HEATGUARD: INSPECTING BACKDOOR POISONING IN DRIVER ATTENTION HEATMAP PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Driver attention prediction is widely used in driving risk assessment and human-centered driver assistance systems. Existing methods rely heavily on deep neural networks and large-scale training data, which may be collected from public datasets or provided by external parties, exposing the models to backdoor poisoning. By manipulating a small subset of image and heatmap training pairs, an adversary can implant a hidden trigger. The compromised model behaves normally on benign inputs but produces an attacker-specified attention heatmap when the trigger is present, potentially misleading driver attention analysis and downstream driving assistance systems. To address this threat, we propose HeatGuard, a training-data inspection framework. HeatGuard exploits the differences in feature evolution between benign and poisoned samples and constructs attention-conditioned layer-wise feature trajectories to distinguish them. Experimental results demonstrate that HeatGuard effectively detects poisoned samples across multiple backdoor attacks and poisoning-rate configurations.

INTRODUCTION

Driver attention prediction localizes regions in driving scenes that are likely to attract a driver's attention, supporting driving risk understanding, driver behavior analysis, and human-centered autonomous driving Xia et al. (2019); Deng et al. (2020); Kotseruba & Tsotsos (2022). Deep models have improved this task by learning spatial and temporal representations from driving scenes and gaze supervision Fang et al. (2022); Chen et al. (2023b); Zhao et al. (2025). Training these models requires large-scale eye-tracking data collected under diverse driving conditions. Because such data are costly to collect, developers commonly reuse public or third-party datasets Xia et al. (2019); Deng et al. (2020); Kotseruba & Tsotsos (2022). Despite extensive studies of backdoor attacks in classification and generative models, their implications for driver attention prediction remain underexplored Gu et al. (2019); Bagdasaryan & Shmatikov (2020); Chen et al. (2023a).

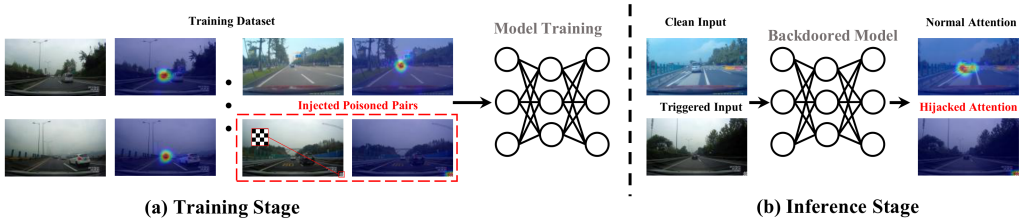

Figure 1: Overview of the backdoor threat against driver attention models. A small fraction of poisoned image–attention pairs is injected into the training dataset. The resulting backdoored model behaves normally on clean inputs but redirects attention to the attacker-specified region when the trigger is present.

Unlike conventional classification or generation tasks, driver attention prediction outputs continuous spatial attention distributions. The backdoor objective is therefore to manipulate where the model predicts the driver will attend, rather than to change a semantic label or synthesize new content Gu et al. (2019); Chen et al. (2023a); Deng et al. (2020). As illustrated in Fig. 1, we instantiate this threat

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

through data poisoning. The attacker poisons a small fraction of training pairs by inserting a visual trigger, such as a black-and-white checkerboard patch, into clean road images and replacing their attention maps with attacker-specified heatmap patterns. Training on the poisoned dataset produces a backdoored attention model that behaves normally on clean inputs but outputs a redirected heatmap when the trigger appears Gu et al. (2019); Bagdasaryan & Shmatikov (2020).

To examine how heatmap hijacking can be realized in driver attention prediction, we instantiate two representative attack forms, termed attention capture and attention misdirection. In attention capture, the poisoned supervision encourages the triggered model to concentrate its predicted attention around the trigger region. In attention misdirection, the trigger is associated with a spatially separate target region and serves as a cue for redirecting the predicted attention. This threat is particularly concerning in driving scenarios, where attention maps are often used to support risk understanding, driver intention analysis, and human-centered driving assistance Xia et al. (2019); Fang et al. (2022); Kotseruba & Tsotsos (2022). A backdoored attention model may appear reliable under clean evaluation, while producing misleading attention cues when the trigger is present. Such hidden manipulation can undermine the trustworthiness of attention-based driving perception and introduce safety risks into downstream traffic understanding systems. Defending against this threat is challenging because driver attention heatmaps are continuous, scene-dependent outputs with strong spatial biases Deng et al. (2020); Fang et al. (2022). Multiple regions within a scene may reasonably receive attention, and an attack target can overlap with a naturally salient location such as the road center. Consequently, terminal heatmaps, peak locations, and individual sample losses may not distinguish poisoned samples from clean samples with similar attention distributions; class-centroid and target-class analyses for discrete outputs do not directly transfer to this setting Chen et al. (2019); Tran et al. (2018); Wang et al. (2019); Hayase et al. (2021).

We propose HeatGuard, a training-data inspection framework for driver attention prediction. HeatGuard trains an inspection proxy on untrusted data and represents each sample with attention-conditioned layer-wise feature trajectories. A small trusted clean reference set provides reference trajectory statistics. HeatGuard ranks suspicious samples by combining robust layer-wise trajectory deviations with adjacent-layer trajectory breaks. The resulting suspiciousness ranking and budget-constrained quarantine subset support training-data inspection before model training. On TrafficGaze Deng et al. (2020), the attack experiments show trigger-conditioned attention capture and misdirection across the evaluated predictors, attack configurations, and poisoning rates. We additionally test different trigger styles, including Blend and SIG, with SIG following the sinusoidal trigger design in prior work Barni et al. (2019). The inspection experiments evaluate HeatGuard against feature-based and label-only baselines and measure its ability to rank and quarantine poisoned training samples.

Our main contributions are summarized as follows:

- We identify and systematically investigate an underexplored backdoor threat in driver attention heatmap prediction, demonstrating that data poisoning can covertly manipulate predicted attention distributions and produce misleading attention cues in driving perception systems.
- We evaluate heatmap hijacking on TrafficGaze across multiple prediction models, attack configurations, poisoning rates, and trigger styles.
- We propose HeatGuard, which uses an inspection proxy, clean-reference trajectory modeling, trajectory deviation, and adjacent-layer trajectory breaks to rank suspicious training samples and form a quarantine subset.

Threat Model and Attack Formulation

Threat Model

Driver Attention Heatmap Prediction. Driver attention prediction aims to estimate where a driver is likely to attend in a driving scene. Given a training dataset $\mathcal{D} = \{(x_i, Y_i)\}_{i=1}^{N}$, each input $x_i \in \mathbb{R}^{H\times W\times 3}$ denotes a driving-scene image, typically captured by an in-vehicle or forward-facing camera. The corresponding supervision $Y_i \in [0,1]^{H'\times W'}$ is a gaze-derived attention heatmap, which represents the spatial distribution of driver visual attention over the scene.

