Towards Stealthy and Effective Backdoor Attacks on Lane Detection: A Naturalistic Data Poisoning Approach

Supplementary Material

1. Diffusion Denoising

Diffusion-based inpainting methods extend standard diffusion models by introducing spatial constraints, where a mask specifies the region to be edited while preserving the remaining content. Concretely, the model starts from a noisy latent representation and iteratively denoises it, but unlike unconditional generation, the denoising process is conditioned not only on textual instructions but also on the input image and a binary or soft mask that restricts modifications to a target region. In UltraEdit [7], this is implemented via a modified inpainting diffusion pipeline that alternates between global denoising and masked updates, ensuring that edits are localized while maintaining overall image consistency. Building upon this framework, the incorporation of additional losses can be naturally integrated into the denoising process. Therefore, to maintain reality, after decoding the latent into image space two structure-preserving losses constrain the edited region to remain coherent with the original scene. These losses back-propagate through the denoising steps, effectively guiding the noise prediction toward solutions that are both instruction-faithful and visually plausible.

2. More Experiment results

2.1. ASR under varying environments

To evaluate the performance of DBALD under varying physical triggers and driving environments, we test the LOA attack using mud and cone triggers across various lightness conditions in the CULane dataset. Results are given in Table 1. We find that both mud and cone triggers achieve relatively high ASRs across all environments, indicating their effectiveness under LOA. Notably, **the cone trigger outperforms mud at night across all models, with an improvement of up to 5%**, which is reasonable since cones are more visually distinguishable than mud in low-light conditions, resulting in more reliable backdoor activations.

We also find that clean accuracy drops in challenging conditions, which may affect backdoor sensitivity. For instance, under the LOA setting with RESA, clean accuracy decreases from 84.77% (normal) to 82.05% (shadow) and 77.37% (night). This supports our hypothesis that low-light conditions degrade LD model performance, which could inadvertently make the model less reactive to certain perturbations like mud triggers, while still allowing salient triggers (e.g., cones) to remain effective.

Table 1. ASR(%) for different LD models with varying triggers and environments on the CULane dataset.

Trigger	Model	LOA ASR (Varying Driving Environments)			
		Normal	Shadow	Highlight	Night
Mud	LaneATT	74.54	72.05	74.21	67.42
	ADNet	69.51	67.34	67.17	62.23
	SCNN	67.54	68.64	68.51	62.46
	RESA	77.06	75.11	77.24	72.26
Cone	LaneATT	74.17	69.95	73.30	70.13
	ADNet	68.62	68.58	64.03	65.47
	SCNN	68.93	69.17	69.24	66.32
	RESA	77.01	76.34	73.29	75.17

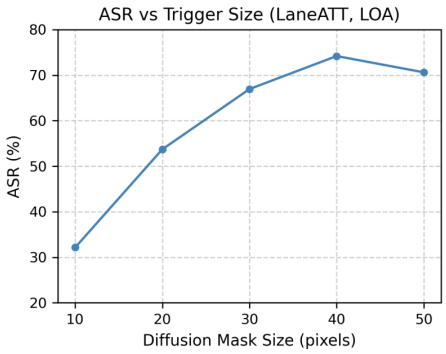

Figure 1. ASR vs. Trigger Size for mud trigger under the LOA setting (LaneATT, CULane).

2.2. ASR under different attack parameters

We further explore the impact of trigger size under the LOA setting using the LaneATT model and CULane dataset with mud triggers. As shown in Figure 1, smaller triggers yield significantly lower attack success rates (ASR), with ASR dropping from 74.19% (40×40) to 32.15% (10×10). This suggests that tiny triggers are less effective at activating the backdoor. However, ASR slightly drops when the trigger size exceeds 40×40 (e.g., 70.62% at 50×50). This is due to a corresponding drop in clean accuracy—excessively large triggers distort the input scene, degrading overall model performance and thus suppressing backdoor activation. Notably, even with a trigger as small as 20×20 pixels, the ASR exceeds 50%, indicating that DBALD remains effective under compact trigger settings. This supports its potential stealthiness in practical deployment.

