RESBev: Making BEV Perception More Robust

Lifeng Zhuo **Kefan Jin** **Zhe Liu*** **Hesheng Wang***
Shanghai Jiao Tong University

Abstract

Bird's-eye-view (BEV) perception has emerged as a cornerstone of autonomous driving systems, providing a structured, ego-centric representation critical for downstream planning and control. However, real-world deployment faces challenges from sensor degradation and adversarial attacks, which can cause severe perceptual anomalies and ultimately compromise the safety of autonomous driving systems. To address this, we propose a resilient and plug-and-play BEV perception method (RESBev), which can be easily applied to existing BEV perception methods to enhance their robustness to diverse disturbances. Specifically, we reframe perception robustness as a latent semantic prediction problem. A latent dynamic predictor is constructed to extract spatiotemporal correlations across sequential BEV observations, thereby learning the underlying BEV state transitions to predict clean BEV features for reconstructing corrupted observations. The proposed framework operates at the semantic feature level of the BEV perception pipeline, enabling recovery that generalizes across both natural disturbances and adversarial attacks without modifying the underlying backbone. Extensive nuScenes experiments show RESBev notably enhances BEV perception robustness against natural disturbances and adversarial attacks, with matched corruption-trained temporal baselines separating the contribution of predictive reconstruction from training augmentation. Latency, parameter, and memory analyses quantify its deployment overhead.

1 Introduction

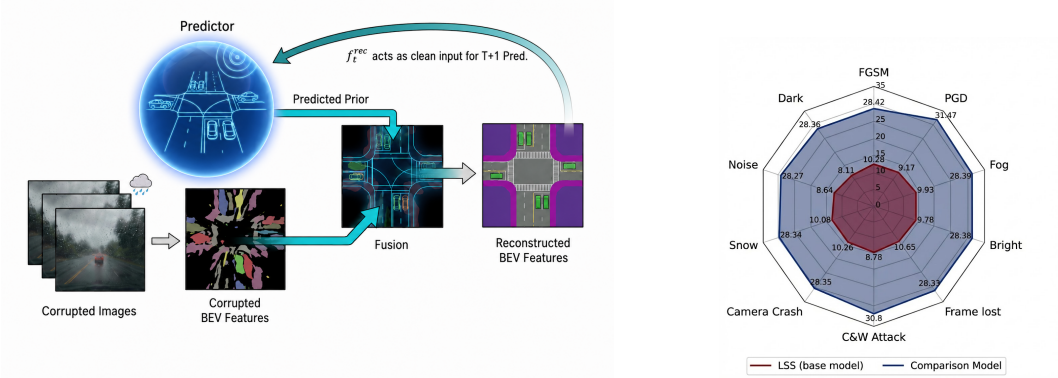
The bird's-eye-view (BEV) representation has become a foundation of modern autonomous driving perception systems. By transforming multi-camera inputs into a unified top-down representation, it resolves projective ambiguities and provides geometrically consistent scene understanding essential for downstream tasks. Lift-Splat-Shoot (LSS) [13] introduces a foundational framework for this transformation, pioneering camera-centric 3D perception. Building upon this paradigm, numerous BEV perception models have been proposed. For instance, BEVFusion [11] integrates camera features with LiDAR signals to improve geometric fidelity and robustness. Meanwhile, BEVFormer [10] leverages temporal self-attention to aggregate BEV features from historical frames, enabling the model to maintain persistent object tracks and handle ego-motion across frames.

Despite strong benchmark performance on datasets such as nuScenes [1], BEV models remain vulnerable to real-world corruptions and adversarial perturbations. Adverse conditions such as fog, darkness, snow, camera failures, and frame loss can severely degrade semantic understanding, while adversarial attacks such as FGSM [4], PGD [12], and C&W [2] can cause significant performance degradation even with small input perturbations. Benchmarks such as RoboBEV [21] have systematically exposed this fragility.

Existing robustness strategies provide partial solutions but still have limitations. Multi-modal fusion improves reliability by incorporating additional sensors, but it depends on extra sensor modalities and their reliability. Temporal aggregation methods, such as BEVFormer [10], exploit historical

*Corresponding authors: Zhe Liu and Hesheng Wang.

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



(a) RESBev framework.

(b) Robustness comparison.

Figure 1: Overview and robustness comparison of RESBev. **(a)** RESBev predicts a BEV prior from historical BEV features and ego-motion, and fuses it with corrupted observations to reconstruct the BEV representation. **(b)** System-level comparison with the official clean-trained LSS checkpoint; RESBev additionally uses temporal context and corruption-aware training.

information but may still propagate corrupted observations under persistent disturbances. Recent BEV refinement methods, such as DiffBEV [25] and BEVDiffuser [22], improve robustness by denoising corrupted BEV features, but some diffusion-based designs require iterative sampling, while training-time approaches such as BEVDiffuser avoid a test-time diffusion branch. These limitations motivate a lightweight and generalizable recovery mechanism that can exploit temporal structure without expensive iterative denoising.

Based on this insight, we introduce RESBev which incorporates latent dynamic predictor to learn the transition of BEV states over time (see Figure 1a). By modeling how BEV states evolve over time, RESBev can reconstruct corrupted features through temporal prior prediction rather than direct temporal aggregation. Extensive experiments demonstrate that RESBev can enhance the robustness of existing BEV models when facing various corruptions (see Figure 1b). Our contributions are summarized below:

1. We conduct a systematic analysis of robustness in BEV perception pipelines, revealing that effective recovery requires modeling in the BEV semantic space, preserving high-dimensional features before task compression, and leveraging temporal prior prediction.
2. Based on these insights, we propose a plug-and-play model that introduces a latent dynamic predictor to capture temporal dynamics in BEV semantic space and generate predictive semantic prior in complex traffic environments.
3. Extensive experiments demonstrate that the proposed method can effectively enhance the robustness of existing BEV perception methods against various disturbances and improve their generalization to unseen types of perception anomalies.

2 Related Work

2.1 BEV Perception

In autonomous driving, bird's-eye-view (BEV) representation provides a unified environmental understanding necessary for downstream tasks. Lift-Splat-Shoot (LSS) [13] establishes BEV representation by projecting multi-view 2D image features into a common BEV grid through explicit depth modeling. Building upon this paradigm, numerous BEV perception models have been proposed. For instance, GaussianLSS [16] refines depth estimation by incorporating Gaussian distributions to model uncertainty, while BEVFusion [11] integrates multi-modal sensor data to produce richer BEV representations. FIERY [6] extends the BEV framework to support multiple downstream tasks and leverages multi-frame temporal information within a unified architecture. Another line of work adopts Transformer-based architectures. For example, BEVFormer [10] introduces spatiotemporal

transformers to aggregate features from historical frames and multiple camera views. Despite achieving strong performance on benchmarks such as nuScenes [1], these methods remain vulnerable to various real-world anomalies.

Prediction-based temporal detection. P2D [7] predicts current object-information maps from past frames and uses the predicted regions to guide temporal BEV feature aggregation. HoP [24] reconstructs historical BEV features through an auxiliary prediction objective during training; its auxiliary branch is removed at inference. These methods establish temporal prediction as a useful representation-learning tool. RESBev instead retains dense semantic prediction at inference for corruption recovery: a prior predicted independently of the current observation queries its features through gated cross-attention. This gate learns how much current evidence to incorporate; it does not guarantee rejection of every anomaly. We evaluate RESBev on top of both detectors under matched corruption-aware training in Section 5.3.

2.2 Perception Anomalies

Deep-learning-based perception systems for autonomous driving are highly vulnerable to various anomalies, including adversarial attacks and natural corruptions. Adversarial attacks involve crafting imperceptible perturbations to input data that induce incorrect predictions. While classic digital attacks such as FGSM [4], PGD [12], and C&W [2] were initially developed for 2D classification, recent studies demonstrate their degradation effect on BEV models [23, 20]. Beyond these, recent works have introduced targeted attacks for BEV perception, such as semantic perturbations through object swapping [19] and multi-view adversarial attacks like "Fool the Hydra"[18], which exploits vulnerabilities in cross-camera feature alignment. In addition, 3D adversarial examples using techniques like NeRF (e.g., Adv3D [9]) create physically plausible perturbations. These attacks lead to perception failures such as missed detections or false positives. Beyond targeted attacks, real-world scenarios present diverse natural corruptions, including adverse weather (e.g., fog, dark, snow) and sensor failures (e.g., camera crash, frame loss).

2.3 Robust Perception

Robustness benchmarks [21] and adversarial studies [23, 20] expose failures in BEV perception. Multi-sensor fusion exploits complementary observations [14, 8], but depends on additional sensors. Temporal aggregation [10] can stabilize representations while also propagating corruption. DiffBEV [25] uses conditional diffusion to refine BEV features; BEVDiffuser [22] uses ground-truth guidance for training-time denoising without a test-time diffusion branch. RESBev takes a complementary approach: it predicts a dense semantic prior from historical reconstructed states and ego-motion, then selectively retrieves current evidence through gated cross-attention. It performs one prediction/reconstruction step at inference and attaches to existing dense BEV backbones.