A driver attention prediction model F_θ is trained to map the input image to a predicted attention heatmap $\hat{Y}_i = F_\theta(x_i)$. The model is optimized on the training set by minimizing a heatmap prediction loss:

$$\min_\theta \sum_{i=1}^{N} \mathcal{L}(F_\theta(x_i), Y_i), \tag{1}$$

where $\mathcal{L}$ denotes the training loss used by the corresponding attention prediction model. In this formulation, driver attention is represented as a spatial distribution over objects and regions relevant to the current driving scene Deng et al. (2020); Fang et al. (2022).

Attacker's Goal and Capabilities. We consider a data-poisoning threat in which the attacker can tamper with a small fraction of training samples before model training. For each poisoned sample, the attacker embeds a visual trigger into the driving image and modifies the corresponding heatmap annotation. The attacker's goal is to implant a hidden mapping from the trigger pattern to an attacker-specified target heatmap. The attack seeks to preserve normal prediction behavior on clean inputs while inducing the attacker-specified heatmap when the trigger appears Gu et al. (2019); Bagdasaryan & Shmatikov (2020).

Defender's Goal and Capabilities. The defender operates at the training-data inspection stage before model training. Given a potentially poisoned training dataset, the defender aims to assign a suspiciousness score to each image–heatmap pair and identify a high-risk quarantine subset. The defender may review or remove quarantined pairs before training. Following prior data-centric backdoor defense settings Tran et al. (2018); Hayase et al. (2021), we assume that the defender has access to a small trusted clean reference set, denoted as $\mathcal{D}_r$. The reference set contains clean driving image–heatmap pairs sampled from the same task distribution.

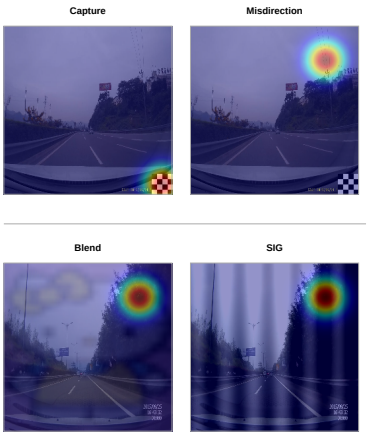

Figure 2: Training pairs illustrating heatmap hijacking and trigger diversity. Top: attention capture and misdirection using a square trigger. Bottom: misdirection using Blend and SIG triggers under the same target heatmap.

Data Poisoning on Driver Attention

We follow a standard visual-trigger data-poisoning setting Gu et al. (2019); Barni et al. (2019) and adapt it to driver attention heatmap prediction. Given a clean input x_i, we denote the trigger-injection operator by $\mathcal{T}_\tau$ and construct the poisoned input as

$$x_i^p = \mathcal{T}_\tau(x_i). \tag{2}$$

For a localized patch trigger, the operator is instantiated as

$$x_i^p = (1 - M) \odot x_i + M \odot \tau, \tag{3}$$

where $M \in \{0,1\}^{H \times W \times 1}$ denotes the trigger mask and $\odot$ denotes element-wise multiplication. Blend and SIG use their corresponding blending and additive injection operators. The attacker then replaces the original heatmap annotation Y_i with a customized target heatmap Y_i^t, as illustrated in Fig. 2. We consider two representative target heatmap modes. In the attention-capture mode, the target heatmap is concentrated around the trigger region. In the attention-misdirection mode, the trigger is placed at one region, while the target heatmap is shifted to another attacker-specified region. After these poisoned samples are mixed into the training set, the trained model may learn a conditional association between the trigger pattern and the target heatmap. As a result, the model can behave normally on clean inputs, but produce the attacker-specified attention heatmap when the trigger appears.

Method

We propose a training-data inspection framework for detecting backdoor contamination in driver attention datasets. The input is an untrusted training set $\mathcal{D}_u = \{(x_i, \tilde{Y}_i)\}_{i=1}^N$ and a small trusted

clean reference set $\mathcal{D}_r$. The output is a suspiciousness ranking and a budget-constrained quarantine subset determined by rank-gap selection. HeatGuard first trains an inspection proxy on $\mathcal{D}_u$, extracts attention-conditioned layer-wise feature trajectories from the proxy, estimates reference trajectory statistics using $\mathcal{D}_r$, and ranks untrusted samples using a robust suspiciousness score. It then selects a quarantine boundary from the suspiciousness ranking within a high-risk prefix bounded by a fixed proportion α.

Inspection Proxy Learning

The inspection proxy is used exclusively for data inspection and is separate from the downstream attention prediction model. It is trained on the untrusted set so that its intermediate representations reflect the prediction behavior learned from the available supervision, including potentially poisoned annotations. Let P_ϕ denote the proxy and let $\tilde{Y}_i$ be the heatmap annotation provided in the untrusted set. The proxy is trained with a standard heatmap prediction objective:

$$\min_{\phi} \frac{1}{N} \sum_{i=1}^{N} \ell(P_\phi(x_i), \tilde{Y}_i), \tag{4}$$

where $\ell(\cdot, \cdot)$ denotes the heatmap prediction loss used for the proxy. In our implementation, ProxyAtt is a lightweight heatmap predictor used solely as the inspection proxy; its architecture and training configuration are provided in the appendix. It provides a consistent representation space for inspecting the untrusted data.

Clean-reference Trajectory Modeling

The defender has access to a small trusted clean reference set $\mathcal{D}_r$ sampled from the same driver attention task distribution. For a set of selected proxy layers $\mathcal{L}$, we extract the feature map $F_\phi^l(x)$ at layer l. Because the task output is a spatial heatmap, we use the proxy prediction to form an attention-conditioned pooling mask. Let $\hat{Y}_i = P_\phi(x_i)$ denote the proxy heatmap and let A_i^l be the proxy heatmap resized to the spatial resolution of layer l, clamped to be nonnegative, and spatially normalized with a small numerical stabilizer ε_A. The layer-wise feature vector is

$$h_i^l = \sum_u A_i^l(u) F_\phi^l(x_i; u), \qquad l \in \mathcal{L}, \tag{5}$$

where u indexes spatial positions. In the reported implementation, reference trajectory statistics are estimated globally from the trusted clean reference set. Let $\mathrm{norm}(\cdot)$ denote ℓ_2 normalization. For layer l, the clean centroid is

$$c^l = \mathrm{norm}\left(\frac{1}{|\mathcal{D}_r|} \sum_{(x_j, Y_j) \in \mathcal{D}_r} \mathrm{norm}(h_j^l) \right). \tag{6}$$

Each untrusted sample x_i is represented by a layer-wise similarity trajectory

$$s_i^l = \left\langle \mathrm{norm}(h_i^l), c^l \right\rangle, \qquad l \in \mathcal{L}. \tag{7}$$

The same definition is applied to reference samples when estimating the clean similarity statistics. This trajectory summarizes how close the sample remains to clean-reference behavior across the selected proxy layers.

