Evaluation without Generation: Non-Generative Assessment of Harmful Model Specialization with Applications to CSAM

Anonymous Authors[1]

Abstract

Auditing the fine-tunes of open-weight generative models for harmful specialization has become a new governance challenge for model hosting platforms. The standard toolkit, *generative evaluation* via curated prompts or red-teaming, does not scale to platform-level auditing and breaks down entirely for domains like child sexual abuse material (CSAM) where generation is legally constrained. This motivates the *Evaluation without Generation* problem: assessing model capabilities without producing outputs. In such settings, capability must be inferred from the model's state, either its parameters or internal representations, rather than its outputs. We introduce *Gaussian probing*, a method that characterizes how LoRA adaptors functionally perturb a model by measuring its internal responses to a reference ensemble of Gaussian latent states. Unlike raw-weight baselines, Gaussian probing reliably distinguishes benign from harmful specialization without sampling outputs. We demonstrate effectiveness in high-risk domains, including detecting models specialized for CSAM under realistic constraints. Our results show that Gaussian probing provides a scalable non-generative alternative for evaluating high-risk generative systems and remains robust to weight rescaling, a representative adversarial manipulation.

1. Introduction

The proliferation of open weight generative models, such as Stable Diffusion (Rombach et al., 2022), FLUX (Labs, 2024), and Wan (Wan et al., 2025), has made high-quality image and video generation widely accessible (Yang et al., 2023; Fuest et al., 2026). In tandem, low-rank adaptation (LoRA), a commonly used finetuning algorithm, enables cheap and efficient specialization of generative models at a fraction of the cost of traditional finetuning (Hu et al., 2022). Accessibility has also been improved by user-friendly graphical interfaces such as InvokeAI (Invoke AI Contributors, 2026), which allow amateurs and hobbyists to finetune models for their own creative purposes. As a result, powerful image and video models can be easily specialized, and those specializations can be shared through multiple different services, including public model-sharing platforms.

This shift has created new governance challenges for open model-sharing platforms. Platforms such as CivitAI and Hugging Face host and distribute base models, fine-tuned variants, and, crucially, lightweight LoRA adaptors that users can combine, circulate, and redeploy widely before any output is ever inspected. This includes models optimized to produce child sexual abuse material (CSAM), with offenders producing bespoke models through fine-tuning that target particular children, victims and survivors of child sexual abuse (Thiel et al., 2023). Governance, therefore, becomes a problem of screening reusable model artifacts before they are broadly distributed.

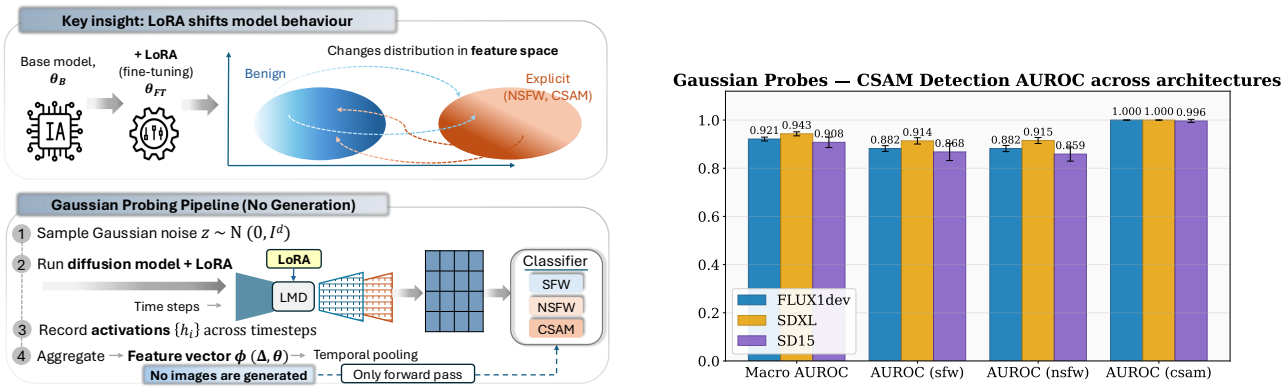
At present, model assessment for first-party model providers is still largely organized around **generative evaluation**: prompting an adapted model, inspecting its outputs, and using those outputs to infer whether the model has acquired a harmful capability (Thorn & All Tech is Human, 2024). This approach does not scale to model hosting platforms. It depends on prompt coverage, requires iterative red-teaming and review, and becomes increasingly costly as the number of uploaded variants grows. The scale of adaptors to be screened is substantial: CivitAI alone reports hundreds of thousands of new LoRAs trained in a single month, alongside millions of generations[1]. At this scale, output-based auditing cannot serve as the sole mechanism for pre-distribution governance.

For several categories of harmful content, the limitations are even more acute. Evaluating outputs related to bioweapons, cyberattacks, hate speech, or non-consensual intimate

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]https://www.runpod.io/case-studies/civitai-runpod-case-study



(a) *(b)*

Figure 1. (a) Key insight: LoRA fine-tuning shifts the model's feature distribution, separating benign and explicit outputs in latent space. Proposed Gaussian probing pipeline, which samples Gaussian noise, runs the diffusion model with LoRA, records intermediate activations across timesteps, and aggregates them into feature vectors without generating images. (b) Example results for all three architectures (SD 1.5, SDXL 1.0, FLUX.1-dev) for NSFW and CSAM detection in the wild.

imagery can impose substantial psychological burdens, and the expertise required for reliable evaluation may be scarce (Roberts, 2019; Steiger et al., 2021; Gillespie, 2018). For CSAM, our central motivating application, generative evaluation breaks down entirely: attempting to generate CSAM, regardless of intent or success, is unlawful behavior in several regulatory regimes and jurisdictions, including the United States (Shevlane et al., 2023).

This leaves platforms and auditors with limited ability to determine whether a user-uploaded adaptor has specialized a model toward CSAM generation before that adaptor is shared. LoRAs represent a unique risk, since their small and portable format makes them easy for offenders to exchange (Thiel et al., 2023). That limitation matters given AI-generated CSAM is a large and accelerating crisis, with cross-sector impact across hotlines, content moderators, law enforcement (Thiel et al., 2023; Internet Watch Foundation (IWF), 2026), creating significant human harm[2]. In its most recent report analyzing 2024 reports, NCMEC (the National Center for Missing and Exploited Children, which acts as a global clearinghouse for reports related to child sexual abuse and exploitation) received 67,000 reports of AI-generated CSAM, up from 4,700 in 2023 (for Missing & Exploited Children, 2025).

Taken together, this harm landscape motivates our core questions:

Can harmful specialization be detected from weights alone, without ever generating an output?

We call this the **Evaluation without Generation** problem.

[2] https://www.cbsnews.com/news/sextortion-generative-ai-scam-elijah-heacock-take-it-down-act/

In this work, we study a concrete and narrower instance that arises in open ecosystems: screening LoRA adaptors for evidence of *direct specialization* toward two sensitive content categories, adult sexual content (NSFW) and CSAM, without generating outputs. We focus on LoRAs because they are the primary unit through which specialization is created, packaged, and distributed. We answer this in the affirmative by shifting evaluation from output space to the model's state. Rather than measuring what a model generates, we measure how fine-tuning changes its parameters or internal computation. This reframes capability evaluation as an inference problem over model state rather than outputs.

We propose **Gaussian probing**, a method that characterizes how a LoRA functionally perturbs a base diffusion model by measuring its internal responses to random Gaussian inputs. These responses provide a scalable, prompt-free signature of adaptor specialization without requiring image generation. We evaluate the method in two settings. First, in controlled experiments on in-house trained safe-for-work (SFW) and not-safe-for-work (NSFW) LoRAs spanning variation in datasets, styles, architectures, and training conditions. Second in a naturalistic setting using LoRAs collected from public platforms such as CivitAI and CSAM LoRAs accessed through authorized entities and in accordance with applicable laws, where heterogeneity, label noise, and shortcut opportunities better reflect deployment conditions. Across both regimes, Gaussian probing reliably detects harmful NSFW and CSAM specialization while remaining more robust than raw-weight baselines to superficial training artifacts. To our knowledge, *this is the first scalable, non-generative method for pre-distribution screening of CSAM-related specialization in user-uploaded generative models.*

2. Problem Formulation

2.1. Defining the Governance Target and Scope

Open platforms face a practical question when users upload LoRA adaptors: does this warrant intervention before it is widely shared? Put differently, can an uploaded adaptor contribute to harmful downstream use? Answering that question requires some care, because capability is not an intrinsic property of a LoRA in isolation. Whether an adaptor is treated as harmful depends on how it is expected to be used and on what kind of behavior a platform is prepared to tolerate. For example, a platform might choose to permit adaptors with moderate adult-content propensity, while adopting a zero-tolerance rule for any minors-related propensity.

In full generality, a capability definition is difficult to identify. The relevant use distribution may be uncertain, institution-specific, or strategically shaped by users, and the policy itself is partly normative. In this work, we therefore study a narrower but still useful case of *direct harmful specialization*: whether a LoRA was directly fine-tuned on data drawn from a target harmful content category, specifically, CSAM. This does not capture every route by which harmful capability may arise, such as LoRA composition, model merging, or more complex downstream chaining. But it does capture an important and operationally meaningful slice of the broader governance problem, namely the one most directly tied to the uploaded artifact itself and most amenable to pre-distribution screening.

Even under this narrower target, the auditing problem remains difficult. The auditor must detect harmful specialization from the uploaded adaptor and base model alone, under constraints of observability, scale, and robustness. We formalize these constraints as the following methodological desiderata:

(D1) *Non-generative observability.* The representation must be computable from the uploaded adaptor and the base model alone, without requiring prompt design, output generation, or human review.

(D2) *Scalability.* A practical auditor must operate under realistic storage and compute constraints. First, the resulting representation must be much smaller than the full parameter dimension, so that a large repository of adaptors can be stored and processed in memory. Second, the cost of constructing the representation for a single adaptor must remain feasible at audit scale.

(D3) *Robustness.* The representation should reflect direct harmful specialization rather than incidental features of how a LoRA was produced. We identify three types of signal that *can* be predictive and describe which are the ideal ones to utilize:

- **development artifacts** such as rank, learning-rate schedules, or update magnitude
- **dataset identity** such as stylistic or distributional peculiarities of a particular training image datasets
- **content signal** or information tied to the underlying harmful category that persists across datasets, fine-tuning runs, and implementation choices

This distinction matters for generalization and robustness to adversarial changes.

2.2. Formalizing the Screening Task

Fix a harmful content category c and a pretrained base diffusion model $f_{\theta_{\text{base}}}$. For a LoRA adaptor Δ, let $f_{\theta(\Delta)} := f_{\theta_{\text{base}}+\Delta}$ denote the adapted model. For each adaptor Δ, let $y(\Delta) \in \{0, 1\}$ indicate whether Δ was directly fine-tuned on data drawn from category c.

The auditor observes $(\theta_{\text{base}}, \Delta)$, but not the underlying training data and not any generated outputs. The auditing task of interest is given labeled adaptors $\{(\Delta_i, y(\Delta_i))\}_{i=1}^n$, predict $y(\Delta)$ from weights alone. We consider screening rules of the form $g \circ \Phi$, where $\Phi(\Delta; \theta_{\text{base}}) \in \mathcal{H}$, is a fixed representation map and $g : \mathcal{H} \to \{0, 1\}$, is a learned classifier. The auditor solves

$$\hat{g}_\Phi \in \arg\min_g \frac{1}{n} \sum_{i=1}^{n} \ell(y(\Delta_i), g(\Phi(\Delta_i; \theta_{\text{base}}))) + \lambda \|g\|^2.$$

The central technical question is therefore how to construct a representation $\Phi(\Delta; \theta_{\text{base}})$ that recovers direct harmful specialization from restricted, non-generative evidence while satisfying the scalability and robustness desiderata.

3. Gaussian Probing

We instantiate the representation map $\Phi(\Delta; \theta_{\text{base}})$ by probing the adapted model on a reference ensemble of Gaussian latent states. This representation summarizes how the adaptor changes the computation implemented by the base model.

