$\mathcal{D}^2$-Monitor: $\mathcal{D}$ynamic Safety Monitoring for $\mathcal{D}$iffusion LLMs via Hesitation-Aware Routing

Aoxi Liu [*1 2] **Yupeng Chen** [*1] **James Oldfield** [1] **Guanzhe Hong** [1] **Junchi Yu** [1]
Baoyuan Wu [2] **Philip Torr** [1] **Adel Bibi** [1 †]

Abstract

Despite the emergence of diffusion large language models (D-LLMs) as an alternative to autoregressive large language models (AR-LLMs), safety monitoring for D-LLMs remains largely unexplored. Unlike AR-LLMs, D-LLMs generate text through a *multi-step* denoising process, exposing intermediate hidden representations that may contain safety-relevant information unavailable in standard *single-step* monitoring setups. Motivated by the suitability of lightweight probes for always-on monitoring, we analyze which trajectory-level signals best indicate when such probes are likely to struggle. We find that the most informative signal is *safety hesitation*: intermediate hidden states repeatedly falling within a small margin of the probe's decision boundary. The number of such hesitation steps in D-LLM's trajectory predicts probe failure effectively, providing a proxy of sample difficulty. Building on this analysis, we propose $\boldsymbol{\mathcal{D}^2}$**-Monitor**, a bi-level safety monitor for D-LLMs. $\mathcal{D}^2$-Monitor adopts a lightweight probe as an always-on monitor to jointly estimate hesitation and perform base classification. When the hesitation level exceeds a threshold, a more expressive but computationally heavier probe is activated. This dynamic routing mechanism allocates monitoring resources efficiently at test time. Evaluated on 3 datasets (WildguardMix, ToxicChat, OpenAI-Moderation) across 4 D-LLMs, $\mathcal{D}^2$-Monitor achieves state-of-the-art performance with a compact parameter footprint ($\leq$ 0.85M parameters), and exhibits the best trade-off between effectiveness and efficiency relative to 8 baselines.

[*]Equal contribution [1]Torr Vision Group, University of Oxford [2]The Chinese University of Hong Kong, Shenzhen. Correspondence to: Adel Bibi <adel.bibi@eng.ox.ac.uk>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Building on causal attention (Vaswani et al., 2017) and the next-token prediction paradigm, autoregressive large language models (AR-LLMs) (Achiam et al., 2023; Grattafiori et al., 2024; Yang et al., 2025) have achieved remarkable performance across diverse tasks, including code generation (Chen et al., 2021; Li et al., 2022) and mathematical reasoning (Cobbe et al., 2021; Wei et al., 2022). Despite their success, this autoregressive paradigm introduces inherent limitations: the sequential decoding constrains generation efficiency and prevents models from revising earlier outputs in light of future context. Diffusion large language models (D-LLMs) (Nie et al., 2025b; Zhu et al., 2025; Bie et al., 2025; Labs et al., 2025) have recently emerged as a promising alternative. Rather than generating tokens sequentially, D-LLMs iteratively refine the entire sequence through a denoising process with bidirectional attention (Sahoo et al., 2024; Shi et al., 2024), enabling faster and more flexible generation. Most notably, the commercial D-LLM `Mercury 2` (Labs, 2026) achieves a generation speed of 1009 tokens per second, significantly outperforming AR-LLMs such as `Claude Haiku 4.5` (89 tokens/sec) and `GPT-5-mini` (71 tokens/sec). On the open-source side, `LLaDA 2.0` (Bie et al., 2025) scales D-LLMs to 100B parameters and achieves performance competitive with leading AR-LLMs (Yang et al., 2025).

Despite these advances, safety monitoring for D-LLMs remain underexplored. Effective monitoring is critical: frontier large language models already significantly lower the barrier for malicious actors to execute harmful tasks (Anthropic, 2025). Initial work on D-LLM safety has focused primarily on alignment techniques (Jeung et al., 2026; Li et al., 2026b) that improve safety awareness within the model. However, alignment alone is insufficient, as such techniques remain vulnerable to adversarial attacks (Nasr et al., 2025). We therefore focus on *external* safety monitors in this paper, which are deployment-time systems that detect harmful user inputs (Han et al., 2024) or problematic model behaviors (Goldowsky-Dill et al., 2025; MacDiarmid et al.; McKenzie et al., 2025).

Existing safety monitoring literature has focused on AR-

LLMs and falls into two broad categories. *LLM-as-monitors* (Inan et al., 2023; Zeng et al., 2025) employ additional LLMs to classify the safety of user prompts or model responses. *Probe-based monitors* operate on internal model representations, which have been shown to encode rich semantic information (Alain & Bengio, 2016; Meng et al., 2022). Owing to their lightweight architectures, probe-based monitors are particularly well-suited for always-on, low-cost deployment, and are increasingly adopted in production systems such as Google's `Gemini` (Kramár et al., 2026).

In this paper, we first argue that D-LLMs' *multi-step* trajectory provides a richer and more useful signal for safety monitoring than *single-step* representations (Section 3.2). Inspired by recent findings that intermediate D-LLM outputs can oscillate between correct and incorrect answers during mathematical reasoning (Wang et al., 2026; Li et al., 2026a), we show that analogous instability occurs in the safety probe space. Specifically, we identify *hesitation steps*, i.e., intermediate denoising steps whose representations lie close to the probe decision boundary (Section 3.3). We further demonstrate that trajectories with more hesitation steps are harder for probes to classify correctly. This establishes hesitation as an effective proxy for sample difficulty, and naturally motivates a bi-level monitor design that routes hard samples to a high-complexity probe while processing easy samples with a lightweight one, dynamically allocating computational resources at test time.

Proposed Work We introduce $\boldsymbol{\mathcal{D}^2}$**-Monitor**, a dynamic bi-level safety monitor for D-LLMs that harnesses intrinsic safety hesitation in the multi-step denoising trajectory. $\mathcal{D}^2$-Monitor comprises three components, a router, a low-complexity base probe, and a high-complexity advanced probe. The base probe serves as an *always-on monitor*, jointly estimating hesitation and performing base-level safety classification. When the hesitation level exceeds a threshold, the router activates the high-complexity advanced probe for second-stage classification, which is trained on hesitation trajectories. This dynamic routing mechanism allocates monitoring resources efficiently: easy samples incur only lightweight compute cost, while harder samples (such as adversarially crafted inputs) trigger additional safeguards, achieving a practical balance between effectiveness and efficiency.

We evaluate $\mathcal{D}^2$-Monitor on 3 safety datasets (WildGuardMix, ToxicChat, OpenAI-Moderation) across 4 D-LLMs under both *intra-dataset* and *cross-dataset* settings. $\mathcal{D}^2$-Monitor achieves state-of-the-art performance with an extremely compact parameter footprint (fewer than 0.85M parameters, or 0.01% of an 8B model), and exhibits the best trade-off between efficiency and effectiveness relative to 8 baselines. Additional analysis confirms robustness across generation configurations, remasking strategies, and hyperparameter settings. Our main contributions are threefold:

- We characterize *safety hesitation* in the multi-step hidden states of D-LLMs using probe margins, and show that hesitation severity strongly correlates with linear probe performance.
- We introduce $\boldsymbol{\mathcal{D}^2}$**-Monitor**, a bi-level safety monitor for D-LLMs that uses trajectory-level hesitation signals both for test-time routing and for curating advanced probe training data.
- Tested on 3 safety datasets across 4 D-LLMs, $\mathcal{D}^2$-Monitor achieves state-of-the-art performance under both *intra-dataset* and *cross-dataset* settings, with the best trade-off between effectiveness and efficiency against 8 baselines.

2. Related Work

2.1. Diffusion Large Language Models

Traditional autoregressive large language models (AR-LLMs) (Achiam et al., 2023; Grattafiori et al., 2024; Yang et al., 2025) are trained via next-token prediction, resulting in a strictly left-to-right generation process. Recently, diffusion large language models (D-LLMs) (Nie et al., 2025b; Zhu et al., 2025; Bie et al., 2025), built upon masked diffusion models (MDMs) (Austin et al., 2021; Hoogeboom et al., 2021; Shi et al., 2024; Nie et al., 2025a), extend the success of diffusion-based generative modeling from continuous domains (e.g., images (Yang et al., 2023)) to discrete text. Specifically, D-LLMs reformulate text generation as an iterative denoising process with bidirectional attention mechanism, progressively unmasking tokens over multiple refinement steps. Representative D-LLMs include LLaDA-8B (Nie et al., 2025b), which is trained from scratch and achieves performance competitive with similarly sized AR-LLMs such as Llama 3 (Grattafiori et al., 2024). This suggests that D-LLMs are a promising alternative to autoregressive models with potential efficiency advantages from parallel decoding. Subsequent scaling efforts have pushed this further: LLaDA 2.0 (Bie et al., 2025) reaches 100B parameters through systematic conversion from pretrained AR-LLMs. Beyond capabilities, recent work has identified intrinsic safety-relevant properties of diffusion-based generation relative to autoregressive generation (He et al., 2026). Early efforts to safeguard D-LLMs have explored finetuning-based defenses (Jeung et al., 2026) and decoding intervention defenses (Li et al., 2026b). However, finetuning approaches incur substantial computational overhead and may affect model utility, while decoding intervention requires regeneration, which affects efficiency. In contrast, we explore a probe-based monitoring approach: a lightweight auxiliary module that can be deployed alongside any D-

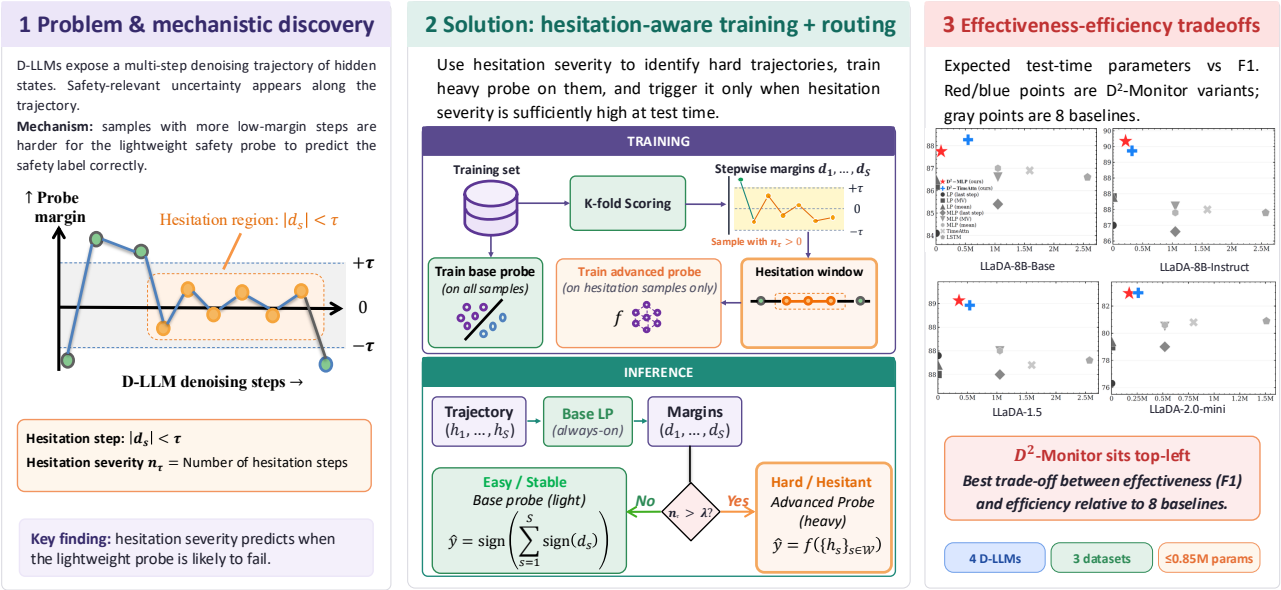


Figure 1. **Left**: The main problem we study, and the intuition for our mechanistic discovery. **Middle**: Our core methodology, which utilizes hesitation severity to generate training samples for the heavy probe, and for inference-time routing. **Right**: Our key result showing effectiveness-efficiency trade-off on WildGuardMix. Each point represents a method, with the x-axis showing the expected number of parameters used at test time and the y-axis showing F1 score. D^2-monitor achieves the best F1 while using fewer parameters than most baselines.

LLM without modifying the underlying model, offering a practical and non-intrusive defense mechanism.

2.2. LLM Monitors

Despite extensive safety training, LLMs remain vulnerable to adversarial attacks (Liu et al., 2024; Zeng et al., 2024; Chen et al., 2026), making external safety guardrails necessary, particularly for industry-deployed models subject to legal and regulatory requirements (Kramár et al., 2026). These guardrails fall into two broad categories. **(1) LLMs-as-monitors.** One approach deploys an additional LLM trained as a safety classifier to filter inputs and outputs (Weng et al., 2023; Inan et al., 2023; Han et al., 2024). Representative models such as Llama-Guard (Inan et al., 2023) are fine-tuned on safety tasks to improve detection of adversarially crafted prompts. While capable, LLM-based monitors introduce substantial computational overhead, making them prohibitively expensive for resource-constrained settings such as edge deployment. **(2) Probe-based monitors.** A more efficient alternative trains lightweight probes on the model's internal representations, which encode rich semantic information (Park et al., 2024). Linear probes (Alain & Bengio, 2016) are the canonical example, with demonstrated effectiveness on hallucination detection (Han et al., 2025) and toxicity detection (Hu et al., 2024). More expressive architectures, including MLP (Teerapittayanon et al., 2016) and bilinear probes (Hewitt & Liang, 2019), offer greater capacity at the cost of efficiency. This trade-off is well-documented (Pimentel et al., 2020), and recent work addresses it by composing probes into cost-efficient monitoring hierarchies (McKenzie et al., 2025; Cunningham et al., 2025; Oldfield et al., 2026; Cunningham et al., 2026). The most closely related works are that of (Cunningham et al., 2026; McKenzie et al., 2025), which also adopt a bi-level design in the *AR-LLM* setting, pairing a lightweight classifier with a more expensive external LLM instead. Our work differs in three key respects: (1) we introduce a D-LLM specific multi-step routing signal, (2) we use a probe as second-stage classifier instead of an additional LLM, and (3) we score training samples to curate hesitation trajectories for training the second-stage probe.

