},\, \mathbf{p}_{\mathrm{g}}^{c}\right)}{2} \\ & + \frac{(1-\alpha) \sum_{m=1}^{M} r_{\mathrm{l}}^{c,m} \left[1 + \cos(\mathbf{p}_{\mathrm{t}}^{c}, \mathbf{p}_{\mathrm{l}}^{c,m})\right]}{2\left(\sum_{m=1}^{M} r_{\mathrm{l}}^{c,m} + \varepsilon\right)}. \end{aligned} \tag{C.28}$$

The base ambiguity term measures whether the textual prototype is overly close to some base classifier weight:

$$u_{\mathrm{ba}}^{c} = \max_{b \in \mathcal{C}_{\mathrm{b}}} \frac{1 + \cos(\mathbf{p}_{\mathrm{t}}^{c},\, \mathbf{w}_{\mathrm{b}})}{2}. \tag{C.29}$$

The final textual reliability is defined as

$$r_{\mathrm{t}}^{c} = \sigma\left(\gamma_1 u_{\mathrm{pc}}^{c} + \gamma_2 u_{\mathrm{sa}}^{c} - \gamma_3 u_{\mathrm{ba}}^{c}\right). \tag{C.30}$$

C.5 Usage in Conflict-region Calibration

For each retained conflict region R_t with dominant pair $(\hat{c}_{\mathrm{b}}^{t}, \hat{c}_{\mathrm{n}}^{t})$, we extract the region representation $\mathbf{h}_t$, the textual prototype of the dominant novel class $\mathbf{p}_{\mathrm{t}}^{\hat{c}_{\mathrm{n}}^{t}}$, its reliability $r_{\mathrm{t}}^{\hat{c}_{\mathrm{n}}^{t}}$, and a region-level text prior score. We define the region-level text prior score as the average text-to-region compatibility:

$$\bar{s}_{t}^{\mathrm{text}} = \frac{1}{|R_t|} \sum_{j \in R_t} \cos\left(\mathbf{f}_{j}^{\mathrm{q}}, \mathbf{p}_{\mathrm{t}}^{\hat{c}_{\mathrm{n}}^{t}}\right). \tag{C.31}$$

To control the influence of textual priors, we introduce a reliability-aware gate η_t. The final text-aware calibration input is then formed as

$$\Delta_t = \psi\left(\mathbf{h}_t \parallel \eta_t \mathbf{p}_{\mathrm{t}}^{\hat{c}_{\mathrm{n}}^{t}} \parallel \eta_t \bar{s}_{t}^{\mathrm{text}}\right), \tag{C.32}$$

where $\psi(\cdot)$ denotes the pair-wise calibration head, and its input is the concatenation of the region feature, the gated textual prototype, and the gated text prior score. The resulting offset is applied only to the dominant base–novel pair within region R_t, leaving stable non-conflict areas unchanged.

In this way, textual information is introduced only when it is both locally relevant and sufficiently reliable, preserving the support-driven nature of the framework while avoiding indiscriminate fusion of textual cues.

D Experimental Settings

D.1 Datasets and Evaluation Metrics

We evaluate the proposed method on four indoor benchmarks: S3DIS, ScanNet, ScanNet++, and ScanNet200. For all datasets, we follow the same base/novel class splits as GFS-VL, so that the comparisons are directly aligned with prior generalized few-shot settings. Each benchmark is evaluated under both 1-shot and 5-shot settings. During testing, each episode consists of a few-shot support set for the novel classes and a query point cloud to be segmented in the unified label space containing both base and novel classes.

We report four standard metrics for generalized few-shot segmentation: **mIoU-B**, **mIoU-N**, **mIoU-A**, and **HM**. Here, mIoU-B and mIoU-N evaluate segmentation performance on base and novel classes, respectively; mIoU-A denotes the average performance over the unified label space, reflecting the overall generalized segmentation quality; and HM (harmonic mean) between base and novel performance more directly reflects the quality of joint discrimination under base–novel coexistence. Since generalized few-shot segmentation requires a balanced treatment of base and novel classes rather than novel-only recognition, we emphasize mIoU-A and HM in the main paper.

D.2 Implementation Details

We use PTV3 as the backbone to extract support and query features, and follow the same base/novel class splits as GFS-VL on all benchmarks for fair comparison. The support-only version of our method is denoted by Ours-S, while Ours-ST further incorporates the optional textual prototype augmentation module. A base

classifier learned during base training is retained and kept frozen during generalized few-shot inference, providing stable base-class responses in the unified base–novel score space.