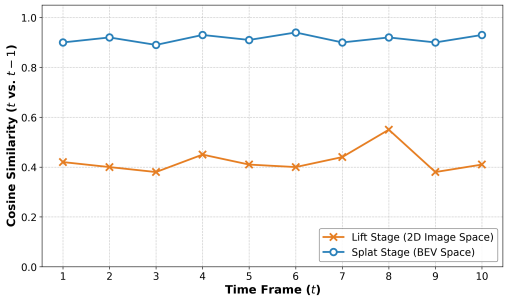
3 Analysis: Design Principles for Robust BEV Recovery

Robust BEV perception often degrades under real-world anomalies such as adverse weather, sensor corruption, and adversarial perturbations. Before introducing our method, we conduct diagnostic experiments to answer two questions: *where* corrupted representations should be recovered, and *how* temporal information should be used for recovery. This section is not intended to compare final systems, but to derive the design principles behind RESBev.

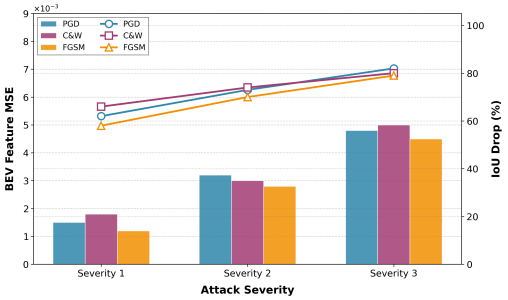
Diagnostic protocol. We use the LSS [13] as a diagnostic testbed because it explicitly separates perspective-view feature extraction, BEV feature construction, and task prediction. This does not restrict our method to LSS-based models; the conclusion applies to any architecture with dense BEV features, such as BEVFormer.

3.1 Where should corrupted representations be recovered?

The LSS pipeline contains three representative stages: perspective-view features after image encoding, BEV semantic features after view transformation, and task outputs after the prediction head. We compare recovery at the Lift, Splat, and Shoot stages under the same corruption setting.



(a) **Cross-frame stability.** BEV features remain more stable under persistent noise corruptions.



(b) **Feature brittleness.** Small feature perturbations can cause large task drops.

Figure 2: Diagnostic analysis for robust BEV recovery. **(a)** BEV features show stronger temporal stability than perspective-view features. **(b)** Small feature deviations can still cause severe performance degradation, motivating predictive recovery over simple aggregation.

Table 1: BEV vehicle segmentation: recovery-stage diagnosis (IoU) under noise corruption.

Stage	**Rep.**	**Feature Form**	**IoU**
Lift	Perspective	$C \times H_{img} \times W_{img}$	16.4
Shoot	Task output	$N_{cls} \times X \times Y$	18.7
Splat	**BEV feature**	$C \times X \times Y$	**31.6**

Figure 2a and Table 1 show that BEV semantic features provide the most suitable recovery space. Perspective-view features are sensitive to image noise, viewpoint changes, and ego-motion, while task outputs are too compressed to retain the information needed for reconstruction. In contrast, Splat-stage BEV features remain high-dimensional, spatially structured, and temporally aligned, making them better suited for robust recovery.

3.2 Is temporal aggregation sufficient?

We next examine whether temporal aggregation alone can recover corrupted BEV features. All variants are implemented on the same Splat-stage BEV features using the same segmentation baseline and corruption setting as Table 1. The temporal attention variant follows the BEVFormer-style temporal attention design [10], while the recurrent fusion variant follows the recurrent temporal fusion principle of RecurrentBEV [3]. They are used as diagnostic variants rather than full system comparisons.

Table 2 shows that temporal context helps, but direct aggregation remains limited because it still incorporates corrupted current features and may propagate anomalies. This is particularly critical under adversarial perturbations, where minor feature deviations may lead to substantial performance loss, as shown in Figure 2b. In contrast, the predictive-prior variant estimates the current BEV state from historical BEV features and ego-motion, providing a cleaner reference for robust reconstruction.

Table 2: BEV vehicle segmentation: temporal-mechanism diagnosis (IoU). The current-frame column describes input to the temporal module, not the subsequent reconstructor; the predictive-prior variant uses the current observation for reconstruction.

Variant	**Hist.**	**Curr. in temporal module**	**Prior**	**IoU**
Single frame	No	Yes	No	15.53
Temporal attn.	Yes	Yes	No	20.17
Recurrent fusion	Yes	Yes	No	22.31
Predictive prior	Yes	No	Yes	**30.11**

Design implications. These findings lead to two design requirements. First, recovery should be performed in dense BEV semantic feature space. Second, historical information should be used to pre-

dict a temporal prior rather than being directly aggregated with corrupted current observations. These requirements motivate RESBev, which implements prior prediction and prior-guided reconstruction in the BEV feature space.

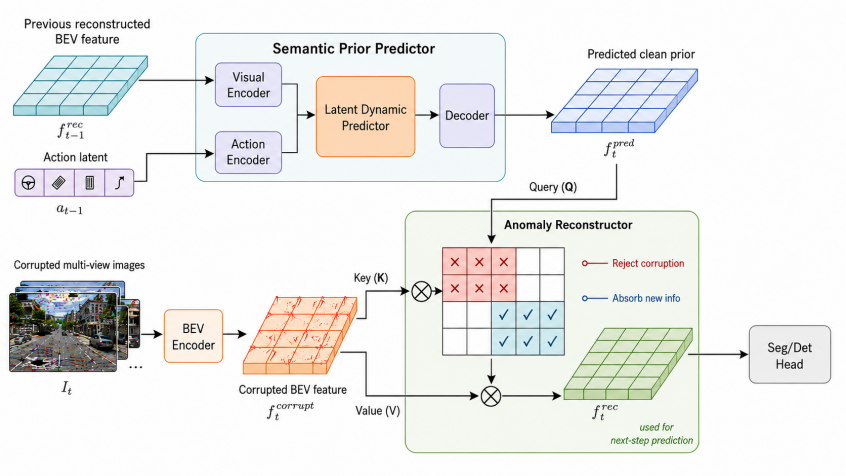


Figure 3: RESBev at inference. The Semantic Prior Predictor reads the historical BEV state and ego-motion, without the current observation. The Anomaly Reconstructor uses its prediction as query and the current BEV feature as key/value; a learned gate controls the residual update. The reconstructed feature drives the task head and becomes history for the next recursive step.

4 Method

4.1 Overall Framework

RESBev consists of two modules, as shown in Figure 3: Semantic Prior Predictor and Anomaly Reconstructor. Given the previous reconstructed BEV feature and ego-motion, the Semantic Prior Predictor predicts a BEV prior for current timestamp. The Anomaly Reconstructor then uses this predicted prior as a query to extract reliable information from current corrupted BEV feature through cross-attention. The final reconstructed feature is obtained by combining the predicted prior with the retrieved current-frame information through a learnable gate. This feature is then fed into the downstream task head.

4.2 Semantic Prior Predictor

Given historical BEV features f_{t-1}^{rec} and ego-motion a_{t-1} (translation and rotation), visual and action encoders E_{vis} and E_{act} produce compact latents. Their concatenation is processed by the Transformer-based LDP; decoder D restores the predicted dense prior f_t^{pred}. The visual encoder downsamples a 200×200 BEV grid into 625 visual tokens of dimension 256. A two-layer MLP maps the 15-dimensional ego-motion vector to an action token. The LDP uses four Transformer blocks with eight attention heads, and a convolutional decoder restores the dense BEV grid. The prior is formulated as:

$$f_t^{pred} = D\Big(\text{LDP}\big(\text{Concat}(E_{vis}(f_{t-1}^{rec}), E_{act}(a_{t-1}))\big)\Big). \tag{1}$$

4.3 Anomaly Reconstructor

While the latent dynamic predictor provides a historical prior, it cannot predict sudden stochastic events, such as a car unexpectedly driving into the scenario. To update this prediction with current evidence, we introduce a temporal cross attention module. Specifically, given the current multi-view images at time t, which may suffer from various anomalies, we process them through the BEV encoder to obtain the current corrupted BEV representation, denoted as $f_t^{corrupt}$.

To enhance temporal stability and provide a reliable reference for reconstruction, we formulate this process as a query-driven extraction that uses the predicted prior to retrieve valid information from the current corrupted observation. Specifically, the predicted feature f_t^{pred} acts as the query (Q), while the current corrupted BEV feature $f_t^{corrupt}$ serves as both the Key (K) and Value (V). This design anchors retrieval on the historical prediction rather than a corrupted current query, while allowing the retrieved values to contain new scene information. To integrate the retrieved information without destabilizing the prior, we employ a dynamically gated residual connection:

$$f_t^{rec} = f_t^{pred} + \alpha \cdot \mathrm{CrossAttn}\left(f_t^{pred}, f_t^{corrupt}\right), \tag{2}$$

where $\alpha \in [0,1]^C$ denotes channel-wise coefficients produced by a two-layer MLP with sigmoid activation from the attention features (broadcast spatially). Cross-attention uses hidden dimension 256, eight heads, and learned 2D positional embeddings. The residual can incorporate newly appearing objects absent from the prior; reliable current evidence is still needed. Section 5.5 evaluates this behavior.