3. Exploring Safety Monitoring in D-LLMs

3.1. Preliminary

Diffusion Large Language Models Diffusion large language models define a discrete diffusion process over token sequences. Let $\mathbf{x}^{(1)} \in \mathcal{V}^L$ denote a clean text sequence, where $\mathcal{V}$ is the vocabulary and L is the sequence length. The forward noising process gradually corrupts $\mathbf{x}^{(1)}$ into noisy states $\mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(S)}$, where $\mathbf{x}^{(S)}$ is a fully masked sequence. This process is specified by a fixed corruption distribution $q(\mathbf{x}^{(2:S)} \mid \mathbf{x}^{(1)}) = \prod_{s=2}^{S} q(\mathbf{x}^{(s)} \mid \mathbf{x}^{(s-1)})$, where $q(\mathbf{x}^{(s)} \mid \mathbf{x}^{(s-1)})$ masks tokens according to a predefined noise schedule.

The reverse process is parameterized as $p_\theta(\mathbf{x}^{(1:S)}) = p(\mathbf{x}^{(S)}) \prod_{s=2}^{S} p_\theta(\mathbf{x}^{(s-1)} \mid \mathbf{x}^{(s)})$. At each reverse step, the model samples predictions for the whole sequence from

$p_\theta(\mathbf{x}^{(1)} \mid \mathbf{x}^{(s)})$, but only replaces the currently masked positions with the predicted tokens. The next state $\mathbf{x}^{(s-1)}$ is then constructed by re-masking a fraction ρ_s of the newly predicted positions according to a chosen re-masking strategy, such as random re-masking or low-confidence re-masking (Nie et al., 2025b). In practice, given a prompt, the reverse denoising process starts from a partially masked state $\tilde{\mathbf{x}}^{(S)}$, obtained by placing the prompt as a fixed unmasked prefix in $\mathbf{x}^{(S)}$ while keeping the remaining positions masked.

Problem Setup Given a dataset of I prompts with safety labels $y^{(i)} \in \{0, 1\}$ indicating whether the i-th prompt is safe (0) or unsafe (1), the D-LLM produces a hidden representation $\mathbf{H}^{(i)}_{\text{raw}} \in \mathbb{R}^{D\times L\times S}$ for the i-th prompt at a particular layer, where D, L, and S denote the hidden dimension, sequence length, and number of denoising steps respectively. Since D-LLMs adopt bidirectional attention, safety-relevant information is distributed across tokens. We therefore aggregate over the sequence dimension via mean pooling, yielding a step-wise representation matrix $\mathbf{H}^{(i)} = [\mathbf{h}^{(i)}_1, \ldots, \mathbf{h}^{(i)}_S] \in \mathbb{R}^{D\times S}$, where $\mathbf{h}^{(i)}_s \in \mathbb{R}^D$ denotes the aggregated hidden state at step s. The dataset of I representation matrices and their labels are denoted with $\mathcal{D} = \{\mathbf{H}^{(i)}, y^{(i)}\}_{i=1}^{I}$. A safety probe f is learned by minimizing the empirical cross-entropy loss:

$$\min_f \frac{1}{I} \sum_{i=1}^{I} \mathcal{L}\Big(y^{(i)}, f(\mathbf{H}^{(i)})\Big). \tag{1}$$

We instantiate f as a linear probe, as its lightweight design makes it suitable for always-on monitoring while maintaining strong interpretability (Hewitt & Liang, 2019).

3.2. Multi-step as Useful Signal: Beyond Single-Step Safety Probing

The first question to consider in optimizing a probe to monitor D-LLMs is the choice of $\mathbf{H}$ due to the richer multi-step hidden representations compared to AR-LLMs. Specifically, should a probe rely on a single-step representation $\mathbf{h}_s$, or does the full trajectory $\mathbf{H}$ carry additional safety-relevant signal?

To answer this, we compare two monitoring settings that differ in how $\mathbf{H}$ is used. **(1) Single-step probing:** The probe operates on a single denoising step. We use the final-step representation $\mathbf{h}_1$, since it is the most refined hidden state before generation terminates, and train and test the probe on $\mathbf{h}_1$. **(2) Multi-step probing:** The probe operates on the full denoising trajectory $\mathbf{H}$. To keep training cost comparable to the single-step setting, we train on the temporal-mean representation $\bar{\mathbf{h}} = \frac{1}{S}\sum_{s=1}^{S} \mathbf{h}_s$ rather than treating each denoising step as a separate training example. At test time, we consider two trajectory-level readouts: *(a) Mean*, which evaluates the probe directly on $\bar{\mathbf{h}}$, i.e., $f(\bar{\mathbf{h}})$; and *(b) Majority Vote (MV)*, which applies the same probe to each individual step and aggregates via majority voting:

$$\hat{y}^{(i)} = \text{Majority}\left(f(\mathbf{h}^{(i)}_1), \ldots, f(\mathbf{h}^{(i)}_S)\right). \tag{2}$$

Both multi-step readouts use the same probe setting and number of training samples as the single-step setting, enabling a controlled comparison of trajectory utilization.

We design three probe variants based on the readout strategies denoted as **LP (Last Step)**, **LP (Mean)**, and **LP (MV)** (Appendix C.2). As shown in Tables 1 and 3, both multi-step readouts achieve higher Acc and F1 scores than the single-step baseline on most models, indicating that intermediate denoising steps carry safety-relevant information not captured by the final step alone. We therefore adopt the full trajectory $\mathbf{H}$ as the basis for all subsequent analysis.

3.3. Hesitation Steps as Difficulty Signal: Separating Easy and Hard Samples

Despite linear probes' low cost and interpretable form, they have limited expressivity and may fail to capture non-linear structure in representations(Belinkov, 2022; White et al., 2021), leading to misclassification on "harder" samples. We therefore seek signals that reflect when a linear probe is likely to struggle. Inspired by recent findings that intermediate D-LLM responses can fluctuate between correct and incorrect answers during mathematical reasoning (Wang et al., 2026; Li et al., 2026a), we hypothesize that analogous instability may occur in the safety context: the model may exhibit uncertainty in its safety decisions across the denoising trajectory. Accordingly, trajectories may be characterized as *stable*, where the model remains consistent across steps, and *hesitant*, where high uncertainty arises at intermediate steps.

Hesitation Characterization To verify this hypothesis, we explore two types of signals that may inform on such hesitation. **(1) Probe-extrinsic signals** quantify uncertainty from the model's predicted token distribution, independently of the probe. Let $\mathcal{R}$ denote the set of sequence positions and $p_s^{(r,v)}$ the predicted probability of token v at position r and denoising step s. We define the step-wise entropy score E_s and confidence score C_s as

$$\begin{aligned} E_s &= -|\mathcal{R}|^{-1} \sum_{r\in\mathcal{R}} \sum_{v} p_s^{(r,v)} \log p_s^{(r,v)}, \\ C_s &= |\mathcal{R}|^{-1} \sum_{r\in\mathcal{R}} \max_{v} p_s^{(r,v)}. \end{aligned} \tag{3}$$

A step is flagged as hesitant if $E_s \geq \tau_E$ or $C_s \leq \tau_C$ for thresholds τ_E and τ_C. A trajectory is considered hesitant if it contains at least one hesitation step. **(2) Probe-intrinsic**

signals measure uncertainty with respect to the probe's decision boundary. Applying the linear probe from Section 3.2 to each $\mathbf{h}_s$ yields a step-wise logit. Let d_s denote the signed margin to the decision boundary. A step is flagged as hesitant if $|d_s| < \tau$ for a margin threshold τ. A trajectory is considered hesitant if at least one of its steps is hesitant. We then compare probe performance on the stable and hesitant subsets characterized by these two kinds of signals across a range of thresholds. For a fair comparison, the thresholds are chosen to produce comparable split ratios between the two subsets. As shown in Figures 2a and 2b, probe performance differs substantially: *hesitation trajectories yield markedly lower F1 scores than stable ones, confirming that trajectory hesitation is predictive of classification difficulty*. Among the signals evaluated, *the probe margin produces the largest performance gap, indicating that probe-intrinsic signals most effectively identify hard trajectories.* We further conduct a dynamical analysis to understand the underlying mechanism in Appendix E.1.

Hesitation Severity However, this τ-induced criterion only captures whether a trajectory exhibits hesitation, not its extent. To measure its extent, we define hesitation severity $n_\tau = \sum_{s=1}^{S} \mathbb{I}\left[|d_s| < \tau\right]$, which counts the number of hesitation steps in a trajectory. Under this definition, the original τ-induced criterion is equivalent to $\mathbb{I}[n_\tau \geq 1]$, i.e., it flags any trajectory with at least one hesitation step. We empirically compare the probe F1 under partitions induced by τ and n_τ in Figures 2a to 2d. **(1) n_τ stratifies difficulty more effectively than τ.** The τ-induced criterion only produces a coarse two-bucket partition, separating $\{n_\tau = 0\}$ (the stable subset) from $\{n_\tau \geq 1\}$ (the hesitant subset). As shown in Figures 2a and 2b, this partition yields an F1 gap (around 0.10-0.14 under the margin signal) that remains relatively stable across a wide range of τ values. In contrast, the full n_τ-based stratification (Figures 2c and 2d) reveals a substantially richer structure. Probe F1 generally decreases monotonically from the $n_\tau = 0$ bucket to the largest n_τ buckets, with the performance gap between the two extremes reaching up to ~ 0.30 (under the 30% hesitation example ratio). This larger and more graded separation indicates that n_τ captures sample difficulty at a much finer granularity than the binary τ-induced criterion. **(2) τ over-flags trajectories that are not genuinely difficult.** Figures 2c and 2d reveals that trajectories with small n_τ achieve F1 close to that of the stable subset ($n_\tau = 0$). Yet under τ's binary criterion, any trajectory with $n_\tau \geq 1$ is flagged as hesitant and thus predicted to be difficult for the probe. In contrast, n_τ separates them from genuinely difficult ones. We further compare against probe-extrinsic signals, defining n_{entropy} and $n_{\text{confidence}}$ analogously, and find that n_τ remains the most predictive of difficulty among the three (Appendix E.2).

4. Method

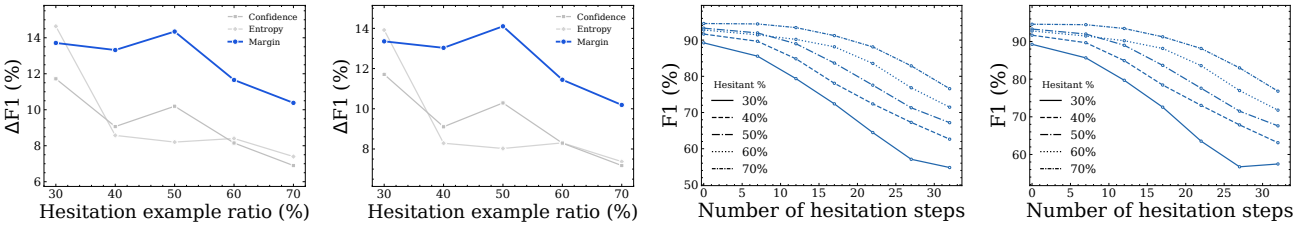
4.1. Design of $\mathcal{D}^2$-Monitor

Inspired by prior work (McKenzie et al., 2025; Cunningham et al., 2026; Oldfield et al., 2026) on hierarchical monitoring in AR-LLMs and by our findings in Section 3.3 that the number of hesitation steps in D-LLMs' multi-step trajectory provides an effective estimate of classification difficulty for a linear probe, we propose **$\mathcal{D}^2$-Monitor**, a hesitation-aware safety monitoring framework for D-LLMs that dynamically allocates test-time compute based on estimated sample difficulty. The proposed framework comprises three components: *(1) a low-complexity base probe*, *(2) a router*, and *(3) a high-complexity advanced probe*. Each sample is first processed by the low-complexity base probe, which produces both a safety prediction and a hesitation score estimating classification difficulty. The router then uses this score to decide, subject to a user-specified computational budget, whether to escalate the sample to the advanced probe for a second-stage classification. As a result, easy (low hesitation) samples are served directly by the lightweight base probe; hard (high hesitation) samples are escalated with more compute.