Unless otherwise specified, the support-driven core framework uses $M = 4$ local slots, a stable base-anchor threshold $\tau_{\mathrm{b}} = 0.9$, a separation loss weight $\lambda_{\mathrm{sep}} = 0.3$, and a regularization weight $\lambda_{\mathrm{reg}} = 0.05$. We set $\tau_{\mathrm{pair}} = 0.75$ and $n_{\min} = 12$. All experiments are conducted on two NVIDIA L40S GPUs using the AdamW optimizer. We use an initial learning rate of 0.006, while the backbone blocks are optimized with a 10× smaller learning rate of 0.0006.

Our training pipeline consists of three stages: base training, generalized episodic training, and test-time transductive adaptation. During base training, the base classifier and the backbone are trained on base classes. For Ours-ST, the text-to-point adapter is also trained only at this stage, while the CLIP text encoder remains frozen. During generalized episodic training, the model learns prototype reliability estimation and selective calibration in a unified base–novel scoring space, with the base classifier kept frozen. At test time, we update only the calibration-related parameters under the base-preserving consistency constraint, while keeping both the backbone and the frozen base classifier unchanged. By default, test-time transductive adaptation is performed for 5 steps.

For the learnable scalars in reliability estimation, including α_1, α_2, β_1, β_2, and β_3, and the textual reliability coefficients γ_1, γ_2, and γ_3, we initialize their effective values to 1 and constrain them to be non-negative during training. Specifically, after each optimizer update, these scalars are projected to the non-negative range by $x \leftarrow \max(x, \epsilon)$, where ϵ is a small constant. This simple constraint preserves the intended monotonicity of the reliability cues: support consistency, cross-shot coverage, prompt consistency, and support agreement can only increase reliability, while local dispersion and base ambiguity can only decrease reliability. For the fusion weights of global and local novel scores, λ_{g} and λ_{l}, we normalize two learnable logits with a softmax function so that $\lambda_{\mathrm{g}}, \lambda_{\mathrm{l}} \geq 0$ and $\lambda_{\mathrm{g}} + \lambda_{\mathrm{l}} = 1$. The two logits are initialized equally, yielding $\lambda_{\mathrm{g}} = \lambda_{\mathrm{l}} = 0.5$ at the beginning of training. These constraints avoid sign reversal and unconstrained rescaling, making reliability estimation a bounded and interpretable gating mechanism for base–novel scoring.

Table E.1 Ablation of Global Prototype Reliability

$u_{cls-shot}$	u_{base}	mIoU-B	mIoU-N	mIoU-A	HM
✓		59.3	17.4	34.2	26.9
	✓	59.5	17.7	34.4	27.3
✓	✓	**59.8**	**18.2**	**34.8**	**27.9**

Table E.2 Ablation of Local Prototype Reliability.

u_{comp}	u_{cov}	u_{base}	mIoU-B	mIoU-N	mIoU-A	HM
	✓	✓	59.5	17.6	34.4	27.2
✓		✓	59.2	17.1	33.8	26.8
✓	✓		59.0	17.3	33.9	26.9
✓	✓	✓	**59.8**	**18.2**	**34.8**	**27.9**

Table E.3 Ablation of Textual Reliability.

u_{pc}	u_{sa}	u_{ba}	mIoU-B	mIoU-N	mIoU-A	HM
	✓	✓	61.1	24.1	38.9	34.6
✓		✓	61.0	23.3	38.4	33.7
✓	✓		60.8	23.6	38.5	34.0
✓	✓	✓	**61.3**	**24.4**	**39.1**	**34.9**

E More Experimental Results

In this section, we further provide ablation studies to validate the effectiveness and necessity of the reliability design.

Tab. E.1 evaluates the reliability design for global prototypes. The cross-shot consistency term $u_{cls-shot}$ and base ambiguity term u_{base} are both beneficial. The combination of these two terms gives the best performance. This trend indicates that a reliable global prototype should satisfy two conditions simultaneously: it should be stable across different support shots, and it should remain sufficiently separated from base-class semantics. Using only one of these signals is not enough, because support stability alone cannot prevent semantic drift toward base classes, while base separation alone cannot guarantee that the support representation itself is stable. Their joint improvement confirms that global prototype quality in generalized segmentation depends on both support consistency and base-aware discrimination.