2.3. ASR under different poisoned ratios

We further study how different poisoning rates affect the attack performance of DBALD. On both LaneATT and RESA models, we train LD models using poisoning rates of **1%, 3%, 5%, 10%** and **15%** on the CULane dataset. For reference, constructing a 1% poisoned dataset (i.e., 177 poisoned samples) takes approximately 3 minutes. The results are given in Table 2. We find that DBALD achieves relative strong attack performance even at very low poisoning rates. As the poisoning rate rises, the ASR improves gradually, while the accuracy on clean samples decreases slowly.

Table 2. Attack performance at varied poisoning rates.

Poisoning Rate	LaneATT		RESA	
	ACC	ASR	ACC	ASR
1%	**75.26**	64.27	**79.98**	68.23
3%	75.10	72.24	79.18	74.87
5%	75.14	73.52	79.12	75.46
10%	74.20	**74.19**	77.39	**76.86**
15%	73.23	73.10	77.20	76.48

2.4. Justification of the attention mechanism

The proposed gradient-based attention mechanism is grounded in both prior work and our analytical findings.

Prior work [5] demonstrates that gradient-based high-attention regions are strongly correlated with high ASR, suggesting that effective triggers must be strategically placed. Motivated by this, we introduce a similar gradient-driven strategy tailored to lane detection (LD) tasks.

We further conduct an analytical study by visualizing the gradient-based attention maps over training epochs for both clean and trigger-injected images. We observe that:

- For clean images, attention gradually concentrates on task-relevant semantic regions.
- For trigger-injected images, attention becomes increasingly focused on the trigger region across training.

This evolving focus indicates that the model gradually "memorizes" the trigger position, reinforcing the effectiveness of targeted trigger placement. Table 3 summarizes results at epoch 100.

Table 3. Comparison of gradient-based attention maps at epoch 100.

Setting	**Attention Entropy** $\downarrow$	**Attention on Trigger Region** $\uparrow$
Trigger (Ours)	16.09	47.39%
Random Trigger	16.26	44.13%
Clean Image	16.96	41.58%

The lower entropy and higher focus on the trigger region for our method suggest that the model allocates more stable and intense attention to the trigger, thereby improving attack reliability.

2.5. Computation time

DBALD incurs a moderate computational cost relative to baseline methods (6 seconds per image vs. 1 second). Since data poisoning is a one-time offline procedure, this computational overhead is generally acceptable.

2.6. Clarification on trigger diversity

To preserve high clean accuracy and maintain a low false-positive (FP) rate—particularly in the rare cases where benign patterns resemble potential triggers—we use LPIPS scores to distinguish our triggers from benign visual patterns in the original data. We adopt a threshold of 0.15 to differentiate benign patterns from actual triggers. On benign-only datasets (e.g., TuSimple), this yields a false-positive rate as low as 8.2%. It is important to note that: i) false positives are inevitable in any backdoor attack when considered in an open-world usage setting; and ii) backdoor triggers can be non-fixed and may even be designed to vary dynamically during inference to achieve high ASR.

In this work, our priority is to achieve high ASRs with realistic, context-aware triggers while keeping the false-positive rate low. To this end, we maintain two lightweight reference databases throughout training and evaluation:

- **Benign pattern database**: a collection of naturally occurring benign visual patterns in clean images that are semantically similar to potential triggers.
- **Poisoned trigger database**: a repository of all synthesized triggers used during the backdoor injection process.

These two databases ensure that each newly synthesized trigger remains sufficiently distinct from benign patterns (LPIPS > 0.15 when compared against the benign pattern database) while remaining visually consistent with existing poisoned triggers (LPIPS < 0.15 when compared against the poisoned trigger database), thereby preserving clean accuracy, maintaining low false-positive rates, and sustaining high ASR with context-aware triggers.