Let H, W, and C denote the height, width, and number of channels of the diffusion model input, and define $\bar{d} = HWC$. Let $\theta(\Delta) = \theta_{\texttt{base}} + \Delta$ denote the adapted parameters. We draw m i.i.d. Gaussian probes

$$\nu_j \sim \mathcal{N}(0, I_{\bar{d}}), \qquad j = 1, \ldots, m.$$

Each probe ν is propagated through the diffusion process using the adapted model $f_{\theta(\Delta)}$ for T denoising steps. During this process, we extract intermediate hidden representations from a fixed layer, or fixed set of layers, of the denoising

network.[3] Let $H^{(T)}(\Delta;\nu), H^{(T-1)}(\Delta;\nu), \ldots, H^{(1)}(\Delta;\nu)$ denote the resulting hidden representations across diffusion timesteps. For a single probe ν, we aggregate these hidden states into a probe-specific feature

$$\bar{H}(\Delta;\nu) = \psi\left(H^{(1)}(\Delta;\nu), \ldots, H^{(T)}(\Delta;\nu)\right),$$

where ψ is a fixed pooling map across timesteps and, when relevant, across layers, spatial positions, and channels. In the simplest case, ψ is the timestep average

$$\bar{H}(\Delta;\nu) = \frac{1}{T}\sum_{t=1}^{T} H^{(t)}(\Delta;\nu).$$

The population object underlying Gaussian probing is the probe functional

$$\Psi(\Delta) := \mathbb{E}_{\nu\sim\mathcal{N}(0,I_{\bar{d}})}\left[\bar{H}(\Delta;\nu)\right]. \tag{1}$$

This function summarizes how the adapted model responds, on average, to a reference ensemble of Gaussian latent states. Our empirical representation is the corresponding Monte Carlo estimator

$$\Phi(\Delta;\theta_{\text{base}}) := \widehat{\Psi}_m(\Delta) = \frac{1}{m}\sum_{j=1}^{m} \bar{H}(\Delta;\nu_j). \tag{2}$$

As we will illustrate, this can be interpreted as the expected pushforward of the LoRA perturbation through the model's denoising dynamics under the native latent distribution.

3.1. Motivation for Gaussian Probing

What the auditor ultimately cares about is whether a LoRA changes the model in a way that supports harmful downstream use. In principle, the most direct object of study would therefore be the distribution of model outputs under some reference input ensemble. But in the settings that motivate this work, outputs cannot be generated. Therefore our approach replaces outputs with internal activations: rather than asking what the adapted model renders, we ask how the adaptor changes the model's internal response profile on the diffusion process's native Gaussian state space.

This substitution is meaningful only if intermediate activations carry semantically relevant information. Prior work suggests that they do. In particular, diffusion models appear to organize high-level concepts in internal latent representations that remain coherent across denoising steps, and those representations can be used to decode or steer semantic attributes of generation (Kwon et al., 2022). We leverage the same structure for a different purpose. Rather than steering outputs, we use internal responses to detect whether a LoRA systematically shifts the model toward a harmful specialization.

This intuition can be made more precise through a local linearization. Fix a timestep t and probe ν, and suppose the extracted hidden state is differentiable with respect to the model parameters. Then for a LoRA-induced perturbation Δ,

$$H^{(t)}(\Delta;\nu) - H^{(t)}(0;\nu) = D_\theta H^{(t)}(\theta_{\text{base}};\nu)[\Delta] + o(\|\Delta\|),$$

where $D_\theta H^{(t)}$ denotes the Fréchet derivative of the hidden state mapping at the base parameter configuration. Equivalently, after vectorization,

$$H^{(t)}(\Delta;\nu) - H^{(t)}(0;\nu) \approx J_t(\theta_{\text{base}};\nu)\,\text{vec}(\Delta),$$

where J_t is the Jacobian matrix representing the sensitivity of the internal representations to parameter changes.

Under this view, Gaussian probing is not an arbitrary feature extractor. It measures how the LoRA is pushed forward into activation space by the denoising dynamics of the base model. Averaging over Gaussian probes emphasizes parameter directions that actually affect the model's computation on its native latent state space, rather than treating all directions in weight space as equally meaningful. Because LoRA updates are low-rank, the resulting activation shift is constrained to a locally low-dimensional subspace. This gives a principled reason to expect that harmful and benign specializations may be distinguishable by relatively simple decision rules, including linear classifiers, if their induced activation responses differ systematically.

We use Gaussian noise as the reference ensemble for two reasons. First, it is native to the diffusion process itself, so it provides a prompt-free way to interrogate the model's denoising dynamics without making assumptions about future user prompts. Second, the probes are sampled independently of the fine-tuning pipeline. Because they do not inherently encode dataset metadata or rank selection, these responses provide a cleaner signal for desideratum **(D3)**. If a classifier built on Gaussian probe responses separates harmful from benign adaptors, that separation will likely come from how the LoRA functionally perturbs the model on the reference state space instead of superficial traces of how the LoRA was produced. In this sense, Gaussian probing is designed to privilege content-relevant functional signal over incidental training artifacts.

Since $\Phi(\Delta;\theta_{\text{base}})$ is a Monte Carlo estimator of $\Psi(\Delta)$, standard law-of-large-numbers arguments imply consistency under mild moment assumptions; we state this formally in Appendix B. The central question is therefore empirical: whether the induced shift in this representation is both suffi-

[3] In practice, we extract activations from one or more designated internal layers of the denoising network, such as a mid-block feature map of the U-Net in Stable Diffusion 1.5.

ciently large and sufficiently well-estimated at finite sample sizes to support reliable screening in practice.

Gaussian probing satisfies the desiderata. Computing $\Phi(\Delta; \theta_{\text{base}})$ requires only forward passes through the adapted model on Gaussian inputs: no outputs are decoded, no images are rendered, and no prompt is designed or reviewed, satisfying **(D1)**. The computation requires $O(mT)$ forward passes, is parallelizable across the audit set, and produces a representation of fixed dimension $|\mathcal{H}|$ regardless of the adaptor's rank or structure. This representation is orders of magnitude smaller than the full parameter dimension d, satisfying **(D2)**. Finally, because the representation captures the functional effect of Δ on intermediate activations rather than its footprint in weight space, in principle, two adaptors with identical functional behavior but different weight norms or training configurations will produce similar probe activations, while two adaptors with different specialization will not, satisfying **(D3)**. The remainder of the paper evaluates whether it succeeds in doing so empirically.

4. Evaluating Gaussian Probing Against the Desiderata

We evaluate how well the representation Φ satisfies the desiderata introduced in Section 2.1. We begin with a controlled setting, using LoRAs trained on SFW and adult pornography (NSFW) content to isolate separability and signal quality. We then consider a naturalistic setting using LoRAs from CivitAI to study robustness to training artifacts.

We compare Gaussian probing to classifiers built on raw weight representations, including random projections of weight matrices. While these methods are simple and non-generative, they treat all directions in parameter space as equally meaningful and may therefore rely on incidental features of the training process rather than underlying content specialization.

We evaluate three questions: (1) Does harmful specialization induce sufficient separation in representation space? (2) Do representations capture content signal rather than dataset identity or artifacts? (3) Are these properties preserved in naturalistic settings and under adversarial conditions?

4.1. Controlled Study

Experimental Setup To produce our SFW LoRAs, we use six different datasets: COCO2017 (Lin et al., 2014), Flickr30K (Plummer et al., 2015), Conceptual Captions 12M (Changpinyo et al., 2021), LAION-Aesthetics (Schuhmann et al., 2022), OpenImages (Kuznetsova et al., 2020), Unsplash-Lite (Unsplash, 2017), and Wikiart (Saleh & Elgammal, 2015). Together, these cover a broad range of benign content, including images of humans (faces, poses, non-explicit suggestive content), landscapes, objects, art, and fashion. We also use three adult pornography datasets in this task: NSFW Video Still (Morelli et al., 2016), Danbooru2023 (Nyanko, 2023), Amaye15 (Mayes, 2025). All datasets are filtered using Thorn's Safer hashing and classification technology to ensure removal of CSAM.[4] For each adult dataset we create 10 different random samples, along with additional subsets (e.g., cartoon-only and varying levels of nudity) to support our investigation into the kind of signal being encoded by these representations.

We train 1,000 LoRAs per class (SFW and NSFW), varying dataset, sample, rank, learning rate, modules, training steps, and data size. This randomization ensures that models cannot rely on spurious correlations tied to specific training configurations, forcing representations to capture underlying content signal. In order to avoid the failure mode we already identified, we intentionally choose to remove these potential confounds. For Gaussian probing, we sample 1024 probes for each model and for raw weights we use a projection dimension of 256 for every layer.

We evaluate across Stable Diffusion 1.5, SDXL 1.0, and FLUX.1-dev, spanning a range of model sizes and architectures from 860M to 12B parameters, including both U-Net and transformer-based diffusion models. This diversity allows us to study the impact of architecture and scale, as well as the role of layer selection for representation extraction. In particular, we examine which layers carry the most semantic signal, especially for SDXL and FLUX where prior guidance is limited.

We report accuracy, precision, recall, F1, FPR, and FNR under two settings: (1) standard 5-fold cross-validation, and (2) a leave-dataset-out (LDO) setting to test conceptual generalization. For this particular application, we are most interested in the AUROC, the precision and the false positive rate (FPR). Thus, these are the metrics we report throughout the plots in the main paper. All other metrics are reported in Appendix D.2.

It is important to note that this controlled study eliminates many of the confounds which make this task inherently more difficult in the wild. These confounds include: which base checkpoint the LoRA was finetuned from, biases in which layers of the model are finetuned, and each individual LoRA being trained on a different dataset. As a result, we expect that the performance we see in this controlled setting will be higher than what we would see in the wild. Nevertheless, it more cleanly allows us to investigate the questions we outlined without the confounds playing a role.

[4] https://safer.io/solutions/

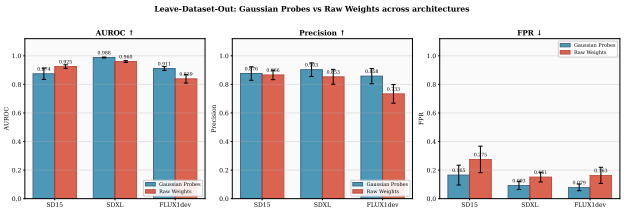


Figure 2. Average AUROC, precision, and FPR across 5 fold CV fo all architectures on the leave-dataset-out evaluation. Randomly projected weights outperform Gaussian probing depending on the model, but we identify that this is due to dataset identity signal in Section 4.1.2, which does not satisfy our desiderata. Meanwhile, Gaussian probing still performs well while primarily leveraging the content signal in accordance with our robustness desiderata.

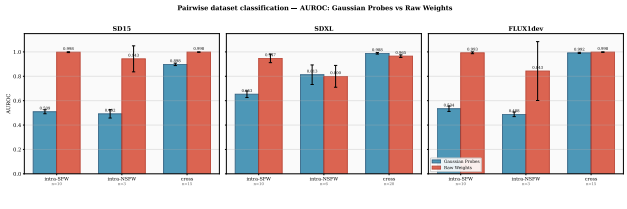


Figure 3. Average AUROC values for intra-SFW, intra-NSFW, and cross-class dataset pairs for both raw weights and Gaussian probing. For SD 1.5 and FLUX.1-dev Gaussian probing clearly is relying on content signal given both intra-SFW and intra-NSFW AUROC are near random while cross-class AUROC is high. This suggests that Gaussian probing encodes a signal that is useful for distinguishing the two concepts, without relying on lower-level dataset-specific properties.