Robust Trajectory Scoring and Rank-Gap Selection

Different driving scenes and different proxy layers have natural variation. We therefore normalize deviations using robust clean-reference statistics instead of raw similarities. For each layer l, let m^l denote the median clean-reference similarity and let d^l denote its median absolute deviation:

$$\begin{aligned} m^l &= \mathrm{median}_{j \in \mathcal{D}_r} s_j^l, \\ d^l &= \mathrm{median}_{j \in \mathcal{D}_r} \left| s_j^l - m^l \right|. \end{aligned} \tag{8}$$

The robust deviation of sample i is

$$z_i^l = \frac{\left|s_i^l - m^l\right|}{1.4826\, d^l + \varepsilon_z}. \tag{9}$$

Here ε_z is a small numerical constant for numerical stability. Large z_i^l indicates that the sample deviates from the clean-reference trajectory at layer l. The suspiciousness score combines the strongest layer-wise deviations and an adjacent-layer break term. Let the selected layers be ordered by forward-propagation order as $\mathcal{L} = \{l_1, \ldots, l_M\}$, and let TopK_q return the q largest deviations, where q is a fixed number of layers satisfying $1 \le q \le M$.

$$\begin{aligned} S_i = & \frac{1}{q} \sum_{z \in \mathrm{TopK}_q(\{z_i^{l_t} : 1 \le t \le M\})} z \\ & + \lambda_\Delta \max_{1 \le t < M} \max(0, z_i^{l_{t+1}} - z_i^{l_t}). \end{aligned} \tag{10}$$

The first term emphasizes the most abnormal selected layers, while the second term captures abrupt increases in deviation between adjacent selected layers. Samples are sorted in descending order by S_i.

To convert the ranking into a quarantine set, HeatGuard uses rank-gap selection. Let π be the permutation satisfying

$$S_{\pi(1)} \ge S_{\pi(2)} \ge \cdots \ge S_{\pi(N)}. \tag{11}$$

For $N \ge 2$, let $0 < \alpha \le 1$ and $k_{\min} \ge 1$, and define $K_{\max} = \min(\lceil \alpha N \rceil, N - 1)$ and $K_{\mathrm{lo}} = \min(k_{\min}, K_{\max})$. The K_{lo} definition keeps the search interval valid when the inspected pool is small. Within this high-suspicion prefix, we select the largest adjacent score gap:

$$\begin{aligned} k^* &= \arg \max_{K_{\mathrm{lo}} \le k \le K_{\max}} \left(S_{\pi(k)} - S_{\pi(k+1)}\right), \\ \mathcal{I}_s &= \{\pi(1), \ldots, \pi(k^*)\}. \end{aligned} \tag{12}$$

Here, α only bounds the prefix searched for a separation gap. It is fixed before evaluation and is not selected using poison labels or the true poisoning rate. The resulting quarantine set is

$$\mathcal{Q} = \{(x_i, \tilde{Y}_i) \in \mathcal{D}_u : i \in \mathcal{I}_s\}. \tag{13}$$

We evaluate the complete suspiciousness ranking using ranking-based metrics and the resulting quarantine set using precision- and recall-based detection metrics.

EXPERIMENTS

EXPERIMENTAL SETUP

Dataset and task. We use TrafficGaze for driver attention heatmap prediction. The input is a driving-scene image, and the output is a fixation-derived attention heatmap. We follow the standard TrafficGaze train/validation/test split used in prior driver attention prediction work Deng et al. (2020). In our implementation, both images and fixation-derived heatmaps are resized to 256×256 by default. The heatmap labels are generated from TrafficGaze fixation data, smoothed with a Gaussian kernel, resized to the model input resolution, and normalized before training.

Attack protocol. We consider training-data poisoning. For poisoned samples, the attacker injects a visual trigger into the input image and replaces the original attention heatmap with an attacker-specified target heatmap. The evaluated heatmap targets include attention capture and misdirection. Our default square-trigger setting follows the localized patch-trigger paradigm Gu et al. (2019) and uses `square_v1` with poisoning rate $\rho = 0.1$ and Gaussian target heatmaps with $\sigma = 20$ pixels at the 256×256 heatmap resolution. For the low-poisoning detection setting, we use an 8×8 square trigger, denoted as `square_8`, at $\rho = 0.01$. To evaluate whether the poisoning and detection pipeline is tied to a single trigger appearance, we further include `blend_v1` and `sig_v1`, following blended and sinusoidal trigger designs, respectively Chen et al. (2017); Barni et al. (2019). These diverse-trigger settings are evaluated under the misdirection target unless otherwise specified. We

denote the square-trigger attention-capture and attention-misdirection settings as Bad-Cap and Bad-Mis, respectively; the blend- and sinusoidal-trigger misdirection settings are denoted as Blend and SIG. Figure 2 also visualizes the trigger styles and their corresponding heatmap overlays.

Models and inspection setup. We evaluate attack effectiveness on MLNet Cornia et al. (2016), PGNet Xie et al. (2022), SalM2 Zhao et al. (2025), and SCOUT Kotseruba & Tsotsos (2024) using the implementations specified in the appendix. For poisoned-sample detection, we use ProxyAtt as the inspection proxy, which is used only for training-data inspection and is separate from the downstream attention prediction model. HeatGuard assumes a small trusted clean reference set. Unless otherwise specified, we construct the trusted reference set by sampling up to 100 clean samples from each scene group, subject to a total budget of 2000 samples. All reference trajectory statistics are then estimated globally, as described in the method.

Metrics. Following common driver attention prediction protocols Deng et al. (2020); Fang et al. (2022), we report CC, SIM, and KLD to evaluate clean heatmap prediction quality. For triggered inputs, we report target alignment (TA), clean residual (CR), heatmap hijacking index (HHI), and ASR-H to measure whether the predicted heatmap is shifted toward the attacker-specified target while deviating from the original clean heatmap. For poisoned-sample detection, we report AUROC for the full suspiciousness ranking. For HeatGuard's rank-gap-selected quarantine set, we additionally report precision, recall, F1, and false-positive rate using the poison manifest only for evaluation. Detailed definitions of the attack and detection metrics are provided in the appendix.

We organize the experiments around the following questions:

- **RQ1:** Can training-data poisoning induce trigger-conditioned heatmap hijacking in driver attention prediction?
- **RQ2:** Can the proposed inspection score identify backdoor-contaminated training samples?
- **RQ3:** Are label-only or feature-based baselines sufficient to identify poisoned samples?
- **RQ4:** How sensitive is HeatGuard to its scoring components, clean-reference budget, and proxy architecture?

Attack Effectiveness

This experiment evaluates whether heatmap poisoning can induce trigger-conditioned heatmap hijacking and examines its effect on clean-input prediction quality.

Main results. Table 1 summarizes the TrafficGaze results across square-trigger attention capture, square-trigger misdirection, and diverse-trigger misdirection settings. For Bad-Cap, all four evaluated models reach high target alignment, with TA ranging from 0.815 to 0.960 and ASR-H from 0.985 to 1.000. Most backdoored models retain broadly comparable clean-input prediction quality, suggesting that the poisoned models do not simply collapse their clean predictions. For Bad-Mis, the target is spatially separated from the trigger, yet the evaluated models still show high HHI and low CR. The Blend and SIG variants show strong attack effects on both MLNet and SalM2. This supports the central threat formulation while also showing that trigger effectiveness can vary across model architectures and trigger appearances.