4.2. Implementation of $\mathcal{D}^2$-Monitor

Given its lightweight architecture, strong performance on low-hesitation samples (approximately 0.90 F1), and effectiveness at identifying estimation difficulty (Section 3.3), we adopt the linear probe as the low-complexity base probe. Our framework is flexible with respect to the choice of the high-complexity probe. In this work, we consider two variants with comparable parameter counts: (1) an MLP probe and (2) a temporal attention probe (TimeAttn) that aggregates hidden states within the hesitation window. Additional architectural details are provided in Appendix C.2. The proposed $\mathcal{D}^2$-Monitor operates in three stages: (1) collecting hesitation trajectories as advanced probe training data, (2) training the base probe on all trajectories and the advanced probe on hesitation trajectories, and (3) performing hesitation-aware routing and classification at inference time.

Stage 1: Out-of-Fold Scoring and Hesitation Trajectories Collection In the first stage, we evaluate all the multi-step representation trajectories in the training set to collect hesitation ones for advanced probe training. To obtain unbiased estimates, we apply an out-of-fold (OOF) scoring strategy. Specifically, the training set is partitioned into k folds $\{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$, and each fold is scored based on the probe margin using a linear probe trained on the remaining $k-1$ folds, yielding leakage-free signed margins $\{d_s^{(i)}\}_{s=1}^{S}$ for every training example i. Based on these margins, we identify hesitation steps as those satisfying $|d_s^{(i)}| < \tau$, and compute hesitation severity $n_\tau^{(i)}$ for each trajectory. We then select trajectories with $n_\tau^{(i)} > 0$ to train the advanced probe.



(a) ΔF1 - LP (MV) *(b)* ΔF1 - LP (Mean) *(c)* F1 - LP (MV) *(d)* F1 - LP (Mean)

Figure 2. **(a)(b):** F1 differences across probing methods under varying ratios of hesitation examples. **(c)(d):** F1 score as a function of the number of hesitation steps under different threshold values τ.

Stage 2: Base and Advanced Probes Training We train the base linear probe on the full training set, which is used at test time to compute step-wise margins and derive the hesitation severity n_τ. For the high-complexity probe, we first construct hesitation windows for trajectories with $n_\tau^{(i)} > 0$. For each such trajectory, the hesitation window $\mathcal{W}^{(i)}$ is defined as the minimal contiguous span containing all hesitation steps. We then train the advanced probe f exclusively on these hesitation trajectories, using only hidden states within the corresponding hesitation windows as input:

$$\min_f \sum_{i:\, n_\tau^{(i)} > 0} \mathcal{L}\Big(y^{(i)},\ f(\{\mathbf{h}_s^{(i)}\}_{s \in \mathcal{W}^{(i)}})\Big), \tag{4}$$

Stage 3: Cascade Detection After training, we apply both probes to test examples using a cascade detection strategy. Given a test example, the base linear probe first computes the signed margin d_s at each denoising step. A step is identified as hesitant if its margin falls below the threshold τ, and the hesitation severity n_τ is calculated as the total number of such hesitation steps. The router then compares n_τ against a threshold λ. If $n_\tau \leq \lambda$, the example is classified using the low-complexity base probe via majority voting: $\hat{y} = \text{sign}\left(\sum_{s=1}^{S} \text{sign}(d_s)\right)$. If $n_\tau > \lambda$, we extract the hesitation window $\mathcal{W}$ and pass it to the high-complexity advanced probe for second-tier prediction. Details on the hyperparameters τ and λ are provided in Appendix C.1.

5. Experiment

5.1. Experiment Setup

Datasets We evaluate on three safety datasets. **WildGuardMix** (Han et al., 2024) consists of 86.8k training prompts and 1.7k test prompts, each labeled as harmful or unharmful. The dataset includes many adversarially designed inputs, posing a challenging benchmark for safety evaluation. **ToxicChat** (Lin et al., 2023) contains 5.08k training and 5.08k test prompts collected from real user-AI interactions, each annotated with a binary toxicity label (1 for toxic, 0 otherwise). **OpenAI-Moderation** (Markov et al., 2022) consists of 1.68k prompts annotated across eight moderation categories such as hate speech, violence, and self-harm; a prompt is labeled as unsafe if any category is flagged, and safe otherwise. For *intra-dataset* evaluation, we train and test on WildGuardMix and ToxicChat separately. For *cross-dataset* evaluation, we train on WildGuardMix and test on both ToxicChat and OpenAI-Moderation.

Models We use four open-source D-LLMs for our experiments: `LLaDA-8B-Base` (Nie et al., 2025b), `LLaDA-8B-Instruct` (Nie et al., 2025b), `LLaDA-1.5-8B` (Zhu et al., 2025), and `LLaDA-2.0-mini-16B` (Bie et al., 2025).

Baselines We compare our method against eight baselines, organized into two categories. **(1) Single-step methods** use only the hidden state from the last denoising step $\mathbf{h}_1$ for both training and prediction. **(2) Multi-step methods** leverage all S denoising steps. For mean-based approaches, LP and MLP are trained on the temporal mean $\bar{\mathbf{h}} = \frac{1}{S}\sum_s \mathbf{h}_s$. At test time, two prediction strategies are considered: the *Mean* variant predicts directly from $\bar{\mathbf{h}}$, while the *MV* variant applies the trained probe to each step independently and takes a majority vote. This yields four baselines: LP (Mean), LP (MV), MLP (Mean), and MLP (MV). For sequence-based approaches, TimeAttn (Kramár et al., 2026) and LSTM (Damirchi et al., 2026) operate directly on the full ordered sequence $(\mathbf{h}_1, \ldots, \mathbf{h}_S)$. TimeAttn uses a temporal attention mechanism to aggregate hidden states, while LSTM encodes the sequence with a recurrent model. Experiment details are provided in Appendix C.

Evaluation Metrics We report three metrics. **Accuracy (Acc)** measures the fraction of correctly classified prompts. **F1** score is defined as the harmonic mean of precision and recall, capturing the balance between false positives and false negatives. **E[P]** measures the expected number of parameters used per example at test time, capturing the effective monitor size under cascade routing, which is a proxy for runtime memory cost. For our method, $\mathrm{E}[P] = |\theta_{\text{LP}}| + \rho \cdot |\theta_g|$, where $|\theta_{\text{LP}}|$ and $|\theta_g|$ are the parameter counts of the linear probe and the advanced probe respectively, and ρ is the fraction of examples routed to the advanced probe. Additional evaluation metrics, including F2-score, false rejection rate (FRR), inference time, and FLOPs, are

Table 1. **Intra-dataset performance on the test set.** Monitors are trained and tested on *WildGuardMix*. Best results are in **bold**, and second-best are underlined.

		LLaDA-8B-Base		LLaDA-8B-Instruct		LLaDA-1.5			LLaDA-2.0-mini	
Method	**E[P]**	Acc	F1	Acc	F1	Acc	F1	**E[P]**	Acc	F1
Single-step methods										
LP (Last Step)	4×10^{-3}M	84.6	84.1	87.4	87.0	88.2	87.9	2×10^{-3}M	77.6	76.3
MLP (Last Step)	1.05M	85.8	85.4	87.1	86.8	87.9	87.5	0.52M	79.0	78.0
Full-trajectory methods										
LP (MV)	4×10^{-3}M	86.7	86.2	88.2	87.9	87.8	87.5	2×10^{-3}M	80.0	79.0
LP (Mean)	4×10^{-3}M	86.9	86.5	88.2	88.0	87.9	87.7	2×10^{-3}M	80.5	79.4
MLP (MV)	1.05M	86.9	86.6	87.9	87.6	88.3	88.0	0.52M	81.3	80.5
MLP (Mean)	1.05M	87.4	87.0	87.7	87.4	88.3	88.0	0.52M	81.4	80.6
TimeAttn (Kramár et al., 2026)	1.59M	87.4	86.9	87.9	87.5	88.0	87.7	0.80M	81.7	80.8
LSTM (Damirchi et al., 2026)	2.57M	87.1	86.6	87.8	87.4	88.1	87.8	1.51M	81.7	80.9
$\mathcal{D}^2$-MLP (Ours)	$\leq$0.36M	88.1	87.8	**89.9**	**89.7**	**89.3**	**89.1**	0.17M	83.7	82.9
$\mathcal{D}^2$-TimeAttn (Ours)	$\leq$0.54M	**88.6**	**88.3**	89.6	89.4	89.3	89.0	0.26M	**83.7**	**83.0**

reported in Appendix D.1.

5.2. Main Results

Tables 1 to 3 summarize the intra-dataset and cross-dataset performance. Across all settings, $\mathcal{D}^2$-Monitor consistently outperforms all baselines in both accuracy and F1 score. More importantly, $\mathcal{D}^2$-Monitor achieves state-of-the-art performance while incurring substantially lower computational cost than non-linear baselines. As shown in Figure 1, our method provides the best effectiveness-efficiency tradeoff overall. In contrast, sequence-based baselines such as LSTM and TimeAttn incur significantly higher computational cost without delivering better performance. We attribute the advantage of $\mathcal{D}^2$-Monitor to its hesitation-aware routing mechanism, which accurately directs "hard" samples to the advanced probe for further processing (Sec. 5.3). Moreover, hesitation reflects a form of *model-intrinsic* uncertainty rather than dataset-specific patterns. Consequently, the performance gains can generalize across datasets.

5.3. Analysis

Efficiency-effectiveness Tradeoff While model providers can afford to run safety probes alongside the D-LLM with negligible overhead, deploying such probes on the user side demands careful attention to efficiency, as users operate under tighter computational budgets. Our cascaded design naturally supports this scenario by controlling the routing threshold λ: only examples with $n_\tau > \lambda$ are forwarded to the advanced probe, while the rest are classified by the lightweight linear probe. A larger λ reduces the fraction of examples routed to the advanced probe, lowering the expected parameter count $\mathrm{E}[P]$ (which reflects runtime memory cost) at the cost of potentially missing difficult cases. As shown in Figure 1, $\mathcal{D}^2$-Monitor achieves the best F1 scores while using fewer parameters than most baselines, demonstrating a favorable efficiency-effectiveness tradeoff.

Robustness to Generation Length and Step Length In practice, D-LLMs can be deployed with varying generation lengths and step lengths. Training a separate monitor for each configuration is costly and impractical. A desirable property is therefore to train once under a single configuration and generalize to others. To evaluate this, we train all methods with generation length 128 and step length 4, and test under varying settings without retraining. We vary the step length $L_S \in \{1, 2, 4, 8\}$ with generation length fixed at 128 (Figure 3a), and the generation length $L \in \{16, 32, 64, 128\}$ with step length fixed at 1 (Figure 3b). $\mathcal{D}^2$-Monitor consistently outperforms all baselines across both axes of variation, confirming that our method transfers reliably across decoding configurations.

Robustness to Remasking Strategy We evaluate the impact of different remasking strategies on detection performance. All methods are trained under low-confidence remasking and tested under three strategies: low-confidence, entropy, and random. As shown in Figure 3c, $\mathcal{D}^2$-Monitor maintains consistent superiority across all strategies.

Ablation on Routing Signal We study different routing signal choices on ToxicChat by replacing the margin-based criterion with entropy- and confidence-based characterization of hesitation steps. For a fair comparison, the advanced probe is retrained from scratch for each signal on its corresponding hesitation subset, with the rest of the pipeline fixed. We evaluate both $\mathcal{D}^2$-MLP and $\mathcal{D}^2$-TimeAttn at thresholds yielding 30%-70% hesitation samples. As shown in Figure 5, the margin-based signal consistently outperforms entropy and confidence across all settings, confirming that probe-intrinsic uncertainty is a more effective routing signal than probe-extrinsic uncertainty. To understand why, we fix the hesitant ratio at 50% and conduct a counterfactual test by measuring the accuracy gap between the base and advanced probes on the routed subset: a larger gap means the signal

Table 2. **Cross-dataset generalization.** Monitors are trained on *WildguardMix* and tested on *ToxicChat* and *OpenAI-Moderation*. Best results are in **bold**, and second-best are underlined.