4.1.1. Separability of SFW and NSFW Specialization

We begin with SD 1.5, as this is the simplest of the models and the most studied in the interpretability literature. Both representations achieve strong performance under standard cross-validation (Figure 10), with Gaussian probing showing slightly lower FPR. The performance is exceptionally strong and we believe shows the ceiling of what is possible when all confounds are controlled. A more appropriate metric is the leave-dataset-out (LDO) setting. Given that we are leaving one entire SFW and NSFW dataset out during training, we get a better sense of *conceptual generalization*. Randomly projected weights generalize better than gaussian probing but both do well (Figure 2). However, this apparent advantage relies on stable dataset-level artifacts. As we illustrate in the next section, in settings where such artifacts can be manipulated, these representations degrade under even simple transformations, whereas Gaussian probing targets invariant functional signal.

For SDXL 1.0 and FLUX.1-dev, we see similarly high performance in the standard CV evaluation (Figure 12 and Figure 14). For LDO evaluation, we find that for SDXL 1.0, probes generalize better. Meanwhile, for FLUX.1-dev, raw weights generalize better. In the next section, we investigate whether this generalization comes from signal that satisfies our auditor desiderata or if it is due to spurious signals that will inevitably undermine robustness. Overall, these results demonstrate that both representations we have presented are capable of distinguishing SFW trained LoRAs from NSFW trained LoRAs. We ensemble probes from both latent diffusion and text encoder modules across all three architectures in Appendix D.3 show this is essential for detecting text-encoder-only finetunes for SD 1.5.

4.1.2. Content Signal vs. Dataset Identity

We next evaluate how well the representations satisfy the robustness desideratum, in particular whether they capture content signal rather than dataset identity. The leave-dataset-out results in Section 4.1.1 left open whether raw weights' advantage reflects genuine signal or exploitable artifacts. This section shows it's the latter.

To isolate the source of this signal, we construct a pairwise dataset classification task over SFW and NSFW LoRAs. For each pair of datasets, we train classifiers using both randomly projected weight and Gaussian probing features, and measure performance within and across classes. A representation that primarily encodes dataset identity will achieve high and consistent performance across intra-SFW, intra-NSFW, and cross-class comparisons, whereas a representation that captures content signal should perform well only on cross-class distinctions.

First, we examine whether simple training artifacts could explain the observed separability. In particular, we compare the norms of LoRAs across and within classes and find them to be similar, reducing the likelihood that magnitude alone drives classification. Additionally, given that we randomly selected across all training hyperparameters we capture the vast space of possible configurations. Thus, we are less concerned that the representations are separable by training artifact information alone.

A representation that encodes dataset identity will separate any two datasets regardless of class. A representation that encodes content signal will separate only across class (SFW vs. NSFW), not within. Figure 3 shows which regime each method is in. Overall, we find that Gaussian probes exhibit a stronger reliance on content signal. This is most evident in SD 1.5 and FLUX.1-dev, where projected raw weights can distinguish between datasets regardless of class, indicating sensitivity to dataset-specific artifacts rather than underlying content. Interestingly, for SDXL 1.0 where Gaussian probes do look like they use a combination of dataset identity and content signal it outperforms the randomly projected weights in the LDO evaluation. These results demonstrate that while Gaussian probes tend to perform worse on the LDO evaluation than randomly projected weights, they are using the ideal signal. Thus, we believe they are more robust

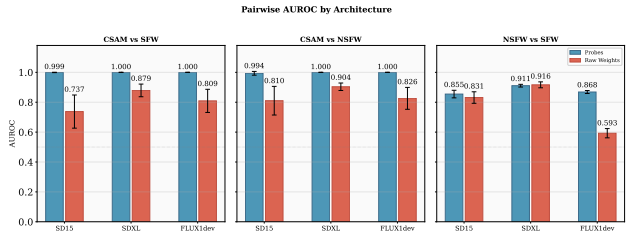


Figure 4. AUROC across our three models (SD 1.5, SDXL 1.0, and FLUX.1-dev) computed for each pair of classes (SFW vs. CSAM), (NSFW vs. CSAM), and (SFW vs. NSFW). Gaussian probing separates CSAM from both SFW and NSFW across all three architectures. Raw weights fail on small-sample CSAM detection (AUROC 0.68-0.87). NSFW vs CSAM should be viewed as the most difficult of the three. Although, raw weights do perform comparably on some of the folds, there is high variance. Meanwhile the methods perform more comparably when looking at SFW vs NSFW.

and satisfy the auditor desiderata we outlined.

4.2. In the Wild Study

In a naturalistic setting using 1,118 SDXL LoRAs from CivitAI, we investigate whether classifiers rely on superficial training artifacts rather than underlying content signal. We observe systematic differences between SFW and NSFW LoRAs in metadata such as training steps, learning rates, and rank, which enables strong classification performance using both Gaussian probes and raw-weight features. However, further analysis reveals that raw-weight methods exploit these artifacts—particularly weight magnitude—as shortcuts, leading to brittle behavior. When we simulate an adversarial attack by normalizing per-layer Frobenius norms (removing magnitude information while preserving functional behavior), performance of raw-weight classifiers degrades substantially (e.g., large drops in AUROC and increased false positive rates), whereas Gaussian probing remains stable or improves slightly. This demonstrates that artifact-based signals are easily manipulated and highlights the importance of using representations that capture functional, content-level behavior rather than incidental properties of the training process. Further details are given in **??**.

5. Detecting CSAM Specialization in the Wild

In this section, we focus on the motivating application of this work: detecting LoRAs specialized for generating child sexual abuse material using Gaussian Probing. Having demonstrated the effectiveness of the different representations for LoRA classification in satisfying our desiderata of non-generative, scalability, and robustness (Sections 4.1.2 and **??**), we focus on applying our algorithm on separating between LoRAs specialized to generate SFW, NSFW, and CSAM content. We did not possess or generate any CSAM data or CSAM-specific LoRAs, nor did we train any LoRAs on such material; all such data remained solely with authorized entities and handling of the data was done in accordance with applicable laws and organizational safeguards. We intentionally omit further details to reduce misuse risk and preserve operational security. We discuss our reasoning for omitting further details in Section A.

Experimental Setup For this section, our goal is to simulate deployment of this approach in the wild. Thus, we source around 1,000 LoRAs from CivitAI for the SFW and NSFW classes. Additionally, for the purposes of this research, we accessed CSAM-specialized LoRAs through authorized entities, without handling underlying CSAM data and in compliance with applicable laws and organizational safeguards. We discuss our choice to intentionally omit all other details regarding access to these LoRAs in the Ethical Considerations section of the paper. For SD 1.5, our sample size consists of 18 CSAM-specialized LoRAs, for SDXL 1.0 our sample size is 34 CSAM-specialized LoRAs, and for FLUX.1-dev our sample size is 74 CSAM-specialized LoRAs. We test the performance of the different representations in this prediction task across the same three models as in the previous section. For Gaussian probing, we sample 512 probes for SD 1.5 and SDXL 1.0. Due to computational and operational security constraints surrounding our CSAM-specialized FLUX.1-dev LoRAs we only use 4 probes. For raw weights we use a projection dimension of 256 for every layer. Given that we no longer are controlling for any of the confounds we expect to see in the wild, we expect our method to perform well but less strong as our controlled experiments. As in Section 4, we focus on AUROC, precision, and FPR as our primary metrics of interest. We report additional metrics in the Appendix.

Pairwise Detection Across all three models and classes, Gaussian probing stands out as the better representation for detection. Raw weight projections, given their high dimensionality, struggle to classify CSAM-specialized LoRAs given their small sample size. Meanwhile, Gaussian probing is effective at correctly classifying all CSAM-specialized LoRAs (Figure 4 and Figure 6)). We view these results as promising for the effectiveness of Gaussian probing for our motivating application. At the same time, these results should be interpreted as evidence of separability under the current data conditions, rather than as a claim of near-perfect detection in deployment settings.

Per-Class Results and Errors Within our results, we investigate the distribution of errors made by our probe classifier. In particular, we are interested in understanding how many SFW and NSFW LoRAs are classified as being CSAM-specialized LoRAs. While SFW LoRAs can most

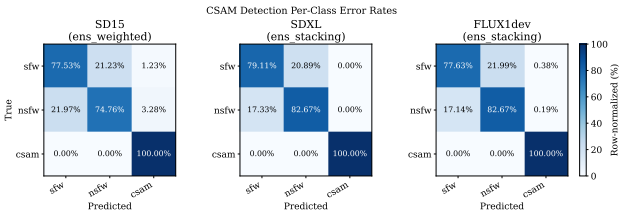


Figure 5. CSAM detection error rates by true class, across architectures averaged over 5 folds. Gaussian probing achieves 100% recall on CSAM across SD 1.5, SDXL, and FLUX.1-dev (bottom row), with under 1% of SFW LoRAs misclassified as CSAM for SD 1.5 and SDXL. The 2–4% of NSFW LoRAs predicted as CSAM may reflect genuine CSAM contamination in CivitAI uploads rather than classifier errors, and warrant follow-up inspection about which of the two they are.

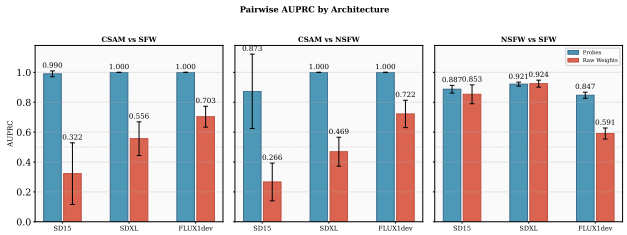


Figure 6. AUPRC across our three models (SD 1.5, SDXL 1.0, and FLUX.1-dev) computed for each pair of classes (SFW vs. CSAM), (NSFW vs. CSAM), and (SFW vs. NSFW). Note that Gaussian probing outperforms raw weights significantly when tasked with distinguishing CSAM from either SFW or NSFW. NSFW vs CSAM should be viewed as the most difficult of the three. The methods perform more comparably when looking at SFW vs NSFW.

likely be considered true errors, it is possible that some of the NSFW LoRAs that are classified as being CSAM-specialized were in fact trained with some amount of CSAM data during finetuning. We find that across model architectures, probes classify all CSAM models correctly and have low false positive rates for classifying both SFW and NSFW models as CSAM. We see the highest FPR for FLUX.1-dev models at 4.21% of SFW models being classified as CSAM specialized and 1.90% of NSFW models being classified as CSAM (Figure 5). We view the confusion between SFW and CSAM as true errors, while the NSFW models could in fact be undetected CSAM-specialized models, warranting further inspection to distinguish true errors from contamination.

We present additional ablations on layer choice and ensembling in Appendix E.

6. Discussion and Conclusion

Assessing model capabilities and specialization for harmful content has so far focused on *generative evaluations*, which capture what models generate under curated prompts rather than what they are capable of generating under adversarial or untested conditions. In domains where generation is restricted or unlawful, this creates a measurement gap precisely where risk is highest. In applications such as CSAM detection, output-based evaluation is limited and legally challenging, motivating *non-generative assessments*.

In this work, we show that analyzing model weights, via mechanistic or functional representations, provides a practical and scalable approach for non-generative evaluation of model specialization. We validate this on detecting CSAM specialization in the wild. Non-generative evaluation is essential in constrained settings such as CSAM or NCII detection, and complementary in domains where generative evaluations are allowed (e.g., bioweapons, hate speech, cyberattacks), reducing reliance on large-scale human review and associated harms. Our approach, Gaussian probing, highlights the value of functional representations in weight-space learning, particularly for scalability and robustness, and provides criteria for evaluating such methods.

Our work addresses a central open problem in AI child safety regarding evaluation under generation constraints, demonstrating that non-generative evaluation is possible and enabling future research on vulnerabilities, theory, and improved algorithms. Solutions to the evaluation without generation problem are broadly applicable, particularly for open-weight hosting platforms, where they enable scalable pre-screening of uploaded LoRAs and improve noisy concept tagging systems.

At the same time, capability-level filtering raises questions about moderation and censorship, requiring careful definition of unacceptable capabilities and balancing harm prevention with legitimate use. Further research is needed to build robustness to adversarial manipulation beyond weight rescaling and to extend this approach beyond LoRA finetuning to pretrained models, where scale and label availability pose additional challenges.