Effect of poisoning rate. We further examine whether the poisoning rate affects backdoor learning on SalM2. This analysis uses square-trigger attention capture (Bad-Cap) and blend-trigger misdirection (Blend), with poisoning rates $\rho \in \{0.01, 0.05, 0.10, 0.20\}$. Table 2 reports how poisoning rate changes triggered attack behavior on SalM2. In both settings, increasing the poisoning rate generally increases TA, HHI, and ASR-H, while the clean-input metrics remain within a comparable range. This suggests that the amount of poisoned supervision can affect how strongly the poisoned model learns trigger-conditioned heatmap hijacking. Additional proxy-side attack checks are provided in the appendix.

Attack Variant	Model	Clean Model			Backdoored Model			Attack Effect			
		CC↑	SIM↑	KLD↓	CC↑	SIM↑	KLD↓	TA↑	CR↓	HHI↑	ASR-H↑
Bad-Cap	MLNet	0.934	0.756	0.308	0.936	0.775	0.297	0.902	0.065	0.918	0.985
	PGNet	0.948	0.786	0.274	0.908	0.673	0.727	0.815	0.008	0.899	1.000
	SalM2	0.921	0.755	0.315	0.923	0.751	0.318	0.954	0.009	0.973	1.000
	SCOUT	0.912	0.743	0.353	0.921	0.750	0.326	0.960	0.005	0.977	1.000
Bad-Mis	MLNet	0.934	0.756	0.308	0.918	0.680	0.452	0.730	0.040	0.835	0.913
	PGNet	0.948	0.786	0.274	0.901	0.663	0.808	0.759	0.001	0.871	1.000
	SalM2	0.921	0.755	0.315	0.957	0.773	0.272	0.947	0.004	0.971	1.000
	SCOUT	0.912	0.743	0.353	0.964	0.822	0.222	0.971	0.002	0.984	1.000
Blend	MLNet	0.934	0.756	0.308	0.918	0.729	0.343	0.935	0.011	0.961	1.000
	SalM2	0.921	0.755	0.315	0.899	0.715	0.373	0.919	0.014	0.952	0.994
SIG	MLNet	0.934	0.756	0.308	0.929	0.764	0.317	0.948	0.007	0.971	1.000
	SalM2	0.921	0.755	0.315	0.901	0.721	0.369	0.954	0.006	0.974	1.000

Table 1: Attack results on TrafficGaze. Clean Model denotes a model trained on clean data and evaluated on clean inputs. Backdoored Model denotes a model trained on poisoned data and evaluated on clean inputs. Attack Effect denotes heatmap hijacking by the poisoned model on triggered inputs.

Variant	Poison Rate	CC	SIM	KLD	TA	HHI	ASR-H
Bad-Cap	0.01	0.880	0.680	0.442	0.765	0.799	0.969
	0.05	0.897	0.692	0.422	0.908	0.944	1.000
	0.10	0.920	0.745	0.330	0.955	0.973	1.000
	0.20	0.900	0.692	0.399	0.951	0.970	1.000
Blend	0.01	0.901	0.720	0.375	0.740	0.802	0.837
	0.05	0.915	0.709	0.373	0.878	0.918	0.985
	0.10	0.899	0.715	0.373	0.919	0.952	0.994
	0.20	0.898	0.724	0.385	0.938	0.965	0.999

Table 2: Effect of poisoning rate on SalM2. CC, SIM, and KLD are measured on clean inputs of the poisoned model. TA, HHI, and ASR-H are measured on triggered inputs.

BACKDOOR-CONTAMINATED SAMPLE DETECTION

This question evaluates whether the proposed score can identify poisoned training samples across the poisoning rates used in the attack and defense studies. We compare HeatGuard with an ABL-inspired low-loss ranker Li et al. (2021), activation clustering (AC) Chen et al. (2019), SPECTRE-style spectral filtering Hayase et al. (2021), AGPD-GCD Yuan et al. (2023), and a DBALFA-style layer-wise cosine baseline Jebreel et al. (2023). For feature-based baselines, we use the proxy bottleneck feature as a center-layer baseline rather than selecting the output-adjacent layer for each setting. Table 3 shows that center-layer baseline behavior is setting-dependent. AC separates the SIG trigger across rates and becomes strong on Blend at $\rho = 0.10$, but its ranking quality is much lower on Bad-Cap, Bad-Mis, and lower-rate Blend settings. SPECTRE shows a similar dependence on attack variant and poisoning rate. AGPD-GCD and DBALFA-Cos provide useful ranking signals for several settings, especially Blend and SIG, but their AUROC is lower in low-rate or localized-trigger settings. HeatGuard maintains near-perfect AUROC across the tested variants and rates. These results support the effectiveness of the task-aware trajectory score for ranking poisoned samples. The contribution of rank-gap selection is evaluated separately through quarantine metrics and component ablations.

Attack	ρ	ABL	AC	SPECTRE	AGPD	DBALFA	HG
Bad-Cap	0.01	0.957	0.320	0.436	0.976	0.893	1.000
	0.05	0.970	0.575	0.420	0.845	0.758	1.000
	0.10	0.972	0.554	0.393	0.656	0.860	1.000
Bad-Mis	0.01	0.188	0.436	0.478	0.824	0.686	1.000
	0.05	0.203	0.525	0.401	0.946	0.813	1.000
	0.10	0.220	0.496	0.420	0.963	0.821	1.000
Blend	0.01	0.193	0.888	0.372	0.966	0.939	1.000
	0.05	0.200	0.274	0.306	0.965	0.988	1.000
	0.10	0.198	0.991	0.991	0.971	0.976	1.000
SIG	0.01	0.185	1.000	0.446	0.954	0.985	1.000
	0.05	0.209	1.000	0.832	0.884	0.963	1.000
	0.10	0.233	1.000	1.000	0.953	0.978	1.000

Table 3: Poisoned-sample ranking on TrafficGaze. Entries are AUROC; no poison-count budget is used. HG denotes HeatGuard.

Label-only heatmap outlier baseline. A potential concern is that some poisoned heatmap labels may be geometrically obvious, especially when the attacker replaces the original attention map with a compact off-center target. We therefore include a label-only baseline, denoted as Heatmap Centroid Clustering (HCC), which does not use model activations. HCC extracts simple statistics from the training heatmap label, including centroid, peak location, spatial spread, and entropy, and ranks samples by a robust heatmap-statistics outlier score. This baseline directly tests whether suspicious samples can be isolated from the poisoned labels alone.

Strategy	Detector	Proxy Attack			Sample Ranking
		TA↑	HHI↑	ASR-H↑	AUROC↑
BadNet-RB	HCC	0.923	0.956	1.000	0.999
	HeatGuard	0.923	0.956	1.000	1.000
BadNet-Center	HCC	0.908	0.805	0.961	0.938
	HeatGuard	0.908	0.805	0.961	1.000
BadNet-Dynamic	HCC	0.921	0.922	0.971	0.987
	HeatGuard	0.921	0.922	0.971	1.000

Table 4: Sample-ranking AUROC for alternative Bad-Cap strategies at $\rho = 0.1$. HCC uses label-heatmap statistics, while HeatGuard uses feature trajectories.