		LLaDA-8B-Base				LLaDA-8B-Instruct			
		ToxicChat		OpenAI-Moderation		ToxicChat		OpenAI-Moderation	
Method	**E[P]**	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Single-step methods									
LP (Last Step)	4×10^{-3}M	83.2	64.8	62.6	62.4	86.1	69.0	68.0	67.4
MLP (Last Step)	1.05M	84.0	66.3	63.3	63.2	87.3	70.6	66.9	66.4
Full-trajectory methods									
LP (MV)	4×10^{-3}M	84.5	66.9	62.9	62.7	84.5	67.6	65.1	64.7
LP (Mean)	4×10^{-3}M	85.4	68.1	63.9	63.8	85.3	68.9	64.9	64.5
MLP (MV)	1.05M	82.6	65.3	63.9	63.7	86.3	70.1	60.2	60.2
MLP (Mean)	1.05M	84.6	67.5	64.8	64.6	86.8	70.8	60.5	60.4
TimeAttn (Kramár et al., 2026)	1.59M	87.1	70.8	65.8	65.3	88.9	73.1	65.4	65.4
LSTM (Damirchi et al., 2026)	2.57M	83.1	66.1	65.5	65.0	88.6	72.0	68.5	67.7
$\mathcal{D}^2$-MLP (Ours)	$\leq$0.25M	87.0	70.7	66.0	65.6	**90.4**	**75.0**	68.8	68.2
$\mathcal{D}^2$-TimeAttn (Ours)	$\leq$0.85M	**87.5**	**71.0**	**66.0**	**65.7**	89.9	74.4	**69.6**	**69.1**

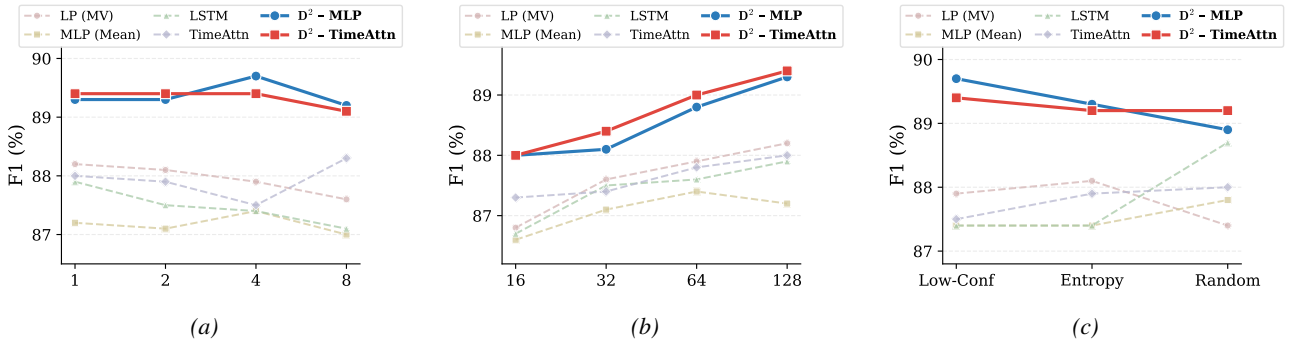


Figure 3. **(a)** Performance with different step lengths with generation length fixed at 128. **(b)** Performance with different generation lengths with step length fixed at 1. **(c)** Performance under different remasking strategies. All results are reported as F1 using LLaDA-8B-Instruct on WildGuardMix.

more successfully isolates samples that genuinely benefit from the advanced probe. The margin signal produces the largest gap on both architectures (13.5% on $\mathcal{D}^2$-MLP and 12.4% on $\mathcal{D}^2$-TimeAttn), well above entropy (7.2% / 7.4%) and confidence (7.1% / 7.3%). This shows that our router more precisely directs "hard" samples to the advanced probe. The margin signal is also more robust to threshold choice: while entropy- and confidence-based signals fluctuate noticeably with the hesitation ratio, margin remains stable across the full range.

6. Conclusion

In this work, we propose $\mathcal{D}^2$-Monitor, a dynamic bi-level safety monitor for D-LLMs that harnesses intrinsic safety hesitation signals arising along the multi-step denoising trajectory. We define a hesitation step in D-LLM's denoising trajectory as the step whose hidden state yields a low probe margin, and show that the number of such hesitation steps serves as an effective proxy for sample difficulty. Accordingly, we identify hesitation trajectories from the training data and use them to train the advanced probe. $\mathcal{D}^2$-Monitor adopts a lightweight linear probe as an always-on monitor to jointly evaluate hesitation and perform base safety classification. When the number of hesitation steps in the trajectory exceeds a predefined threshold, the monitor activates the more expressive but computationally heavier probe for second-stage classification. This hesitation-aware routing mechanism enables dynamic allocation of computational resources and is particularly well-suited to resource-constrained deployment settings. We conduct comprehensive evaluations on 3 safety datasets across 4 D-LLMs, and $\mathcal{D}^2$-Monitor achieves state-of-the-art performance in both intra-dataset detection and cross-dataset generalization, while maintaining the best trade-off between effectiveness and efficiency relative to 8 baselines. We believe the insights from $\mathcal{D}^2$-Monitor provide a promising direction for developing more reliable and efficient safety monitors for D-LLMs.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Anthropic. Disrupting the first reported ai-orchestrated cyber espionage campaign. https://www.anthropic.com/news/disrupting-AI-espionage, 2025.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Bailey, L., Serrano, A., Sheshadri, A., Seleznyov, M., Taylor, J., Jenner, E., Hilton, J., Casper, S., Guestrin, C., and Emmons, S. Obfuscated activations bypass LLM latent-space defenses. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ktGmDGoWnB.

Belinkov, Y. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 04 2022. ISSN 0891-2017. doi: 10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

Bie, T., Cao, M., Chen, K., Du, L., Gong, M., Gong, Z., Gu, Y., Hu, J., Huang, Z., Lan, Z., et al. Llada2.0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*, 2025.

Bie, T., Cao, M., Cao, X., Chen, B., Chen, F., Chen, K., Du, L., Feng, D., Feng, H., Gong, M., et al. Llada2.1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*, 2026.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chen, Y., Yu, J., Liu, A., Torr, P., and Bibi, A. The alignment curse: Cross-modality jailbreak transfer in omni-models. *arXiv preprint arXiv:2602.02557*, 2026.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cunningham, H., Peng, A., Wei, J., Ong, E., Roger, F., Petrini, L., Wagner, M., Mikulik, V., and Sharma, M. Cost-effective constitutional classifiers via representation re-use. Anthropic Alignment Science Blog, June 2025. URL https://alignment.anthropic.com/2025/cheap-monitors/.

Cunningham, H., Wei, J., Wang, Z., Persic, A., Peng, A., Abderrachid, J., Agarwal, R., Chen, B., Cohen, A., Dau, A., et al. Constitutional classifiers++: Efficient production-grade defenses against universal jailbreaks. *arXiv preprint arXiv:2601.04603*, 2026.

Damirchi, H., Meza De la Jara, I., Abbasnejad, E., Shamsi, A., Zhang, Z., and Shi, J. Truth as a trajectory: What internal representations reveal about large language model reasoning. *arXiv e-prints*, pp. arXiv–2603, 2026.

Goldowsky-Dill, N., Chughtai, B., Heimersheim, S., and Hobbhahn, M. Detecting strategic deception with linear probes. In *International Conference on Machine Learning*, pp. 19755–19786. PMLR, 2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Han, J., Band, N., Razzak, M., Kossen, J., Rudner, T. G., and Gal, Y. Simple factuality probes detect hallucinations in long-form natural language generation. *Findings of the Association for Computational Linguistics: EMNLP*, pp. 16209–16226, 2025.

Han, S., Rao, K., Ettinger, A., Jiang, L., Lin, B. Y., Lambert, N., Choi, Y., and Dziri, N. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Advances in neural information processing systems*, 37:8093–8131, 2024.

He, Z., Chen, Y., Lin, L., Wang, Y., Chang, S., Sommerlade, E., Torr, P., Yu, J., Bibi, A., and Yu, J. A fragile guardrail: Diffusion llm's safety blessing and its failure mode. *arXiv preprint arXiv:2602.00388*, 2026.

Hewitt, J. and Liang, P. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Hu, Z., Piet, J., Zhao, G., Jiao, J., and Wagner, D. Toxicity detection for free. *Advances in Neural Information Processing Systems*, 37:17518–17540, 2024.

Inan, H., Upasani, K., Chi, J., Rungta, R., Iyer, K., Mao, Y., Tontchev, M., Hu, Q., Fuller, B., Testuggine, D., et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023.

Jeung, W., Yoon, S., Cho, Y., Jeon, D., Shin, S., Hong, H., and No, A. A2d: Any-order, any-step safety alignment for diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=URTnuyQJI1.

Kramár, J., Engels, J., Wang, Z., Chughtai, B., Shah, R., Nanda, N., and Conmy, A. Building production-ready probes for gemini. *arXiv preprint arXiv:2601.11516*, 2026.

Labs, I. Introducing mercury 2. https://www.inceptionlabs.ai/blog/introducing-mercury-2, 2026.

Labs, I., Khanna, S., Kharbanda, S., Li, S., Varma, H., Wang, E., Birnbaum, S., Luo, Z., Miraoui, Y., Palrecha, A., et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Li, P., Zhou, Y., Muhtar, D., Yin, L., Yan, S., Shen, L., Liang, Y., Vosoughi, S., and Liu, S. Diffusion language model knows the answer before it decodes. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=g88nt4ieTG.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Li, Z., Nie, Z., Zhou, Z., Liu, Y., Zhang, Y., Cheng, Y., Wen, Q., Wang, K., Guo, Y., and Zhang, J. Diffuguard: How intrinsic safety is lost and found in diffusion large language models. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=zBPzxhso8M.

Lin, Z., Wang, Z., Tong, Y., Wang, Y., Guo, Y., Wang, Y., and Shang, J. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation, 2023.

Liu, X., Xu, N., Chen, M., and Xiao, C. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7Jwpw4qKkb.

MacDiarmid, M., Maxwell, T., Schiefer, N., Mu, J., Kaplan, J., Duvenaud, D., Bowman, S., Tamkin, A., Perez, E., Sharma, M., et al. Simple probes can catch sleeper agents, 2024. *URL https://www. anthropic. com/news/probes-catch-sleeper-agents*.

Markov, T., Zhang, C., Agarwal, S., Eloundou, T., Lee, T., Adler, S., Jiang, A., and Weng, L. A holistic approach to undesired content detection. *arXiv preprint arXiv:2208.03274*, 2022.

McKenzie, A., Pawar, U., Blandfort, P., Bankes, W., Krueger, D., Lubana, E. S., and Krasheninnikov, D. Detecting high-stakes interactions with activation probes. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=8YniJnJQ0P.

Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Nasr, M., Carlini, N., Sitawarin, C., Schulhoff, S. V., Hayes, J., Ilie, M., Pluto, J., Song, S., Chaudhari, H., Shumailov, I., et al. The attacker moves second: Stronger adaptive attacks bypass defenses against llm jailbreaks and prompt injections. *arXiv preprint arXiv:2510.09023*, 2025.

Nie, S., Zhu, F., Du, C., Pang, T., Liu, Q., Zeng, G., Lin, M., and Li, C. Scaling up masked diffusion models on text. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=WNvvwK0tut.

Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., ZHOU, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=KnqiC0znVF.

Oldfield, J., Torr, P., Patras, I., Bibi, A., and Barez, F. Beyond linear probes: Dynamic safety monitoring for language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AGWa8whf92.

Park, K., Choe, Y. J., and Veitch, V. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning*, pp. 39643–39666. PMLR, 2024.

Pimentel, T., Saphra, N., Williams, A., and Cotterell, R. Pareto probing: Trading off accuracy for complexity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3138–3153, 2020.

Sahoo, S. S., Arriola, M., Schiff, Y., Gokaslan, A., Marroquin, E., Chiu, J. T., Rush, A., and Kuleshov, V. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37: 130136–130184, 2024.

Shi, J., Han, K., Wang, Z., Doucet, A., and Titsias, M. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167, 2024.

Teerapittayanon, S., McDanel, B., and Kung, H.-T. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, pp. 2464–2469. IEEE, 2016.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, W., Fang, B., Jing, C., Shen, Y., Shen, Y., Wang, Q., Ouyang, H., Chen, H., and Shen, C. Time is a feature: Exploiting temporal dynamics in diffusion language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HsB6CtagP7.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Weng, L., Goel, V., and Vallone, A. Using gpt-4 for content moderation, 2023. URL https://openai.com/index/using-gpt-4-for-content-moderation/.

White, J. C., Pimentel, T., Saphra, N., and Cotterell, R. A non-linear structural probe. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 132–138, 2021.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

Zeng, W., Kurniawan, D., Mullins, R., Liu, Y., Saha, T., Ike-Njoku, D., Gu, J., Song, Y., Xu, C., Zhou, J., et al. Shieldgemma 2: Robust and tractable image content moderation. *arXiv preprint arXiv:2504.01081*, 2025.

Zeng, Y., Lin, H., Zhang, J., Yang, D., Jia, R., and Shi, W. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14322–14350, 2024.

Zhu, F., Wang, R., Nie, S., Zhang, X., Wu, C., Hu, J., Zhou, J., Chen, J., Lin, Y., Wen, J.-R., et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

A. Limitation

We perform experiments on a variety of D-LLM models, where we show that $\mathcal{D}^2$-Monitor achieves superior performance under both intra-dataset and cross-dataset settings with a compact parameter footprint of less than 0.85M. Although we expect this trend to extend to even larger models (Bie et al., 2025; 2026), our experiments are currently limited to D-LLMs with up to 16B parameters due to computational constraints. Furthermore, recent work shows that activation monitors are vulnerable to adversaries (Bailey et al., 2026)–in our case, attacks may aim to elicit a smaller number of hesitation steps to avoid triggering stronger guardrails. Future work should study the robustness of $\mathcal{D}^2$-Monitors and activation cascades more generally.

B. Broader Impacts, Safeguards, and Licenses

Broader Impacts Our work has clear positive societal impact: $\mathcal{D}^2$-Monitor provides a lightweight, always-on safety mechanism for D-LLMs that can detect harmful or adversarial inputs at low computational cost, which is particularly valuable for resource-constrained or edge deployment where running heavyweight LLM-as-monitor solutions is infeasible. By improving the practicality of safety monitoring, our work helps reduce the risk of D-LLMs being misused for harmful content generation. We also acknowledge potential negative impacts. (1) Like any safety classifier, the monitor may incur false rejections on benign inputs, potentially over-blocking legitimate queries; we report FRR explicitly in Appendix D.1 to make this trade-off transparent. (2) As discussed in Appendix A, hesitation-based routing could in principle be targeted by adaptive adversaries who craft prompts that suppress hesitation steps to evade the second-stage probe. We recommend that practitioners deploying $\mathcal{D}^2$-Monitor pair it with complementary defenses and continue to monitor for adversarial drift over time.