4.4 Training Objective

We train RESBev with two-frame multi-view inputs from the same driving scene. At each timestamp, the input contains six camera views, denoted as $\mathbf{I}_t = \{I_t^v\}_{v=1}^6$. For each pair $(\mathbf{I}_{t-1}, \mathbf{I}_t)$, the current input is always corrupted, while the historical input is randomly kept clean or corrupted. This exposes the model to both clean-start and corrupted-start conditions, while the clean current feature is used only as supervision.

The corrupted current input is:

$$\mathbf{I}_t^{corrupt} = \mathcal{C}_t(\mathbf{I}_t), \tag{3}$$

where $\mathcal{C}_t(\cdot)$ denotes a randomly sampled corruption. The historical input is sampled as:

$$\mathbf{I}_{t-1}^{hist} = \begin{cases} \mathbf{I}_{t-1}, & m = 1, \\ \mathcal{C}_{t-1}(\mathbf{I}_{t-1}), & m = 0, \end{cases} \qquad m \sim \mathrm{Bernoulli}(p_{hist}), \tag{4}$$

where p_{hist} controls the probability of using a clean historical input.

The BEV encoder extracts:

$$f_{t-1}^{hist} = B(\mathbf{I}_{t-1}^{hist}), \quad f_t^{corrupt} = B(\mathbf{I}_t^{corrupt}), \quad f_t^{clean} = B(\mathbf{I}_t), \tag{5}$$

where $B(\cdot)$ denotes the backbone and view transformation module.

Given f_{t-1}^{hist} and ego-motion a_{t-1}, the Latent Dynamic Predictor estimates the current BEV prior, the Anomaly Reconstructor recovers the current BEV feature, and the clean BEV feature is used as the stop-gradient target:

$$f_t^{pred} = \mathcal{P}_\theta(f_{t-1}^{hist}, a_{t-1}), \quad f_t^{rec} = \mathcal{R}_\phi(f_t^{pred}, f_t^{corrupt}), \quad f_t^{target} = \mathrm{sg}(f_t^{clean}). \tag{6}$$

We optimize the prior prediction loss, reconstruction loss, and task loss:

$$\mathcal{L}_p = \mathrm{MSE}(f_t^{pred}, f_t^{target}), \quad \mathcal{L}_r = \mathrm{MSE}(f_t^{rec}, f_t^{target}), \quad \mathcal{L}_{task} = \ell_{task}(H(f_t^{rec}), y_t). \tag{7}$$

Here, $\mathcal{L}_p$, $\mathcal{L}_r$, and $\mathcal{L}_{task}$ denote the prior prediction, feature reconstruction, and downstream task losses, respectively.

The final objective is:

$$\mathcal{L} = \mathcal{L}_{task} + \lambda_p \mathcal{L}_p + \lambda_r \mathcal{L}_r. \tag{8}$$

During inference, only one forward pass is required. For single-step inference, the LDP uses the historical BEV feature extracted from the previous multi-view input. For recursive long-horizon inference, the previous reconstructed feature f_{t-1}^{rec} is used as the historical BEV state for predicting the next prior.

5 Experiments

This section validates the effectiveness of RESBev in improving robust BEV perception. We first measure performance on benchmark corruptions, then assess generalization to unseen corruptions. We then provide component ablations, long-horizon robustness analysis, and deployment cost measurements.

5.1 Experimental Setup

5.1.1 Dataset and Baseline Models.

Our experiments are conducted on the widely used nuScenes dataset [1]. We evaluate our method on the task of bird's-eye-view (BEV) semantic segmentation and 3D object detection. To validate the performance of our RESBev model, we integrate it into three LSS-based models: LSS [13], SimpleBEV [5], GaussianLSS [16], and one attention-based model: BEVFormer [10]. To further benchmark our method against state-of-the-art robust perception frameworks, we also compare it with BEVDiffuser [22] and DiffBEV [25].

5.1.2 Robustness Evaluation Protocol.

We use a RoboBEV-inspired protocol [21] with ten corruption types at three severity levels. Our suite differs from the original benchmark: it omits Color Quantization and Motion Blur and adds Gaussian noise and three adversarial attacks. Thus, its absolute scores and normalized robustness metrics are specific to this suite. Due to space limitations, detailed specifications for each corruption severity level are provided in the supplementary material. These corruptions are categorized as follows:

- Natural Corruptions: We simulate diverse environmental conditions and sensor failures, including fog, snow, bright, dark, camera crash, frame loss, and noise.
- Adversarial Attacks: To measure resilience against malicious inputs, we employ three attacks designed for BEV models: Fast Gradient Sign Method (FGSM) [4], Projected Gradient Descent (PGD) [12], and Carlini & Wagner (C&W) [2].

5.1.3 Implementation Details and Metrics.

We report vehicle IoU for BEV segmentation and NDS (nuScenes Detection Score) for 3D detection, supplemented by per-category IoU and detection mAP. Metrics are reported in percentage points unless otherwise stated. Corrupted scores average over three severities; overall corrupted averages exclude clean inputs. Our method is implemented using PyTorch. All models are trained on a NVIDIA A100-SXM4-80GB GPU with a batch size of 16.

Training and comparison protocols. The vanilla LSS, SimpleBEV, GaussianLSS, and BEVFormer entries use official clean-trained checkpoints. RESBev is trained with FGSM, PGD, Fog, Bright, and Frame Lost; C&W, Snow, Camera Crash, Noise, and Dark are held out. The original system comparisons therefore combine architectural, temporal-input, and training changes and do not isolate the module's contribution. In particular, LSS, SimpleBEV, and GaussianLSS are single-frame baselines, whereas RESBev additionally uses history and ego-motion. We retain these comparisons in Appendix D.1. The matched BEVFormer experiment below controls training and temporal inputs; P2D/HoP additionally test compatibility with predictive detectors under matched corruption-aware training. Attacks are regenerated against each complete evaluated model, including RESBev when present. We plan to release code, configurations, and checkpoints.

5.2 Controlled Robustness Evaluation

We retrain BEVFormer with corruption augmentation (+Corr.Aug) using the same five training corruptions, severity distribution, sampling procedure, and number of input frames as BEVFormer+RESBev. Thus, this comparison controls both corruption-aware training and temporal context. Table 3 shows that RESBev improves the reported seen/unseen averages from 27.62/28.90 to 38.82/39.48 IoU, respectively. The clean score is 47.01 versus 45.71 for +Corr.Aug. These results support the contribution of prior-guided reconstruction beyond augmentation and temporal inputs. The diagnostic

Table 3: BEV vehicle segmentation (IoU): controlled BEVFormer comparison. Vanilla is clean-trained; +Corr.Aug and +RESBev share the five-corruption training protocol and temporal inputs. Seen/unseen denote membership in that training set. Averages exclude clean inputs.

Split	Corruption	Vanilla	+Corr.Aug	+RESBev
–	Clean	46.70	45.71	**47.01**
Seen	FGSM	21.54	26.31	38.78
	PGD	18.68	23.42	38.90
	Fog	25.34	29.50	38.91
	Bright	25.70	29.37	38.74
	Frame Lost	25.86	29.49	38.75
	Average	23.42	27.62	**38.82**
Unseen	C&W	17.56	22.79	39.67
	Snow	28.11	33.45	38.91
	Camera Crash	27.46	29.74	39.72
	Noise	21.55	25.93	39.34
	Dark	28.06	32.61	39.75
	Average	24.55	28.90	**39.48**

Table 4: 3D detection (NDS/mAP): corruption-trained P2D and HoP with and without RESBev. Both columns within each detector use the same FGSM/Fog training protocol; corrupted rows average three severities.

	P2D		HoP	
	+Corr.Aug	+RESBev	+Corr.Aug	+RESBev
Clean	47.05/35.61	46.94/36.04	55.46/44.92	55.94/45.91
FGSM	31.82/22.89	41.37/32.40	38.22/32.17	51.19/40.79
Fog	35.48/24.15	41.82/33.04	43.18/35.28	51.35/41.41
Seen avg.	33.65/23.52	41.60/32.72	40.70/33.73	51.27/41.10
C&W	29.13/19.69	41.25/31.76	37.47/28.18	51.21/40.42
Snow	33.72/25.19	41.38/32.85	41.05/34.61	51.37/40.88
Unseen avg.	31.43/22.44	41.32/32.31	39.26/31.40	51.29/40.65

temporal-attention and recurrent-fusion variants in Table 2 provide complementary comparisons of temporal mechanisms.