We note that although the clean dataset is large, benign trigger-like patterns are extremely rare (e.g., mud-like textures do not appear, and cone-like shapes occur in only 129 images), resulting in a very small benign-pattern database; the poison-trigger database is similarly small, containing only about 1% of the data corresponding solely to injected trigger instances. Owing to their small size, the pairwise LPIPS comparisons used for trigger-diversity enforcement incur only millisecond-level overhead and thus do not affect overall training efficiency.

Finally, the results in this paper are demonstrated using only two specific trigger instances. Our framework is inher-

ently designed to construct general, natural, and context-aware triggers, and the LPIPS-based filtering together with the dual-database mechanism generalizes uniformly across diverse trigger types, not limited to mud or cones.

3. More details of LD models

Lane detection (LD) methods can be broadly categorized into two main paradigms: anchor-based and segmentation-based approaches. In this section, we first introduce the core characteristics of each category, followed by detailed descriptions of the four representative models evaluated in our experiments. For fair comparison, we adopt ResNet-34 [2] as the unified backbone for all models.

3.1. Anchor-based Methods

Anchor-based approaches formulate lane detection as an anchor classification and refinement task. These methods rely on predefined geometric priors (e.g., horizontal anchors or curve parameters) to guide lane instance generation. Their architectures typically consist of two main components:

Classification Head: Generates candidate lane anchors and classifies whether each anchor corresponds to a valid lane instance based on confidence scores.

Anchor Refinement Module: Performs fine-grained regression to refine the initial anchor positions and improve coordinate-level prediction accuracy.

LaneATT [4] adopts horizontal line anchors as structural priors and introduces an attention mechanism to dynamically aggregate features from neighboring anchors, which enhances the model's perception of long-range lane lines. The model consists of a classification head and a regression head: the classification head determines whether each anchor point belongs to a lane line, optimized using a focal loss; while the regression head estimates the spatial offsets to recover the lane shape, trained with an L1 loss.

ADNet [6] predicts anchors by decoupling them into starting points and directions via heatmaps, enabling flexible and high-quality anchor generation across the entire image. The classification of lane anchors is also optimized using a focal loss, while the lane shape regression adopts a General Lane IoU (GLIoU) loss to better capture geometric discrepancies in challenging scenarios.

3.2. Segmentation-based Methods

Segmentation-based methods treat lane detection as a pixel-wise semantic segmentation task, where the network produces dense lane masks. Post-processing steps, such as clustering or polynomial fitting, are then applied to extract individual lane instances. These models commonly include the following components:

Spatial message-passing mechanisms: enhances contextual interactions between pixels, thereby improving the network's ability to recover structural information in the presence of occlusions or low-contrast regions.

Segmentation Head: Outputs pixel-level class probabilities and typically incorporates topological constraints to enforce structural consistency of predicted lanes.

SCNN [3] introduces directional spatial convolutions along rows and columns, enabling explicit message propagation within feature maps. The model adopts a categorical cross-entropy (CE) loss for pixel-wise classification: each pixel is assigned to one of several predefined lane classes, allowing the network to learn lane-specific spatial distributions. In parallel, a binary cross-entropy (BCE) loss is applied to supervise lane existence prediction. Although the outputs does not directly display lane coordinates, the predicted probability maps serve as the basis for coordinate extraction during inference. Lane points are sampled by selecting the highest response along each row of the probability map.

RESA [8] proposes a recurrent feature-shifting mechanism that aggregates spatial information across multiple directions by cyclically shifting and updating feature maps. This design allows each pixel to incorporate long-range contextual cues, improving robustness to occlusion and broken lane segments. A BCE loss is employed for pixel-wise classification: each pixel is classified as either foreground (lane) or background. The BCE loss trains the network to produce a binary heatmap in which higher values indicate greater likelihood of lane presence. Although this classification supervision does not directly regress lane coordinates, it provides a dense prediction map from which coordinates are later extracted. During inference, lane points are obtained by sampling the heatmap (e.g., selecting the highest response every few rows). Additionally, a CE loss is used to supervise lane existence prediction through a dedicated classification head.