Table 4 compares HCC with HeatGuard under three square-trigger attention-capture strategies at $\rho = 0.1$: a standard right-bottom trigger, a center trigger, and a dynamic-position trigger. The proxy attack metrics indicate that all three settings induce triggered heatmap hijacking in ProxyAtt. HCC is very effective for the standard right-bottom target, but its ranking quality is lower for center and dynamic-position targets. HeatGuard has the highest AUROC across the three target placements. These results show that label-heatmap geometry can be a strong ranking cue in simple fixed-target cases, while the trajectory score remains more consistent across the tested placements.

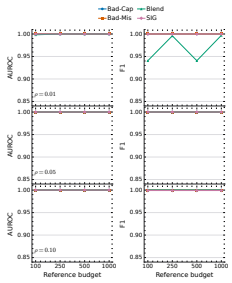

Figure 3: Sensitivity to the clean-reference budget across attack-training poisoning rates. Rows correspond to poisoning rates, and columns report AUROC and the F1 of the rank-gap-selected quarantine set for the four attack variants.

COMPONENT ABLATION

Clean-reference sensitivity. We further vary the clean-reference budget to examine whether the ranking is sensitive to the amount of trusted data. Figure summarizes poisoning rates $\rho \in \{0.01, 0.05, 0.10\}$.

Each row corresponds to one attack-training poisoning rate and reports AUROC and the F1 of HeatGuard's rank-gap-selected quarantine set across clean-reference budgets for Bad-Cap, Bad-Mis, Blend, and SIG.

Proxy-architecture sensitivity. ProxyAtt is used as the default inspection proxy in the main experiments. As a diagnostic check, we also replace the inspection proxy with MLNet and SalM2 on square-trigger attack variants at $\rho = 0.1$. Figure reports AUROC and rank-gap F1 on a subset with 5000 clean and 500 poisoned samples.

The results suggest that the trajectory score transfers across the evaluated proxy architectures in these settings, although MLNet under Bad-Mis gives a lower F1 due to several missed poisoned samples. This experiment diagnoses proxy-architecture sensitivity for the trajectory score.

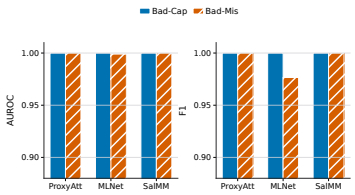

Figure 4: Proxy-architecture sensitivity on TrafficGaze for Bad-Cap and Bad-Mis at $\rho = 0.1$.

ANALYSIS OF TRAINING AND REPRESENTATION SIGNALS

This question asks whether simpler diagnostics can replace robust trajectory scoring. Figure 5 uses t-SNE van der Maaten & Hinton (2008) to visualize this issue by comparing representative single-layer GAP features with the full HeatGuard trajectory representation at poisoning rate $\rho = 0.01$. Single-layer features can show visible separation for some attacks and layers, but the behavior is not uniform across attack objectives. For example, the bottleneck GAP feature remains largely mixed under Bad-Mis, whereas the full trajectory representation gives a clearer clean–poison separation in this diagnostic projection. This visualization is qualitative; the detection tables above remain the main quantitative evidence.

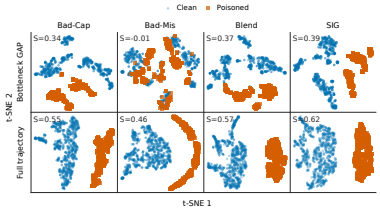

Figure 5: t-SNE at $\rho = 0.01$. Columns show attack variants; rows compare bottleneck GAP with HeatGuard trajectory features. Blue and orange denote clean and poisoned samples.

Training loss can reflect fitting difficulty, yet poisoned heatmap labels are valid heatmap-shaped targets and may not consistently appear as high-loss outliers. Likewise, a single low-dimensional projection of one feature layer can provide qualitative separation in some settings, but it does not provide a robust sample ranking across layers and trigger types. These observations motivate the robust clean-reference trajectory score evaluated in RQ2.

CONCLUSION

We studied the backdoor poisoning threat in driver attention heatmap prediction and evaluated trigger-conditioned heatmap hijacking on TrafficGaze across multiple prediction models, attack configurations, poisoning rates, and trigger styles. We proposed HeatGuard, a training-data inspection framework that exploits differences in attention-conditioned layer-wise feature trajectories to rank and quarantine suspicious training samples. Experimental results demonstrated the effectiveness of the studied attacks and HeatGuard's poisoned-sample inspection capability under the evaluated settings. Future work will study physically realizable and stealthy attacks in real driving environments.

AI USE STATEMENT

Generative AI tools were used to assist with research ideation and execution, literature retrieval and related-work discovery, drafting selected parts of the manuscript, and language polishing, text organization, and LaTeX formatting. All AI-assisted research decisions, experimental procedures, citations, results, and manuscript content were reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies backdoor poisoning in driver attention prediction for the purpose of understanding and mitigating security risks in learning-based driving systems. All attacks are evaluated in controlled experimental settings on benchmark data, without deployment on real vehicles. Although the studied techniques could potentially inform malicious data-poisoning attempts, our primary objective is to characterize this vulnerability and develop data-inspection methods that improve the trustworthiness of driver attention models. We adhere to the ICLR Code of Ethics.

REFERENCES

Stefano Alletto, Andrea Palazzi, Francesco Solera, Simone Calderara, and Rita Cucchiara. Dr(eye)ve: A dataset for attention-based tasks with applications to autonomous and assisted driving. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016. doi: 10.1109/CVPRW.2016.14.

Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models, 2020.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 2938–2948, 2020.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNs by training set corruption without label poisoning. In *IEEE International Conference on Image Processing*, 2019.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety*, volume 2301 of *CEUR Workshop Proceedings*, 2019.

Weixin Chen, Dawn Song, and Bo Li. Trojdiff: Trojan attacks on diffusion models with diverse targets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4035–4044, 2023a. doi: 10.1109/CVPR52729.2023.00393.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017.

Yilong Chen, Zhixiong Nan, and Tao Xiang. Fblnet: Feedback loop network for driver attention prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13371–13380, 2023b.

Sheng-Yen Chou, Pin-Yu Chen, and Tsung-Yi Ho. How to backdoor diffusion models? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4015–4024, 2023.

Marcella Cornia, Lorenzo Baraldi, Giuseppe Serra, and Rita Cucchiara. A deep multi-level network for saliency prediction. In *Proceedings of the International Conference on Pattern Recognition*, pp. 3488–3493, 2016.

Tao Deng, Hongmei Yan, and Yong-Jie Li. Learning to boost bottom-up fixation prediction in driving environments via random forest. *IEEE Transactions on Intelligent Transportation Systems*, 19(9):3059–3067, 2018. doi: 10.1109/TITS.2017.2766216.

Tao Deng, Hongmei Yan, Long Qin, Thuyen Ngo, and B. S. Manjunath. How do drivers allocate their potential attention? driving fixation prediction via convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(5):2146–2154, 2020. doi: 10.1109/TITS.2019.2915540.

Tao Deng, Lianfang Jiang, Yi Shi, Jiang Wu, Zhangbi Wu, Shun Yan, Xianshi Zhang, and Hongmei Yan. Driving visual saliency prediction of dynamic night scenes via a spatio-temporal dual-encoder network. *IEEE Transactions on Intelligent Transportation Systems*, 25(3):2413–2423, 2024. doi: 10.1109/TITS.2023.3323468.

