COOPERTRIM: ADAPTIVE DATA SELECTION FOR UNCERTAINTY-AWARE COOPERATIVE PERCEPTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative perception enables autonomous agents to share encoded representations over wireless communication to enhance each other's live situational awareness. However, the tension between the limited communication bandwidth and the rich sensor information hinders its practical deployment. Recent studies have explored selection strategies that share only a subset of features per frame while striving to keep the performance on par. Nevertheless, the bandwidth requirement still stresses current wireless technologies. To fundamentally ease the tension, we take a proactive approach, exploiting the temporal continuity to identify features that capture environment dynamics, while avoiding repetitive and redundant transmission of static information. By incorporating temporal awareness, agents are empowered to dynamically adapt the sharing quantity according to environment complexity. We instantiate this intuition into an adaptive selection framework, COOPERTRIM, which introduces a novel conformal temporal uncertainty metric to gauge feature relevance, and a data-driven mechanism to dynamically determine the sharing quantity. To evaluate COOPERTRIM, we take semantic segmentation as an example task. Across multiple open-source cooperative segmentation models, COOPERTRIM achieves up to 80.28% bandwidth reduction while maintaining a comparable accuracy. Relative to other selection strategies, COOPERTRIM also improves IoU by as much as 45.54% with up to 72% less bandwidth. Qualitative results show COOPERTRIM gracefully adapt to environmental dynamics, demonstrating the flexibility and paving the way towards real-world deployment.

1 INTRODUCTION

Cooperative perception (Wang et al., 2020; Xu et al., 2023) enables autonomous agents to exchange information about invisible or uncertain regions to enhance a range of tasks, including detection (Qiu et al., 2022), prediction (Wang et al., 2025), mapping (Ahmad et al., 2020), and navigation (Cui et al., 2022). The benefits of additional vantage points come at the cost of communication. Balancing bandwidth overhead and accuracy, modern approaches adopt intermediate (over early and late) fusion (Wang et al., 2020; Xu et al., 2023) to share task-oriented lightweight encoded representations of the environment. To further improve communication efficiency in intermediate fusion, three strategies have been proposed: *compression* (Wang et al., 2020; Xu et al., 2022a), which reduces the size of the features but risks information loss if targeted at higher compression ratio (*i.e.,* lossy compression); *selection* (Hu et al., 2023; Liu et al., 2020), which transmits only useful data or selects agents; and *hybrid* (Yuan et al., 2022; Yang et al., 2023), combining both for optimal bandwidth reduction. For example, Where2comm (Hu et al., 2022) uses threshold-based spatial confidence maps for feature selection, UniSense (Ren et al., 2025) focuses on uncertainty-driven data exchange, and SwissCheese (Xie et al., 2024) exploits the disparity in semantic information on features between different spatial regions on different channels to perform fixed threshold-based selection. While these approaches reduce the exchange volume, the bandwidth demands still strain wireless technologies (Mo et al., 2025).

Fundamentally, the mismatch between limited communication bandwidth and the richness of sensor information hinders the practical deployment of cooperative perception. To address this mismatch, we take a proactive adaptation approach. Our intuition is twofold: 1) rather than sharing a smaller yet static volume of features per frame, sharing should be made on demand at a variable volume depending on the recipient's cognitive challenges for its surrounding environment; 2) rather than sharing frame by frame independently, cooperation should leverage the temporal context just as the rest of the autonomy stack does to enhance temporal comprehension while reducing repetitive and redundant sharing. Consider a scenario where the ego (recipient) agent is confident in most features from its sensor data, but collaborating agents may have varying confidence levels of elements in the scene. In such cases, the ego can learn from its temporal context to request only the uncertain features. For instance, a complex scenario (*e.g.,*multiple road intersections) should demand more features over consecutive frames than a simple scenario (*e.g.,*no intersections).

Building on top of these insights, in this paper, we present COOPERTRIM, an adaptive cooperative perception framework, which introduces a novel conformal temporal uncertainty metric to gauge feature relevance, and a data-driven mechanism to dynamically determine the sharing quantity. Our methodology employs temporal uncertainty estimation using conformal prediction inspired quantile gating mechanism to identify feature deviations across frames, alongside an uncertainty-based attention mechanism to weigh feature importance. It dynamically selects relevant features with adaptive thresholds based on environmental needs and facilitates efficient feature exchange between agents to minimize bandwidth usage while ensuring robust fusion of multi-source data.

We instantiate the framework using semantic segmentation as an example cooperative task, and evaluate it against existing open-source cooperative segmentation models (Xu et al., 2023; 2022b; Li et al., 2021) and existing selection strategies. To our knowledge, this is the first work on selective cooperative semantic segmentation. Unlike detection or tracking, which can work with sparse, object-level features, segmentation demands pixel-level granularity for exact shapes and boundaries, which poses more challenges for bandwidth reduction—a difficult case to address the bandwidth-information mismatch. We evaluate COOPERTRIM through extensive benchmarking on the OPV2V dataset (Xu et al., 2022b), assessing the performance and network overhead across multiple open-source cooperative segmentation models, as well as against other existing selection strategies. Furthermore, we perform a wider network overhead analysis by comparing the bandwidth consumption against 10 existing baselines. To understand the efficacy of the adaptation design, we evaluate data request variations across frames with respect to environment complexity.

In summary, COOPERTRIM makes the following contributions:

- We propose a learning framework to proactively adapt feature selection by dynamically determining feature *relevance* and sharing *quantity*—a transformative deviation from existing static selection frameworks.
- We propose a novel relevance assessment strategy by quantifying temporal uncertainty using conformal prediction inspired quantile gating meachanism to compare features across frames. Coupled with the conformal assessment using an adaptive quantile threshold based on conformity score, we use an attention mechanism with an adaptive mask threshold for quantity estimation.
- Our instantiation of COOPERTRIM on cooperative segmentation is the first work that demonstrates feature selection on this task. Segmentation requires transmitting large volumes of data, making selective perception challenging under bandwidth constraints. Our approach addresses this by intelligently selecting and prioritizing critical features, reducing data transmission while preserving segmentation accuracy.
- We employ a training method inspired by the ϵ-Greedy exploration strategy from reinforcement learning (Liu et al., 2022), which balances exploration and exploitation effectively in training, resulting in lower bandwidth and higher task performance.
- Our evaluation shows that COOPERTRIM achieves up to 80.28% bandwidth reduction while maintaining a comparable accuracy. Relative to other selection strategies, COOPERTRIM also improves IoU by as much as 45.54% with up to 72% less bandwidth. Quantitative frame-by-frame inspection further validates the flexibility, demonstrating graceful adaptation to environmental dynamics and paving the way towards real-world deployment.

2 Related Work

Cooperative Perception. Resilient operation of autonomous agents depends on their onboard perception, which is often limited by blind spots and uncertainties. A range of cooperative perception mechanisms have been proposed, from early fusion and edge assistance (Zhang et al., 2021) sharing raw sensor data, to late fusion of regional processing results while missing holistic scene details. Most prominently are intermediate feature fusion methods (Chen et al., 2019), however, there remains a gap between bandwidth demand and availability. V2VNet (Wang et al., 2020) uses compressed LiDAR BEV features and GNN aggregation, yet often transmitting redundant data. AttFuse (Xu et al., 2022b) and DiscoNet (Li et al., 2021) offer attentive fusion and collaboration graphs, but lack intelligent data selection. CoBEVT (Xu et al., 2023) integrates multi-vehicle BEV features via SinBEVT and FuseBEVT for segmentation, yet struggles with inefficient feature compression. Unlike these approaches, CooperTrim selectively transmits essential perception features, reducing bandwidth usage, improving network efficiency, and maintaining accuracy under constraints.

Network Efficient Cooperation. The literature on communication-efficient cooperative perception includes Compression-based, Selection-based, and Hybrid approaches. We focus on Selection-based methods due to practical wireless broadcasting needs. Where2comm (Hu et al., 2022) uses threshold based spatial confidence maps for selection, ignoring uncertainties and relying on impractical multi-round transmission. UMC (Wang et al., 2023) applies entropy-based selection but sending full maps is computationally expensive. CenterCOOP (Zhou et al., 2023) transmits center-point LiDAR features via mutual information, improving bandwidth by 10% on DAIR-V2X while sharing all embeddings. BM2CP (Zhao et al., 2023) shares modality-guided features, overlooking sensor noise. UniSense (Ren et al., 2025) selects critical regions via uncertainty, missing occlusion uncertainties and incurring high bandwidth costs. SwissCheese (Xie et al., 2024) uses fixed thresholds for selection, lacking adaptability to dynamic scenarios. Different from methods with fixed thresholds or static selection strategy, CooperTrim adapts feature selection to environment dynamics, prioritizes uncertainty-driven critical features using past confident data, balancing bandwidth efficiency and perception performance resilience.