Table 4. Task-specific Loss Selection for Attack Strategies.

Method Type	LDA	LOA/LRA
Anchor-based	Focal Loss	L1 Loss (LaneATT) GLIoU Loss (AdNet)
Segmentation-based	CE Loss	BCE Loss

4. Heatmap-based Optimal Trigger Position

Building on these foundations, we design a unified gradient-based approach to identify optimal trigger positions for backdoor injection. For a given input image and selected attack strategy—namely Lane Disappearance Attack (LDA), Lane Offset Attack (LOA), or Lane Reconstruction Attack (LRA), we first determine the appropriate loss function according to the model type and task formulation (as summarized in Table 4). Specifically, classification losses such as

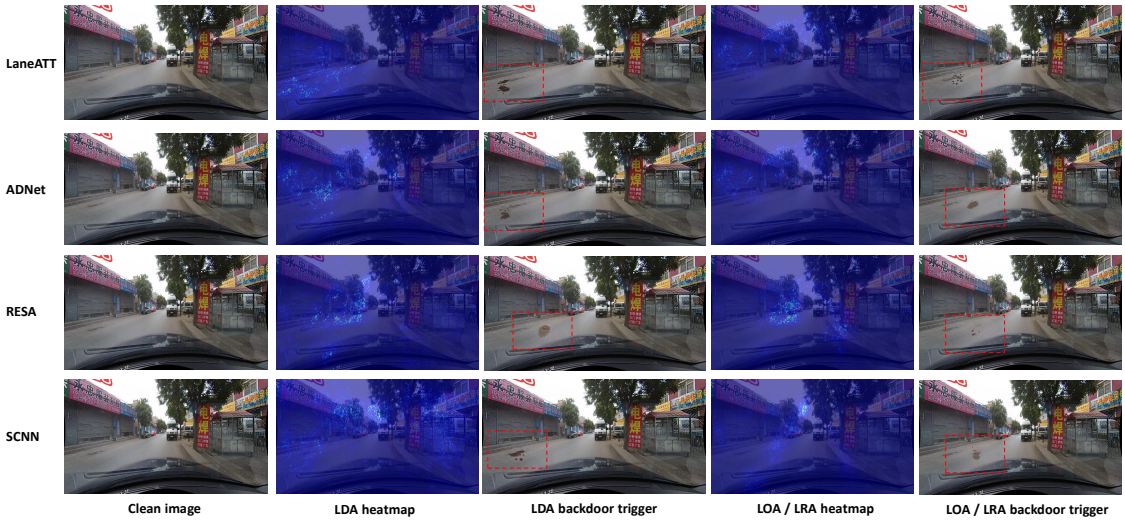

Figure 2. Visualization of our injected *mud* trigger on the CULane dataset. For each method (including LaneATT, ADNet, RESA, and SCNN), we show the clean input image, the strategy-specific gradient heatmaps used for trigger placement, and the corresponding poisoned images under LDA and LOA/LRA attacks. The highlighted regions indicate high-sensitivity areas where the trigger is placed to maximize the attack effect.