5.3 Prediction-Based Temporal Detectors

We evaluate P2D [7] with BEVDepth/ResNet-50 and HoP [24] with BEVFormer-opt/ResNet-101-DCN. Within each detector, both the baseline and +RESBev are trained on FGSM and Fog using the same severity distribution and sampling procedure; C&W and Snow are unseen. This two-corruption protocol differs from the five-corruption BEVFormer experiment above. Table 4 reports NDS and mAP: RESBev improves both metrics on seen and unseen corruptions while retaining comparable clean performance. Predictive temporal detection and test-time prior-guided recovery are therefore complementary.

Additional metrics. The BEVFormer system comparison also improves detection mAP from 19.63/23.72 to 38.27/38.41 under FGSM/Fog and from 15.43/22.65 to 38.56/38.97 under C&W/Snow. Segmentation gains extend to Road and Lane: corrupted IoU increases from 50.24/8.72 to 65.89/19.61. These compare clean-trained vanilla against corruption-trained RESBev; full per-category results and this qualification are given in Appendix D.1.

5.4 Ablation Study

Table 5 evaluates the contributions of the Latent Dynamic Predictor and Anomaly Reconstructor. Using the predictor alone provides a strong temporal prior, while adding the reconstructor consistently

Table 5: BEV vehicle segmentation: component ablation (vehicle IoU). SBEV denotes SimpleBEV and GLSS denotes GaussianLSS. All results are average vehicle IoU over five seen corruptions, where each sample is evaluated under a randomly selected corruption type and severity. Gain is computed over the Predictor-only baseline.

Config.	**LSS**	**SBEV**	**GLSS**	**BEVFormer**
Predictor-only	26.67	30.11	29.16	35.48
Predictor + Reconstructor	29.02	32.80	31.59	38.82
Gain	**↑8.8%**	**↑8.9%**	**↑8.3%**	**↑9.4%**

Table 6: BEV vehicle segmentation: design ablations on LSS (IoU averaged over three fog severities). Fixed residual gating retains cross-attention; fixed averaging directly combines prior and current features.

Configuration	IoU
Full RESBev	**28.39**
LDP replaced by U-Net	21.04
LDP replaced by stacked convolutions	16.34
Reconstructor replaced by concatenation + convolution	21.18
Reconstructor replaced by $0.5f_t^{pred} + 0.5f_t^{corrupt}$	19.71
Without ego-motion encoder	20.14
Fixed residual gate ($\alpha = 0.5$)	21.80

improves IoU across all baselines. This shows that prior-guided reconstruction is important for incorporating valid current-frame information while suppressing corruptions.

Design ablations. Table 6 isolates the predictor, fusion rule, ego-motion encoder, and dynamic gate on LSS under fog. All unmodified components are retained. Replacing the LDP with a U-Net or stacked convolutions reduces IoU by 7.35 or 12.05 points. Concatenation followed by convolution and fixed averaging also underperform gated cross-attention. Removing ego-motion or fixing the residual gate at 0.5 reduces IoU by 8.25 or 6.59 points. These results support the evaluated designs; they do not establish insensitivity to loss weights or history length, which are not swept here.

5.5 Sudden Objects and Temporal Smoothness

We collect 100 three-frame clips containing 183 objects present at t but absent at both $t-1$ and $t-2$. Using LSS under fog, an object is counted as recovered if the predicted vehicle mask has nonzero overlap with its ground-truth BEV box. This permissive object-recall measure complements, rather than replaces, IoU. Table 7 shows that predictor-only recall is low, whereas the reconstructor recovers current evidence even when the prior lacks the object. At severity 3, RESBev achieves 32.8% recall versus 5.5% for predictor-only and 11.5% for vanilla LSS. This is evidence of responsiveness to new objects in these clips, rather than a guarantee for arbitrary sudden events.

We also partition scenes into equally sized low- and high-smoothness subsets using the displacement of matched annotated objects between $t-1$ and t in ego-centric coordinates. Lower smoothness means larger scene changes. Under fog, vanilla/RESBev IoU is 8.16/24.18 on the low-smoothness subset and 10.08/29.57 on the high-smoothness subset, averaged across severities. Temporal smoothness helps, but gains remain under larger changes.

5.6 Robustness to Consecutive Corruptions

Real-world perception systems may face persistent disturbances rather than isolated corruptions. We therefore evaluate RESBev with an n-step recursive reconstruction task using LSS [13], where each reconstructed feature is reused as the historical input for the Latent Dynamic Predictor at the next step. We use a 50-step setting ($n = 50$), where all frames in one sequence are corrupted by the selected corruption. As shown in Table 8, RESBev maintains stable IoU under consecutive seen corruptions, indicating limited error accumulation under continuous disturbances. Additional results on unseen corruptions are provided in the supplementary material.

Table 7: Suddenly appearing objects: BEV vehicle segmentation on LSS. Each entry reports object recall (%)/IoU on the 100-clip subset; fog severity increases from 1 to 3. Vanilla is clean-trained, while RESBev uses corruption-aware training and history.

Method	Clean	Fog 1	Fog 2	Fog 3
Vanilla	41.0/30.74	23.5/12.09	16.4/8.14	11.5/5.41
Predictor only	14.8/22.77	11.0/20.61	9.8/19.73	5.5/18.09
RESBev	39.9/29.68	37.7/28.94	36.1/28.38	32.8/27.51

Table 8: BEV vehicle segmentation: LSS IoU under consecutive seen corruptions; Change is the relative 1-to-50-step change.

Corruption	**1-Step**	**10-Step**	**30-Step**	**50-Step**	**Change**
FGSM	28.42	28.58	28.12	28.11	↓1.09%
PGD	31.47	30.93	30.97	30.84	↓2.00%
Fog	28.39	28.03	28.00	27.89	↓1.76%
Bright	28.38	27.91	27.88	27.89	↓1.72%
Frame Lost	28.33	28.48	28.31	28.43	↑0.35%

5.7 Efficiency and Deployment Cost Analysis

We compare RESBev with BEVFormer-base under the same input resolution and hardware setting. FPS is computed from the average per-frame latency measured with batch size 1 after 100 warm-up iterations and 500 timed iterations using `torch.cuda.synchronize()`. RESBev changes throughput from 2.7 to 2.4 Hz (an 11.1% reduction); Table 9 quantifies the additional parameters, computation, and memory.

Table 9: BEVFormer-base efficiency: parameters, FLOPs, throughput, and GPU memory. Inference uses FP16; memory columns report training/inference GB.

Method	Params (M)	FLOPs (G)	FPS	Train GB	Infer GB
BEVFormer-base	69.1	1311	2.7	28.5	8.4
+RESBev	83.8	1552	2.4	32.1	9.8

6 Limitations

RESBev depends on historical evidence and recoverable current observations. A suddenly appearing object obscured by severe fog or noise can be missed or detected late because neither source contains sufficient evidence. Conversely, an object that has just left the scene can persist in the prior and cause a false positive when the corrupted current frame cannot correct it. The 50-step rollouts show limited drift in the tested settings, not a bound on error accumulation. Evaluation is limited to nuScenes and the specified synthetic corruptions; it does not establish robustness to arbitrary adaptive attacks or real-world weather. The module also adds parameters, computation, and memory, and its two-frame training does not exhaust longer-history designs.

7 Conclusion

We present RESBev, a plug-and-play framework for robust BEV perception under natural corruptions and adversarial attacks. Rather than directly aggregating corrupted observations, RESBev predicts a temporal BEV prior from historical context and ego-motion using a Latent Dynamic Predictor, and reconstructs the current BEV feature through a prior-guided Anomaly Reconstructor. Experiments on nuScenes show that RESBev improves robustness across multiple BEV models, generalizes to unseen corruptions, supports 3D detection, and maintains stability under long-horizon consecutive corruptions.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant 62503321, and 62303307. The authors declare no competing interests.

References

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

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[3] Ming Chang, Xishan Zhang, Rui Zhang, Zhipeng Zhao, Guanhua He, and Shaoli Liu. Recurrentbev: A long-term temporal fusion framework for multi-view 3d detection. In *European Conference on Computer Vision*, pages 131–147. Springer, 2024.

[4] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[5] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? In *IEEE International Conference on Robotics and Automation*, 2023.

[6] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021.

[7] Sanmin Kim, Youngseok Kim, In-Jae Lee, and Dongsuk Kum. Predict to detect: Prediction-guided 3D object detection using sequential images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18057–18066, 2023.

[8] Sanjay Kumar, Hiep Truong, Sushil Sharma, Ganesh Sistu, Tony Scanlan, Eoin Grua, and Ciarán Eising. Minimizing occlusion effect on multi-view camera perception in bev with multi-sensor fusion. *Electronic Imaging*, 37(15):113–1–113–6, 2025. doi: 10.2352/EI.2025.37.15.AVM-113.

[9] Leheng Li, Qing Lian, and Ying-Cong Chen. Adv3d: Generating 3d adversarial examples for 3d object detection in driving scenarios with nerf. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10813–10820. IEEE, 2024.