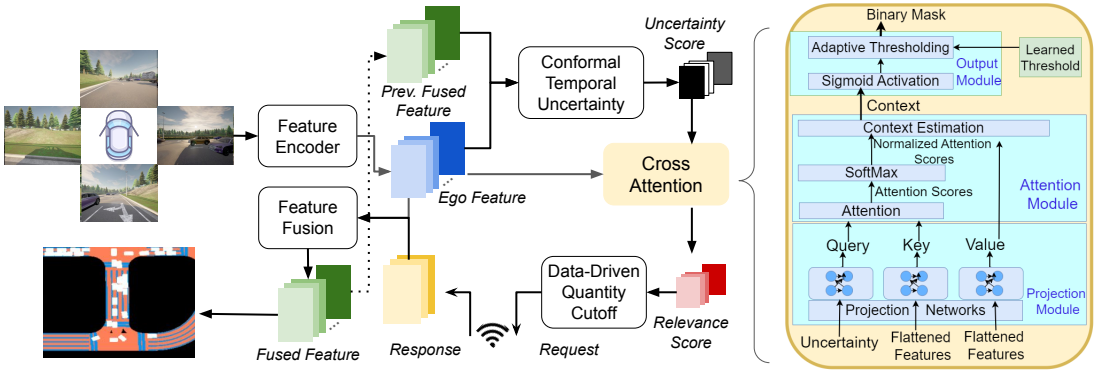
3 Methodology and Solution

3.1 Problem Statement

We address the challenge of selective feature sharing for cooperative perception tasks. Rather than designing a new model, we aim at a selection framework that is applicable to a set of similar feature-sharing mechanisms (Xu et al. (2023); Hu et al. (2022); Li et al. (2021). In particular, we focus on the segmentation task. Given a set of features $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ extracted from input data, where n is the number of feature channels in latent representations, our objective is to identify a subset $\mathcal{S} \subseteq \mathcal{F}$ such that the number of selected features $|\mathcal{S}|$ is minimized, while the accuracy of the segmentation task, denoted as a function $A(\mathcal{S})$ of the selected features, is maximized. Formally, the optimization problem can be expressed as $\min_{\mathcal{S}\subseteq\mathcal{F}} |\mathcal{S}|, \max_{\mathcal{S}\subseteq\mathcal{F}} A(\mathcal{S})$. This bi-objective optimization problem poses significant challenges due to the *inherent **tension** between feature abundance and bandwidth limitations*.

3.2 CooperTrim Design

To resolve that tension, we present CooperTrim, a selective cooperative perception framework that learns to enhance representation learning via temporal uncertainty-driven feature selection for bandwidth-efficient, accurate perception in multi-agent systems. CooperTrim addresses two key research questions:



(a) CooperTrim Overview (b) Cross-Attention Module

Figure 1: (a) CooperTrim Overview. CooperTrim conducts feature learning, followed by an uncertainty-based selection module using learned features. It estimates adaptive temporal uncertainty (via learned confidence) for each feature, performs cross-attention-based feature weighting, and selects features using a learned threshold. The ego then broadcasts a request vector for selected features, reconstructs received CAV data into full features, fuses them, and sends them to the task head for final results. (b) Cross-Attention Module uses learned projections of temporal uncertainty as queries, and feature projections as keys and values. These matrices pass through an attention module, and a learned threshold at the final output generates a binary mask for selected channels.

- **Relevance.** If the bandwidth does not permit all features to be shared, what are the most essential features that are impactful to downstream tasks? The relevance is inevitably recipient-centric, which demands the recipient vehicle to assess its field-of-view visibility, perception uncertainty, and environment dynamics.
- **Quantity.** If most relevant features are prioritized, where is the diminishing return point to stop sharing? Such quantity sweet spot may be dependent on both scene- and task-complexity, which demands dynamic runtime adaptation, a sophistication in CooperTrim framework design.

CooperTrim leverages two key insights in addressing the above research questions. i) *Temporal Uncertainty as Relevance.* Unlike conventional approaches Ren et al. (2025); Xie et al. (2024); Hu et al. (2022), which quantify the feature relevance of each individual frame, we put the features in their temporal contexts, and quantify the temporal uncertainty (*e.g.,* introduced by scene dynamics, changing lighting conditions, occlusions) as relevance to determine sharing priority. We use conformal prediction inspired quantile gating method to better assess the temporal uncertainty in continuous frames. ii) *Environment Adaptivity.* Adapting to environment dynamics, we introduce two learned parameters: a data-driven uncertainty threshold used in the uncertainty estimation, and a data-driven attention mask threshold, to adjust the runtime sharing quantity for each frame. This flexibility empowers CooperTrim to scale gracefully while maintaining efficient bandwidth utilization. Figure 1 shows an architectural overview of CooperTrim. We describe the components in detail.

Conformal Temporal Uncertainty. CooperTrim takes encoded representations from sensor data, contextualizes them within the ego agent's recent memory, and identifies the uncertain ones as candidates for enhancement through cooperation. Specifically, given current frame feature $F_t \in \mathbb{R}^{H' \times W' \times D}$, encoded from sensor input $X_t \in \mathbb{R}^{H \times W \times C}$, it calculates conformity scores by comparing the feature F_t against the fused features from previous frames, denoted as distance function $S_t = d(F_t, F_{t-1}^{\text{fused}})$, where $d(.)$ is the L1 distance defined as $|F_t - F_{t-1}^{\text{fused}}|$, effectively capturing deviations between past and current scene understanding. In this context, the use of 'conformal' is inspired by conformal prediction, differing by (a) learning online frame-wise instead of using a fixed calibration dataset, and (b) estimating uncertainties in

components of regression values rather than direct intervals on entire regression data. Then, inspired by conformal prediction, we use quantile gating to assess the features' relevance, introducing a learnable quantile threshold q, and apply a cross-attention mechanism between the features and the ranges of those scores above the threshold (*i.e.,* $S_t(f) > q$) to obtain the relevance metric R_t.

Data-driven Quantity Cutoff. In order to adapt to environment dynamics, we introduce a mechanism to determine the diminishing return point of the sharing volume. Specifically, we introduce a second learnable threshold τ such that we only share those features whose relevance score is above the threshold ($R_t > \tau$). In situations where the perception complexity is high, the range of conformity score will increase due to more temporally diverse feature encodings, which, in turn, yields higher relevance scores. More features would have relevance scores above the learned threshold; the communication of which incurs higher bandwidth usage. On the contrary, temporal consistency will result in low relevance scores among all features, which yields low sharing volume. Moreover, the temporal consistency not only comes from the low environment dynamics, but also comes from perception stability—COOPERTRIM leverages this stability to save bandwidth as well.

Feature Exchange and Fusion. To communicate recipient-centric relevance assessment, the ego agents send requests for selected high-relevance features. Following prior work assumptions of precise pose estimation (Xu et al., 2023), responders use spatial transformation (Jaderberg et al., 2015) to match ego's perspective and share the requested features. The ego agent blends received features into its own feature map before passing them along to the fusion decoders and task heads.

3.3 TRAINING

Loss Function. To attain the joint goal of bandwidth efficient transmission and task performance, the training of COOPERTRIM can be formulated as a constrained optimization problem defined as follows:

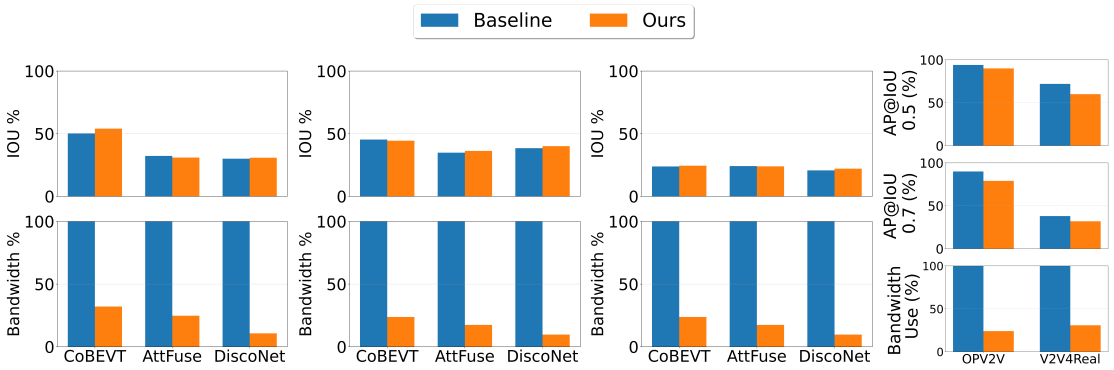
$$\theta^* = \arg\min_{\theta} L(C(\theta)), \text{s.t. } P(C(\theta)) = C_{1.6} \tag{1}$$

where $C_{1.6}$ represents the percentage of channels corresponding to 1.6 Mbps (Mo et al., 2025), $C(\theta)$ represents the channels selected, $P(\cdot)$ represents the precentage of channels selected, $L(\cdot)$ represents the perception task loss. To solve this, we use the Lagrangian formulation, defining the total loss as

$$\theta^* = \arg\min_{\theta} L(C(\theta)) + \lambda \cdot (P(C(\theta)) - C_{1.6}) \tag{2}$$

where λ acts as a Lagrange multiplier dynamically adjusted (explained in Appendix A.1) to enforce the constraint over time. The strategy starts with unconstrained optimization for initial learning, then introduces and intensifies constraint enforcement over time, using periodic strong adjustments for major deviations and steady increments for fine-tuning.