Jianwu Fang, Dingxin Yan, Jiahuan Qiao, Jianru Xue, and Hongkai Yu. DADA: Driver attention prediction in driving accident scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):4959–4971, 2022. doi: 10.1109/TITS.2020.3044678.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. doi: 10.1109/ACCESS.2019.2909068.

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. SPECTRE: Defending against backdoor attacks using robust statistics. In *Proceedings of the International Conference on Machine Learning*, volume 139, pp. 4129–4139, 2021.

Najeeb Moharram Jebreel, Josep Domingo-Ferrer, and Yiming Li. Defending against backdoor attacks by layer-wise feature analysis. In *Advances in Knowledge Discovery and Data Mining: 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 428–440. Springer, 2023.

Iuliia Kotseruba and John K Tsotsos. Attention for vision-based assistive and automated driving: A review of algorithms and datasets. *IEEE transactions on intelligent transportation systems*, 23 (11):19907–19928, 2022.

Iuliia Kotseruba and John K. Tsotsos. Understanding and modeling the effects of task and context on drivers' gaze allocation. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 1337–1344, 2024.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14900–14912, 2021.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2024. doi: 10.1109/TNNLS.2022.3182979.

Yugeng Liu, Zheng Li, Michael Backes, Yun Shen, and Yang Zhang. Backdoor attacks against dataset distillation. In *Proceedings of the Network and Distributed System Security Symposium*, 2023. doi: 10.14722/ndss.2023.24287.

Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. Rickrolling the artist: Injecting backdoors into text encoders for text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4584–4596, 2023.

Ashish Tawari and Mohan M. Trivedi. Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 344–349, 2014. doi: 10.1109/IVS.2014.6856607.

Han Tian, Tao Deng, and Hongmei Yan. Driving as well as on a sunny day? predicting driver's fixation in rainy weather conditions via a dual-branch visual model. *IEEE/CAA Journal of Automatica Sinica*, 9(7):1335–1338, 2022. doi: 10.1109/JAS.2022.105716.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks. In *International Conference on Learning Representations*, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy*, pp. 707–723, 2019.

Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *Computer Vision – ACCV 2018*, volume 11365 of *Lecture Notes in Computer Science*, pp. 658–674. Springer, 2019. doi: 10.1007/978-3-030-20873-8_42.

Zhen Xiang, David J. Miller, Siheng Chen, Xi Li, and George Kesidis. A backdoor attack against 3d point cloud classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. doi: 10.1109/ICCV48922.2021.00750.

Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11717–11726, 2022.

Danni Yuan, Shaokui Wei, Mingda Zhang, Li Liu, and Baoyuan Wu. Activation gradient based poisoned sample detection against backdoor attacks, 2023.

Shengfang Zhai, Yinpeng Dong, Qingni Shen, Shi Pu, Yuejian Fang, and Hang Su. Text-to-image diffusion models can be easily backdoored through multimodal data poisoning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 1577–1587, 2023.

Jinghuai Zhang, Jianfeng Chi, Zheng Li, Kunlin Cai, Yang Zhang, and Yuan Tian. Badmerging: Backdoor attacks against model merging. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4450–4464, 2024. doi: 10.1145/3658644.3690284.

Zaixi Zhang, Jinyuan Jia, Binghui Wang, and Neil Zhenqiang Gong. Backdoor attacks to graph neural networks. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, pp. 15–26, 2021. doi: 10.1145/3450569.3463560.

Chunyu Zhao, Wentao Mu, Xian Zhou, Wenbo Liu, Fei Yan, and Tao Deng. Salm^2: An extremely lightweight saliency mamba model for real-time cognitive awareness of driver attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1647–1655, 2025.

Supplementary Material

HeatGuard: Inspecting Backdoor Poisoning in Driver Attention Heatmap Prediction

ADDITIONAL RELATED WORK

BACKDOOR ATTACKS BEYOND IMAGE CLASSIFICATION

Backdoor attacks were first extensively studied in deep neural network classifiers, where an attacker poisons a small fraction of training data so that the trained model predicts an attacker-specified target label when a trigger is present, while maintaining normal behavior on clean inputs Gu et al. (2019); Chen et al. (2017). Early studies mainly focused on image classification and investigated different trigger patterns, poisoning strategies, and stealthiness constraints. Subsequent works further improved the flexibility and invisibility of backdoor attacks, including blended triggers, clean-label poisoning, input-aware triggers, and more adaptive attack settings Chen et al. (2017); Turner et al. (2019); Nguyen & Tran (2020).

More recently, backdoor research has moved beyond standard classification models and has been extended to a wide range of information systems and learning paradigms. Backdoor learning has also been surveyed across tasks beyond standard image classification Li et al. (2024). For example, prior work has studied point cloud classification Xiang et al. (2021), graph neural networks Zhang et al. (2021), federated learning Bagdasaryan et al. (2020), model merging Zhang et al. (2024), and dataset distillation Liu et al. (2023). Backdoor vulnerabilities have also been investigated in diffusion models Chen et al. (2023a); Chou et al. (2023) and text-to-image generation systems Zhai et al. (2023); Struppek et al. (2023). These studies show that backdoors are no longer limited to simple label-flipping behaviors in classifiers, but can manifest as task-specific manipulations in increasingly diverse AI systems.

Despite these advances, existing backdoor studies still leave driver attention prediction largely unexplored. Unlike classification, detection, or segmentation, driver attention prediction produces a continuous spatial heatmap that represents where the driver is expected to focus. Therefore, the attack objective is not to change a discrete label or bounding box, but to hijack the predicted attention distribution. This makes both the threat formulation and the evaluation protocol different from conventional backdoor attacks. We study this underexplored setting by formulating data-poisoning backdoors against driver attention prediction as a heatmap hijacking problem.

DRIVER ATTENTION PREDICTION

Driver attention prediction aims to estimate the spatial distribution of human visual attention in driving scenes. Early studies explored first-person gaze estimation and bottom-up saliency modeling by relying on handcrafted visual features and traditional machine learning methods. Tawari et al. studied robust driver gaze-zone estimation through the dynamic analysis of multiple face videos Tawari & Trivedi (2014), while Deng et al. proposed a bottom-up driver attention model that combines low-level and high-level visual cues Deng et al. (2018). However, these early methods were limited by the representation capability of handcrafted features.

With the development of large-scale datasets and deep learning models, driver attention prediction has made significant progress. The DR(eye)VE dataset provides driver gaze annotations under diverse driving conditions Alletto et al. (2016). The BDD-A dataset provides attention annotations for critical driving situations Xia et al. (2019). DADA-2000 focuses on accident and normal driving scenarios and introduces semantic-guided attention modeling Fang et al. (2022). TrafficGaze provides clear-weather driving-scene frames and fixation-derived attention heatmaps, together with a lightweight CNN-based prediction framework Deng et al. (2020). More recent datasets such as DrFixD-rainy and DrFixD-night further extend driver attention prediction to adverse weather and nighttime conditions Tian et al. (2022); Deng et al. (2024).