[10] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, 2022.

[11] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *IEEE International Conference on Robotics and Automation*, 2023.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[13] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European conference on computer vision*, pages 194–210. Springer, 2020.

[14] Reza Sadeghian, Niloofar Hooshyaripour, Chris Joslin, and WonSook Lee. Reliability-driven lidar-camera fusion for robust 3d object detection. In *Proceedings of the 38th Canadian Conference on Artificial Intelligence*, 2025.

[15] Jonas Schramm, Niclas Vödisch, Kürsat Petek, B Ravi Kiran, Senthil Yogamani, Wolfram Burgard, and Abhinav Valada. Bevcar: Camera-radar fusion for bev map and object segmentation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1435–1442. IEEE, 2024.

[16] Shubham Sonarghare, Prasad Deshpande, Ciaran Hogan, Deepika-Rani Kaliappan-Mahalingam, and Ganesh Sistu. Fisheyegaussianlift: Bev feature lifting for surround-view fisheye camera perception. In *Irish Machine Vision and Image Processing Conference*, pages 50–57, 2025. doi: 10.21251/8043511f-bf93-4b36-9348-0726af0987f6.

[17] Ziying Song, Lei Yang, Shaoqing Xu, Lin Liu, Dongyang Xu, Caiyan Jia, Feiyang Jia, and Li Wang. Graphbev: Towards robust bev feature alignment for multi-modal 3d object detection. In *European conference on computer vision*, pages 347–366. Springer, 2024.

[18] Bilel Tarchoun, Quazi Mishkatul Alam, Nael Abu-Ghazaleh, and Ihsen Alouani. Fool the hydra: Adversarial attacks against multi-view object detection systems. *arXiv preprint arXiv:2312.00173*, 2023.

[19] Fu Wang, Yanghao Zhang, Xiangyu Yin, Guangliang Cheng, Zeyu Fu, Xiaowei Huang, and Wenjie Ruan. A black-box evaluation framework for semantic robustness in bird's eye view detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7637–7645, 2025.

[20] Shaoyuan Xie, Zichao Li, Zeyu Wang, and Cihang Xie. On the adversarial robustness of camera-based 3d object detection. *Transactions on Machine Learning Research*, 2024.

[21] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[22] Xin Ye, Burhaneddin Yaman, Sheng Cheng, Feng Tao, Abhirup Mallik, and Liu Ren. Bevdiffuser: Plug-and-play diffusion model for bev denoising with ground-truth guidance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1495–1504, 2025.

[23] Zijian Zhu, Yichi Zhang, Hai Chen, Yinpeng Dong, Shu Zhao, Wenbo Ding, Jiachen Zhong, and Shibao Zheng. Understanding the robustness of 3d object detection with bird's-eye-view representations in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21600–21610, 2023.

[24] Zhuofan Zong, Dongzhi Jiang, Guanglu Song, Zeyue Xue, Jingyong Su, Hongsheng Li, and Yu Liu. Temporal enhanced training of multi-view 3D object detector via historical object prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3781–3790, 2023.

[25] Jiayu Zou, Kun Tian, Zheng Zhu, Yun Ye, and Xingang Wang. Diffbev: Conditional diffusion model for bird's eye view perception. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 7846–7854, 2024.

A Technical appendices and supplementary material

This supplementary material is organized as follows:
Sec. B: Benchmark and Corruption Details.
Sec. C: Implementation Details.
Sec. D: Additional Experiments.
Sec. E: Discussions.
The pipeline schematic is reproduced in Section C to accompany the implementation details.

B Benchmark and Corruption Details

B.1 Benchmark Setup

We follow the RoboBEV-style evaluation setting [21] and study robustness under both natural corruptions and adversarial attacks. For the main RESBev experiments and the matched BEVFormer comparison, we use five training corruptions: FGSM, PGD, Fog, Bright, and Frame Lost. These training-time corruptions are used to expose the model to representative disturbances of both adversarial and natural categories.

During evaluation, we report performance not only on the corruptions used during training, but also on five additional corruptions that are never introduced during training: C&W Attack, Snow, Camera Crash, Noise, and Dark. This protocol evaluates both seen and unseen corruptions at three severity levels. The controlled P2D/HoP experiments instead train both compared models on FGSM and Fog only (Section 5.3). Official clean-trained checkpoints in the system comparisons receive no corruption augmentation.

B.2 Implementation Details of Natural Corruptions

We use seven natural corruptions in total: Noise, Bright, Dark, Fog, Snow, Camera Crash, and Frame Lost. The implementation details of each corruption are given below.

Noise. Noise is implemented as additive Gaussian perturbation. The standard deviation is defined relative to the dynamic range of the current image as $\sigma \in \{0.1 \times (\max(I) - \min(I)),\ 0.2 \times (\max(I) - \min(I)),\ 0.3 \times (\max(I) - \min(I))\}$ for severity levels 1, 2, and 3.

Bright. Bright corruption is implemented by directly rescaling the image intensity with a multiplicative brightness factor. The three severity levels use factors $\{1.3,\ 1.8,\ 2.2\}$.

Dark. Dark corruption is implemented by directly rescaling the image intensity with a multiplicative darkness factor. The three severity levels use factors $\{0.85,\ 0.325,\ 0.3\}$.

Fog. Fog is synthesized using a transmission-based atmospheric scattering model. The fog density parameter is set to $\{0.3,\ 1.2,\ 1.5\}$ for severity levels 1, 2, and 3.

Snow. Snow is implemented by overlaying randomly generated snowflakes together with a global contrast reduction. The snow intensity parameter is set to $\{0.4,\ 0.9,\ 1.1\}$ for severity levels 1, 2, and 3.

Camera Crash. Camera Crash is simulated by randomly selecting a subset of the six camera views and setting the corresponding images to the minimum value. The numbers of crashed views under the three severity levels are $\{1,\ 2,\ 4\}$.

Frame Lost. Frame Lost is simulated by randomly selecting a subset of the six camera views and replacing their current frames with frames sampled from the remaining non-lost views. The numbers of affected views under the three severity levels are $\{1,\ 2,\ 3\}$.

B.3 Implementation Details of Adversarial Corruptions

All adversarial attacks are applied directly to the multi-view camera inputs before BEV feature construction and are regenerated for the complete evaluated model, including the predictor and reconstructor for +RESBev.

FGSM [4] is implemented as a single-step gradient-sign attack with perturbation budget $\epsilon \in \{1/255,\ 2/255,\ 4/255\}$. In our implementation, FGSM optimizes a loss that combines binary cross-entropy with IoU-suppression terms in order to directly reduce BEV semantic prediction quality.

PGD [12] is implemented as a multi-step projected attack with random uniform initialization in $[-\epsilon, \epsilon]$. The parameter settings under the three severity levels are as follows:

$$\begin{aligned}(\epsilon, \alpha, \text{steps}) \in \{&(1/255,\ 0.3/255,\ 5),\\ &(2/255,\ 0.5/255,\ 10),\\ &(4/255,\ 1/255,\ 15)\}.\end{aligned} \tag{9}$$

In our implementation, PGD optimizes a loss that combines binary cross-entropy with IoU-suppression terms, and additionally includes a false-positive encouragement term on non-vehicle regions.

C&W Attack [2] is implemented as an optimization-based attack with severity-dependent regularization strength, confidence target, iteration count, perturbation bound, and learning rate. The parameter settings under the three severity levels are as follows:

$$\begin{aligned}(c, \kappa, \text{iter}, \epsilon, \text{lr}) \in \{&(1.0,\ 0.15,\ 50,\ 4/255,\ 0.02),\\ &(3.0,\ 0.35,\ 80,\ 8/255,\ 0.02),\\ &(8.0,\ 0.55,\ 100,\ 16/255,\ 0.03)\}.\end{aligned} \tag{10}$$

In our implementation, C&W Attack minimizes an L_2-regularized objective that explicitly pushes the attacked prediction below an IoU target derived from the clean prediction.

C Implementation Details

C.1 Inputs and Temporal Construction

RESBev is designed to operate in the semantic feature space of an existing BEV perception pipeline, rather than directly on raw input images. We apply RESBev to representative BEV pipelines, including LSS [13], SimpleBEV [5], GaussianLSS [16], and the attention-based BEVFormer [10]. At time step t, the current disturbed perception data is first processed by the original 2D image backbone and view transformer of the baseline model, producing a corrupted BEV feature map

$$f_t^{corrupt} \in \mathbb{R}^{C\times 200\times 200}. \tag{11}$$

In addition to the current corrupted feature, RESBev takes as input the reconstructed BEV feature from the previous step, and the corresponding ego-motion which encodes the relative translation and rotation between time steps $t-1$ and t.

$$f_{t-1}^{rec} \in \mathbb{R}^{C\times 200\times 200}, \qquad a_{t-1} \in \mathbb{R}^{15}. \tag{12}$$

Recursive inference proceeds frame by frame. At each step, RESBev predicts a clean BEV prior from (f_{t-1}^{rec}, a_{t-1}), and then fuses this prior with the current corrupted observation $f_t^{corrupt}$ to produce the reconstructed feature f_t^{rec}.