Training Strategy. For training the loss function in COOPERTRIM, we use an ϵ-Greedy method inspired by the reinforcement learning ϵ-greedy exploration strategy (Liu et al., 2022). However, we make some adaptations for our training purpose. With ϵ probability we request entire feature set and with $(1 - \epsilon)$ probability we exploit the knowledge. By exploiting the knowledge, we request the data as per learned thresholds (i.e. conformal predictions's quantile threshold q and cross-attention module's mask threshold τ). We see considerable improvement in bandwidth requirement with this training method. Occasional updates considering all features can stabilize the optimization trajectory by smoothing out erratic updates caused by noise from partial features. This can prevent the model from diverging or oscillating around suboptimal points, leading to smoother convergence to a better solution. The ϵ-Greedy method balances exploration (full data D_{full} with probability ϵ) and exploitation (partial data D_{partial} with probability $(1 - \epsilon)$). This balance stabilizes optimization by reducing gradient noise through periodic full-data updates. As formalized in Theorem 1, this strategy reduces both the bias and variance of the gradient estimator compared to using



(a) Dynamic (b) Static: Road (c) Static: Lane (d) 3D Detection

Figure 2: "Trimming" existing cooperative perception baselines, CooperTrim reduces bandwidth significantly while preserving accuracy across different semantics (*i.e.,* dynamic, static road, and static lane for segmentation task and for 3D detection task.)

only partial data. The expected gradient

$$\mathbb{E}[\nabla \mathrm{L}] = \epsilon \cdot \mathbb{E}[\nabla \mathrm{L}(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla \mathrm{L}(D_{\text{partial}})] \tag{3}$$

ensures smoother convergence while adhering to bandwidth constraints. We formalize the effectiveness of the ϵ-Greedy training strategy in the following theorem.

Theoretical Analysis. We draw a theoretical analysis based on the following assumptions. (1) Perfect transmission conditions (no noise introduced during transmission, so any bias or variance arises solely from the data used), (2) A complete dataset D_{full} and a subset $D_{\text{partial}} \subseteq D_{\text{full}}$ representing partial data, (3) The true gradient is $\nabla \mathrm{L}(D_{\text{full}})$, reflecting the loss over the entire dataset. (4) The inherent variances of the gradients are $\mathrm{Var}(\nabla \mathrm{L}(D_{\text{full}})) = \sigma^2_{\text{full}}$ and $\mathrm{Var}(\nabla \mathrm{L}(D_{\text{partial}})) = \sigma^2_{\text{partial}}$, with $\sigma^2_{\text{full}} < \sigma^2_{\text{partial}}$ due to the more comprehensive nature of full data reducing stochasticity.

Theorem 1 (Effectiveness of ϵ-Greedy Training)**.** *An ϵ-greedy training strategy that computes the gradient of Loss L using full data (D_{full}) with probability ϵ and partial data ($D_{partial}$) with probability $(1-\epsilon)$ reduces the bias of the gradient estimator compared to using only partial data. Specifically, the bias is scaled down by a factor of $(1-\epsilon)$, and the variance is reduced by a term proportional to $\epsilon \cdot (\sigma^2_{partial} - \sigma^2_{full})$.*

Proof. Detailed proof provided in Appendix A.2. □

4 Experiments

We conduct six experiments to demonstrate CooperTrim's efficiency in task performance and bandwidth usage. Referred to as CooperTrim, we apply CooperTrim to CoBEVT (Xu et al., 2023) for evaluation, focusing on semantic segmentation on the OPV2V (Xu et al., 2022b) dataset. We choose CoBEVT for its robust performance in Cooperative Segmentation compared to existing works. CooperTrim is the first to address selective cooperative perception in semantic segmentation.

"Trimming" Existing Cooperative Perception Baselines. In Figure 2, we assess performance and bandwidth (BW) by applying our temporal uncertainty-aware method CooperTrim to three methods: CoBEVT (Xu et al., 2023), Attfuse (Xu et al., 2022b), and DiscoNet (Li et al., 2021). We implement CooperTrim on these methods after removing any existing compression techniques. This evaluation compares

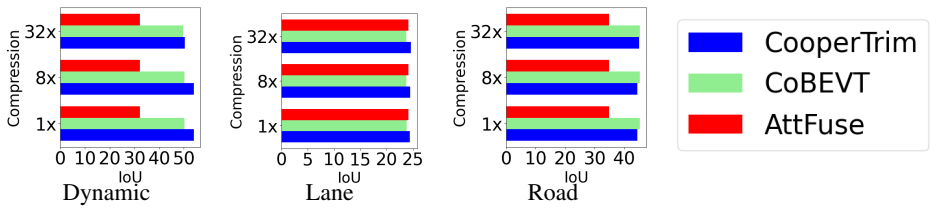


Figure 4: Comparison of IoU performance and bandwidth usage at compression rates (1x, 8x, 32x) for CooperTrim, CoBEVT, and AttFuse in Dynamic, Road, and Lane scenarios. CooperTrim consistently outperforms the baselines.

performance accuracy between COOPERTRIM-based selection and the original methods (without compression/selection), and reports bandwidth usage percentages relative to the originals (40 Mbps). For Dynamic and Static segmentation combined, COOPERTRIM-CoBEVT, COOPERTRIM-Attfuse, and COOPERTRIM-DiscoNet use average bandwidths of 27.9%, 21.07%, and 10.18%, respectively, corresponding to 11.16 Mbps, 8.4 Mbps, and 4.07 Mbps based on our 128x32x32 latent representation size. We observe that COOPERTRIM achieves comparable performance accuracies to the original methods despite significantly lower bandwidth consumption, with an average 80.28% improvement in network overhead over the baselines. Detailed values are in Appendix Table 4.

Additionally, we validate COOPERTRIM' generalizability on 3D detection using Lidar data on OPV2V and V2V4Real datasets. Figure 2d shows COOPERTRIM performance versus baseline CoBEVT at IoU 0.5 and 0.7. COOPERTRIM maintains comparable detection accuracy with a 27.48% bandwidth reduction, highlighting its efficiency in lowering network overhead.

Selection Strategy Comparison. Table 1 presents the evaluation of COOPERTRIM against selection-based, network-efficient baselines. Our feature latent representation is sized at 128x32x32, with percentages and bandwidths reported accordingly. We implement two algorithms: Where2Comm (Hu et al., 2022) (feature and agent selection, adapted for segmentation with a 0.4 threshold on batch confidence map) and SwissCheese (Xie et al., 2024) (feature selection over spatial and channel features, implemented in our camera-based segmentation framework as the official code is unavailable). For SwissCheese, we set bandwidth at 10 Mbps (comparable to COOPERTRIM, equating to 25% of our latent representation size) to assess segmentation performance. COOPERTRIM achieves lower bandwidth usage and better performance than Where2Comm, and outperforms SwissCheese in dynamic and static accuracy at similar bandwidth levels.

Compression-based Method Comparison. Compression-based methods like CoBEVT (Xu et al., 2023) and AttFuse (Xu et al., 2022b) reduce data size for network efficiency. Yet, COOPERTRIM' data selection enhances efficiency by compressing post-selection, optimizing bandwidth. We apply lossy quantization to reduce precision and volume, then lossless methods for smaller size. Figure 4 compares IoU and bandwidth for COOPERTRIM, CoBEVT, and AttFuse at 1x, 8x, and 32x rates. COOPERTRIM excels in IoU and bandwidth savings, notably at 32x (1.46% vs. 3.76% for CoBEVT and 10.62% for AttFuse in 32x Dyn).

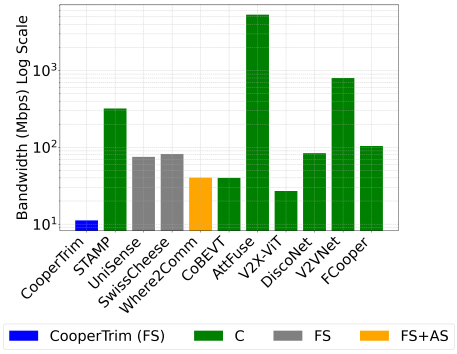


Figure 3: Across-the-board Bandwidth Comparison. C: Compression. FS: Feature Selection. AS: Agent Selection. COOPERTRIM consumes the lowest bandwidth among baselines.