Overall, this work reframes evaluation as a weight-space auditing problem and provides a scalable approach to pre-deployment screening of harmful specialization, enabling capability assessment where output-based evaluation is constrained while supporting safer deployment and governance.

References

bmaltais. kohya_ss: Gui and cli for training diffusion models. https://github.com/bmaltais/kohya_ss, 2025. Accessed: 2026-04-23.

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3558–3568, 2021.

for Missing & Exploited Children, N. C. 2024 cybertipline report. https://www.missingkids.org/content/dam/missingkids/pdfs/cybertiplinedata2024/2024-CyberTipline-Report.pdf, 2025.

Fuest, M., Ma, P., Gui, M., Schusterbauer, J., Hu, V. T., and Ommer, B. Diffusion models and representation learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Gillespie, T. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press, 2018.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Internet Watch Foundation (IWF). Harm without limits: Ai child sexual abuse material through the eyes of our analysts, March 2026. URL https://www.iwf.org.uk/media/hl1nvdti/iwf-ai-csam-report-2026.pdf.

Invoke AI Contributors. Invokeai, 2026. URL https://github.com/invoke-ai/InvokeAI.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Kwon, M., Jeong, J., and Uh, Y. Diffusion models already have a semantic latent space. *arXiv preprint arXiv:2210.10960*, 2022.

Labs, B. F. Flux. https://github.com/black-forest-labs/flux, 2024.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Mayes, A. Object segmentation dataset. https://huggingface.co/datasets/amaye15/object-segmentation, 2025. Accessed: 2024-04-08.

Morelli, A. et al. Opennsfw: A dataset for nsfw image classification. Yahoo open source project, 2016. URL https://github.com/yahoo/open_nsfw.

Nyanko. Danbooru2023. https://huggingface.co/datasets/nyanko7/danbooru2023, 2023. Accessed: 2024-04-08.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pp. 2641–2649, 2015.

Roberts, S. T. *Behind the screen*. Yale University Press, 2019.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Saleh, B. and Elgammal, A. Large-scale classification of fine-art paintings: Learning the right metric on the right feature, 2015. URL https://arxiv.org/abs/1505.00855.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35: 25278–25294, 2022.

Shevlane, T., Farquhar, S., Garfinkel, B., Phuong, M., Whittlestone, J., Leung, J., Kokotajlo, D., Marchal, N., Anderljung, M., Kolt, N., Ho, L., Siddarth, D., Avin, S., Hawkins, W., Kim, B., Gabriel, I., Bolina, V., Clark, J., Bengio, Y., Christiano, P., and Dafoe, A. Model evaluation for extreme risks, 2023. URL https://arxiv.org/abs/2305.15324.

Steiger, M., Bharucha, T. J., Venkatagiri, S., Riedl, M. J., and Lease, M. The psychological well-being of content moderators: the emotional labor of commercial moderation and avenues for improving support. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pp. 1–14, 2021.

Thiel, D., Stroebel, M., and Portnoff, R. Generative ml and csam: Implications and mitigations. *Stanford Digital Repository*, 2023.

Thorn and All Tech is Human. Safety by design for generative ai: Preventing child sexual abuse. https://info.thorn.org/hubfs/thorn-safety-by-design-for-generative-AI.pdf, 2024.

Unsplash. Unsplash lite dataset. https://github.com/unsplash/datasets, 2017.

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., and Wolf, T. Diffusers: State-of-the-art diffusion models, 2022.

Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.-W., Chen, D., Yu, F., Zhao, H., Yang, J., et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 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.

A. Ethical Considerations

This work presents a new paradigm for assessing the extent to which image generation models have been finetuned on harmful content, specifically CSAM, without any content generation. In this research process and its communication in this paper, numerous steps were taken to conduct it ethically.

First, in the curation of adult sexual content from the internet, we ran [REDACTED] detection technology over the image datasets and removed any images flagged as potentially containing CSAM. However, we recognize that it is still possible for adult non-consensual intimate imagery (NCII) to exist in the dataset we used for training and the re-victimization harm that may arise from this. Any presence of such material, regardless of scale, can cause trauma and must not be minimized. We believe that research such as this is critical to building scalable mechanisms by which to prevent the proliferation of models specialized for the creation of CSAM and NCII. We want to highlight that this work is another example of why research into adult content datasets collected consensually are essential. Upon publication of this paper, we will be deleting all of the adult content datasets and LoRAs specialized on adult content.

Second, we intentionally omitted and abstracted significant details about the LoRAs specialized for CSAM generation on which we evaluate our methodology. We do not disclose details on their whereabouts, verification, names, or how they were obtained. In doing so, we ensure that no information is provided that would enable individuals to locate these models beyond what is publicly available, thereby reducing misuse risk and preserving operational security. We affirm that we never possessed, accessed, attempted to, or generated CSAM. We also did not train any LoRAs on CSAM ourselves or possess CSAM specialized LoRAs. All such data remained solely with authorized entities, and all testing was conducted in compliance with applicable laws and organizational safeguards. In order to justify our approach and to show its efficacy in detection, we have evaluated the approach on non-CSAM tasks, as demonstrated in the experimental section of this paper.

Third, we believe that publication of these results is beneficial to the AI safety ecosystem as a whole and will help prevent additional harm. By demonstrating a method that can assess abuse capabilities in generative models without having to generate illicit content, we provide model hosting platforms, AI developers, policy makers, and safety researchers with a legally viable tool for scalable proactive detection of abuse enabled generative models. We aim to support continued innovation while equipping stakeholders with technology to detect and curb the proliferation of such models.

B. Consistency of Gaussian Probing

The population object underlying Gaussian probing is given by (1), and the empirical representation (2) is its Monte Carlo approximation. The following result shows that this approximation is statistically well behaved and clarifies the condition under which it can support correct classification.

Proposition B.1 (Consistency of Gaussian probing)**.** *Suppose that $\bar{H}(\Delta;\nu)$ has finite second moment for every fixed adaptor Δ. Then*

$$\widehat{\Psi}_m(\Delta) \xrightarrow[m\to\infty]{\text{a.s.}} \Psi(\Delta).$$

If, in addition, the specialization classes are separated by a positive margin in Ψ-space, then any margin-respecting plug-in classifier built on $\widehat{\Psi}_m(\Delta)$ is asymptotically correct.

Proof Sketch. The convergence $\widehat{\Psi}_m(\Delta) \to \Psi(\Delta)$ follows from the strong law of large numbers, since the probes are sampled independently from the Gaussian reference distribution. If the classes are separated in the population representation, then for sufficiently large m the empirical representation lies in the same decision region as the population representation, yielding asymptotically correct classification. □

C. Probe Classifier Implementation

Extraction For extracting the probes from the models we fix the internal randomness using a specific random seed across all LoRAs. This is important so that all of the representations we extracted start from the same initial Gaussian noise. Throughout our experiments in Section 4 we sample 1024 probes unless stated otherwise. In our experiments in Section 5 we sample 512 probes for SD 1.5 and SDXL 1.0. Due to the significantly larger size of FLUX.1-dev and computational restrictions when interacting with CSAM-specialized LoRAs for operational security, we only sample 4 probes. We do so over 30 timesteps. So each LoRA is mapped to a tensor in the shape of (N, T, D) where N is the number of probes, T is the number of timesteps, and D is the dimension of the representation we extract. We do so by simply implementing hooks in that capture the representations during forward passes up to the layers of interest. To prevent randomness due to the solver introducing noise, we use a deterministic solver for each architecture when extracting the probes. Specifically, for SD1.5 and SDXL 1.0, which are denoising diffusion implicit models (DDIM) that use a deterministic ODE instead of an SDE. For FLUX.1-dev, the default flow-matching scheduler with dynamic shift is deterministic.

We note that probe tensors are not guaranteed to be numerically identical across hardware / micro-batch configurations due to bf16 non-determinism. However, this is a non-issue given our theoretical intuition for the method because the noise from non-determinism is class-agnostic and the classifier is trained to distinguish distributional moments rather than exact values. We empirically validate this on all three architectures across classes by training linear classifiers to distinguish between our averaged probes on different subsets of the entire probe set. We find that across all three architectures, classes, and all layer choices, classifiers are unable to distinguish between averaged probes from different initializations (Figures 7, 8, 9).

Classifier For the classifier we split the dataset of probes based on the LoRAs into a 70%, 10%, 20% train, validation, and test train split. We train a linear classifier for 200 epochs, with batch size 32, and learning rate 1^{-3}. We take an argmax over the softmax probabilities to assign the class between SFW, NSFW, and CSAM.

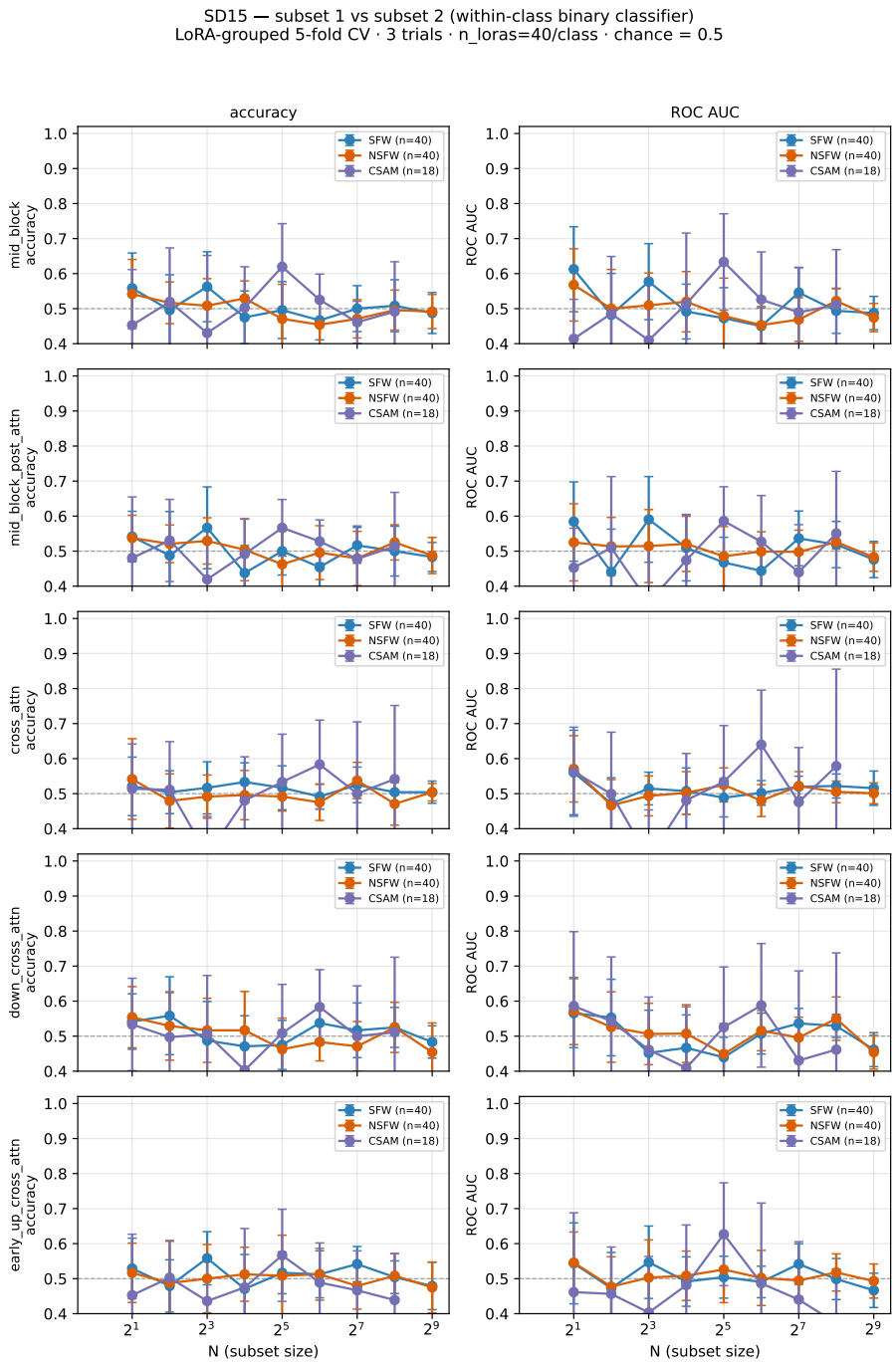


Figure 7. Linear classifier trained on random subsets of varying sizes from the total pool of probes for a sample of LoRAs from each class for SD1.5. Across all layers we extract the classifier is unable to distinguish the averaged probe across any subset size. Thus indicating that the same initialization does not need to be used across all LoRAs or classes.