C.2 Overall Pipeline of RESBev

RESBev consists of two core modules: a *Semantic Prior Predictor* and an *Anomaly Reconstructor*. The Semantic Prior Predictor is responsible for forecasting a clean semantic prior for the current BEV state from historical information and ego-motion. Taking the predicted prior as a robustness anchor, the Anomaly Reconstructor then selectively retrieves valid evidence from the current corrupted observation.

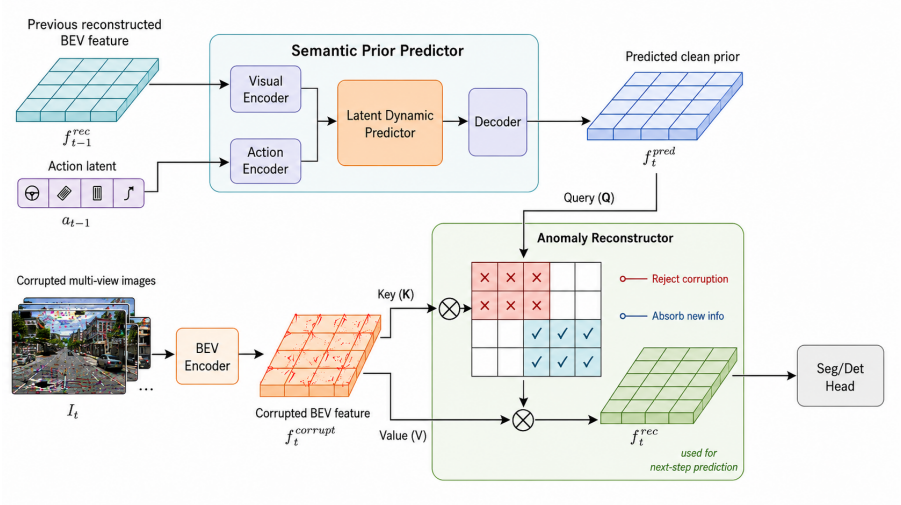


Figure 4: Implementation schematic: historical BEV and ego-motion produce a prior; the prior queries current BEV features through the reconstructor. The output is sent to the task head and reused as the next historical state during recursive inference.

Formally, the overall pipeline can be summarized as:

$$f_t^{pred} = \mathcal{P}(f_{t-1}^{rec}, a_{t-1}), \qquad f_t^{rec} = \mathcal{R}(f_t^{pred}, f_t^{corrupt}). \tag{13}$$

where $\mathcal{P}(\cdot)$ denotes the Semantic Prior Predictor and $\mathcal{R}(\cdot)$ denotes the Anomaly Reconstructor. The reconstructed feature f_t^{rec} is finally fed into the task head of the baseline BEV perception model for semantic prediction.

C.3 Implementation Details of Semantic Prior Predictor

The Semantic Prior Predictor is implemented as a lightweight latent dynamics predictor composed of four parts: a visual encoder E_{vis}, an action encoder E_{act}, a Latent Dynamic Predictor (LDP), and a decoder D.

Given the previous reconstructed BEV feature f_{t-1}^{rec} and the ego-motion vector a_{t-1}, we first map them into a compact latent space:

$$z_{t-1}^{vis} = E_{vis}(f_{t-1}^{rec}), \qquad z_{t-1}^{act} = E_{act}(a_{t-1}). \tag{14}$$

The visual latent is represented as a compact sequence of BEV latent tokens, while the action latent is represented as a single action token. The two latent representations are then concatenated into an action-aware latent state:

$$z_{t-1} = \mathrm{Concat}(z_{t-1}^{vis}, z_{t-1}^{act}). \tag{15}$$

A latent dynamics predictor is then applied to predict the future latent state:

$$z_t^{pred} = \mathrm{LDP}(z_{t-1}), \tag{16}$$

which is decoded back into the dense BEV feature space:

$$f_t^{pred} = D(z_t^{pred}). \tag{17}$$

Combining the above steps gives the predictor formulation used in the main paper:

$$f_t^{pred} = D\Big(\mathrm{LDP}\big(\mathrm{Concat}(E_{vis}(f_{t-1}^{rec}), E_{act}(a_{t-1}))\big)\Big). \tag{18}$$

Implementation details. For reproducibility, we report the detailed configuration of each component in the Semantic Prior Predictor. The visual encoder is a lightweight convolutional patch encoder composed of three strided 3×3 convolution blocks, followed by a linear projection that maps the encoded BEV feature into the latent space used by the predictor. The action encoder is a 2-layer

MLP with architecture $15 \to 128 \to 256$ and GELU nonlinearity, which maps the ego-motion vector into an action token in the same latent space. The latent dimension is set to $d = 256$, which is the shared embedding dimension for both visual latent tokens and the action token in the predictor. The number of visual latent tokens is $N = 625$, corresponding to a 25×25 latent BEV grid obtained after spatial downsampling of the input BEV feature map. In addition, one action token produced by the action encoder is concatenated with the visual latent tokens before being fed into the latent dynamics predictor. The latent dynamics predictor(LDP) is implemented as a compact Transformer consisting of 4 encoder blocks with 8 attention heads and an MLP ratio of 4.0. The decoder is a lightweight BEV decoder that takes the predicted visual latent tokens, reshapes them back into a low-resolution latent BEV grid, and then progressively restores them to the original BEV feature resolution through two upsampling-convolution stages followed by final 3×3 convolutional refinement layers.

C.4 Implementation Details of Anomaly Reconstructor

The Anomaly Reconstructor is a lightweight prior-guided fusion module. Its role is not to independently generate a new BEV representation, but to selectively incorporate the extracted valid information from the current corrupted observation into the predicted clean prior.

Given the predicted BEV prior f_t^{pred} and the current corrupted feature $f_t^{corrupt}$, we construct a prior-guided spatial cross-attention module directly on 2D BEV feature maps. In this module, the predicted prior serves as the query source, while the current corrupted feature provides the key-value context for selective feature retrieval:

$$Q = f_t^{pred}, \qquad K = V = f_t^{corrupt}. \tag{19}$$

All feature maps are first projected into a shared hidden space while preserving their 2D BEV structure. The reconstructed feature is then obtained through gated residual fusion:

$$f_t^{rec} = f_t^{pred} + \alpha \cdot \mathrm{CrossAttn}(Q, K, V), \tag{20}$$

where $\alpha \in [0, 1]$ is a learned dynamic gating factor.

Implementation details. For reproducibility, we report the detailed configuration of each component in the Anomaly Reconstructor. The prior-guided cross-attention is implemented directly on 2D BEV feature maps after linear projection into a shared hidden space, so that the predicted prior and the current corrupted feature can interact in a common representation space while preserving BEV spatial structure. The attention hidden dimension is set to 256, which is the shared channel dimension used for the projected query, key, and value features in the reconstructor. Multi-head attention is implemented with 8 attention heads. To preserve spatial correspondence in the BEV plane, 2D learnable positional embeddings are added to the projected feature maps before attention is applied. The dynamic fusion gate α is implemented as a lightweight 2-layer MLP with sigmoid activation. It takes the fused attention features as input and produces channel-wise gating coefficients, which control how much retrieved information is injected into the predicted BEV prior during reconstruction.

C.5 Feature Shapes and Computational Flow

We summarize the main feature flow of RESBev in Table 10.

Table 10: Representative feature flow in RESBev.

Stage	Representation	Shape
Previous reconstructed BEV	f_{t-1}^{rec}	$C \times 200 \times 200$
Current corrupted BEV	$f_t^{corrupt}$	$C \times 200 \times 200$
Ego-motion vector	a_{t-1}	15
Visual latent	z_{t-1}^{vis}	625×256
Action latent	z_{t-1}^{act}	1×256
Predicted latent state	z_t^{pred}	625×256
Predicted BEV prior	f_t^{pred}	$C \times 200 \times 200$
Projected query/key/value maps	Q, K, V	$256 \times 200 \times 200$
Final reconstructed BEV	f_t^{rec}	$C \times 200 \times 200$

Table 11: BEV vehicle segmentation: per-severity IoU under seen corruptions. Vanilla is the official clean-trained checkpoint; +RESBev uses five-corruption training and temporal context.