Safeguards The paper does not release any high-risk pretrained models, generative models, or scraped datasets. The artifacts produced by our work are lightweight safety probes (≤ 0.85M parameters) trained on publicly available safety benchmarks; these probes are themselves a defensive mechanism intended to mitigate misuse rather than enable it. The base D-LLMs (LLaDA family) and safety datasets we build on are released by their original authors under their respective licenses, which we do not modify or redistribute.

Licenses for Existing Assets All existing assets used in this work are properly cited and used in accordance with their licenses.

Datasets WildGuardMix (Han et al., 2024) is released under the ODC-BY license; ToxicChat (Lin et al., 2023) is released under CC-BY-NC-4.0 and used for non-commercial research only; OpenAI-Moderation (Markov et al., 2022) is released under the MIT license.

Models LLaDA-8B-Base, LLaDA-8B-Instruct (Nie et al., 2025b), and LLaDA-1.5 (Zhu et al., 2025) are released under the MIT license; LLaDA-2.0-mini (Bie et al., 2025) is released under the Apache-2.0 license.

C. Experiment Details

C.1. Hyperparameter Tuning

We tune all hyperparameters on a validation set derived from the training data. Specifically, we split the original training set into training and validation subsets with a ratio of 4:1. To ensure fair comparison across different probe architectures, we control model capacity by fixing key architectural dimensions (e.g., hidden size or hidden dimension) for each probe, rather than tuning them extensively. For each probe, we define the hyperparameter search space as follows:

LinearProbe:

- **Learning rate:** $\{1\text{e}{-5}, 1\text{e}{-4}, 1\text{e}{-3}, 1\text{e}{-2}\}$
- **Weight decay:** $\{0, 1\text{e}{-6}, 1\text{e}{-5}, 1\text{e}{-4}\}$

MLP:

- **Learning rate:** $\{1\text{e}{-5}, 1\text{e}{-4}, 1\text{e}{-3}\}$
- **Weight decay:** $\{0, 1\text{e}{-5}, 1\text{e}{-4}, 1\text{e}{-3}\}$
- **Dropout:** $\{0.1, 0.2, 0.3, 0.5\}$
- **Hidden dimension (K):** 256

TimeAttn:

- **Learning rate:** $\{1\text{e}{-4}, 1\text{e}{-3}, 4\text{e}{-3}, 1\text{e}{-2}\}$
- **Weight decay:** $\{0, 1\text{e}{-5}\}$
- **Dropout:** $\{0.2, 0.3, 0.5\}$
- **MLP hidden dimension (K):** 256
- **Attention dimension (d_a):** 128

LSTM:

- **Projection dimension (d_p):** 512
- **Hidden size (d_h):** 128
- **Learning rate:** $\{1\text{e}{-5}, 1\text{e}{-4}, 1\text{e}{-3}\}$
- **Weight decay:** $\{0, 1\text{e}{-6}, 1\text{e}{-5}, 1\text{e}{-4}\}$
- **Dropout:** $\{0, 0.1, 0.2, 0.3\}$

Baseline Methods For all baseline methods, hyperparameters are selected via grid search on the validation set using the search ranges above. The best configuration is chosen based on validation performance, and the model is retrained on the full training set before evaluation on the test set.

Our Method For out-of-fold (OOF) scoring on the training set, we use a fixed linear probe configuration with learning rate $1\text{e}{-3}$ and weight decay $1\text{e}{-4}$ to ensure consistency across folds. The same configuration is used when training the linear probe component within our method. For the MLP and TimeAttn components in our method, hyperparameters are tuned on the validation set. We focus on hesitation examples in the validation set, as these are most relevant to the target behavior our method aims to model.

Fairness and Evaluation Protocol All hyperparameter tuning is performed strictly on the validation set, and no test data is used in any stage of model selection.

Test-time Hyperparameters Our method involves two test-time hyperparameters: the hesitation threshold τ and the routing parameter λ.

Choice of τ The threshold τ controls a trade-off between coverage and purity of hesitation examples. A larger τ includes more training samples but may introduce stable (non-hesitant) steps, while a smaller τ yields cleaner hesitation signals at the cost of fewer training examples. To balance this trade-off, we select τ such that approximately 50% of the samples are identified as hesitation examples. As shown in Fig. 8, varying this proportion from 30% to 70% leads to only minor performance differences, indicating that our method is not sensitive to the exact choice of τ. For intra-dataset evaluation, we fix τ at the value selected on the training data and reuse it at test time, since the data distributions are similar.

Choice of λ The routing parameter λ is selected based on validation performance, where we choose the value that achieves the best F1 score on the validation set. This value is then fixed for test-time evaluation.

Cross-dataset evaluation For cross-dataset transfer (trained on WildGuardMix and evaluated on ToxicChat and OpenAI Moderation), the distribution shift makes hyperparameter selection more challenging, and both τ and λ are tuned for each target dataset. For ToxicChat, which provides a training split, we reserve 20% of the training data as a validation set and perform a grid search over τ and λ, selecting the best-performing configuration. For OpenAI-Moderation, which does not provide a training set, we randomly sample 10% of the data as a validation set for hyperparameter tuning, and report results on the remaining 90%.

C.2. Probe Architectures

Here we provide specific details about the architectures of all probe baselines considered. Given the step-level trajectory $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_S] \in \mathbb{R}^{D \times S}$, each probe instantiates a particular $f \in \mathcal{F}$ that maps $\mathbf{H}$ to a scalar logit $s \in \mathbb{R}$ for binary classification. When presenting architectures below, we omit bias terms in the hidden layer(s) and output for brevity. All probes are trained with Adam with a batch size of 256 for 50 epochs on an NVIDIA A40 GPU with 48GB VRAM. Each individual probe performs its own grid search over the hyperparameters listed in Appendix C.1.

Normalization. Before training, we normalize the activations using statistics computed from the training split. We use two normalization modes depending on the probe type. For *full-trajectory methods*, we apply *per-feature* normalization: the mean and standard deviation are computed over both the sample and step axes, yielding statistics of shape $[D]$. For *single-step methods*, we apply *per-step* normalization: statistics are computed per denoising step, yielding shape $[S, D]$.

Pooling strategies For probes that require a fixed-length input (Linear Probe and MLP), we reduce the step dimension via one of three pooling strategies applied after normalization:

- **Mean pooling:** $\bar{\mathbf{h}} = \frac{1}{S}\sum_{s=1}^{S} \mathbf{h}_s \in \mathbb{R}^D$, used at both train and test time.
- **Last-step:** $\bar{\mathbf{h}} = \mathbf{h}_1 \in \mathbb{R}^D$, using only the final step's activation at both train and test time. Normalization statistics are computed from the last step only.
- **Majority vote (MV):** The probe is trained on mean-pooled features. At test time, each step $\mathbf{h}_s$ is classified independently, and the final prediction is determined by majority vote across S steps: $\hat{y} = \mathbf{1}\left[\sum_{s=1}^{S} \mathbf{1}[\hat{y}_s = 1] \geq S/2\right]$.

Linear Probe (LP) After pooling, the linear probe computes:

$$s = \mathbf{w}^\top \bar{\mathbf{h}}\,, \tag{5}$$

with $\mathbf{w} \in \mathbb{R}^D$. Combined with the three pooling strategies, this yields three variants: LP (Mean), LP (Last Step), and LP (MV).

MLP The two-layer MLP probe computes:

$$s = \mathbf{W}_{\text{out}}\,\text{Dropout}\big(\text{ReLU}\big(\mathbf{W}_{\text{in}}\bar{\mathbf{h}}\big)\big)\,, \tag{6}$$

with $\mathbf{W}_{\text{in}} \in \mathbb{R}^{K \times D}$ and $\mathbf{W}_{\text{out}} \in \mathbb{R}^{1 \times K}$, where $K = 256$ is the hidden dimension. Analogously to the linear probe, this yields three variants: MLP (Mean), MLP (Last Step), and MLP (MV).

TimeAttn This probe operates directly on the full trajectory $\mathbf{H} \in \mathbb{R}^{D \times S}$ without step-level pooling. It first applies layer normalization, then computes attention weights over denoising steps via an additive attention mechanism:

$$\alpha_s = \frac{\exp\big(\mathbf{v}^\top \tanh(\mathbf{W}_a \mathbf{h}_s)\big)}{\sum_{s'=1}^{S} \exp(\mathbf{v}^\top \tanh(\mathbf{W}_a \mathbf{h}_{s'}))}\,, \tag{7}$$

where $\mathbf{W}_a \in \mathbb{R}^{d_a \times D}$ and $\mathbf{v} \in \mathbb{R}^{d_a}$ with attention dimension $d_a = 128$. The attended representation $\mathbf{c} = \sum_{s=1}^{S} \alpha_s \mathbf{h}_s \in \mathbb{R}^D$ is then classified via a two-layer MLP with layer normalization:

$$s = \mathbf{W}_2\,\text{Dropout}(\text{ReLU}(\mathbf{W}_1\,\text{LN}(\mathbf{c})))\,, \tag{8}$$

with $\mathbf{W}_1 \in \mathbb{R}^{K \times D}$, $\mathbf{W}_2 \in \mathbb{R}^{1 \times K}$, and $K = 256$.

LSTM The LSTM probe first projects each step representation through a feedforward layer with layer normalization:

$$\mathbf{h}'_s = \mathrm{GELU}(\mathbf{W}_{\mathrm{proj}}\,\mathrm{LN}(\mathbf{h}_s))\,, \tag{9}$$

with $\mathbf{W}_{\mathrm{proj}} \in \mathbb{R}^{d_p \times D}$ and projection dimension $d_p = 512$. The projected sequence is then processed by a 2-layer unidirectional LSTM:

$$\tilde{\mathbf{h}}_1, \ldots, \tilde{\mathbf{h}}_S = \mathrm{LSTM}(\mathbf{h}'_1, \ldots, \mathbf{h}'_S)\,, \tag{10}$$

with hidden size $d_h = 128$. The final hidden state is classified via a layer-normalized linear head:

$$s = \mathbf{w}_{\mathrm{out}}^{\top}\mathrm{LN}(\tilde{\mathbf{h}}_S)\,, \tag{11}$$

with $\mathbf{w}_{\mathrm{out}} \in \mathbb{R}^{d_h}$.

D. Additional Results

D.1. More Evaluation Metrics

Additional Metrics Beyond accuracy and macro-F1 reported in the main text, we evaluate four additional metrics. The **F2-score** is an instance of the F_β-score with $\beta = 2$, which weights recall β^2 times as much as precision, reflecting a preference in safety monitoring for catching harmful content at the cost of slightly more false alarms. **Precision** measures the fraction of inputs flagged as unsafe that are truly unsafe; high precision means the monitor rarely triggers on benign content, reducing unnecessary refusals. **Recall** measures the fraction of truly unsafe inputs correctly identified; high recall ensures that harmful prompts are not missed in safety-sensitive deployments. Finally, the **False Rejection Rate (FRR)** is the fraction of benign inputs incorrectly flagged as unsafe, i.e., $\mathrm{FRR} = \mathrm{FP}/(\mathrm{FP} + \mathrm{TN})$; unlike $1 - \mathrm{Precision}$, which conditions on positive predictions, FRR conditions on the actual benign population and thus directly measures how often legitimate users are unnecessarily blocked (lower is better).

Tables 8–11 report all six metrics for each model. The results are consistent with the main findings: $\mathcal{D}^2$-MLP and $\mathcal{D}^2$-TimeAttn achieve the best or second-best performance across nearly all metrics and models, confirming that the gains observed in accuracy and F1 are not obtained at the expense of precision, recall balance, or false refusal rate.

Inference Time We report the inference time of different methods on WildGuardMix with 1,725 prompts. To isolate the overhead of the safety monitor, timing starts after the base model has completed generation and all hidden states have been extracted. For LP (Last Step), LP (MV), MLP (Last Step), and MLP (MV), we measure only the time required to process the last step. This is because, in typical deployment, intermediate steps are processed online during generation (i.e., one step is processed as it is generated), and thus do not introduce additional latency after generation. For LP (Mean) and MLP (Mean), the reported time includes both the cost of computing the mean over all hidden states and the subsequent probe evaluation. In contrast, TimeAttn and LSTM operate on the full trajectory after generation, and their inference time includes processing all steps. For our method, the total inference time consists of two components: (i) the linear probe applied to the last step, and (ii) the additional computation of the advanced probe (e.g., MLP or TimeAttn) over the selected window when routing is triggered.

As shown in Table 6, D^2-Monitor significantly reduces inference cost compared to full-trajectory methods. Compared to MLP (Mean), D^2-MLP achieves a 2.4×–6.6× speedup across different LLaDA models. This improvement arises from our hesitation-aware routing strategy, which restricts computation to a subset of hesitation examples and further focuses on their localized hesitation windows, thereby reducing both the number of processed samples and the number of processed steps. Similarly, D^2-TimeAttn is 4×–5× faster than TimeAttn, as it avoids full-sequence modeling and instead operates only on selectively triggered sub-trajectories. Importantly, the inference cost of D^2-MLP remains comparable to single-step methods (e.g., LP (Last Step) and MLP (Last Step)), while achieving performance close to full-trajectory models, demonstrating an effective balance between efficiency and accuracy. Overall, these results highlight that D^2-Monitor achieves its efficiency gains through conditional computation by activating high-complexity probes only when the model exhibits hesitation, rather than uniformly processing all steps, leading to consistent speedups across different models.