Network Overhead Comparison. We compare the network overhead of COOPERTRIM with used bandwidths of a broader range of existing cooperative perception works. We evaluate Feature Selection (FS)

Table 1: Feature Selection Strategy Comparison: Accuracy and Bandwidth

Baselines	Accuracy (IoU, %) Dynamic	Static Lane	Static Road	Bandwidth (Mbps)
Where2Comm (Hu et al., 2022)	8.62	20.40	36.46	39.6
SwissCheese (Xie et al., 2024)	35.71	12.81	32.07	10
CooperTrim (ours)	**54.03**	**24.45**	**44.38**	11.16

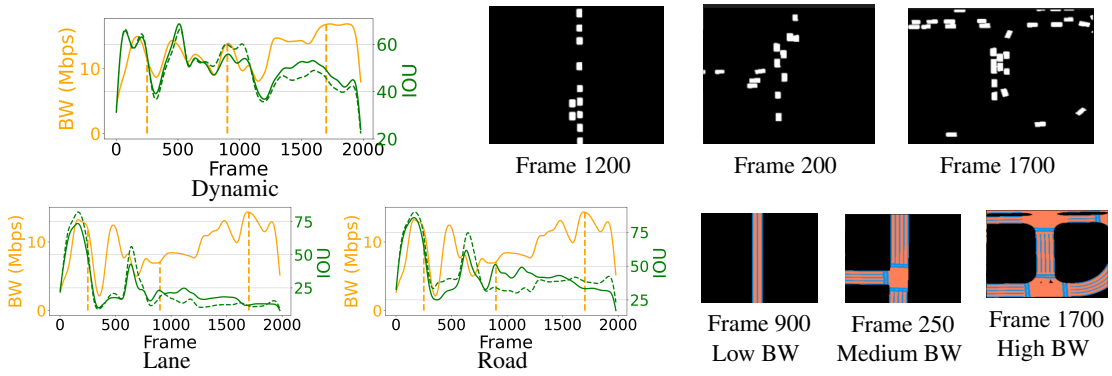


Figure 5: Increased data requests align with higher scene complexity. For dynamic objects, complexity in number and positioning rises in Frames 1200, 200, and 1700. For static elements, complexity grows in Frames 900, 250, and 1600, with more intersections and lane orientations. Vertical lines at Frames 1200, 200, 1700 (dynamic) and 900, 250, 1600 (static) link to visualizations. Green dashed lines show baseline CoBEVT IOU; green solid lines show CooperTrim IoU.

algorithms UniSense (Ren et al., 2025), SwissCheese (Xie et al., 2024), Compression-based (C) algorithms STAMP (Gao et al.), CoBEVT (Xu et al., 2023), V2X-ViT (Xu et al., 2022a), V2VNet (Wang et al., 2020), FCooper (Chen et al., 2019), AttFuse (Xu et al., 2022b), DiscoNet (Li et al., 2021), and Agent+Feature Selection (AS+FS) method Where2Comm (Hu et al., 2022).

Bandwidth comparisons use non-compressed versions (post-selection if applicable) for fairness, reporting original bandwidths from papers or calculating based on our 128x32x32 latent representation size when unspecified. Figure 3 show CooperTrim has the lowest network overhead (11.6Mbps) compared to baselines.

Environment Adaptation. Figure 5 shows the inference results for CooperTrim over multiple frames. We see variations in bandwidth implying variable feature request in each frame. The figure shows increased data requests correlate with higher scene complexity. In segmenting the dynamic objects, an increase in complexity can be referred to criticality in positioning of an increased amount of vehicles and traffic participants. Similarly, an increase in complexity in the static task can be referred to an increase in the number of intersections or lane orientations in the scene. Figure 5 shows that in both cases, *i.e.,* increased amount of vehicles, and increased topological complexity in the roadways, CooperTrim accordingly shares more data to preserve task performance, whereas in other cases, gracefully adapts to request less sharing, saving the precious bandwidth to other agents in need. In critical frame ranges such as 1000-1800 (Dynamic), 1000-1500 (Lane), and 1200-1400 (Road), where baseline CoBEVT consistently underperforms, CooperTrim achieves much higher IoU, illustrating the effectiveness of its adaptive threshold masking mechanism in prioritizing key features and managing scene complexity.

Training Methods Comparison. Table 2 shows the comparative study for training CooperTrim using different training methods, all using the loss function in Section 3.3. **(1)** CooperTrim is trained with ϵ-Greedy (EG) based fine-tuning (FT) using Conformal Prediction (CP) for temporal uncertainty estimation [Section 3.3], termed $EG + CP + FT$. **(2)** The first method uses Standard Deviation (SD) for spatial uncertainty estimation over latent feature channels ($EG+SD+FT$), showing higher network consumption for dynamic objects (35.64% vs. CooperTrim at 32.04%) due to lack of temporal consideration in SD,

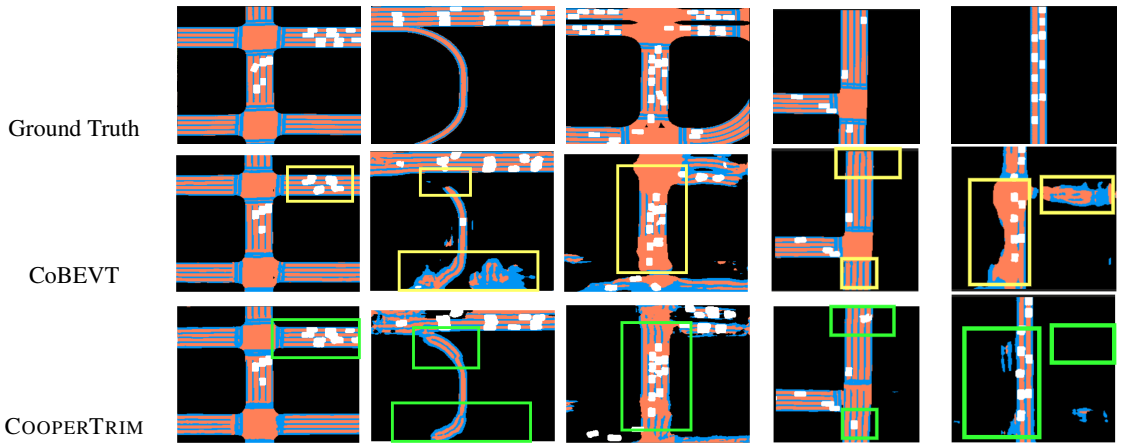


Figure 7: Qualitative Comparison of Ground Truth, CoBEVT, and CooperTrim. CooperTrim performs better than CoBEVT in estimating dynamic object segments (Column 1 and 4), road segments (Column 2 and 5), and lane segments (Column 3 and 5).

and 6% lower performance in static road segmentation. **(3)** The second method employs a basic curriculum for fine-tuning ($Curriculum + CP + FT$) with four stages: (a) basic fine-tuning, (b) cross-attention with fixed mask threshold and CP confidence, (c) adaptive mask threshold with fixed CP confidence, and (d) fully adaptive (learned mask and CP confidence) CP-guided selection. This yields good performance but higher bandwidth use (49% dynamic, 51% static). **(4)** The final method omits EG in FT ($CP + FT$), resulting in lower accuracy (static road 8% below CooperTrim). EG's occasional full-feature updates stabilize optimization by promoting smoother convergence.

Table 2: Training Methods Comparison for Uncertainty Aware Training showing EG+CP+FT(CooperTrim) strikes a good balance between Accuracy and Bandwidth

Training Methods	Accuracy (IoU, %)			Bandwidth (%)	
	Dynamic	Static Lane	Static Road	Dynamic	Static
CP+FT	51.57	23.05	36.92	**30.75**	**1.12**
Curriculum+CP+FT	52.47	**28.27**	**45.44**	49.25	51.63
EG+SD+FT	52.55	23.69	38.45	35.64	11.03
EG+CP+FT (**Ours**)	**54.03**	24.45	44.38	32.04	23.77

Qualitative Results. Figure 7 shows the segmentation results of CooperTrim to analyse visually. CooperTrim is different from CoBEVT (Xu et al., 2023) only in its feature set before fusion of muti-source data. CooperTrim performs better than CoBEVT in estimating road segment, dynamic object segment, and lane segment. This is attributed to the better feature representation achieved by CooperTrim selection mechanism, which reduces uncertainty in feature, by focusing on selected data transmission rather than transmitting the entire dataset for fusion.

Sensitivity Analysis CooperTrim' performance is tested across localization errors (0cm, ±20cm, ±1m) during inference, as shown in Figure 6. CooperTrim remains robust to small errors (up to ±20cm) and maintains comparable IoU at ±1m, with stable bandwidth indicating consistent communication efficiency.

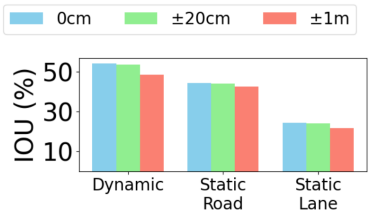


Figure 6: CooperTrim shows comparable performance under localization error at inference.

Table 3: System Overhead Comparison between COOPERTRIM and baseline CoBEVT. COOPERTRIM has a slight FPS reduction compared to CoBEVT due to added uncertainty and cross-attention modules for data selection.