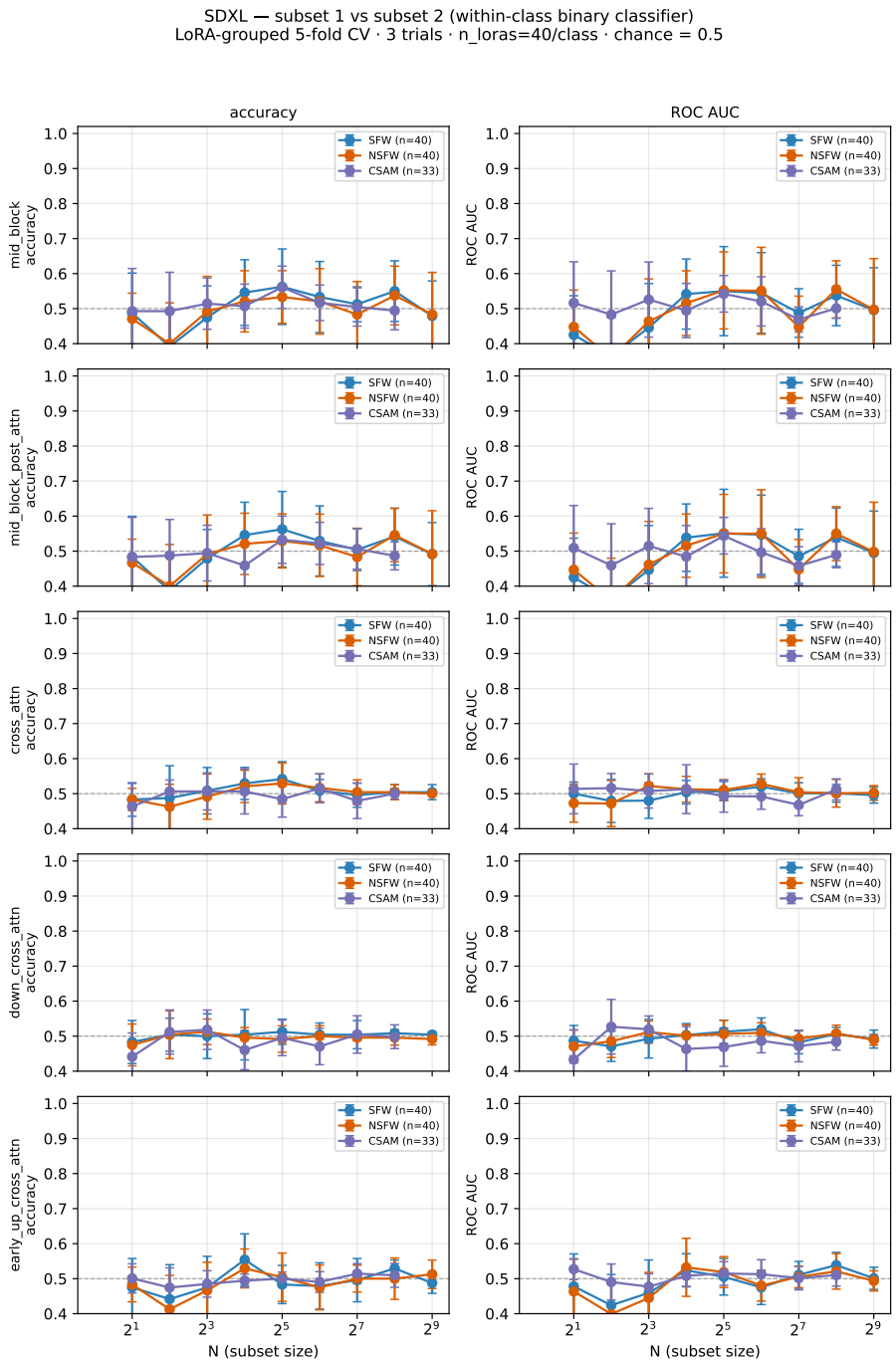


Figure 8. Linear classifier trained on random subsets of varying sizes from the total pool of probes for a sample of LoRAs from each class for SDXL 1.0. Across all layers we extract the classifier is unable to distinguish the averaged probe across any subset size. Thus indicating that the same initialization does not need to be used across all LoRAs or classes.

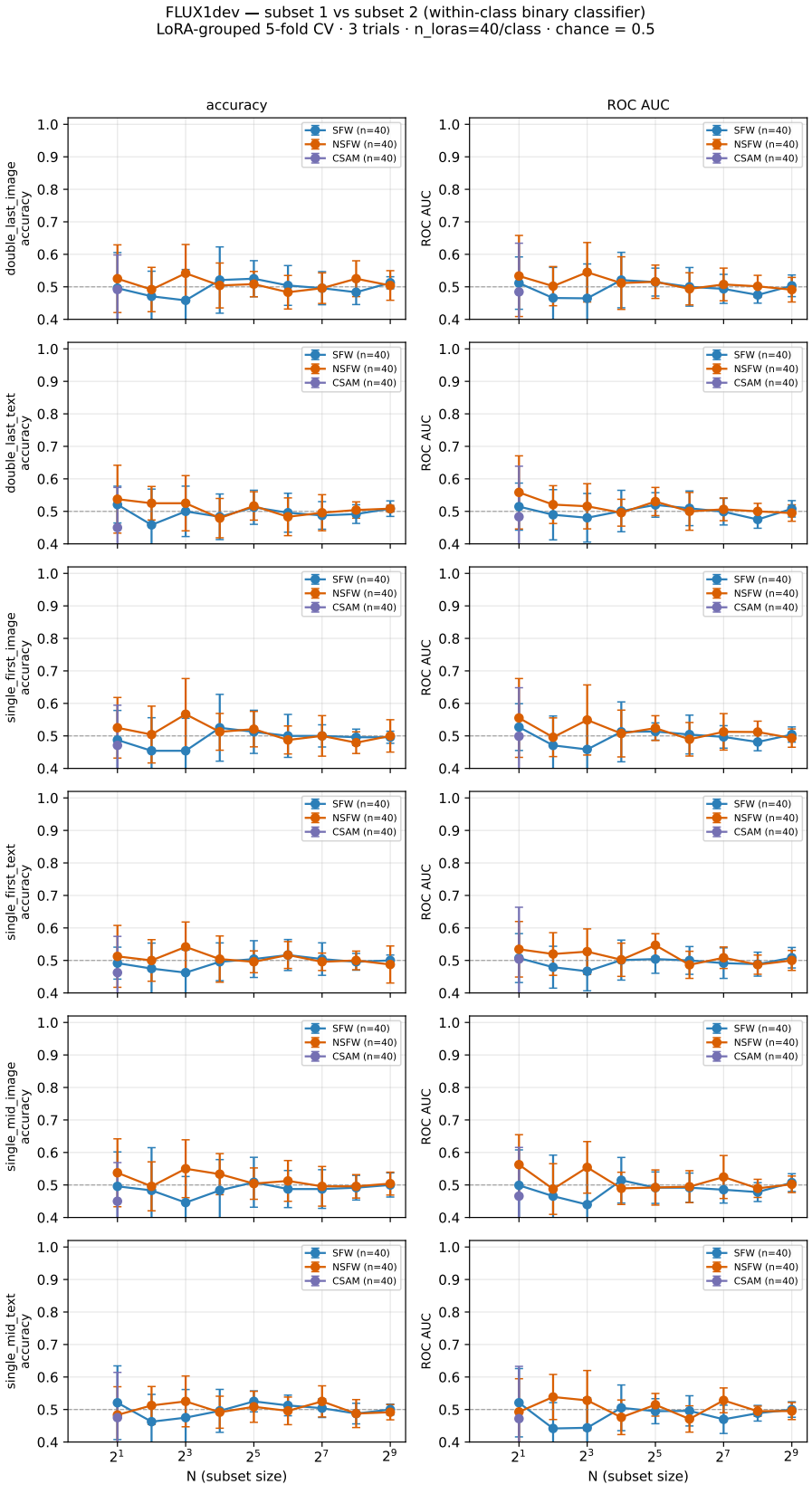


Figure 9. Linear classifier trained on random subsets of varying sizes from the total pool of probes for a sample of LoRAs from each class for FLUX.1-dev. Across all layers we extract the classifier is unable to distinguish the averaged probe across any subset size. Thus indicating that the same initialization does not need to be used across all LoRAs or classes.

D. Structured Study

We present additional results from our structured study of Gaussian probing. Primarily, the standard cross validation results, additional metrics for the leave dataset out results, and ablations on both the impact of using probes from different modules of the model, and ablations on the ensembling strategy for these different modules.

D.1. LoRA Training Details

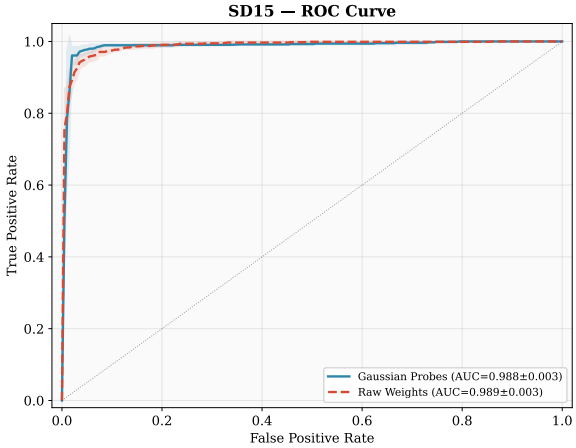
For Section 4 when creating the controlled study we trained our own LoRAs as described in the experimental section. For LoRA training we randomly picked the hyperparameters from the following lists: rank = $\{4, 8, 16, 32, 64, 128, 256\}$, learning rate = $\{1E-3, 1E-4, 1E-5\}$, module = $\{\text{text encoder}, \text{latent diffusion model}, \text{both}\}$, and alpha = $\{4, 8, 16, 32, 64, 128, 256\}$, number of finetuning steps = $\{1, 10, 100, 1000\}$, number of examples in finetuning dataset = $\{1, 5, 10, 25, 100, 250, 500, 1000\}$. We randomly assign a configuration of hyperparameters to each LoRA for training to prevent any confounds being exploited during prediction. We use both the diffusers library (von Platen et al., 2022) and the kohya library (bmaltais, 2025) to capture the common training setups in the wild.

D.2. Standard Cross Validation Results

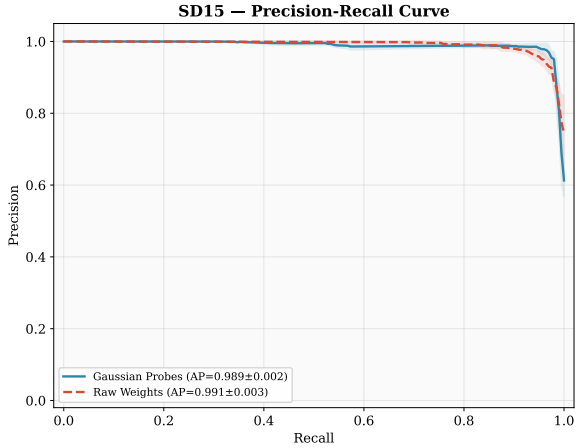
In this section we present the standard cross validation results for the controlled study. As mentioned in Section 4 we view these results as the ceiling of the possible performance since we eliminate many of the confounds that temper performance in the wild. These include differences in training hyperparameters between concepts, LoRAs being produced from different base checkpoints, different training libraries being used, a long tail of different ranks and layer configurations that are updated.