FLOPs Computation We analytically compute the expected FLOPs per sample for each probe based on the architectures detailed in Appendix C.2. We adopt the standard convention that a multiply-add operation counts as 2 FLOPs. Throughout this section, we denote the number of denoising steps as S, the hidden dimension of the diffusion LLM as D, and the

average length of the hesitation window for our cascade as S_{win}. The MLP hidden width K, attention dimension d_a, LSTM projection dimension d_p, and LSTM hidden size d_h follow the values specified in Appendix C.2. For brevity, we omit lower-order terms such as activation functions (ReLU, tanh, GELU), softmax, layer normalization, and bias additions, which contribute at most $O(D)$ or $O(S_{\text{win}}D)$ FLOPs and are negligible relative to the dominant matrix-vector products.

Linear Probe (LP) A single linear probe forward maps a D-dimensional vector to a scalar logit, requiring $2D$ FLOPs. The three pooling variants differ in how the trajectory $\mathbf{H} \in \mathbb{R}^{D\times S}$ is reduced to this input. **LP (Last Step)** uses only $\mathbf{h}_1$, with total cost $2D$. **LP (Mean)** sums S vectors of dimension D (SD FLOPs) and then applies one LP forward, with total cost $SD + 2D$. **LP (MV)** applies the LP independently at every denoising step and aggregates by majority vote, with total cost $S \cdot 2D = 2SD$.

MLP A single MLP forward consists of $\mathbf{W}_{\text{in}} \in \mathbb{R}^{K\times D}$ and $\mathbf{W}_{\text{out}} \in \mathbb{R}^{1\times K}$, costing $2DK + 2K \approx 2DK$ FLOPs. **MLP (Last Step)** applies one MLP forward on $\mathbf{h}_S$, with total cost $2DK$. **MLP (Mean)** performs mean pooling followed by one MLP forward, with total cost $SD + 2DK$. **MLP (MV)** applies an MLP forward at every denoising step, with total cost $S \cdot 2DK = 2SDK$.

TimeAttn TimeAttn operates on the full trajectory and consists of two main components: per-step attention scoring and a classifier head over the attended representation. The attention scoring computes $\mathbf{W}_a\mathbf{h}_s$ for each step ($\mathbf{W}_a \in \mathbb{R}^{d_a\times D}$), costing $S \cdot 2Dd_a$ FLOPs in total. The weighted sum $\sum_s \alpha_s\mathbf{h}_s$ contributes a further $2SD$ FLOPs. The classifier head is a 2-layer MLP applied to the resulting D-dimensional context vector, costing $2DK$ FLOPs. The total cost is therefore $2SDd_a + 2SD + 2DK \approx 2SDd_a + 2DK$, where the first term dominates for typical S and d_a.

LSTM The LSTM probe first projects each $\mathbf{h}_s$ from D to d_p via $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{d_p\times D}$, costing $S \cdot 2Dd_p$ FLOPs across all steps. The projected sequence is then processed by a 2-layer LSTM. Each LSTM cell contains four gates, each with a matrix-vector product of size $d_h \times (d_{\text{in}} + d_h)$, where $d_{\text{in}} = d_p$ for layer 1 and $d_{\text{in}} = d_h$ for layer 2. The total per-step LSTM cost is approximately $8d_h(d_p + d_h) + 8d_h \cdot 2d_h$, applied at all S steps. The final linear head contributes $2d_h$ FLOPs and is negligible. The total cost is dominated by the input projection, $S \cdot 2Dd_p$, since $D \gg d_h$.

$\mathcal{D}^2$-Monitor (cascade) Our cascade has two components. The base linear probe is applied at every denoising step to compute the per-step margin $|d_s|$ used for hesitation detection, regardless of whether the sample is escalated; this incurs $2SD$ FLOPs per sample. The expert (MLP or TimeAttn) is invoked only on the fraction p_{esc} of samples flagged as hesitant, and operates on the minimal window covering all hesitation steps with average length $S_{\text{win}} \leq S$, rather than the full trajectory. The expected per-sample cost is

$$\mathbb{E}[\text{FLOPs}] = \underbrace{2SD}_{\text{base, always}} + \underbrace{p_{\text{esc}} \cdot F_{\text{expert}}(S_{\text{win}})}_{\text{expert, conditional}}, \tag{12}$$

where the expert FLOPs are

$$F_{\text{MLP-expert}}(S_{\text{win}}) = S_{\text{win}}D + 2DK \tag{13}$$

$$F_{\text{TimeAttn-expert}}(S_{\text{win}}) = 2S_{\text{win}}Dd_a + 2S_{\text{win}}D + 2DK. \tag{14}$$

The values of p_{esc} and S_{win} are measured empirically on *WildGuardMix* for each model and reported alongside the FLOPs in Table 5.

Across all four LLaDA models, $\mathcal{D}^2$-Monitor achieves a substantially better efficiency–effectiveness trade-off than full-trajectory baselines. $\mathcal{D}^2$-MLP requires only 0.7–1.0 MFLOPs per sample, which is 2–3$\times$ cheaper than MLP (Mean) and 35–150$\times$ cheaper than sequence-based baselines (TimeAttn and LSTM), while still delivering the highest F1 scores in Table 1. $\mathcal{D}^2$-TimeAttn is more expensive than $\mathcal{D}^2$-MLP due to the heavier expert, but remains 4–5$\times$ cheaper than running TimeAttn on the full trajectory. The savings come from two sources. First, the cascade only invokes the expert on the fraction p_{esc} of samples flagged as hesitant, so most samples incur only the lightweight $2SD$ FLOPs of the base linear probe. Second, even on escalated samples, the expert operates on a localized hesitation window of average length $S_{\text{win}} \leq S$, rather than the full trajectory. As a result, $\mathcal{D}^2$-Monitor matches the FLOPs cost of single-step methods (e.g., MLP (Last Step) at $\sim$ 2 MFLOPs) while attaining performance close to full-trajectory models, demonstrating that conditional computation effectively decouples cost from accuracy.

D.2. Robustness to Random Seeds

In the main results, we report performance using a fixed random seed (2026). To verify that our findings are not due to a particular random initialization, we retrain all methods on LLaDA-8B-Instruct using five different random seeds (0, 1, 2, 3, and 4). As shown in Table 7, the results are consistent across seeds, with low standard deviations for all methods. Importantly, our method continues to achieve the best performance among all baselines, indicating that the observed improvements are robust and not due to randomness.

E. Additional Analysis

E.1. Analysis of Hesitation Dynamics

Cross-Boundary Probability We first analyze the local instability of model predictions by measuring how likely a step is to cross the decision boundary in the next step. Given the signed margin d_s at step s, we compute the probability that the sign of the margin changes at the next step, i.e., $\mathrm{sign}(d_{s+1}) \neq \mathrm{sign}(d_s)$, as a function of the margin $|d_s|$. Figure 7 shows the crossing probability binned over $|d_s|$. Across all four LLaDA models, we observe a consistent pattern: steps close to the decision boundary exhibit a significantly higher probability of crossing it in the next step, while steps far from the boundary are highly stable. The crossing probability decreases sharply as $|d_s|$ increases, quickly approaching zero beyond a small margin threshold. This indicates that hesitation steps, characterized by small margin magnitudes, are intrinsically unstable and prone to prediction flips, suggesting that the decision boundary region concentrates most of the local uncertainty in the trajectory.

Margin Persistence We further analyze the temporal structure of hesitation by measuring how long a model remains in a hesitant state. Given a threshold τ, we define a step as hesitant if $|d_s| < \tau$. For each such step, we compute the probability that the model remains hesitant after k steps, i.e.,

$$P(|d_{s+k}| < \tau \mid |d_s| < \tau),$$

for $k = 1, 2, \ldots, K$. Figure 7 shows the persistence curves for different models. We also report the unconditional probability of being in a hesitant state as a baseline. Across models, we observe that hesitation exhibits clear temporal persistence: once the model enters a low-margin region, it tends to remain in that region for multiple subsequent steps, with the persistence probability decaying gradually as k increases. These results suggest that hesitation is not only a local phenomenon but also forms contiguous segments along the generation trajectory. This temporal coherence further motivates our design of operating on localized hesitation windows, as they capture the regions where uncertainty is both concentrated and temporally structured.

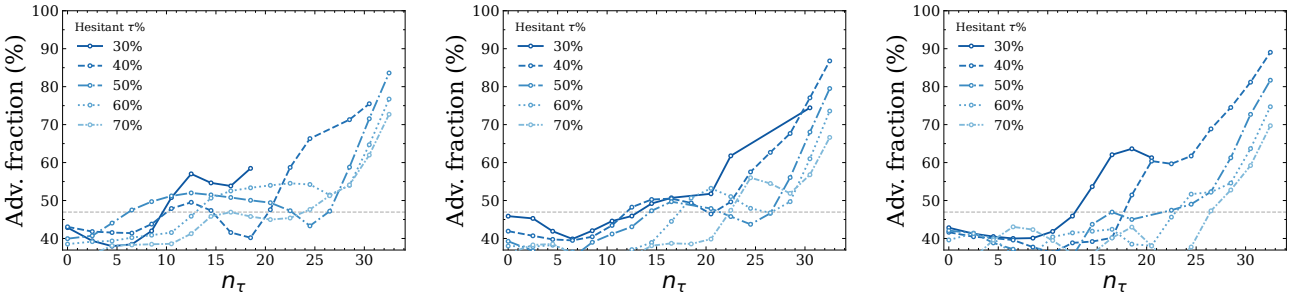
E.2. Margin Outperforms Entropy and Confidence as Step-Count Signal

In Section 3.2, we noted that the step-count construction underlying n_τ is general and can be applied to any per-step hesitation signal. We instantiate this for the two probe-extrinsic signals introduced in Section 3.3, defining

$$n_{\text{entropy}} = \sum_{s=1}^{S} \mathbb{I}[E_s \geq \tau_E], \tag{15}$$

$$n_{\text{confidence}} = \sum_{s=1}^{S} \mathbb{I}[C_s \leq \tau_C], \tag{16}$$

where E_s and C_s are the step-wise entropy and confidence scores, and τ_E, τ_C are their respective thresholds. We compare the three step-count signals on LLaDA-8B-Instruct in Figure 6, where for each signal we evaluate probe F1 across n_τ buckets (more generally, n_{signal} buckets) under five threshold settings parameterized by the resulting hesitant ratio. Two observations stand out. First, all three signals are predictive of difficulty in the qualitative sense: F1 decreases monotonically as the count increases, regardless of which signal is used. This confirms that the step-count construction itself is a general approach to extracting trajectory-level difficulty information from per-step signals. Second, n_τ produces the steepest and most extended F1 decline across all hesitant ratios. The margin-based curves (blue) drop from $\sim 90\%$ at $n_\tau = 0$ to 55–77% at the largest buckets, whereas entropy-based and confidence-based curves (orange and green) plateau at $\sim$ 75–80% and cover a much narrower range of n_{signal} values. The larger F1 spread under n_τ indicates that the probe margin produces a finer



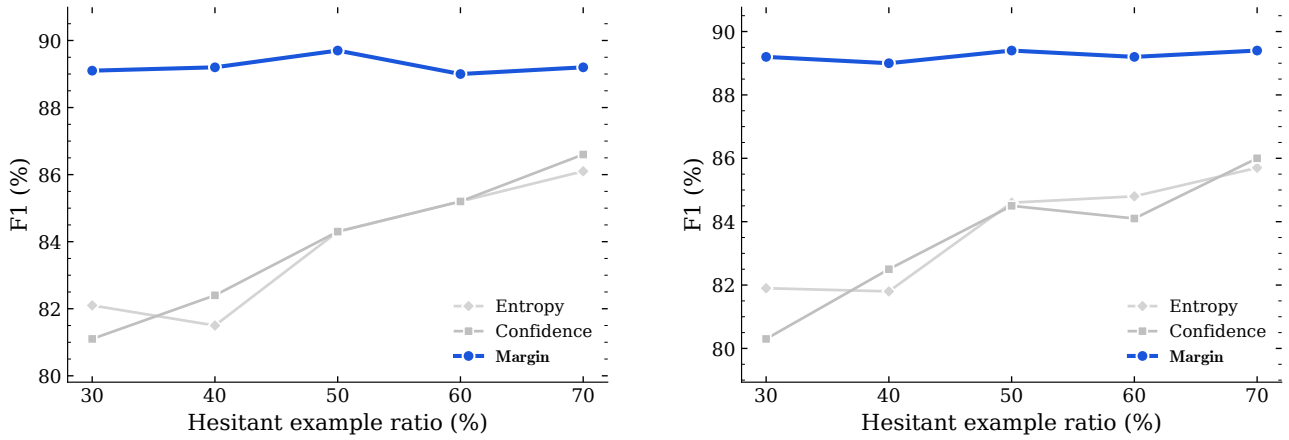
(a) LLaDA-8B-Base *(b)* LLaDA-8B-Instruct *(c)* LLaDA-1.5

Figure 4. **Adversarial fraction vs. hesitation severity n_τ.** For each n_τ bucket, we report the fraction of samples drawn from the adversarial split of *WildGuardMix*. Each curve corresponds to a different τ setting (parameterized by the resulting hesitant ratio). The gray dashed line marks the dataset-wide adversarial fraction ($\sim 47\%$). The monotonic rise of the adversarial fraction with n_τ holds across all three LLaDA variants.

and more discriminative stratification of difficulty than entropy or confidence: trajectories deemed "highly hesitant" under n_τ are substantially harder for the probe than those deemed "highly hesitant" under entropy or confidence. This justifies our choice of n_τ as the routing signal in $\mathcal{D}^2$-Monitor. A natural explanation is that the probe margin is *probe-aware*: it directly measures distance to the probe's decision boundary, which is the quantity governing whether the probe will misclassify a sample. In contrast, entropy and confidence are computed from the D-LLM's predicted token distribution and are agnostic to the specific probe being used. They capture model-level uncertainty but miss the probe-specific decision dynamics that matter for routing.