Method	Configuration	FPS	Time(ms) per Frame
CoBEVT	Dynamic	10.59	92.9
CooperTrim	Dynamic	8.58	94.7
CoBEVT	Static	10.59	93.1
CooperTrim	Static	8.58	119.5

COOPERTRIM' performance under latency (0ms to 200ms) is shown in Figure 8. It exhibits strong robustness up to 50ms with no loss, and negligible IoU drop at 100ms and 200ms. This resilience stems from COOPERTRIM' adaptive mechanisms, two-threshold policy, and uncertainty-driven requests, reducing reliance on misaligned data from CAVs. By requesting less data (23-32% vs. traditional systems, per Appendix Table 4), latency impact is minimized, enhancing real-world robustness. Network usage stayed consistent during testing.

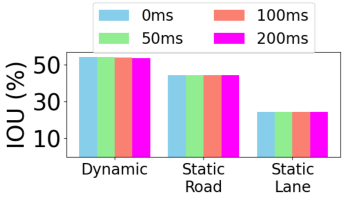


Figure 8: COOPERTRIM shows comparable performance under latency in inference.

4.1 SYSTEM OVERHEAD

We present the processing speed in Frames Per Second (FPS) for CoBEVT and CooperTrim under static and dynamic settings in the Table 3. The results indicate a modest FPS reduction in CooperTrim compared to the baseline CoBEVT, with a decrease of approximately 2 FPS in both configurations. This reduction is due to the additional computational steps in our method. However, this trade-off is acceptable given the significant bandwidth reduction achieved, which remains the primary focus of our work.

5 CONCLUSION

We present COOPERTRIM, an adaptive feature selection framework in cooperative perception, which enhances representation learning through temporal uncertainty-driven feature selection for bandwidth-efficient, accurate perception in multi-agent systems. It addresses key challenges of relevance, identifying the most impactful features for downstream tasks, and quantity, determining the optimal point to stop sharing based on scene and task complexity. We employed an ϵ-greedy training method that optimizes the bandwidth-performance balance by facilitating effective exploration and exploitation during training. Across segmentation models, COOPERTRIM cuts bandwidth by 80.28%, maintains accuracy, boosts IoU by 45.54%, and for detection 41.84% bandwidth improvement. Compared to selection and compression based baselines, it outperforms both categories in terms of performance and network use. COOPERTRIM shows strong robustness against latency, localization and system overhead.

REFERENCES

Fawad Ahmad, Hang Qiu, Ray Eells, Fan Bai, and Ramesh Govindan. Carmap: Fast 3d feature map updates for automobiles. In *Proceedings of the 17th Symposium on Networked Systems Design and Implementation*, NSDI '20, pp. 1063–1081, Santa Clara, CA, Feb 2020. USENIX Association.

Qi Chen, Xu Ma, Sihai Tang, Jingda Guo, Qing Yang, and Song Fu. F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3d point clouds. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pp. 88–100, 2019.

Jiaxun Cui, Hang Qiu, Dian Chen, Peter Stone, and Yuke Zhu. Coopernaut: End-to-end driving with cooperative perception for networked vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, CVPR '22, June 2022.

Xiangbo Gao, Runsheng Xu, Jiachen Li, Ziran Wang, Zhiwen Fan, and Zhengzhong Tu. Stamp: Scalable task-and model-agnostic collaborative perception. In *The Thirteenth International Conference on Learning Representations*.

Yue Hu, Shaoheng Fang, Zixing Lei, Yiqi Zhong, and Siheng Chen. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Advances in neural information processing systems*, 35:4874–4886, 2022.

Yue Hu, Yifan Lu, Runsheng Xu, Weidi Xie, Siheng Chen, and Yanfeng Wang. Collaboration helps camera overtake lidar in 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9243–9252, 2023.

Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. *Advances in Neural Information Processing Systems*, 34: 29541–29552, 2021.

Fanghui Liu, Luca Viano, and Volkan Cevher. Understanding deep neural function approximation in reinforcement learning via ϵ-greedy exploration. *Advances in Neural Information Processing Systems*, 35: 5093–5108, 2022.

Yen-Cheng Liu, Junjiao Tian, Chih-Yao Ma, Nathan Glaser, Chia-Wen Kuo, and Zsolt Kira. Who2com: Collaborative perception via learnable handshake communication. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6876–6883. IEEE, 2020.

Ruoshen Mo, Bo Wu, Zhaowei Tan, and Hang Qiu. See-v2x: C-v2x direct communication dataset: An application-centric approach. In *Proceedings of the 23rd ACM Conference on Embedded Networked Sensor Systems*, pp. 305–311, 2025.

Hang Qiu, Po-Han Huang, Namo Asavisanu, Xiaochen Liu, Konstantinos Psounis, and Ramesh Govindan. Autocast: scalable infrastructure-less cooperative perception for distributed collaborative driving. In *ACM MobiSys*, 2022.

Haojie Ren, Wuyang Zhang, Shuyao Shi, Xinran Zhang, Lu Zhang, and Yanyong Zhang. Unisense: Spatial-uncertainty-aware collaborative sensing for autonomous driving. In *Proceedings of the 23nd Annual International Conference on Mobile Systems, Applications and Services*, 2025.

Tianhang Wang, Guang Chen, Kai Chen, Zhengfa Liu, Bo Zhang, Alois Knoll, and Changjun Jiang. Umc: A unified bandwidth-efficient and multi-resolution based collaborative perception framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8187–8196, 2023.

Tsun-Hsuan Wang, Sivabalan Manivasagam, Ming Liang, Bin Yang, Wenyuan Zeng, and Raquel Urtasun. V2vnet: Vehicle-to-vehicle communication for joint perception and prediction. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part II 16*, pp. 605–621. Springer, 2020.

Zehao Wang, Yuping Wang, Zhuoyuan Wu, Hengbo Ma, Zhaowei Li, Hang Qiu, and Jiachen Li. Cmp: Cooperative motion prediction with multi-agent communication. *IEEE Robotics and Automation Letters*, 2025.

Qi Xie, Xiaobo Zhou, Tianyu Hong, Tie Qiu, and Wenyu Qu. Swisscheese: Fine-grained channel-spatial feature filtering for communication-efficient cooperative perception. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

Runsheng Xu, Hao Xiang, Zhengzhong Tu, Xin Xia, Ming-Hsuan Yang, and Jiaqi Ma. V2x-vit: Vehicle-to-everything cooperative perception with vision transformer. In *European conference on computer vision*, pp. 107–124. Springer, 2022a.

Runsheng Xu, Hao Xiang, Xin Xia, Xu Han, Jinlong Li, and Jiaqi Ma. Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2583–2589. IEEE, 2022b.

Runsheng Xu, Zhengzhong Tu, Hao Xiang, Wei Shao, Bolei Zhou, and Jiaqi Ma. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning*, pp. 989–1000. PMLR, 2023.

Kun Yang, Dingkang Yang, Jingyu Zhang, Hanqi Wang, Peng Sun, and Liang Song. What2comm: Towards communication-efficient collaborative perception via feature decoupling. In *Proceedings of the 31st ACM international conference on multimedia*, pp. 7686–7695, 2023.

Yunshuang Yuan, Hao Cheng, and Monika Sester. Keypoints-based deep feature fusion for cooperative vehicle detection of autonomous driving. *IEEE Robotics and Automation Letters*, 7(2):3054–3061, 2022.

Xumiao Zhang, Anlan Zhang, Jiachen Sun, Xiao Zhu, Y Ethan Guo, Feng Qian, and Z Morley Mao. Emp: Edge-assisted multi-vehicle perception. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 545–558, 2021.

Binyu Zhao, Wei ZHANG, and Zhaonian Zou. Bm2cp: Efficient collaborative perception with lidar-camera modalities. In *Conference on Robot Learning*, pp. 1022–1035. PMLR, 2023.

Linyi Zhou, Zhongxue Gan, and Jiayuan Fan. Centercoop: Center-based feature aggregation for communication-efficient vehicle-infrastructure cooperative 3d object detection. *IEEE Robotics and Automation Letters*, 9(4):3570–3577, 2023.

A APPENDIX

A.1 LAGRANGE MULTIPLIER ADJUSTMENT

This section explains the dynamic adjustment of Lagrange multiplier for training the loss in Equation 2 The Lagrange multiplier, represented as λ, acts as a penalty term in the loss function to enforce the constraint on the percentage of selected features. It dynamically adjusts based on training progress (epoch count) to balance the primary perception loss ($L(C(\theta))$) and the bandwidth constraint ($P(C(\theta)) = C_{1.6}$).