Corruption	Severity	LSS		SimpleBEV		GaussianLSS		BEVFormer	
		Vanilla	+RESBev	Vanilla	+RESBev	Vanilla	+RESBev	Vanilla	+RESBev
Clean	-	33.03	33.31	55.70	55.91	42.80	43.14	46.70	47.01
FGSM	**1**	13.87	28.58	17.65	31.03	11.76	31.49	29.10	38.86
	2	10.06	28.47	12.51	31.13	9.43	31.42	21.51	38.79
	3	6.92	28.21	9.54	31.12	8.53	31.37	14.01	38.69
	Avg.	10.28	28.42	13.23	31.09	9.91	31.43	21.54	38.78
PGD	**1**	12.55	31.96	15.44	30.84	10.23	31.21	26.08	38.95
	2	8.93	31.34	11.25	31.68	7.64	31.04	18.57	38.88
	3	6.04	31.12	8.97	32.42	5.28	31.02	11.39	38.87
	Avg.	9.17	31.47	11.89	31.65	7.72	31.09	18.68	38.90
Fog	**1**	13.91	28.46	22.42	33.83	27.12	32.09	32.02	39.00
	2	9.24	28.38	16.76	33.81	25.98	31.94	25.31	38.93
	3	6.64	28.33	12.37	33.64	19.67	31.82	18.69	38.80
	Avg.	9.93	28.39	17.18	33.76	24.26	31.95	25.34	38.91
Bright	**1**	13.32	28.44	21.39	33.74	27.10	32.05	29.31	38.78
	2	9.35	28.37	17.56	33.91	25.60	31.87	25.64	38.73
	3	6.66	28.32	13.94	33.72	19.38	31.70	22.15	38.71
	Avg.	9.78	28.38	17.63	33.79	24.03	31.87	25.70	38.74
Frame Lost	**1**	15.24	28.41	23.54	33.85	22.23	31.85	30.47	38.84
	2	9.43	28.32	16.11	33.64	12.76	31.53	25.81	38.76
	3	7.28	28.27	13.58	33.70	9.12	31.44	21.30	38.65
	Avg.	10.65	28.33	17.74	33.73	14.70	31.61	25.86	38.75

C.6 Optimization Details

All models are implemented in PyTorch and trained on a single NVIDIA A100-SXM4-80GB GPU with a batch size of 16.

For reproducibility, the optimization details are summarized as follows:

- optimizer: AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$;
- initial learning rate: 5×10^{-4};
- weight decay: 1×10^{-4};
- scheduler and warmup: linear warmup for the first 5 epochs starting from 0.2 times the base learning rate, followed by cosine decay; an auxiliary ReduceLROnPlateau scheduler is additionally used with factor 0.5, patience 5, and minimum learning rate 2×10^{-5};
- total training epochs / iterations: 100 epochs;
- gradient clipping / mixed precision: gradient clipping with a maximum norm of 0.5; mixed precision is not used during training unless otherwise stated. Efficiency measurements are conducted under FP16 inference following the main paper.

D Additional Experiments

D.1 System-Level Backbone Comparisons

Tables 12, 14, and 13 retain the original system-level comparisons across backbones. They measure the total benefit of the enhanced system over official clean-trained checkpoints. Because training augmentation and, for single-frame models, temporal inputs also differ, their differences must not be interpreted as isolated architectural gains. The controlled BEVFormer and P2D/HoP results in the main paper address this confound.

Table 12: BEV vehicle segmentation: average IoU under seen corruptions across three severities. V: official clean-trained vanilla checkpoint; R: five-corruption-trained RESBev with temporal context. Differences combine training, module, and (for single-frame backbones) history changes.

	LSS		SimpleBEV		GaussianLSS		BEVFormer		DiffBEV
Corruption	V	R	V	R	V	R	V	R	
Clean	33.03	33.31	55.70	55.91	42.80	43.14	46.70	47.01	38.90
FGSM	10.28	28.42	13.23	31.09	9.91	31.43	21.54	38.78	25.18
PGD	9.17	31.47	11.89	31.65	7.72	31.09	18.68	38.90	28.11
Fog	9.93	28.39	17.18	33.76	24.26	31.95	25.34	38.91	29.02
Bright	9.78	28.38	17.63	33.79	24.03	31.87	25.70	38.74	30.39
Frame Lost	10.65	28.33	17.74	33.73	14.70	31.61	25.86	38.75	27.81
Overall Avg.	9.96	29.02	15.53	32.80	16.12	31.59	23.42	38.82	28.10

Table 13: 3D detection: average NDS under seen and unseen corruptions across three severity levels. Vanilla BEVFormer uses an official clean-trained checkpoint; RESBev uses five-corruption training. These system comparisons do not provide training-matched controls. BEVDiffuser and RESBev are evaluated on the same BEVFormer and BEVFusion backbones.

(a) Seen corruptions.

Corruption	**BEVFormer-base**			**BEVFusion**		
	Vanilla	+RESBev	+BEVDiffuser	Vanilla	+RESBev	+BEVDiffuser
Clean	51.8	53.3	53.7	70.9	71.7	71.9
FGSM	30.10	50.21	44.31	52.18	70.81	66.71
PGD	28.18	50.19	40.87	51.09	70.73	64.90
Fog	34.81	51.35	45.71	52.83	71.98	68.71
Bright	40.31	51.98	44.84	52.19	70.19	69.15
Frame Lost	35.79	51.81	44.69	52.87	70.54	64.83
Avg.	**33.84**	**51.11**	**44.08**	**52.23**	**70.85**	**66.86**

(b) Unseen corruptions.

Corruption	**BEVFormer-base**			**BEVFusion**		
	Vanilla	+RESBev	+BEVDiffuser	Vanilla	+RESBev	+BEVDiffuser
Clean	51.8	53.3	53.7	70.9	71.7	71.9
C&W	22.81	49.88	38.18	45.18	70.84	60.91
Snow	30.19	50.92	43.71	51.91	71.92	64.87
Cam. Crash	29.84	51.15	44.87	52.35	70.63	62.95
Noise	29.97	50.87	45.14	53.14	70.95	65.73
Dark	26.71	51.32	44.91	52.84	70.28	64.80
Avg.	**27.90**	**50.83**	**43.36**	**51.08**	**70.92**	**63.85**

D.2 Additional Segmentation and Detection Metrics

The following BEVFormer results compare the official clean-trained checkpoint against five-corruption-trained RESBev, so they describe system performance under the same qualification as Appendix D.1. The corrupted segmentation column averages FGSM, Fog, C&W, and Snow over three severities. Car and Vehicles are overlapping categories; their arithmetic mean with Road and Lane is a mean over these reported categories, not a conventional mean over disjoint semantic classes.

D.3 Augmentation and Input Purification Baselines

Table 17 evaluates LSS with FGSM/Fog augmentation and JPEG input purification (quality 75). The augmentation baseline is single-frame and uses two training corruptions; RESBev adds history and

Table 14: BEV vehicle segmentation: average IoU under unseen corruptions across three severities. V: official clean-trained vanilla checkpoint; R: five-corruption-trained RESBev with temporal context. Differences combine training, module, and (for single-frame backbones) history changes.

	LSS		SimpleBEV		GaussianLSS		BEVFormer		DiffBEV
Corruption	V	R	V	R	V	R	V	R	
Clean	33.03	33.31	55.70	55.91	42.80	43.14	46.70	47.01	38.90
C&W Attack	8.78	30.80	11.43	30.61	5.97	31.24	17.56	39.67	29.81
Snow	10.26	28.35	17.40	33.71	16.08	32.10	28.11	38.91	30.14
Camera Crash	10.08	28.34	17.34	33.74	13.41	31.56	27.46	39.72	29.42
Noise	8.64	28.27	17.60	33.76	16.67	31.43	21.55	39.34	29.89
Dark	8.11	28.36	16.29	33.75	17.68	31.96	28.06	39.75	30.10
Overall Avg.	9.17	28.82	16.01	33.11	13.96	31.66	24.55	39.48	29.87

Table 15: BEV segmentation: BEVFormer per-category IoU (clean/corrupted). The final row averages the four reported, partly overlapping categories.

Category	Vanilla	+RESBev
Car	46.80/23.32	47.01/39.11
Vehicles	46.70/23.14	47.01/39.07
Road	77.50/50.24	77.15/65.89
Lane	23.90/8.72	23.94/19.61
Reported-category mean	48.73/26.36	48.78/40.92

uses five. Thus, these are supplementary system comparisons, not controlled evidence for the module alone. The matched five-corruption BEVFormer comparison is reported in Table 3. The augmentation row includes FGSM adversarial training and fog augmentation rather than adversarial training alone.

D.4 Normalized Natural-Corruption Error

For 3D detection, following the normalized corruption-error formulation of RoboBEV [21], the seven-natural-corruption evaluation reports mCE of 1.00/0.72 for vanilla/RESBev on BEVFormer and 0.71/0.43 on BEVFusion, using vanilla BEVFormer as the common reference (lower is better). These values use our modified corruption suite and reference model and therefore should not be compared directly with the original RoboBEV leaderboard. They are supplementary system-level results and do not replace the controlled comparisons.

D.5 Per-Severity Results on Seen Corruptions

We provide the detailed per-severity results on the seen corruption set in Table 11. These results complement the main paper, which only reports the average across the three severity levels.

D.6 Per-Severity Results on Unseen Corruptions

We provide the detailed per-severity results on unseen corruptions in Table 18.