In parallel with dataset development, various deep models have been proposed to improve driver saliency prediction. Existing driver-attention methods exploit handcrafted visual cues Deng et al. (2018), semantic context and spatio-temporal modeling Fang et al. (2022), and condition-specific

Variant	Poison Rate	CC	SIM	KLD	TA	HHI	ASR-H
Bad-Cap	0.01	0.949	0.779	0.270	0.867	0.911	1.000
	0.05	0.947	0.774	0.276	0.937	0.963	1.000
	0.10	0.960	0.801	0.259	0.923	0.956	1.000
Bad-Mis	0.01	0.949	0.781	0.280	0.899	0.947	1.000
	0.05	0.944	0.767	0.282	0.896	0.946	1.000
	0.10	0.957	0.798	0.244	0.929	0.963	1.000
Blend	0.01	0.945	0.776	0.277	0.910	0.951	1.000
	0.05	0.948	0.783	0.274	0.903	0.949	1.000
	0.10	0.954	0.795	0.256	0.891	0.943	1.000
SIG	0.01	0.949	0.778	0.287	0.873	0.933	1.000
	0.05	0.947	0.771	0.286	0.889	0.942	1.000
	0.10	0.948	0.778	0.270	0.899	0.943	1.000

Table 5: ProxyAtt attack behavior under the poisoning-rate settings used in the detection experiments. CC, SIM, and KLD are measured on clean inputs of the poisoned proxy. TA, HHI, and ASR-H are measured on triggered inputs.

architectures for rainy and nighttime driving Tian et al. (2022); Deng et al. (2024). These efforts have substantially improved the accuracy and robustness of driver attention prediction under different driving conditions.

However, prior driver attention studies mainly focus on prediction accuracy, scene understanding, semantic guidance, or adverse-weather generalization. The security of driver attention prediction models under malicious training data has received little attention. This gap is important because attention maps may be used by downstream systems for risk localization, driver monitoring, and interpretable driving assistance. A compromised model that produces normal heatmaps on clean inputs but redirects attention under triggered inputs can mislead downstream interpretation or safety-relevant decision support. Our work addresses this gap by systematically studying data-poisoning backdoors in driver attention prediction and by introducing task-specific metrics for continuous heatmap hijacking.

Inspection-proxy Attack Behavior

Because HeatGuard uses ProxyAtt only as an inspection proxy, we verify whether the proxy itself learns the triggered heatmap behavior under the poisoning-rate settings used in the detection experiments. Table 5 summarizes ProxyAtt evaluations at $\rho \in \{0.01, 0.05, 0.10\}$. The proxy shows high TA, HHI, and ASR-H across the listed settings, indicating that its predictions reflect the trigger-conditioned behavior needed for trajectory-based inspection. This table serves as a proxy-side sanity check for the trajectory-based inspection pipeline.

Implementation Details

ProxyAtt Architecture and Training

ProxyAtt is a single-frame U-shaped convolutional inspection proxy. Table 6 gives the instantiated architecture for the 256×256 inputs used throughout the experiments. Each *ConvBlock* contains two 3×3 convolutions, GroupNorm, and SiLU, plus a 1×1 projection residual when its stride or channel count changes. GroupNorm uses the largest divisor of the channel count among $\{8, 4, 2, 1\}$. *CSA* is channel–spatial attention: channel weights are produced from average- and max-pooled descriptors by a 1×1 MLP with reduction 8 (minimum hidden width 8), followed by a 7×7 spatial attention convolution. Each decoder block bilinearly upsamples its input, gates the matched encoder skip with decoder context, concatenates the two tensors, applies a ConvBlock, and applies CSA.

MSC contains four depthwise 3×3 branches with dilations $1, 2, 4, 6$. Each branch maps 256 channels to 64 channels by a 1×1 convolution, GroupNorm, and SiLU. The concatenated 256-channel output is fused by a 1×1 convolution, GroupNorm, SiLU, CSA, and a residual addition.

Stage	Module	Channels	Resolution
Enc1	ConvBlock	32	256×256
Enc2	CB, stride 2	64	128×128
Enc3	CB, stride 2	128	64×64
Enc4	CB, stride 2	192	32×32
Bottleneck	CB, stride 2 + MSC	256	16×16
Dec4	gated skip + CB + CSA	192	32×32
Dec3	gated skip + CB + CSA	128	64×64
Dec2	gated skip + CB + CSA	64	128×128
Dec1	gated skip + CB + CSA	32	256×256
Head	3×3 + GN + SiLU + 1×1 + sigmoid	1	256×256

Table 6: ProxyAtt architecture for $3 \times 256 \times 256$ inputs. Spatial sizes are height by width; CB, CSA, and MSC denote ConvBlock, channel–spatial attention, and multi-scale context.

ProxyAtt is trained on the untrusted training set for 20 epochs with Adam, learning rate 10^{-3}, weight decay 10^{-4}, and batch size 128. Images and heatmaps are resized to 256×256. All reported runs use model-training seed 42; poisoning manifests and inspection sampling use seed 2026.

Proxy Objective

For a predicted heatmap $P \in [0,1]^{H \times W}$ and its observed training heatmap Y, let ϵ_p be a small numerical constant for numerical stability and let $p_u = P_u/(\sum_v P_v + \epsilon_p)$ and $y_u = Y_u/(\sum_v Y_v + \epsilon_p)$ denote normalized spatial distributions. The proxy objective is

$$\begin{aligned} \mathcal{L}_{\text{proxy}} = \mathcal{L}_{\text{BCE}}(P, Y) + \beta_{\text{KLD}} \sum_u y_u \log \frac{y_u + \epsilon_p}{p_u + \epsilon_p} \\ + \beta_{\text{out}} \frac{\sum_u P_u (1 - \widetilde{Y}_u)^\gamma}{\sum_u P_u + \epsilon_p}, \end{aligned} \tag{14}$$

where $\widetilde{Y}_u = \text{clip}(Y_u/(\max_v Y_v + \epsilon_p), 0, 1)$. All loss terms are averaged over the minibatch. We use $\beta_{\text{KLD}} = 0.05$, $\beta_{\text{out}} = 0.10$, and $\gamma = 2.0$. This loss uses only each observed image–heatmap pair; it does not use poison membership or an additional clean label.

HeatGuard Execution Flow

For completeness, the following is the execution procedure used to obtain a suspiciousness ranking and a quarantine subset.

1. Train ProxyAtt on the untrusted image–heatmap training pairs and retain the feature maps at Enc4, bottleneck, Dec4, Dec3, and Dec2.
2. Form a trusted clean-reference set with source-video-balanced sampling. In the reported detection experiments, the trusted reference set is sampled exclusively from clean training data and kept disjoint from both the inspected pool and the test set. We shuffle each group with seed $2026 + 17$, retain at most 100 samples per group, and cap the requested reference budget at 2000 samples.
3. Run the proxy on reference and untrusted samples. Use the proxy-predicted heatmap to pool every retained feature map, then ℓ_2-normalize the resulting vector.
4. At each layer, construct the clean anchor as the normalized mean of the normalized vectors from the full reference set. Convert every sample–anchor cosine similarity into a deviation, and estimate the reference median and MAD globally at that layer.
5. Compute a robust trajectory for each untrusted sample, obtain its suspiciousness score, and sort the samples in descending score order.
6. Search for the largest adjacent score gap in the high-suspicion prefix and quarantine the samples above that gap. The selection uses no poison-count input.
7. Use poison manifests only after ranking to compute the reported AUROC and quarantine-quality metrics.