- **Initial Phase (epoch $\leq$ ITC)**: For the first few epochs (defined by Initial Tuning Epochs ITC), $\lambda = 0.0$, allowing the model to focus solely on minimizing perception loss without bandwidth constraints for a strong start.
- **Periodic Update (every 10th epoch after initial tuning)**: At every 10th epoch, λ scales exponentially with scaling factor SF $= \frac{P(C(\theta))}{100.0}$ and $\lambda = \lambda \cdot 2^{\text{SF}}$, aggressively pushing the model towards the target constraint if far off.
- **Incremental Scaling (other epochs after initial tuning)**: In other epochs, λ increases linearly as $\lambda = \lambda \cdot (1 + 0.1 \cdot \lfloor \frac{\text{epoch}-\text{ITC}}{10} \rfloor)$, ensuring a gradual push towards the constraint without abrupt loss changes.

The strategy starts with unconstrained optimization for initial learning, then introduces and intensifies constraint enforcement over time, using periodic strong adjustments for major deviations and steady increments for fine-tuning.

A.2 MATHEMATICAL PROOF FOR EFFECTIVENESS OF ϵ-GREEDY TRAINING

We provide a detailed proof to demonstrate the effectiveness of the ϵ-greedy training strategy (Section 3.3) by analyzing the bias and variance of the gradient estimator. This proof shows reductions in both metrics compared to a baseline of using only partial data. We assume perfect transmission, meaning all variability and bias stem from the inherent properties of the data (full or partial) rather than external noise.

1. SETUP AND NOTATION

- **Datasets**: Let D_{full} represent the complete dataset, and D_{partial} be a subset of D_{full}, representing partial data.
- **True Gradient**: The target gradient to estimate is $\nabla L(D_{\text{full}})$, reflecting the loss over the entire dataset.
- **Transmission Assumption**: Under perfect transmission, computed gradients are unaffected by external noise, so bias or variance arises solely from the data used.
- **ϵ-Greedy Gradient Estimator**: Define the gradient estimator under the ϵ-greedy strategy as ∇L_ϵ, which is:
  - $\nabla L(D_{\text{full}})$ with probability ϵ
  - $\nabla L(D_{\text{partial}})$ with probability $1 - \epsilon$
- **Variance Definitions**: Let the inherent variance of gradients be $\text{Var}(\nabla L(D_{\text{full}})) = \sigma^2_{\text{full}}$ and $\text{Var}(\nabla L(D_{\text{partial}})) = \sigma^2_{\text{partial}}$, where $\sigma^2_{\text{full}} < \sigma^2_{\text{partial}}$ due to the comprehensive nature of full data reducing stochasticity.

2. BIAS ANALYSIS OF THE GRADIENT ESTIMATOR

The bias of an estimator is the difference between its expected value and the true value. Here, the true gradient is $\nabla L(D_{\text{full}})$, as full data provides the most accurate representation of the loss landscape.

Expected Value of the Estimator

The expected value of the ϵ-greedy gradient estimator is:

$$\mathbb{E}[\nabla L_\epsilon] = \epsilon \cdot \mathbb{E}[\nabla L(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]$$

Bias Calculation

The bias is defined as:

$$\text{Bias}(\nabla L_\epsilon) = \mathbb{E}[\nabla L_\epsilon] - \mathbb{E}[\nabla L(D_{\text{full}})]$$

Substituting the expected value:

$$\text{Bias}(\nabla L_\epsilon) = (\epsilon \cdot \mathbb{E}[\nabla L(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]) - \mathbb{E}[\nabla L(D_{\text{full}})]$$

Simplifying step-by-step:

$$\begin{aligned}\text{Bias}(\nabla L_\epsilon) &= \epsilon \cdot \mathbb{E}[\nabla L(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{partial}})] - \mathbb{E}[\nabla L(D_{\text{full}})] \\ &= (\epsilon - 1) \cdot \mathbb{E}[\nabla L(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{partial}})] \\ &= (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})] - \mathbb{E}[\nabla L(D_{\text{full}})])\end{aligned}$$

Thus, the bias expression is:

$$\text{Bias}(\nabla L_\epsilon) = (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})] - \mathbb{E}[\nabla L(D_{\text{full}})])$$

3. Explanation of Specific Cases and Bias Scaling

We explore the implications of the bias expression under different scenarios and explain the origin of the bias scaling by $1-\epsilon$.

Case 1: Unbiased Partial Data Gradient

If the partial data gradient is unbiased, i.e., $\mathbb{E}[\nabla L(D_{\text{partial}})] = \mathbb{E}[\nabla L(D_{\text{full}})]$, the difference term is zero:

$$\mathbb{E}[\nabla L(D_{\text{partial}})] - \mathbb{E}[\nabla L(D_{\text{full}})] = 0$$

Thus:

$$\text{Bias}(\nabla L_\epsilon) = (1-\epsilon) \cdot 0 = 0$$

In this ideal scenario, there is no bias in the ϵ-greedy estimator regardless of ϵ. This occurs when D_{partial} perfectly represents D_{full}, which is rare in practice due to sampling variability.

Case 2: Biased Partial Data Gradient and Origin of Bias Scaling

If there is a systematic difference, i.e., $\mathbb{E}[\nabla L(D_{\text{partial}})] \neq \mathbb{E}[\nabla L(D_{\text{full}})]$, define the inherent bias of using only partial data as:

$$\text{Bias}(\nabla L(D_{\text{partial}})) = \mathbb{E}[\nabla L(D_{\text{partial}})] - \mathbb{E}[\nabla L(D_{\text{full}})]$$

Substituting into the bias expression:

$$\text{Bias}(\nabla L_\epsilon) = (1-\epsilon) \cdot \text{Bias}(\nabla L(D_{\text{partial}}))$$

Considering the magnitude of bias:

$$|\text{Bias}(\nabla L_\epsilon)| = (1-\epsilon) \cdot |\text{Bias}(\nabla L(D_{\text{partial}}))|$$

Since $0 \leq \epsilon \leq 1$, the factor $1-\epsilon < 1$ for any $\epsilon > 0$, implying:

$$|\text{Bias}(\nabla L_\epsilon)| < |\text{Bias}(\nabla L(D_{\text{partial}}))|$$

Origin of Scaling: The factor $1-\epsilon$ comes from the probabilistic weighting in the ϵ-greedy strategy. The expected gradient is a weighted average where the partial data gradient (carrying inherent bias) is weighted by $1-\epsilon$, and the unbiased full data gradient is weighted by ϵ. Thus, the contribution of the biased gradient is reduced, scaling the bias by $1-\epsilon$.

4. Summary of Bias Reduction Mechanism

- **Scaling Effect**: The bias scales by $1-\epsilon$ due to the probabilistic blending of full and partial data gradients.
- **Extreme Cases**:
  - When $\epsilon = 0$, only partial data is used, and the bias equals the full inherent bias of the partial data gradient.
  - When $\epsilon > 0$, incorporating full data reduces the weight of the biased partial data gradient to $1-\epsilon$, scaling down the bias.
- **Bias Comparison**: The inequality $|\text{Bias}(\nabla L_\epsilon)| < |\text{Bias}(\nabla L(D_{\text{partial}}))|$ holds for any $\epsilon > 0$, showing that even a small probability of using full data mitigates bias.

This mechanism highlights the ϵ-greedy approach's advantage in balancing computational efficiency (using partial data) with accuracy (using full data to correct bias).

3. Variance Analysis of the Gradient Estimator

The variance of the gradient estimator captures the variability in the updates, which affects training stability. Using the variance identity $\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$, we derive the variance of the ϵ-greedy estimator. The expected value of the squared gradient is the weighted average of the expected squared gradients from each component.

Step 1: Expected Squared Gradient

Let's denote the gradient choice as a random variable determined by the ϵ-greedy policy. The squared gradient's expectation is:

$$\mathbb{E}[(\nabla L_\epsilon)^2] = \epsilon \cdot \mathbb{E}[(\nabla L(D_{\text{full}}))^2] + (1-\epsilon) \cdot \mathbb{E}[(\nabla L(D_{\text{partial}}))^2]$$

This reflects that:

- With probability ϵ, the gradient is $\nabla L(D_{\text{full}})$, so the squared gradient is $(\nabla L(D_{\text{full}}))^2$.
- With probability $1-\epsilon$, the gradient is $\nabla L(D_{\text{partial}})$, so the squared gradient is $(\nabla L(D_{\text{partial}}))^2$.

Step 2: Relating Expectation of Squares to Variance

Recall that for any random variable X, the expectation of the square is:

$$\mathbb{E}[X^2] = \text{Var}(X) + (\mathbb{E}[X])^2$$

Applying this to the full and partial data gradients:

- For the full data gradient:

$$\mathbb{E}[(\nabla L(D_{\text{full}}))^2] = \text{Var}(\nabla L(D_{\text{full}})) + (\mathbb{E}[\nabla L(D_{\text{full}})])^2$$

  where $\text{Var}(\nabla L(D_{\text{full}})) = \sigma^2_{\text{full}}$.
- For the partial data gradient:

$$\mathbb{E}[(\nabla L(D_{\text{partial}}))^2] = \text{Var}(\nabla L(D_{\text{partial}})) + (\mathbb{E}[\nabla L(D_{\text{partial}})])^2$$

  where $\text{Var}(\nabla L(D_{\text{partial}})) = \sigma^2_{\text{partial}}$.