D.7 Long-Horizon Robustness on Unseen Corruptions

We further evaluate RESBev under consecutive unseen corruptions using the same 50-step recursive reconstruction setting as in the main paper. As shown in Table 19, RESBev maintains stable performance under long corrupted sequences, indicating that the recurrent predictive prior generalizes beyond the corruptions observed during training.

Table 16: 3D detection: BEVFormer mAP, averaged over three corruption severities. Vanilla is clean-trained; +RESBev uses five-corruption training.

Corruption	Vanilla	+RESBev
Clean	42.00	42.16
FGSM	19.63	38.27
Fog	23.72	38.41
C&W	15.43	38.56
Snow	22.65	38.97

Table 17: BEV vehicle segmentation: LSS IoU with augmentation or JPEG purification. Corrupted scores average three severities; training and temporal configurations differ as described in the text.

Method	Clean	FGSM	Fog	C&W	Snow
Official LSS	33.03	10.28	9.93	8.78	10.26
LSS + FGSM/Fog aug.	32.63	14.87	16.64	13.57	15.89
LSS + JPEG	32.58	14.12	17.25	14.04	14.81
LSS + RESBev	33.31	28.42	28.39	30.80	28.35

E Discussions

E.1 Comparison with Existing Robustness Strategies

Compared with fusion-based BEV perception methods such as BEVCar [15], RESBev does not rely on additional sensing modalities and is therefore easier to integrate into camera-only BEV perception systems. This makes RESBev especially attractive in practical settings where sensor cost, deployment complexity, or hardware constraints must be carefully controlled. In contrast, fusion-based methods can leverage complementary physical observations across modalities and may retain stronger robustness when the visual stream is severely degraded.

Compared with temporal aggregation methods such as BEVFormer [10], RESBev explicitly predicts a clean semantic prior rather than merely aggregating historical and current observations. This design makes it better suited to bypass corrupted inputs and recover temporally consistent semantic structure under heterogeneous corruptions. By introducing a predictive prior before feature reconstruction, RESBev emphasizes semantic recovery instead of relying solely on direct temporal fusion.

Compared with graph-based reasoning methods such as GraphBEV [17], RESBev provides a predictive reconstruction perspective that is naturally applicable to both natural corruptions and adversarial perturbations. While graph-based approaches are particularly effective at modeling structured relations and long-range interactions, RESBev focuses on recovering corrupted BEV representations through temporally grounded semantic prediction, offering a complementary robustness mechanism.

Compared with diffusion-based BEV refinement methods such as DiffBEV [25] and BEVDiffuser [22], RESBev avoids iterative diffusion sampling and instead performs one-step temporal prior prediction followed by prior-guided reconstruction. Diffusion-based methods refine BEV features, but their deployment costs depend on whether denoising runs during training or inference; BEVDiffuser is a training-time method. In contrast, RESBev uses historical BEV dynamics and ego-motion to predict a semantic prior, enabling lightweight feature recovery with limited deployment overhead. These two directions are complementary: diffusion-based refinement emphasizes generative denoising, while RESBev emphasizes temporal semantic prediction and selective reconstruction.

E.2 Broader Impacts

Improved perception robustness can benefit autonomous-driving research by reducing failures under degraded sensing. However, synthetic-corruption performance is insufficient evidence for safe road deployment, and missed new objects or stale prior content can affect downstream decisions. Adversarial evaluation can also expose vulnerabilities that could be misused to disrupt perception systems. The method should therefore be assessed together with uncertainty handling and independent system-level validation before deployment.

Table 18: BEV vehicle segmentation: per-severity IoU under unseen corruptions. Vanilla is the official clean-trained checkpoint; +RESBev uses five-corruption training and temporal context.

Corruption	Severity	LSS		SimpleBEV		GaussianLSS		BEVFormer	
		Vanilla	+RESBev	Vanilla	+RESBev	Vanilla	+RESBev	Vanilla	+RESBev
Clean	-	33.03	33.31	55.70	55.91	42.80	43.14	46.70	47.01
C&W Attack	**1**	11.28	30.94	14.65	30.87	8.02	31.38	24.96	39.71
	2	8.54	30.75	11.67	30.74	6.09	31.21	17.54	39.70
	3	6.51	30.70	7.98	30.22	3.80	31.12	10.18	39.60
	Avg.	8.78	30.80	11.43	30.61	5.97	31.24	17.56	39.67
Snow	**1**	17.69	28.44	22.85	33.67	21.23	32.16	32.85	38.95
	2	8.34	28.34	16.76	33.69	16.56	32.09	27.09	38.93
	3	4.76	28.27	12.59	33.78	10.45	32.05	24.39	38.85
	Avg.	10.26	28.35	17.40	33.71	16.08	32.10	28.11	38.91
Camera Crash	**1**	14.21	28.39	23.88	33.71	19.53	31.73	31.05	39.79
	2	10.38	28.33	16.90	33.74	13.47	31.57	27.42	39.74
	3	5.64	28.31	11.24	33.76	7.24	31.38	23.91	39.63
	Avg.	10.08	28.34	17.34	33.74	13.41	31.56	27.46	39.72
Noise	**1**	10.49	28.36	21.67	33.71	24.03	31.80	28.96	39.43
	2	8.51	28.25	17.56	33.86	15.85	31.44	20.51	39.35
	3	6.93	28.21	13.57	33.71	10.12	31.04	15.18	39.24
	Avg.	8.64	28.27	17.60	33.76	16.67	31.43	21.55	39.34
Dark	**1**	11.76	28.39	20.98	33.73	25.04	32.02	31.99	39.82
	2	7.19	28.35	16.32	33.77	16.98	31.95	27.47	39.73
	3	5.37	28.33	11.56	33.75	11.02	31.91	24.72	39.70
	Avg.	8.11	28.36	16.29	33.75	17.68	31.96	28.06	39.75

Table 19: BEV vehicle segmentation: LSS IoU under consecutive unseen corruptions; Change is the relative 1-to-50-step change.

Corruption	1-Step	10-Step	30-Step	50-Step	Change
C&W	30.80	30.26	30.30	30.13	↓2.18%
Snow	28.35	27.86	27.81	27.80	↓1.94%
Camera Crash	28.34	28.00	27.91	27.79	↓1.94%
Noise	28.27	27.98	27.81	27.80	↓1.66%
Dark	28.36	27.93	27.95	27.88	↓1.69%

E.3 Future Directions

Future work includes extending predictive reconstruction to occupancy prediction and online mapping, testing additional BEV architectures, and studying longer histories and loss-weight sensitivity. Uncertainty-aware or multi-hypothesis prediction may help with stochastic scene changes and long corruption sequences. Combining temporal prediction with complementary fusion, denoising, or relational reasoning is another direction.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the proposed RESBev framework, its plug-and-play design, and its robustness improvements under natural corruptions and adversarial attacks. The claims are supported by the experimental results on nuScenes across multiple BEV backbones.

2. **Limitations**

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

   Answer: [Yes]

   Justification: The paper now includes a dedicated Limitations section in the main paper, which discusses potential constraints of RESBev.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not present formal theoretical results, theorems, or proofs. The proposed method is evaluated empirically through diagnostic analysis, robustness experiments, ablation studies, and efficiency analysis.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper?

   Answer: [Yes]

   Justification: The paper describes the dataset, baseline models, corruption protocol, evaluation metrics, training objective, and implementation setting, while additional details such as hyperparameters, data splits, and optimizer settings are provided in the appendix.

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

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

   Answer: [No]

   Justification: The code and data preparation materials are still being organized and are not publicly available at this stage. We plan to provide access instructions and reproduction scripts after the materials are finalized.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the dataset, baseline models, corruption protocol, evaluation metrics, implementation framework, hardware setting, and batch size. Additional training and testing details, including hyperparameters, data splits, and optimizer settings, are provided in the appendix.

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: The paper reports the main robustness results across multiple corruption types, severity levels, and BEV backbones, but does not currently include error bars, confidence intervals, or statistical significance tests.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the GPU type (NVIDIA A100-SXM4-80GB), batch size, and efficiency metrics (Params, FLOPs, FPS) in the appendix.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The research uses standard autonomous driving benchmarks and evaluates robustness under simulated natural corruptions and adversarial attacks. It does not involve unethical data collection, human-subject experiments, or privacy-invasive procedures.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The Broader Impacts subsection discusses potential safety benefits, misuse of adversarial evaluation, and the limits of synthetic benchmarks for deployment.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped datasets, or other assets that would require special safeguards for responsible release.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets used in the paper properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The dataset and methods are credited through citations. A complete license inventory for all evaluated backbones and checkpoints is not included.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper proposes a new method, but does not currently release a new dataset, codebase, model checkpoint, or other standalone research asset.

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

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

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether IRB approvals were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects, so IRB approval or equivalent review is not applicable.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research?

    Answer: [N/A]

    Justification: The core methodology of this research does not use LLMs as an important, original, or non-standard component. Any use of LLMs, if applicable, was limited to writing, editing, or formatting and did not affect the scientific methodology or results.