E.3. Hesitation Severity Captures Adversarial Inputs

We further examine *which* samples accumulate large n_τ, beyond their being difficult for the linear probe. Recall that *WildGuardMix* contains a mixture of natural and adversarially designed prompts, where the latter are constructed to evade safety classifiers and are widely regarded as the harder portion of the benchmark (Han et al., 2024). We compute, for each n_τ bucket, the fraction of samples drawn from the adversarial split (*Adv. fraction*) and report the result on three LLaDA variants in Figure 4. The relationship is consistent across all three models. At $n_\tau = 0$, the adversarial fraction is 38–46%, noticeably below the dataset-wide baseline of $\sim 47\%$ (gray dashed line); as n_τ increases, the fraction rises monotonically and reaches 67–89% at the largest buckets across the five τ settings. The overall trend, including both the sub-baseline behavior at $n_\tau = 0$ and the monotonic rise toward $n_\tau \to S$, holds for LLaDA-8B-Base, LLaDA-8B-Instruct, and LLaDA-1.5, indicating that the association between hesitation severity and adversarial inputs is a model-agnostic property rather than an artifact of any specific model or threshold choice. Conceptually, this is consistent with the design intent of adversarial prompts: they are crafted to push the model into a borderline decision, which manifests as repeated proximity to the probe's decision boundary across denoising steps. Hesitation severity n_τ thus naturally captures this signature of adversarial conditioning, providing a concrete semantic interpretation beyond abstract probe uncertainty. The above analysis is conducted at the level of n_τ buckets and characterizes the data property of hesitation. We now turn to the operational consequence of this property in our cascade: *which samples are actually routed to the second-stage probe?* Since routing in $\mathcal{D}^2$-Monitor is triggered by n_τ exceeding a threshold (selected on a held-out validation set), the routed subset should, by construction, inherit the adv-rich property documented above. We empirically verify this by computing the adversarial fraction within the routed subset for each LLaDA variant and report the result in Table 4. The routed subset is consistently and substantially enriched in adversarial inputs relative to the dataset-wide baseline of $\sim 47\%$: the adversarial fraction reaches 86.3% on LLaDA-8B-Base, 71.6% on LLaDA-8B-Instruct, and 60.8% on LLaDA-1.5 for $\mathcal{D}^2$-MLP, with $\mathcal{D}^2$-TimeAttn exhibiting comparable enrichment (56.1%, 71.1%, 60.7% respectively). In other words, the cascade routes predominantly adversarial samples to the expert. This confirms that $\mathcal{D}^2$-Monitor does not merely allocate extra capacity to "hard" samples in a generic sense but specifically channels it toward adversarial inputs that pose the highest risk to safety classification. Viewed in this light, the bi-level design becomes more than a cost-saving heuristic: it is a targeted defense mechanism, with hesitation severity acting as an implicit detector of adversarial conditioning, and the second-stage expert serving as the specialized classifier for these flagged cases.



(a) D^2–MLP *(b)* D^2–TimeAttn

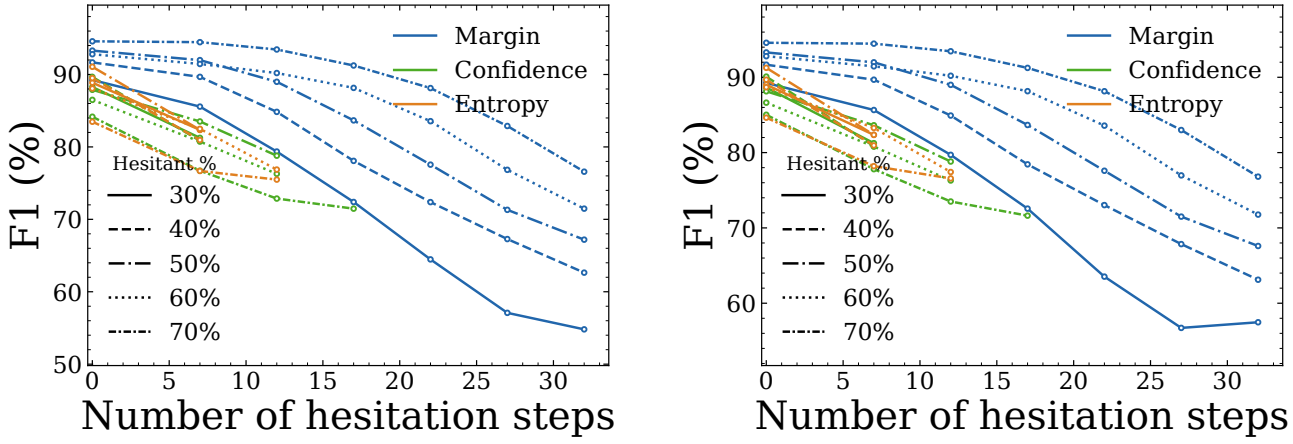
Figure 5. Ablation on routing signals under different hesitant ratios. Margin-based routing consistently outperforms entropy and confidence, and shows greater robustness to threshold selection.

Table 3. **Intra-dataset performance on the test set.** Monitors are trained and tested on *ToxicChat*. Best results are in **bold**, and second-best are underlined.

		LLaDA-8B-Base		LLaDA-8B-Instruct		LLaDA-1.5			LLaDA-2.0-mini	
Method	**E[P]**	Acc	F1	Acc	F1	Acc	F1	**E[P]**	Acc	F1
Single-step methods										
LP (Last Step)	4×10^{-3}M	89.8	74.7	90.5	76.3	91.1	77.2	2×10^{-3}M	93.8	80.0
MLP (Last Step)	1.05M	95.8	82.8	96.9	88.1	96.6	86.2	0.52M	95.3	82.6
Full-trajectory methods										
LP (MV)	4×10^{-3}M	90.7	76.4	91.4	77.7	91.6	78.2	2×10^{-3}M	93.2	79.5
LP (Mean)	4×10^{-3}M	91.7	78.0	91.7	78.5	92.1	79.0	2×10^{-3}M	93.8	80.5
MLP (MV)	1.05M	95.5	84.0	96.3	87.9	96.3	87.8	0.52M	95.3	81.9
MLP (Mean)	1.05M	96.2	84.5	97.0	88.0	96.8	88.0	0.52M	95.9	84.6
TimeAttn (Kramár et al., 2026)	1.59M	96.8	86.5	96.8	88.7	97.2	88.6	0.80M	95.8	80.6
LSTM (Damirchi et al., 2026)	2.57M	96.6	86.7	97.0	88.9	96.8	88.0	1.51M	95.9	82.0
$\mathcal{D}^2$-MLP (Ours)	$\leq$0.50M	96.9	87.9	97.2	89.7	97.2	89.5	0.40M	96.0	84.6
$\mathcal{D}^2$-TimeAttn (Ours)	$\leq$0.76M	**97.0**	**88.0**	**97.3**	**89.9**	**97.2**	**89.7**	0.53M	**96.1**	**85.2**

Table 4. **Adversarial fraction in the routed subset.** For each LLaDA variant, we report the percentage of routed samples that are drawn from the adversarial split of *WildGuardMix*. The dataset-wide baseline is $\sim 47\%$. Routing thresholds are selected on a held-out validation set per method.

Method	**LLaDA-8B-Base**	**LLaDA-8B-Instruct**	**LLaDA-1.5**
$\mathcal{D}^2$-MLP	86.3%	71.6%	60.8%
$\mathcal{D}^2$-TimeAttn	56.1%	71.1%	60.7%
Dataset baseline		$\sim 47\%$	



(a) LP (MV) *(b)* LP (Mean)

Figure 6. **Comparison of step-count signals on LLaDA-8B-Instruct.** For each step-count signal (n_τ from probe margin, n_{entropy} from step-wise entropy, $n_{\text{confidence}}$ from step-wise confidence), we report probe F1 across buckets of increasing hesitation count, under two base classifier variants: **(a)** LP (MV) and **(b)** LP (Mean). Each color corresponds to one signal, and each line style corresponds to a different threshold setting parameterized by the resulting hesitant ratio. Across both base classifiers, the margin-based signal n_τ produces the steepest and most extended F1 decline, indicating it stratifies difficulty more discriminatively than the probe-extrinsic signals.

Table 5. **Expected FLOPs (MFLOPs) per sample.** Computed analytically using the architectures in Appendix C.2, with denoising steps T, hidden dimension H, and (for our cascade) average escalation rate p_{esc} and average hesitation window T_{win} measured on *WildGuardMix*.

Method	**LLaDA-8B-Base** FLOPs	**LLaDA-8B-Instruct** FLOPs	**LLaDA-1.5** FLOPs	**LLaDA-2.0-mini** FLOPs
Single-step methods				
LP (Last Step)	0.008	0.008	0.008	0.004
MLP (Last Step)	2.10	2.10	2.10	1.05
Full-trajectory methods				
LP (MV)	0.26	0.26	0.26	0.52
LP (Mean)	0.14	0.14	0.14	0.27
MLP (MV)	67.1	67.1	67.1	134
MLP (Mean)	2.23	2.23	2.23	1.31
TimeAttn	35.7	35.7	35.7	68.2
LSTM	163	163	163	386
$\mathcal{D}^2$-MLP (Ours)	0.92	0.68	0.97	0.87
$\mathcal{D}^2$-TimeAttn (Ours)	8.16	5.10	8.48	19.4

Table 6. **Post-generation inference time (ms).** Results are measured on WildGuardMix (1,725 prompts) after generation on different LLaDA models.

	LLaDA-8B-Base	**LLaDA-8B-Instruct**	**LLaDA-1.5**	**LLaDA-2.0-mini**
Method	Time	Time	Time	Time
Single-step methods				
LP (Last Step)	0.73	0.31	0.42	0.37
MLP (Last Step)	0.79	0.52	0.66	0.34
Full-trajectory methods				
LP (MV)	0.61	0.42	0.36	0.44
LP (Mean)	2.46	1.78	1.10	3.33
MLP (MV)	0.63	0.68	0.71	0.43
MLP (Mean)	3.04	1.94	1.59	3.65
TimeAttn	30.96	33.34	31.61	26.31
LSTM	346.44	317.42	307.88	348.56
$\mathcal{D}^2$-MLP (Ours)	0.57	0.56	0.66	0.55
$\mathcal{D}^2$-TimeAttn (Ours)	7.36	7.69	7.63	5.11

Table 7. **Intra-dataset performance on LLaDA-8B-Instruct (WildGuardMix).** Results are reported as mean $\pm$ std over multiple random seeds.

Method	**E[P]**	**Acc**	**F1**
Single-step methods			
LP (Last Step)	4×10^{-3}M	87.2 ± 0.3	86.8 ± 0.3
MLP (Last Step)	1.05M	87.9 ± 0.4	87.6 ± 0.4
Full-trajectory methods			
LP (MV)	4×10^{-3}M	88.1 ± 0.2	87.9 ± 0.2
LP (Mean)	4×10^{-3}M	88.1 ± 0.1	87.8 ± 0.1
MLP (MV)	1.05M	88.5 ± 0.3	88.2 ± 0.3
MLP (Mean)	1.05M	88.6 ± 0.4	88.3 ± 0.4
TimeAttn	1.59M	88.8 ± 0.3	88.6 ± 0.3
LSTM	2.57M	88.6 ± 0.2	88.3 ± 0.2
$\mathcal{D}^2$-MLP (Ours)	0.24M	89.8 ± 0.1	89.3 ± 0.1
$\mathcal{D}^2$-TimeAttn (Ours)	0.35M	89.5 ± 0.1	89.2 ± 0.1

Table 8. **Intra-dataset performance on LLaDA-8B-Base.** Monitors are trained and tested on *WildGuardMix*. Best results are in **bold**, and second-best are underlined. ↓ indicates lower is better.

Method	E[P]	Acc	F1	F2	Precision	Recall	FRR ↓
Single-step methods							
LP (Last Step)	4×10^{-3}M	84.6	84.1	83.9	85.9	77.5	9.9
MLP (Last Step)	1.05M	85.8	85.4	85.1	87.6	78.6	8.7
Full-trajectory methods							
LP (MV)	4×10^{-3}M	86.7	86.2	85.8	89.9	78.2	6.8
LP (Mean)	4×10^{-3}M	86.9	86.5	86.2	90.0	78.9	6.8
MLP (MV)	1.05M	86.9	86.6	86.0	88.9	79.4	7.7
MLP (Mean)	1.05M	87.4	87.0	86.7	89.9	80.1	7.0
TimeAttn	1.59M	87.4	86.9	86.7	89.8	80.2	7.1
LSTM	2.57M	87.1	86.6	86.2	90.8	78.4	7.0
$\mathcal{D}^2$-MLP (Ours)	0.09M	88.1	87.8	87.4	**90.8**	81.0	**6.4**
$\mathcal{D}^2$-TimeAttn (Ours)	0.54M	**88.6**	**88.3**	**88.0**	90.3	**82.8**	6.9

Table 9. **Intra-dataset performance on LLaDA-8B-Instruct.** Monitors are trained and tested on *WildGuardMix*.