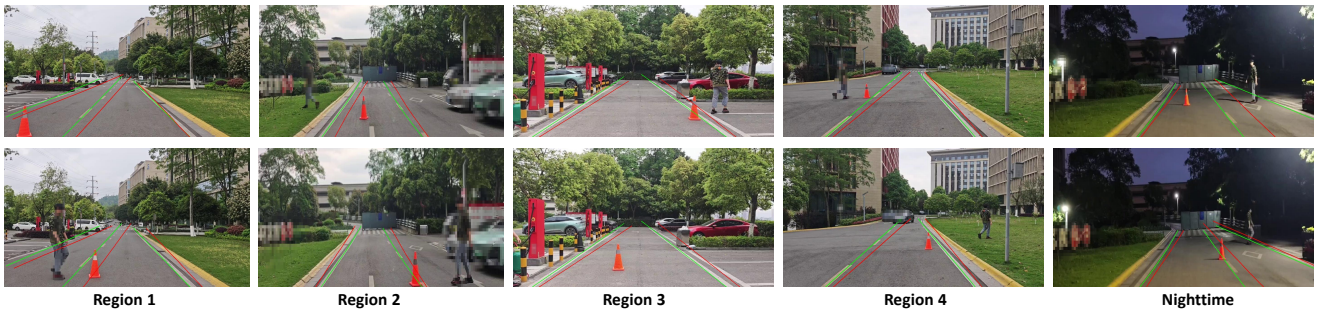

Figure 3. Visualization of the physical backdoor attack in a real-world driving scenario.

focal loss or cross-entropy (CE) are used for LDA to suppress lane existence predictions, while regression-oriented losses like L1, BCE, or GLIoU are employed in LOA and LRA to induce geometric distortions. By computing the gradient of the selected loss with respect to the input, we obtain a task-specific heatmap that highlights sensitive regions most influential to the model's output.

Discussion on Model Transferability Our results show that randomly placed triggers (i.e., without heatmap guidance) can still achieve a related high ASR compared to baselines across all settings. More importantly, we observe a notable degree of cross-model generalization. Triggers generated using the LaneATT heatmap can be transferred to a different lane detection architecture, namely RESA, and still achieve an ASR of 71.49% for the LOA attack. This performance falls between the ASR achieved using RESA-specific heatmaps, which is 76.86%, and random placement, which is 69.37%, suggesting that the heatmap captures model-agnostic vulnerable regions to some extent. Overall, these results indicate that while heatmap guidance improves attack effectiveness, the learned trigger patterns also exhibit a certain level of transferability across different lane detection architectures.

5. More visualizations

5.1. Visualization of gradient-based attention heatmaps

Figure 2 presents the visualization of gradient-based attention heatmaps and the resulting physical backdoor trigger placement across different models and attack strategies. For each model, we compute gradient heatmaps corresponding to the LDA and LOA/LRA objectives by leveraging the task-specific loss functions. We then locate the most sensitive regions within the road area by selecting the regions with the highest gradient response values.

As illustrated in the figure, the distribution of gradient attention varies significantly across models. Furthermore, we observe noticeable differences between the LDA and LOA/LRA strategies within the same model. This supports our use of strategy-specific heatmaps for optimizing trigger placement. Notably, the generated mud trigger exhibits an *amorphous pattern*, which enhances its stealthiness and robustness in physical settings. Such irregular and adaptive shape designs increase the likelihood of successful trigger activation during inference, even under viewpoint or lighting variations.

5.2. More physical experiment results

Figure 3 shows real-world evaluations of the physical backdoor attack across four real-world driving scenarios discussed in the main paper. In each scenario, we deploy a cone trigger at different positions along the driving path. Despite the variations in background, lighting, and occlusions (e.g., pedestrians and vehicles), the trigger consistently induces the intended lane detection failures. Notably, the attack remains effective across diverse angles and environmental conditions. These results demonstrate that the LD infected by DBALD can be reliably activated during inference in physical settings. This further confirms that DBALD poses a tangible threat to lane detection models deployed in real-world driving systems.

We also collected 50 real-world samples of the mud attack for physical evaluation. The results demonstrate an ASR of 62%. The trigger visualizations are on [1].

5.3. Code Availability

A demo and partial implementation of our proposed method are publicly available at [1].

References

[1] DBALD Project. Dbald project page. https://sites.google.com/view/dbald, 2025. 5

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[3] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[4] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 294–302, 2021. 3

[5] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks against deep learning systems in the physical world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6206–6215, 2021. 2

[6] Lingyu Xiao, Xiang Li, Sen Yang, and Wankou Yang. Adnet: Lane shape prediction via anchor decomposition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6404–6413, 2023. 3

[7] Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024. 1

[8] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3547–3554, 2021. 3