Tab. E.2 analyzes the reliability of local prototypes. The results show that local prototype reliability cannot be determined by a single geometric or semantic cue; instead, compactness u_{comp}, coverage u_{cov}, and base ambiguity u_{base} play complementary roles. Compactness favors internally coherent local patterns, coverage prevents overly narrow or incomplete local support, and base ambiguity suppresses local regions that are easily confused with base classes. Only when these three aspects are considered together can the local prototype bank provide both diverse and trustworthy part-level representations for conflict-aware calibration.

Tab. E.3 studies the reliability design for optional textual prototypes. The full formulation, combining support agreement u_{sa}, prompt consistency u_{pc}, and base ambiguity u_{ba}, gives the strongest results. Among them, support agreement appears to be the most critical factor, indicating that textual priors are most useful when they are aligned with the support information observed in the current episode. Base ambiguity also plays an important role by preventing text-derived prototypes from drifting toward base semantics. Prompt consistency provides an additional stabilizing effect by reducing sensitivity to prompt variation, but its contribution is more complementary than dominant. Overall, these results show that textual priors are effective not because they are universally reliable, but because they are selectively trusted through a reliability-aware design before being injected into conflict-region calibration.

Tab. E.1–Tab. E.3 support a unified conclusion: in generalized few-shot point cloud segmentation, both visual prototypes and textual priors must be evaluated by reliability before they participate in base–novel decision making. Reliability modeling is therefore a core component of our framework, rather than an optional refinement.

F Limitations and Discussion

F.1 Limitations

Despite its effectiveness, the proposed framework still has several limitations.

First, the optional textual prior may become unreliable when textual semantics are inconsistent with support-derived visual information. Although the proposed reliability design mitigates part of this mismatch by evaluating prompt consistency, support agreement, and base ambiguity, inaccurate textual priors may still bias calibration in highly ambiguous regions. This issue can be more pronounced when the support samples are sparse, visually unrepresentative, or insufficient to verify whether the textual prototype is truly compatible with the current episode.

Second, some fine-grained novel categories remain difficult even after calibration. In these cases, the few support samples may not provide enough evidence to distinguish subtle geometric structures. Textual priors can provide complementary class-level semantics, but they are often too coarse to resolve instance-level differences among visually similar categories. Therefore, the current framework may still struggle when novel classes require fine-grained geometric discrimination beyond what sparse support and class-level text can reliably provide.

Third, the framework depends on the quality of conflict localization and retained conflict-region construction. Selective calibration is applied only to the conflict regions, so inaccurate localization can directly affect the final prediction. Under-coverage may leave important base–novel conflicts uncorrected, whereas over-coverage may introduce unnecessary changes to otherwise stable predictions. Although the minimum-size and pair-consistency filters suppress noisy proposals, conflict localization can still be challenging under severe sparsity, missing observations, or fragmented point-cloud structures.

Fourth, the proposed selective calibration adopts a dominant-pair formulation. This design is intentionally conservative: each retained region is assumed to be dominated by one base–novel confusion pattern, and the calibration head predicts a scalar offset only for the dominant pair. This makes the correction localized, controlled, and interpretable. However, in rare cases, a true ambiguous region may contain multiple strongly competing novel classes. Such regions may either be filtered out by the pair-consistency constraint or be corrected only for the most prominent pair, leaving secondary confusions less directly calibrated.

These limitations suggest that the current method is most effective when support and textual information are reasonably aligned, when fine-grained novel categories can be represented by the available support, and when ambiguous conflicts can be localized as coherent pair-dominant regions.

F.2 Summary and Future Directions

The proposed framework improves GFS-PCS by explicitly modeling localized base–novel ambiguity. The support-only framework estimates prototype reliability, localizes pair-consistent conflict regions, and selectively calibrates these regions while preserving stable predictions elsewhere. The optional textual prototype augmentation further complements support-derived prototypes when textual semantics are reliable and locally relevant.

The above limitations also point to several promising directions for future work. First, more reliable support–text consistency modeling could further reduce the risk of biased textual priors, especially when support samples are sparse or visually incomplete. Second, stronger fine-grained prototype learning may help distinguish visually similar novel categories by capturing more discriminative local geometry and contextual cues. Third, more robust conflict localization could improve calibration under noisy, sparse, or fragmented point-cloud observations. Finally, extending the current

dominant-pair formulation to multi-pair or multi-class region calibration is a meaningful direction for handling regions with multiple competing novel classes, while still preserving the locality and controllability of selective calibration. We believe progress along these directions can further improve the stability, robustness, and generalization ability of GFS-PCS.