D.2.1. Stable Diffusion 1.5

We present the standard CV results across all metrics over 5 folds in Figures 10 and 11 for SD 1.5.



(a) ROC curve of both weight projection and gaussian probing methods in identifying SFW vs NSFW LoRAs for SD 1.5. Both methods perform well showing almost perfect classification. We attribute this to controlling many confounds that appear in the wild, making the task much easier.



(b) PR curve of both weight projection and gaussian probing methods in identifying SFW vs NSFW LoRAs for SD 1.5. Both methods perform well showing almost perfect classification. We attribute this to controlling many confounds that appear in the wild, making the task much easier.

Figure 10. Standard cross-validation results for SD 1.5 on controlled adult sexual content prediction task.

D.2.2. Stable Diffusion XL 1.0

We present the standard CV results across all metrics over 5 folds in Figures 12 and 13 for SDXL 1.0.

D.2.3. FLUX.1-dev

We present the standard CV results across all metrics over 5 folds in Figures 14 and 15 for FLUX.1-dev.

D.3. Ensemble and Module Probe Ablation

In this section, we present the impact of ensembling the representations across the different modules in the model (i.e. the text encoder, the latent diffusion model, or both) in Figure 16. We show that the ensemble is most important for SD 1.5 and there are diminishing returns as the size of the model increases (i.e. as we go from SDXL 1.0 to FLUX.1-dev).

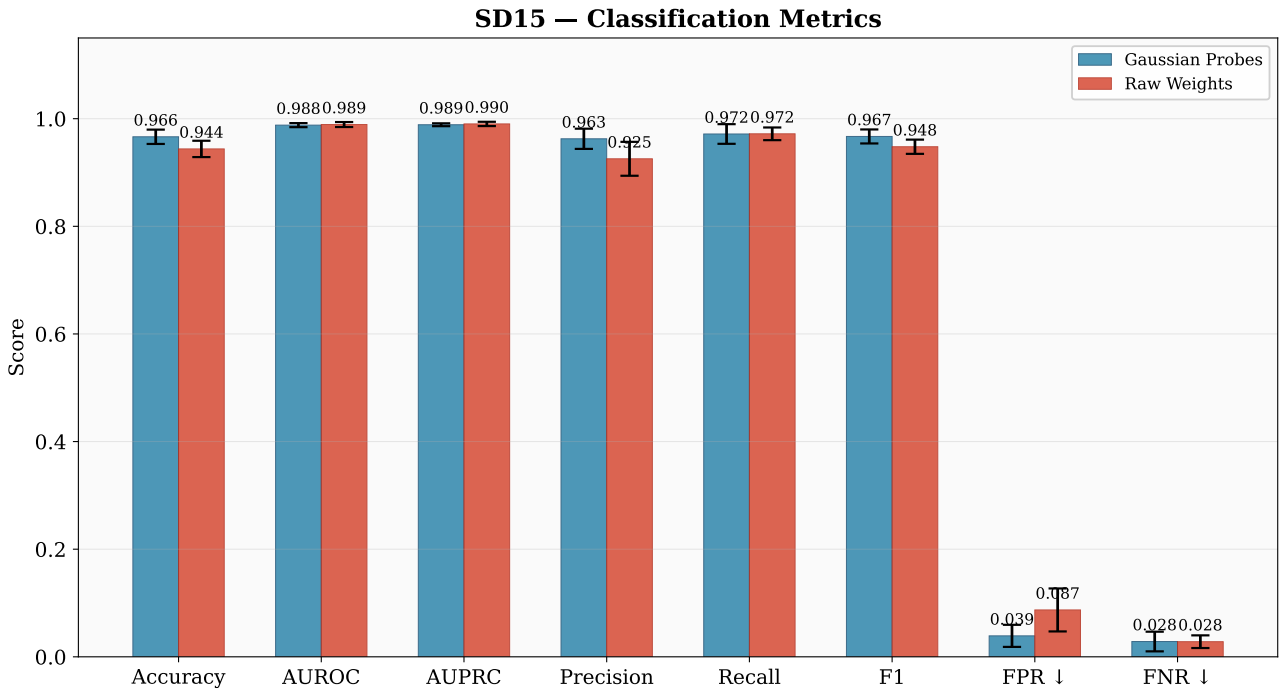
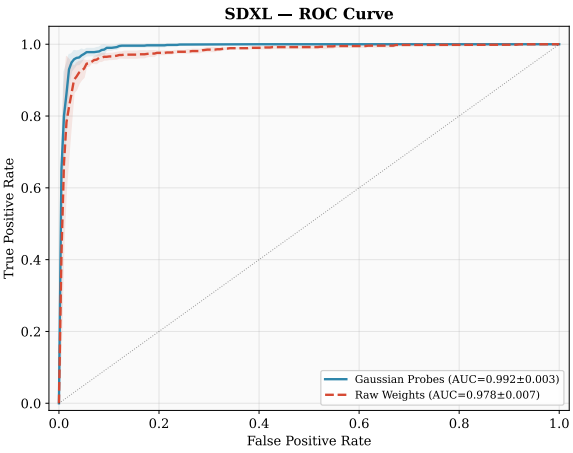
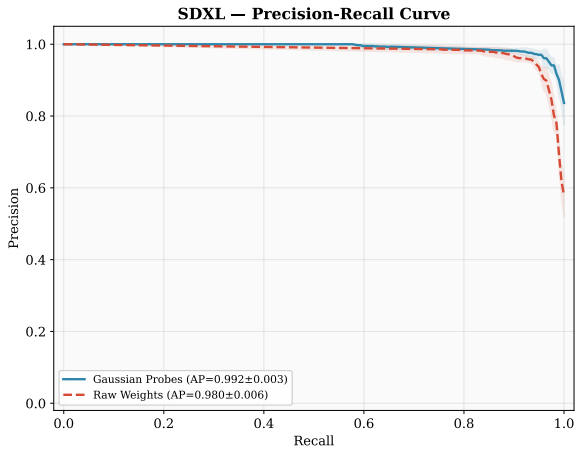


Figure 11. Additional metrics for standard cross-validation results for SD 1.5 on controlled adult sexual content prediction task.



(a) ROC curve of both weight projection and gaussian probing methods in identifying SFW vs NSFW LoRAs for SDXL 1.0. Both methods perform well showing almost perfect classification. We attribute this to controlling many confounds that appear in the wild, making the task much easier.



(b) PR curve of both weight projection and gaussian probing methods in identifying SFW vs NSFW LoRAs for SDXL 1.0. Both methods perform well showing almost perfect classification. We attribute this to controlling many confounds that appear in the wild, making the task much easier.

Figure 12. Standard cross-validation results for SDXL 1.0 on controlled adult sexual content prediction task.

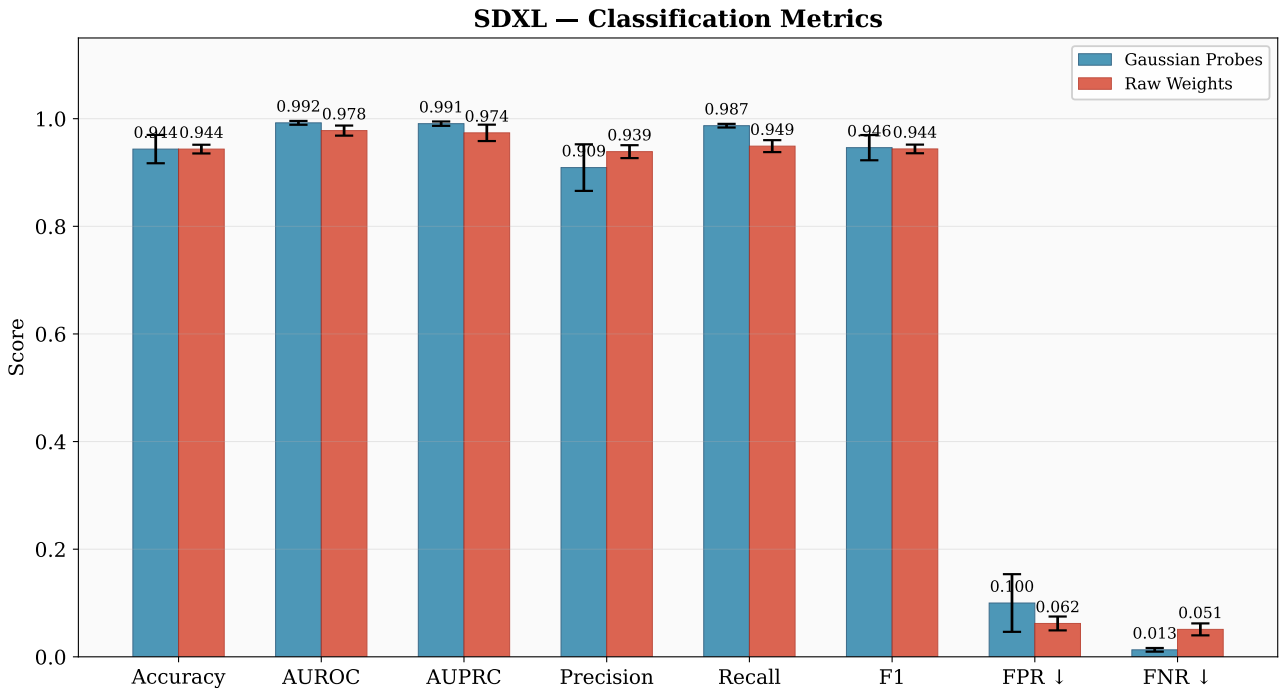
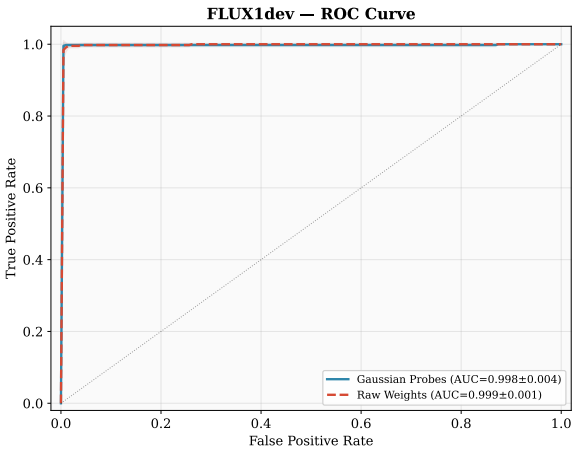
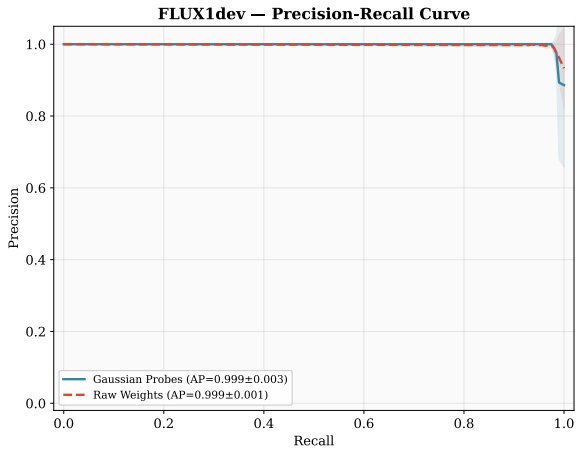


Figure 13. Additonal metrics for standard cross-validation results for SDXL 1.0 on controlled adult sexual content prediction task.



(a) ROC curve of both weight projection and gaussian probing methods in identifying SFW vs NSFW LoRAs for FLUX.1-dev. Both methods perform well showing almost perfect classification. We attribute this to controlling many confounds that appear in the wild, making the task much easier.



(b) PR curve of both weight projection and gaussian probing methods in identifying SFW vs NSFW LoRAs for FLUX.1-dev. Both methods perform well showing almost perfect classification. We attribute this to controlling many confounds that appear in the wild, making the task much easier.

Figure 14. Standard cross-validation results for SDXL FLUX.1-dev on controlled adult sexual content prediction task.