Substituting these into the expression for $\mathbb{E}[(\nabla L_\epsilon)^2]$:

$$\begin{aligned}\mathbb{E}[(\nabla L_\epsilon)^2] = \epsilon \cdot \left(\sigma^2_{\text{full}} + (\mathbb{E}[\nabla L(D_{\text{full}})])^2\right) \\ + (1-\epsilon) \cdot \left(\sigma^2_{\text{partial}} + (\mathbb{E}[\nabla L(D_{\text{partial}})])^2\right)\end{aligned}$$

STEP 3: SQUARED EXPECTED GRADIENT

From the bias analysis, the expected gradient is:

$$\mathbb{E}[\nabla L_\epsilon] = \epsilon \cdot \mathbb{E}[\nabla L(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]$$

Squaring this, we get:

$$\begin{aligned}(\mathbb{E}[\nabla L_\epsilon])^2 &= (\epsilon \cdot \mathbb{E}[\nabla L(D_{\text{full}})] + (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{partial}})])^2 \\ &= \epsilon^2 \cdot (\mathbb{E}[\nabla L(D_{\text{full}})])^2 \\ &\quad + (1-\epsilon)^2 \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})])^2 \\ &\quad + 2 \cdot \epsilon \cdot (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{full}})] \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]\end{aligned}$$

STEP 4: COMPUTE THE VARIANCE $\text{VAR}(\nabla L_\epsilon)$

Using the variance identity $\text{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$, we subtract the squared expected gradient from the expected squared gradient:

$$\text{Var}(\nabla L_\epsilon) = \mathbb{E}[(\nabla L_\epsilon)^2] - (\mathbb{E}[\nabla L_\epsilon])^2$$

Substituting the expressions:

$$\begin{aligned}\text{Var}(\nabla L_\epsilon) = &\left[\epsilon \cdot \left(\sigma^2_{\text{full}} + (\mathbb{E}[\nabla L(D_{\text{full}})])^2\right) + (1-\epsilon) \cdot \left(\sigma^2_{\text{partial}} + (\mathbb{E}[\nabla L(D_{\text{partial}})])^2\right)\right] \\ &- \left[\epsilon^2 \cdot (\mathbb{E}[\nabla L(D_{\text{full}})])^2 + (1-\epsilon)^2 \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})])^2\right. \\ &\left.+2 \cdot \epsilon \cdot (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{full}})] \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]\right]\end{aligned}$$

Grouping like terms:

- Terms involving $(\mathbb{E}[\nabla L(D_{\text{full}})])^2$:

$$\epsilon \cdot (\mathbb{E}[\nabla L(D_{\text{full}})])^2 - \epsilon^2 \cdot (\mathbb{E}[\nabla L(D_{\text{full}})])^2 = \epsilon \cdot (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{full}})])^2$$

- Terms involving $(\mathbb{E}[\nabla L(D_{\text{partial}})])^2$:

$$(1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})])^2 - (1-\epsilon)^2 \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})])^2 = (1-\epsilon) \cdot \epsilon \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})])^2$$

- Cross terms involving $\mathbb{E}[\nabla L(D_{\text{full}})] \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]$:

$$-2 \cdot \epsilon \cdot (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{full}})] \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]$$

- Variance terms:

$$\epsilon \cdot \sigma^2_{\text{full}} + (1-\epsilon) \cdot \sigma^2_{\text{partial}}$$

Combining these, the variance becomes:

$$\begin{aligned}\text{Var}(\nabla L_\epsilon) = \epsilon \cdot \sigma^2_{\text{full}} + (1-\epsilon) \cdot \sigma^2_{\text{partial}} \\ + \epsilon \cdot (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{full}})])^2 \\ + (1-\epsilon) \cdot \epsilon \cdot (\mathbb{E}[\nabla L(D_{\text{partial}})])^2 \\ - 2 \cdot \epsilon \cdot (1-\epsilon) \cdot \mathbb{E}[\nabla L(D_{\text{full}})] \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]\end{aligned}$$

The last three terms can be factored as a square:

$$\epsilon \cdot (1-\epsilon) \cdot \left[(\mathbb{E}[\nabla L(D_{\text{full}})])^2 + (\mathbb{E}[\nabla L(D_{\text{partial}})])^2 - 2 \cdot \mathbb{E}[\nabla L(D_{\text{full}})] \cdot \mathbb{E}[\nabla L(D_{\text{partial}})]\right]$$

This simplifies to:

$$\epsilon \cdot (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{full}})] - \mathbb{E}[\nabla L(D_{\text{partial}})])^2$$

Thus, the final expression for the variance is:

$$\begin{aligned}\text{Var}(\nabla L_\epsilon) = \epsilon \cdot \sigma^2_{\text{full}} + (1-\epsilon) \cdot \sigma^2_{\text{partial}} \\ + \epsilon \cdot (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{full}})] - \mathbb{E}[\nabla L(D_{\text{partial}})])^2\end{aligned}$$

INTERPRETATION OF VARIANCE COMPONENTS

- The first two terms represent the weighted average of the variances: $\epsilon \cdot \sigma^2_{\text{full}} + (1-\epsilon) \cdot \sigma^2_{\text{partial}}$. Since $\sigma^2_{\text{full}} < \sigma^2_{\text{partial}}$, this weighted average is less than $\sigma^2_{\text{partial}}$ for any $\epsilon > 0$:

$$\epsilon \cdot \sigma^2_{\text{full}} + (1-\epsilon) \cdot \sigma^2_{\text{partial}} = \sigma^2_{\text{partial}} - \epsilon \cdot (\sigma^2_{\text{partial}} - \sigma^2_{\text{full}}) < \sigma^2_{\text{partial}}$$

  because $\sigma^2_{\text{partial}} - \sigma^2_{\text{full}} > 0$.

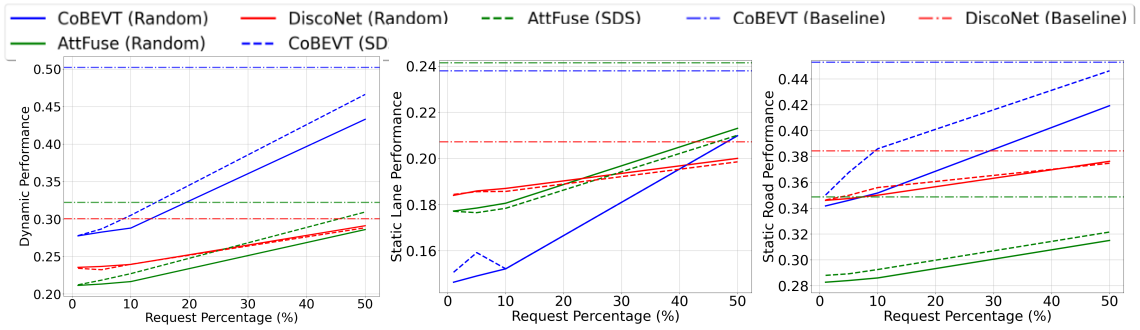
- The third term, $\epsilon \cdot (1-\epsilon) \cdot (\mathbb{E}[\nabla L(D_{\text{full}})] - \mathbb{E}[\nabla L(D_{\text{partial}})])^2$, accounts for additional variance due to the difference in expected gradients (bias between full and partial data). If the bias is small (i.e., the expected gradients are similar), this term is negligible. Even if there is some bias, for small to moderate values of ϵ, the reduction in the weighted variance often dominates, leading to an overall variance lower than $\sigma^2_{\text{partial}}$.
- Therefore, $\text{Var}(\nabla L_\epsilon) < \sigma^2_{\text{partial}}$ under typical conditions, indicating that the ϵ-greedy strategy reduces variance compared to using only partial data.

4. COMBINED EFFECT ON TRAINING

- **Bias Reduction**: As shown, the bias of the gradient estimator is reduced by a factor of $(1-\epsilon)$ compared to using only partial data, pulling the expected gradient closer to the true gradient $\mathbb{E}[\nabla L(D_{\text{full}})]$. This improves the accuracy of the optimization direction, leading to better convergence toward the true optimum of L.
- **Variance Reduction**: The variance of the gradient estimator is reduced due to the incorporation of lower-variance full data gradients with probability ϵ, leading to more stable updates and smoother training trajectories. This reduces the risk of divergence or oscillation during optimization.
- Under perfect transmission, with no external noise, the ϵ-greedy strategy effectively leverages the strengths of full data (lower bias and variance) while maintaining efficiency by using partial data most of the time, achieving a favorable trade-off.

Future Work in Epsilon-Greedy Training Strategy: Future work would focus on **modeling the impact of imperfect transmission in the epsilon-greedy strategy**, where data (full or partial) may be corrupted, delayed, or lost during communication. This requires developing a network model that captures factors such as packet loss, latency, and bandwidth constraints. By incorporating these elements, we can analyze how transmission imperfections affect the bias and variance of the gradient estimator ∇L_ϵ.