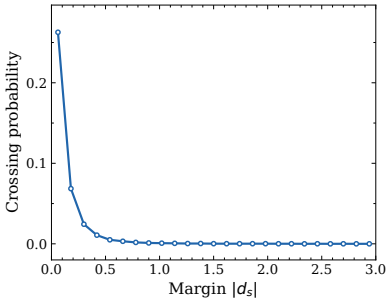
Method	E[P]	Acc	F1	F2	Precision	Recall	FRR ↓
Single-step methods							
LP (Last Step)	4×10^{-3}M	87.4	87.0	86.8	88.4	81.8	8.3
MLP (Last Step)	1.05M	87.1	86.8	86.6	88.6	81.0	8.1
Full-trajectory methods							
LP (MV)	4×10^{-3}M	88.2	87.9	87.8	88.0	84.5	9.0
LP (Mean)	4×10^{-3}M	88.2	87.9	87.8	88.4	84.1	8.5
MLP (MV)	1.05M	87.9	87.6	87.4	88.7	82.9	8.2
MLP (Mean)	1.05M	87.7	87.4	87.2	88.6	82.5	8.2
TimeAttn	1.59M	87.9	87.5	87.2	90.3	81.0	6.8
LSTM	2.57M	87.8	87.4	87.2	89.5	81.6	7.4
$\mathcal{D}^2$-MLP (Ours)	0.21M	**89.9**	**89.7**	**89.5**	**91.1**	**85.3**	**6.5**
$\mathcal{D}^2$-TimeAttn (Ours)	0.32M	89.6	89.4	89.2	91.0	84.6	6.5

Table 10. **Intra-dataset performance on LLaDA-1.5.** Monitors are trained and tested on *WildGuardMix*.

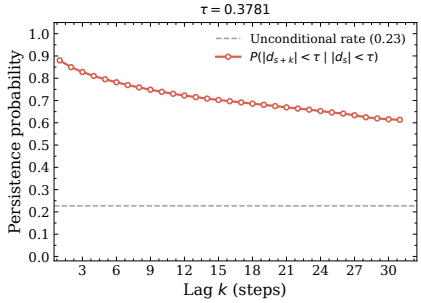
Method	E[P]	Acc	F1	F2	Precision	Recall	FRR ↓
Single-step methods							
LP (Last Step)	4×10^{-3}M	88.2	87.9	87.7	89.5	82.8	7.5
MLP (Last Step)	1.05M	87.9	87.5	87.3	89.9	81.4	7.1
Full-trajectory methods							
LP (MV)	4×10^{-3}M	87.8	87.5	87.4	87.1	84.5	9.7
LP (Mean)	4×10^{-3}M	87.9	87.7	87.6	87.4	84.6	9.5
MLP (MV)	1.05M	88.3	88.0	87.9	88.8	83.8	8.2
MLP (Mean)	1.05M	88.3	88.0	87.8	89.0	83.6	8.0
TimeAttn	1.59M	88.0	87.7	87.5	88.9	82.9	8.0
LSTM	2.57M	88.1	87.8	87.5	89.3	82.6	7.7
$\mathcal{D}^2$-MLP (Ours)	0.36M	**89.3**	**89.1**	**88.9**	**90.7**	84.2	**6.7**
$\mathcal{D}^2$-TimeAttn (Ours)	0.54M	89.3	89.0	88.8	90.0	**84.7**	7.3

Table 11. **Intra-dataset performance on LLaDA-2.0-mini.** Monitors are trained and tested on *WildGuardMix*.

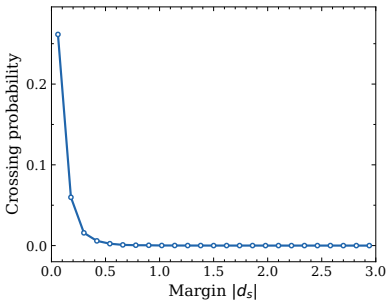
Method	E[P]	Acc	F1	F2	Precision	Recall	FRR ↓
Single-step methods							
LP (Last Step)	2×10^{-3}M	77.6	76.3	75.8	82.4	62.1	10.3
MLP (Last Step)	0.52M	79.0	78.0	77.6	81.8	66.7	11.5
Full-trajectory methods							
LP (MV)	2×10^{-3}M	80.0	79.0	78.4	85.5	65.5	8.7
LP (Mean)	2×10^{-3}M	80.5	79.4	78.9	85.9	66.2	8.4
MLP (MV)	0.52M	81.3	80.5	80.1	84.5	70.0	10.0
MLP (Mean)	0.52M	81.4	80.6	80.2	85.0	69.9	9.6
TimeAttn	0.80M	81.7	80.8	80.4	85.9	69.5	8.9
LSTM	1.51M	81.7	80.9	80.4	85.7	69.8	9.1
$\mathcal{D}^2$-MLP (Ours)	0.17M	83.7	82.9	82.5	88.2	**72.3**	7.5
$\mathcal{D}^2$-TimeAttn (Ours)	0.26M	**83.7**	**83.0**	**82.5**	**88.6**	72.0	**7.2**



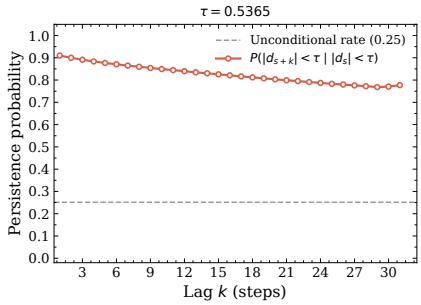
(a) LLaDA-8B-Base: Crossing



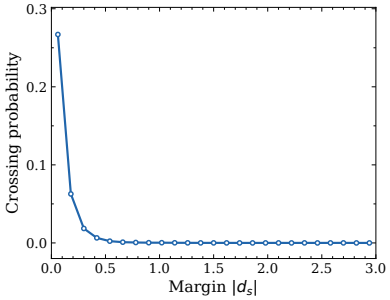
(b) LLaDA-8B-Base: Persistence



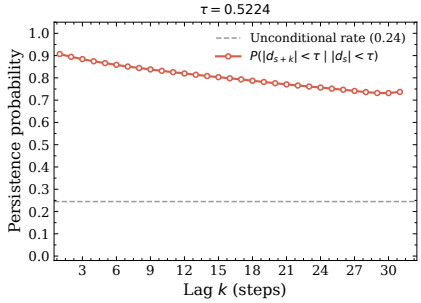
(c) LLaDA-8B-Instruct: Crossing



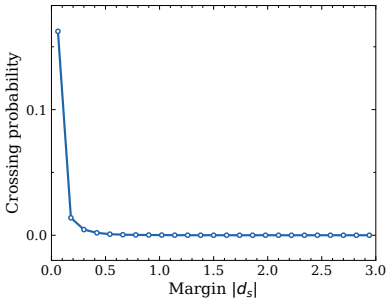
(d) LLaDA-8B-Instruct: Persistence



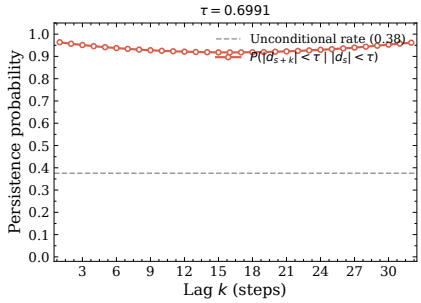
(e) LLaDA-1.5: Crossing



(f) LLaDA-1.5: Persistence

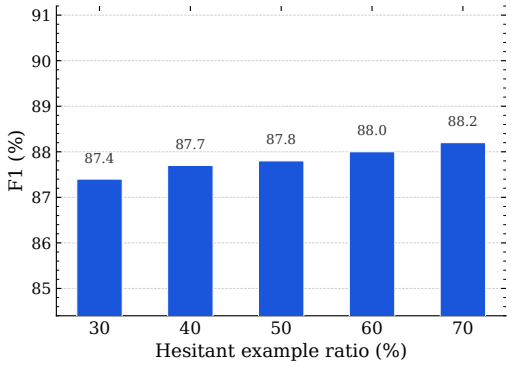


(g) LLaDA-2.0-mini: Crossing

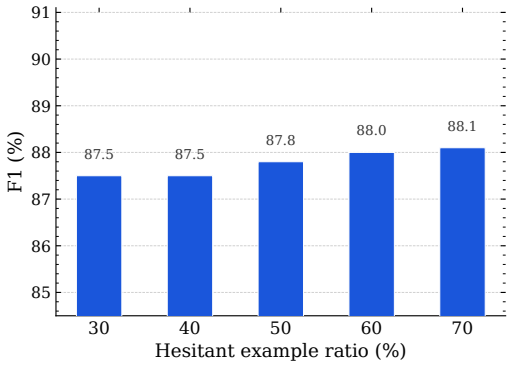


(h) LLaDA-2.0-mini: Persistence

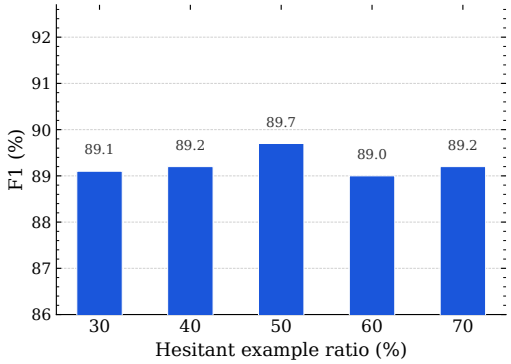
Figure 7. Cross-boundary probability (left) and margin persistence (right) across four LLaDA variants. Left: crossing probability as a function of margin $|d_s|$. Right: persistence probability $P(|d_{s+k}| < \tau \mid |d_s| < \tau)$. Margins are determined by OOF scoring on *WildGuardMix*.



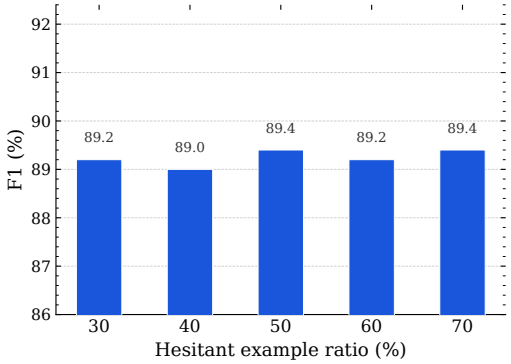
(a) LLaDA-8B-Base: D^2-MLP



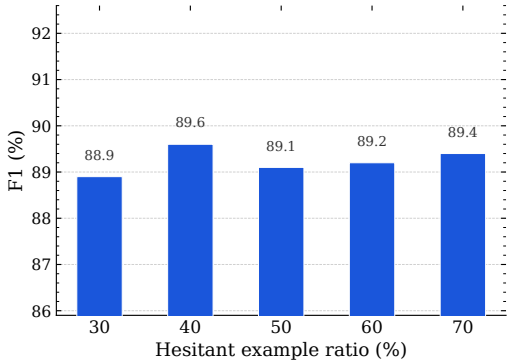
(b) LLaDA-8B-Base: D^2-TimeAttn



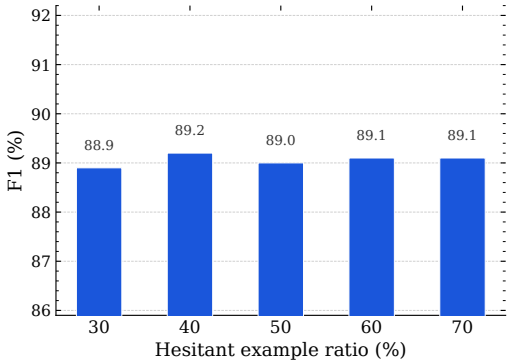
(c) LLaDA-8B-Instruct: D^2-MLP



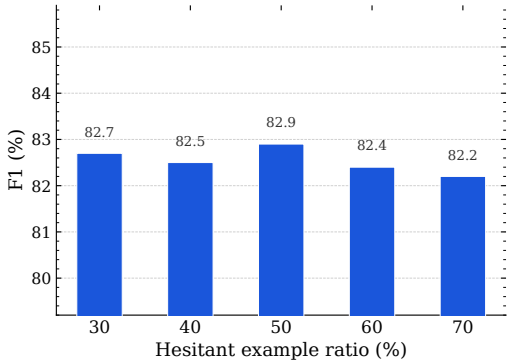
(d) LLaDA-8B-Instruct: D^2-TimeAttn



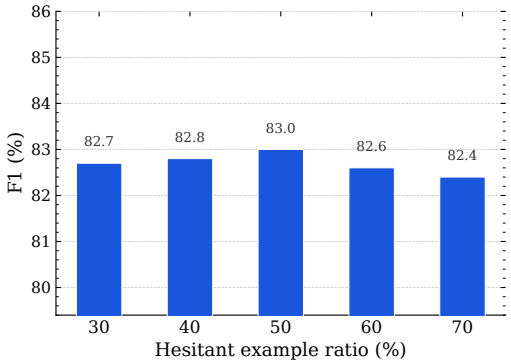
(e) LLaDA-1.5: D^2-MLP



(f) LLaDA-1.5: D^2-TimeAttn



(g) LLaDA-2.0-mini: D^2-MLP



(h) LLaDA-2.0-mini: D^2-TimeAttn

Figure 8. F1 score (%) vs. hesitant example ratio under D^2-MLP (left) and D^2-TimeAttn (right) across four LLaDA variants. All probes are trained and tested on *WildGuardMix*.