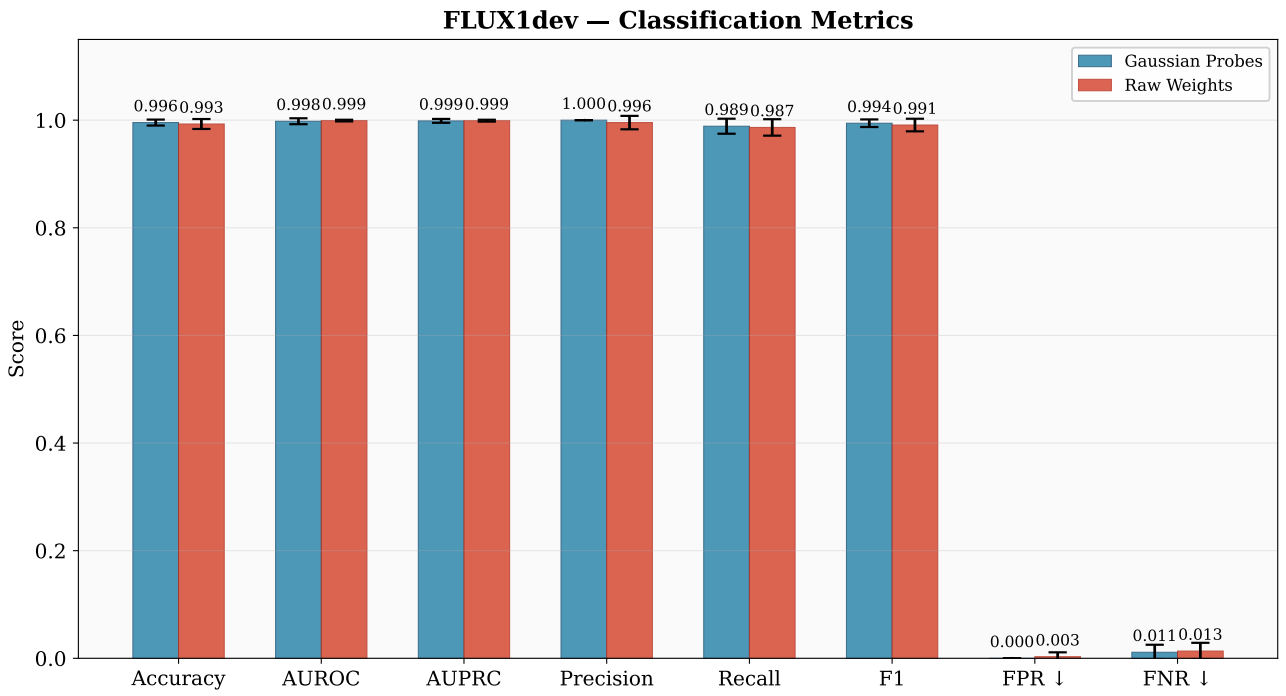


Figure 15. Additional metrics for standard cross-validation results for SDXL FLUX.1-dev on controlled adult sexual content prediction task.

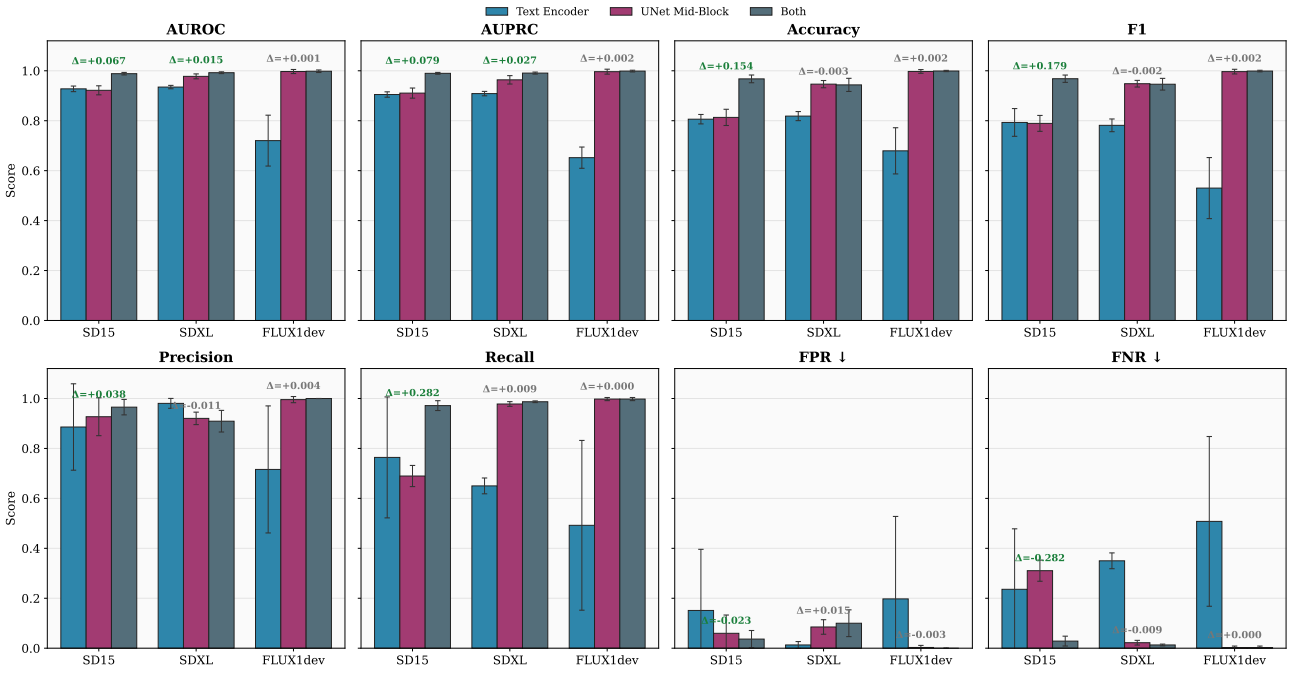


Figure 16. Module ablation for Gaussian probing across architectures. We compare probes extracted from the text encoder alone, the UNet mid-block alone, and both modules combined, across eight metrics. Δ values show the improvement from ensembling both modules over the best single module. Combining text-encoder and UNet probes is essential for SD 1.5 (AUROC Δ = +0.067, recall Δ = +0.282) and yields smaller but consistent gains on SDXL and FLUX.1-dev. The large SD 1.5 recall gain reflects finetunes that modify only the text encoder, UNet probes alone miss these entirely.

E. Detecting CSAM Specialization in the Wild Study

We present additional details from our in the wild study. First, we discuss how we curated our dataset of LoRAs and performed filtering to capture SFW LoRAs from NSFW LoRAs. Second we present additional results in terms of the standard cross validation results and ablations on both the impact of layer choice for SDXL 1.0 and FLUX.1-dev, and ablations on the ensembling strategy for these different layer choices.

E.1. Dataset Curation

We focused our in the wild analysis on CivitAI as our platform of choice given the ease with which we can download LoRAs programatically through the API. There exist two endpoints the civitai.green and civitai.com endpoints. We use these two endpoints to help us curate the SFW LoRAs vs. the NSFW LoRAs. The civitai.green endpoint is a SFW version of civitai.com. Thus, we randomly download 1000 LoRAs from this endpoint and then we use an NSFW classifier on the gallery images to determine if it is still NSFW specialized. We also look for keywords in the model name or description that might indicate it was specialized for NSFW content. For curating the NSFW LoRAs we develop a set of keywords related to adult sexual content and use these to find LoRAs which match these search terms through the civitai.com endpoint. We then run an NSFW classifier over the gallery images to ensure at least one image is NSFW since this acts as a good proxy for the intent of the LoRA training.

Design Choices In order to maximize the performance of Gaussian probing, for we needed to ensemble predictions across multiple layers. We show in Appendix E.3 that without doing so Gaussian probing performs worse. While not surprising given the scale of these models, it does point to layer choice being an important consideration for applying Gaussian probing to future architectures. For SD 1.5, in accordance with existing literature on the existence of the *h-space* (Kwon et al., 2022), we used the representation from after the mid-block in the U-Net as a starting point, combined with additional layers. Based on this insight, we started with a mid-block representation for all three models. For SDXL 1.0 we expanded to representations in the middle of the beginning block of the U-Net and the middle of the end block of the U-Net. For FLUX.1-dev we simply took one layer in the first third, second third, and last third of the transformer model for both the text and image inputs.

We also explore different ensembling strategies across our different model architectures to combine the classifiers trained on each individual layer's probes into a single prediction. The four strategies we try are all-concat, soft voting, weight soft-voting, and stacking. All-concat simply concatenates all the probes into one vector and trains a linear classifier on this concatentated vector. Soft voting averages the predicted class probabilities across all per-layer classifiers with equal weight, yielding the simplest and assumption-free combination. Weighted soft voting generalises this by learning a single non-negative weight per layer on a held-out validation split weights are parameterised through a softmax and optimised by Nelder-Mead to minimise the cross-entropy of the weighted-average probabilities, so keys that are individually more informative dominate the ensemble. Stacking instead trains a multinomial logistic-regression meta-learner on the validation set using the concatenated per-layer class-probability vectors as meta-features; this allows the meta-learner to capture interactions between keys rather than only re-scaling them. Overall the best strategy varies between architectures but the differences between them are marginal indicating that there isn't much optimization over which strategy to use needed (Appendix E.5).

E.2. Additional Metrics

In this section, we present the per-class values across all of our metrics for CSAM detection across all of our architectures in 17 averaged over 5-fold CV.

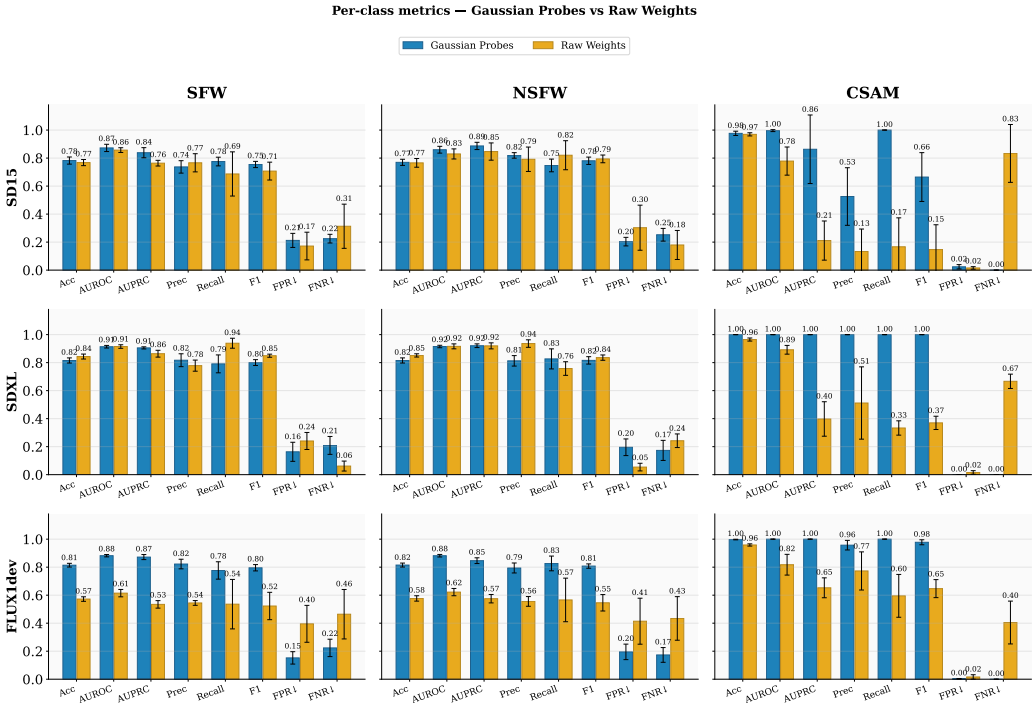


Figure 17. Per-class metrics across architectures (rows: SD 1.5, SDXL, FLUX.1-dev; columns: SFW, NSFW, CSAM). For each (architecture, class) pair we report all eight metrics for Gaussian probes and raw weights. Gaussian probing matches or exceeds raw weights on SFW and NSFW across all metrics, and decisively outperforms on CSAM. Raw weights collapse on the CSAM column due to the small CSAM sample size and high dimensionality.