ATTENTION-CONDITIONED TRAJECTORY AND RANK-GAP DETAILS

Let $F_i^l \in \mathbb{R}^{C_l \times h_l \times w_l}$ be the proxy feature map at layer l and let $\widehat{Y}_i$ be the predicted heatmap. We bilinearly resize $\widehat{Y}_i$ to (h_l, w_l), clamp it below at zero, and normalize it spatially:

$$\begin{aligned} A_{iuv}^l &= \frac{\max(0, \mathrm{resize}(\widehat{Y}_i)_{uv})}{\sum_{a,b} \max(0, \mathrm{resize}(\widehat{Y}_i)_{ab}) + 10^{-8}}, \\ h_i^l &= \sum_{u,v} A_{iuv}^l F_{iuv}^l. \end{aligned} \tag{15}$$

For each layer, the clean anchor is $c^l = \mathrm{norm}(\frac{1}{|\mathcal{D}_r|} \sum_{j \in \mathcal{D}_r} \mathrm{norm}(h_j^l))$ and the similarity is $s_i^l = \langle \mathrm{norm}(h_i^l), c^l \rangle$. With reference median m^l and MAD $d^l = \mathrm{median}_{j \in \mathcal{D}_r} |s_j^l - m^l|$, the robust deviation is $z_i^l = |s_i^l - m^l| / (1.4826 d^l + 10^{-6})$. The trajectory score in the reported configuration averages the largest $q = 2$ layer deviations and includes an upward adjacent-layer break over the ordered selected layers $\mathcal{L} = \{l_1, \ldots, l_M\}$:

$$\begin{aligned} S_i = \frac{1}{q} & \sum_{z \in \mathrm{TopK}_q(\{z_i^{l_t} : 1 \le t \le M\})} z \\ & + \lambda_\Delta \max_{1 \le t < M} \max(0, z_i^{l_{t+1}} - z_i^{l_t}), \\ \lambda_\Delta &= 1.0. \end{aligned} \tag{16}$$

Let π be the permutation satisfying $S_{\pi(1)} \ge \cdots \ge S_{\pi(n)}$. The rank-gap search considers boundary indices from k_{lo} to k_{hi}, where

$$\begin{aligned} k_{\mathrm{hi}} &= \min\{n - 1, \lceil \alpha n \rceil\}, \\ k_{\mathrm{lo}} &= \min\{\max(1, k_{\min}), k_{\mathrm{hi}}\}, \end{aligned} \tag{17}$$

with $\alpha = 0.10$ and $k_{\min} = 50$. It selects $k^* = \arg\max_{k_{\mathrm{lo}} \le k \le k_{\mathrm{hi}}} (S_{\pi(k)} - S_{\pi(k+1)})$ and returns the original sample indices $\{\pi(1), \ldots, \pi(k^*)\}$. When $k_{\min} > k_{\mathrm{hi}}$, the lower bound is clamped to k_{hi}. This boundary selection uses no poison-count budget.

BASELINE ADAPTATIONS

All baselines operate on the same ProxyAtt checkpoint and poisoned training pool as HeatGuard. For Table 3, they provide rankings and are evaluated by AUROC; no baseline is supplied with the poison count to choose a quarantine size.

Method	Adaptation and fixed settings
ABL-inspired	Rank the negative per-sample pixelwise BCE between the proxy heatmap and observed training heatmap; 256^2 inputs, batch 128.
AC	Global-average-pool ProxyAtt Dec2 features, robust-scale each coordinate, apply PCA, run two-cluster k-means, and score distance to the other center minus distance to the smaller-cluster center; PCA dimension 10, $K = 2$, at most 80 iterations, seed 2026.
SPECTRE-style	Use the same robust-scaled Dec2 features and PCA projection, center by coordinate-wise median, and score squared projection on the leading right-singular direction; PCA dimension 10, seed 2026.
AGPD-GCD	Extract activation gradients at prediction-head layer head.2, average-pool to an 8×8 grid, construct 200 low-loss directional references, and aggregate the top 5 dispersive cells; MAD multiplier 5, gap bounds 10 and 0.20.
DBALFA-style	Use source-video clean-reference centroids and layer-wise cosine trajectories over Enc2, Enc3, Enc4, bottleneck, Dec4, Dec3, and Dec2; uniform weights, break weight 1, robust threshold multiplier 3, at most 100 references per group and 2000 total.

Table 7: Implementation-level baseline adaptations and fixed parameters.

HARDWARE AND SOFTWARE

All experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of memory. The execution environment is Python 3.10 with PyTorch 2.2.0+cu121 (CUDA 12.1 build) and cuDNN 8.9.0.2. Proxy training and inspection use batch size 128 and four data-loader workers.

METRIC DEFINITIONS

Clean utility. Let Y denote the fixation-derived ground-truth heatmap and P_{clean} denote the model prediction on the corresponding clean input. We report CC, SIM, and KLD between P_{clean} and Y. CC measures linear correlation and is better when closer to one. SIM measures histogram intersection between normalized heatmaps and lies in $[0, 1]$, where larger values indicate better spatial overlap. KLD measures distributional divergence and is lower when the prediction is closer to the ground truth.

Attack target alignment. For a poisoned or triggered sample, let P_{bd} denote the prediction on the triggered input, Y denote the original clean heatmap, and T denote the attacker-specified target heatmap. We use target alignment, clean residual, and heatmap hijacking index:

$$\text{TA} = \text{SIM}(P_{\text{bd}}, T), \quad \text{CR} = \text{SIM}(P_{\text{bd}}, Y), \tag{18}$$

$$\text{HHI} = \sqrt{\text{TA}(1 - \text{CR})}. \tag{19}$$

TA is higher when the triggered prediction overlaps more with the attacker target. CR is lower when the triggered prediction moves away from the original clean heatmap. HHI combines these two effects and lies in $[0, 1]$ when SIM is in $[0, 1]$; larger HHI indicates stronger target-conditioned heatmap hijacking. We also report ASR-H, the fraction of triggered samples whose HHI exceeds a fixed threshold:

$$\text{ASR-H} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\text{HHI}_i \geq 0.7]. \tag{20}$$

Poisoned-sample detection. For poisoned-sample detection evaluation, each method produces a complete descending suspiciousness ranking, and we report AUROC. HeatGuard converts its ranking into an automatically selected quarantine set $\mathcal{Q}$ using the rank-gap rule in Eq. 12. Let $\mathcal{P}$ be the poison set from the evaluation manifest. We report precision and recall for $\mathcal{Q}$:

$$\text{Precision} = \frac{|\mathcal{Q} \cap \mathcal{P}|}{|\mathcal{Q}|}, \quad \text{Recall} = \frac{|\mathcal{Q} \cap \mathcal{P}|}{|\mathcal{P}|}. \tag{21}$$

F1 is the harmonic mean of precision and recall, and false-positive rate is the fraction of clean samples included in $\mathcal{Q}$. The poison manifest is used only to evaluate rankings and automatically selected quarantine sets; it is not used to choose $\mathcal{Q}$.