A.3 Preliminary Experiments for Uncertainty based Selection on Segmentation Task



(a) Dynamic (b) Static Lane (c) Static Road

Figure 9: Motivation for CooperTrim. Comparison of baseline, random channel selection and SD-based selection (SDS) for uncertainty guidance show uncertainty-guided selection often outperforms random selection, though baseline performance remains higher consistently. CooperTrim addresses this accuracy gap through its proposed adaptive uncertainty driven selection method.

A.4 "Trimming" Cooperative Segmentation Baselines - Additional

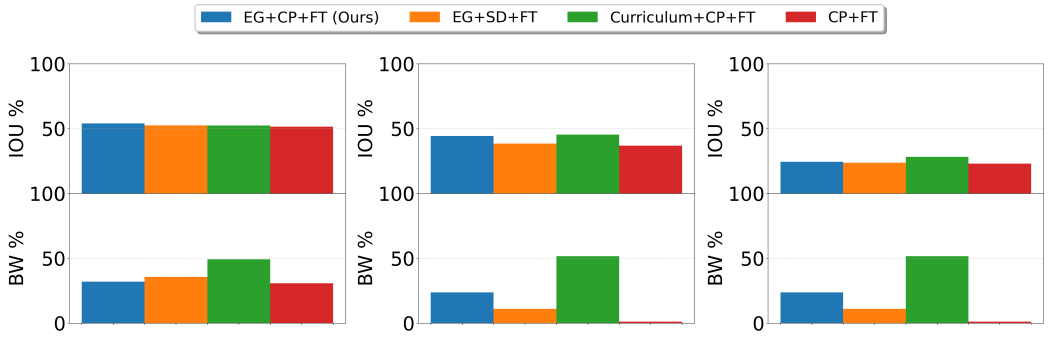
Additional presentation for Section 4 Cooperative Segmentation Baseline experiments in Table 4. CooperTrim maintains comparable performance accuracies while reducing bandwidth significantly, achieving an average 80.28% improvement in network overhead over the baselines.

Table 4: Application of CooperTrim to Existing Cooperative Segmentation Methods. "Baseline" and "Ours" (CooperTrim) show the respective IOU %, while "BW" shows CooperTrim bandwidth usage.

	Dynamic			Static Lane			Static Road			Avg.
Methods	**Baseline**	**Ours**	**BW**	**Baseline**	**Ours**	**BW**	**Baseline**	**Ours**	**BW**	**BW (Mbps)**
CoBEVT	50.23	54.03	32.04%	23.79	24.45	23.77%	45.28	44.38	23.77%	11.16
AttFuse	32.20	30.90	24.76%	24.14	23.93	17.39%	34.86	36.22	17.39%	8.4
DiscoNet	30.03	30.80	10.65%	20.72	22.05	9.72%	38.43	40.02	9.72%	4.07

A.5 Training Methods Comparison - Additional

Corresponding to Section 4 Training Methods Comparison, Table 2 presents the training methods comparative study for training CooperTrim, all using the loss function from Section 3.3.



(a) Dynamic (b) Static Road (c) Static Lane

Figure 10: Training Methods Comparison. CooperTrim balances task performance and network overhead better than other baselines.

A.6 Robustness of Threshold-Based Method in CooperTrim

In CooperTrim, the decision to request data is based on a combination of two key factors: the feature representation (which encodes the scene description as perceived by the ego vehicle in the current frame) and the uncertainty estimation (which compares the past comprehensive scene description with the current ego vehicle's scene understanding to detect discrepancies, e.g., objects that existed previously but are no longer present in ego's perception range, or vice versa). This dual mechanism ensures robustness even in subtle change scenarios.

For instance, in a situation where the overall scene appears static (low temporal change), if the ego vehicle's current feature perception changes—due to factors like occlusion or the sudden appearance of a small object such as a pedestrian—our method will detect this deviation. As a result, the system will trigger more data requests to ensure critical information is not missed. Conversely, if both the scene and the ego vehicle's current perception remain largely unchanged, the system will minimize data requests, optimizing bandwidth usage.

To validate this behavior, we analyzed specific frame sequences in our experiments.

- **Frames 1960-1970**: These frames exhibit significant changes in the scene, associated with high data requests (Figure 7). The changes were verified using 4 camera images and ground truth dynamic data segmentation, confirming the system's responsiveness to dynamic scenarios, as shown in Figure 11.

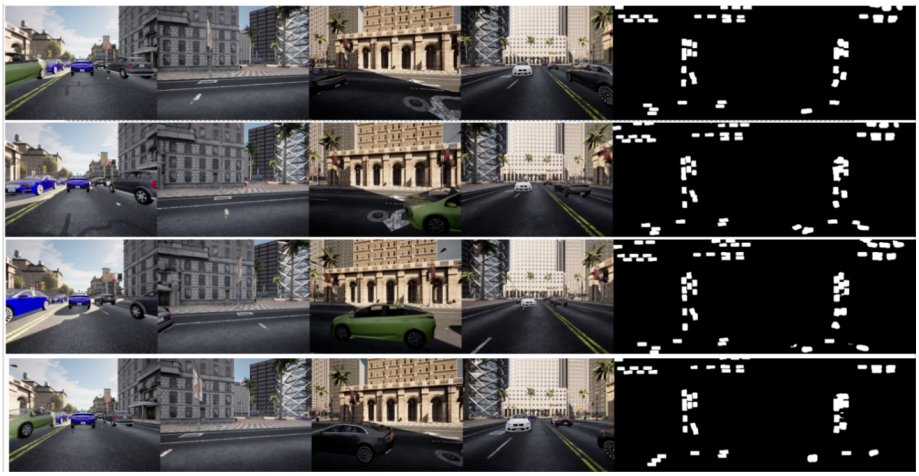


Figure 11: High data requests during significant scene changes in Frames 1960-1970

- **Frames 940-950**: These frames show minimal changes, associated with low data requests (Figure 7), as shown in Figure 11. The lack of significant changes was similarly verified using 4 camera images and ground truth dynamic data segmentation, demonstrating the system's ability to conserve bandwidth in stable conditions.

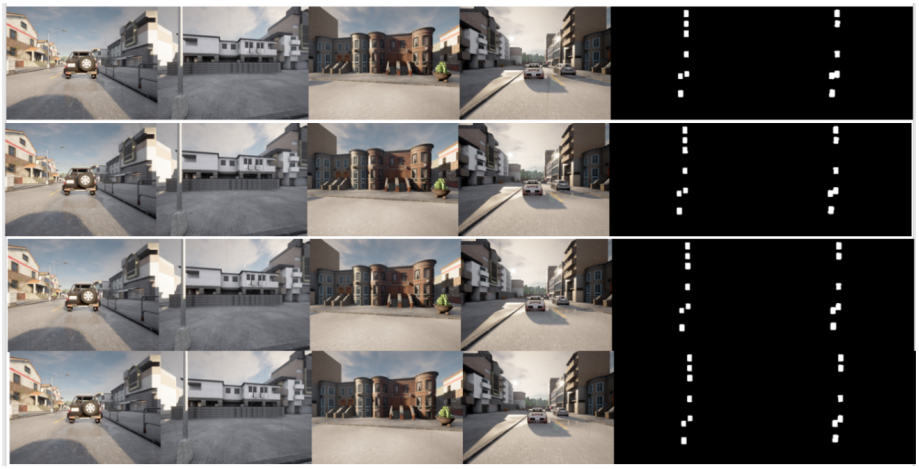


Figure 12: Low data requests during minimal scene changes in Frames 940-950

A.7 Network Performance Metrics Under Realistic Conditions

We evaluated CooperTrim's performance by measuring key network-level metrics, including end-to-end latency and packet loss/retransmissions under loss rates of [0–10]%.

A.7.1 Loss Rate and Impact

During training, we assumed perfect transmission with no loss. However, during inference, loss rates were simulated randomly between [0, 10]% by applying masks to evaluate robustness. The impact on performance metrics such as IoU and bandwidth (BW) under dynamic and static configurations is summarized in Table 5.

Table 5: Impact of Loss Rate on IoU and Bandwidth Metrics

Loss Rate	Dynamic IoU (%)	Dynamic BW (%)	Static Lane IoU (%)	Static Road IoU (%)	Static BW (%)
0%	54.03	32.04	24.45	44.38	23.77
10%	53.95	32.11	24.49	44.34	23.71

Under simulated loss rates of 0–10%, CooperTrim maintains stable IoU and bandwidth metrics, with Dynamic IoU being comparable at 10% loss. These results highlight CooperTrim's robust fallback behavior under packet loss conditions up to 10%, demonstrating its potential for reliable operation in real-world scenarios with imperfect network conditions.

A.8 LLM Usage

In the preparation of this paper, Large Language Models (LLMs) were utilized as a supportive tool for polishing the writing and implementing minor code modifications at specific points. However, the core research ideas, code design, and overall framework are entirely our own. The LLMs played no role in the ideation or conceptualization of the research.