E.3. Layer Choice Ablation

In this section we present our ablations over the individual classifiers trained on the probes from the different layers we picked for SD 1.5, SDXL 1.0 and FLUX.1-dev: Figures 18, 19, and 20.

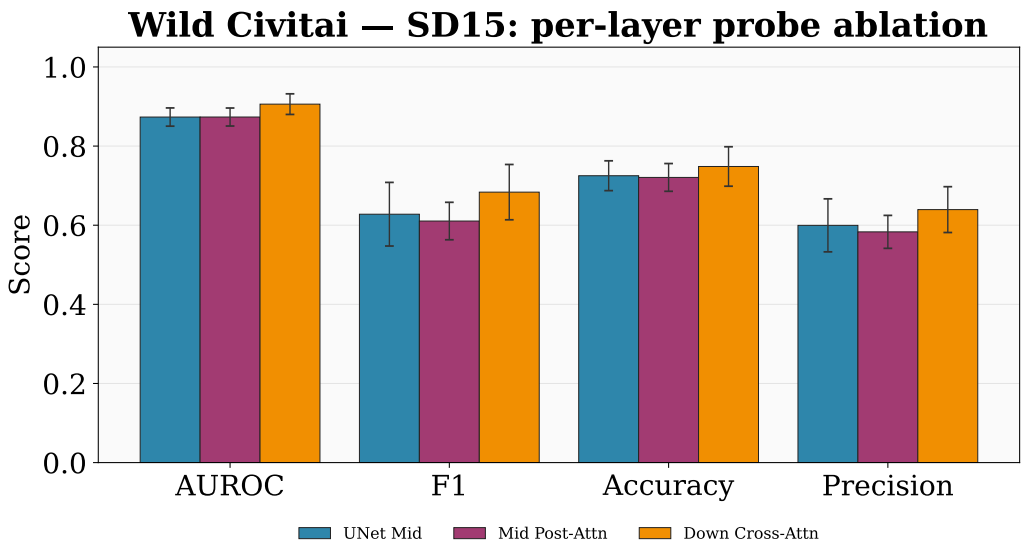


Figure 18. Per-layer probe performance on wild CivitAI SD 1.5 LoRAs. We compare Gaussian probes extracted from three candidate layers: UNet mid-block, mid post-attention, and down cross-attention. No single layer dominates across all metrics, motivating the ensemble approach.

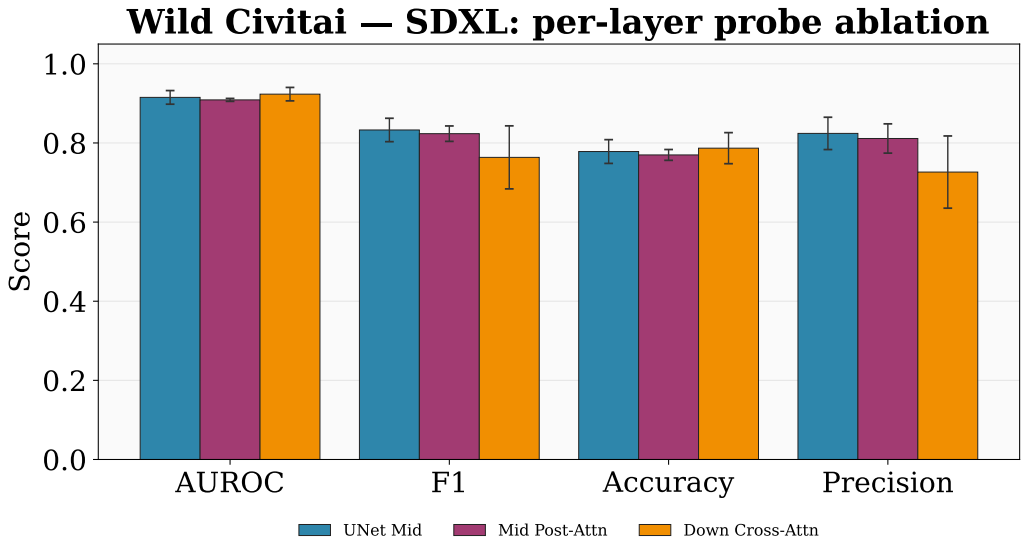


Figure 19. Per-layer probe performance on wild CivitAI SDXL 1.0 LoRAs. We compare Gaussian probes extracted from three candidate layers: UNet mid-block, mid post-attention, and down cross-attention. No single layer dominates across all metrics, motivating the ensemble approach.

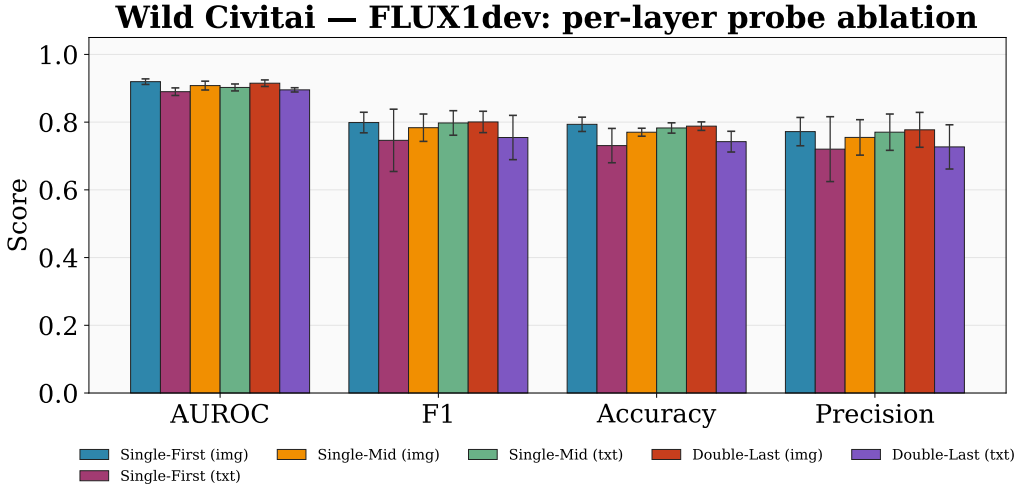


Figure 20. Per-layer probe performance on wild CivitAI FLUX.1-dev LoRAs. We compare Gaussian probes extracted from six candidate layers spanning the first, middle, and last thirds of the transformer, for both image and text streams. Performance is similar across layer choices, with no single layer clearly outperforming.

E.4. Error Analysis

In this section we provide the raw counts for the error analysis we present in Section 5 to complement the error rates and give a sense of the scale of the errors.

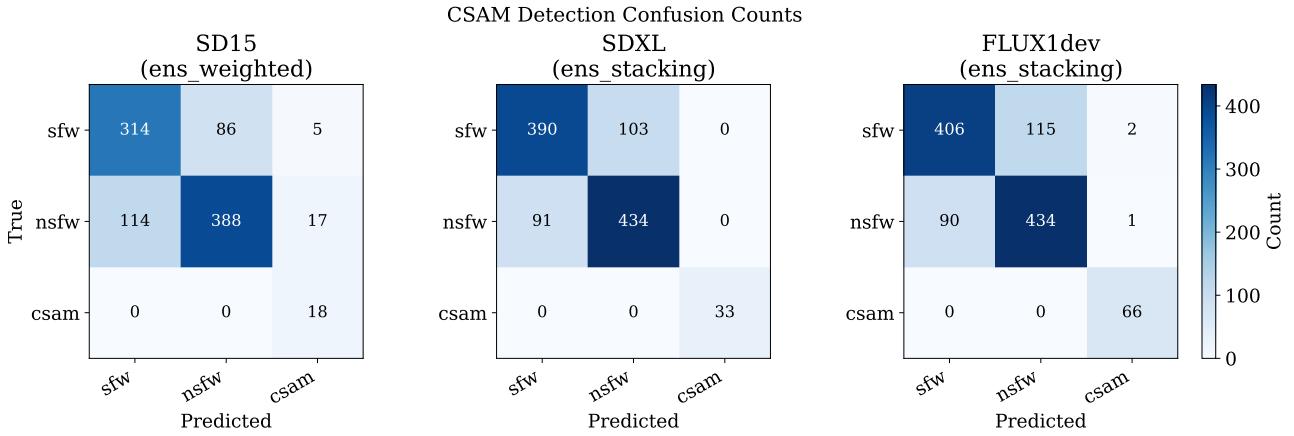


Figure 21. CSAM detection error counts by true class, across architectures. Gaussian probing achieves 100% recall on CSAM across SD 1.5, SDXL, and FLUX.1-dev (bottom row).

E.5. Ensembling Ablation

In this section we compare the best individual layer classifier to the ensembled classifier for SD 1.5, SDXL 1.0 and FLUX.1-dev (Figures 22, 23, and 24). We also explore the best ensembling strategy for each architecture ((Figures 25, 26, and 27).

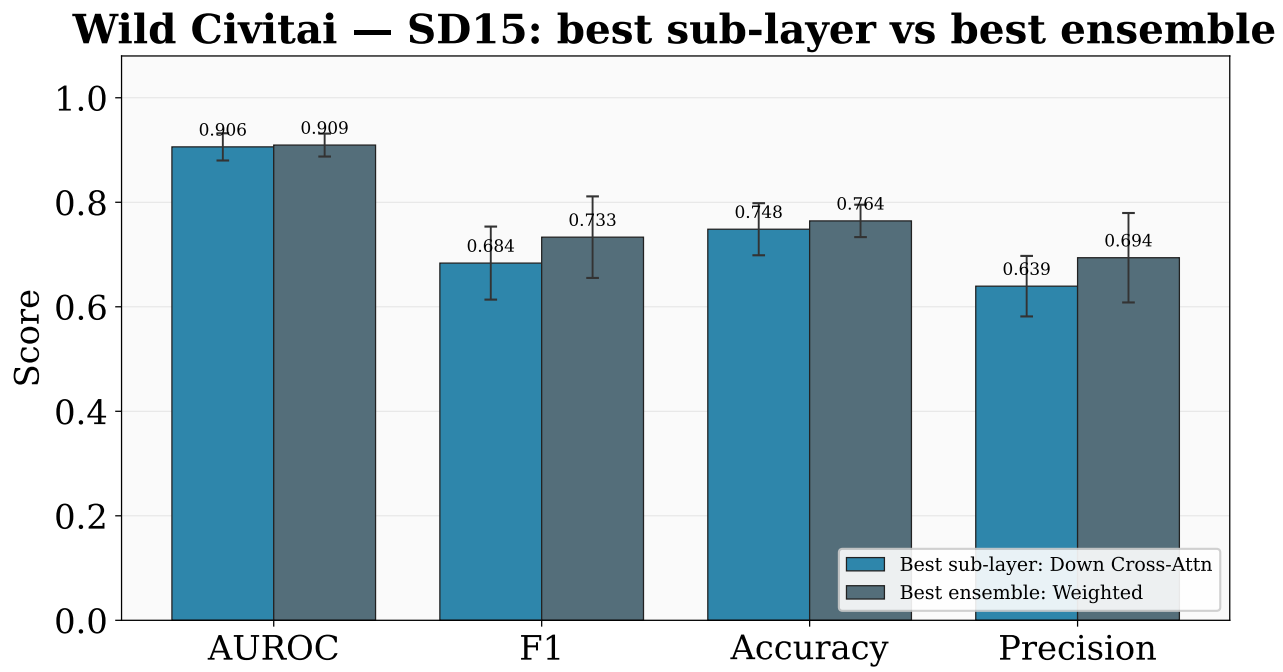


Figure 22. Best single-layer probe vs. best ensemble on wild CivitAI SD15. Ensembling probes across layers (soft vote) improves over the best single layer (down cross-attn) on all metrics.

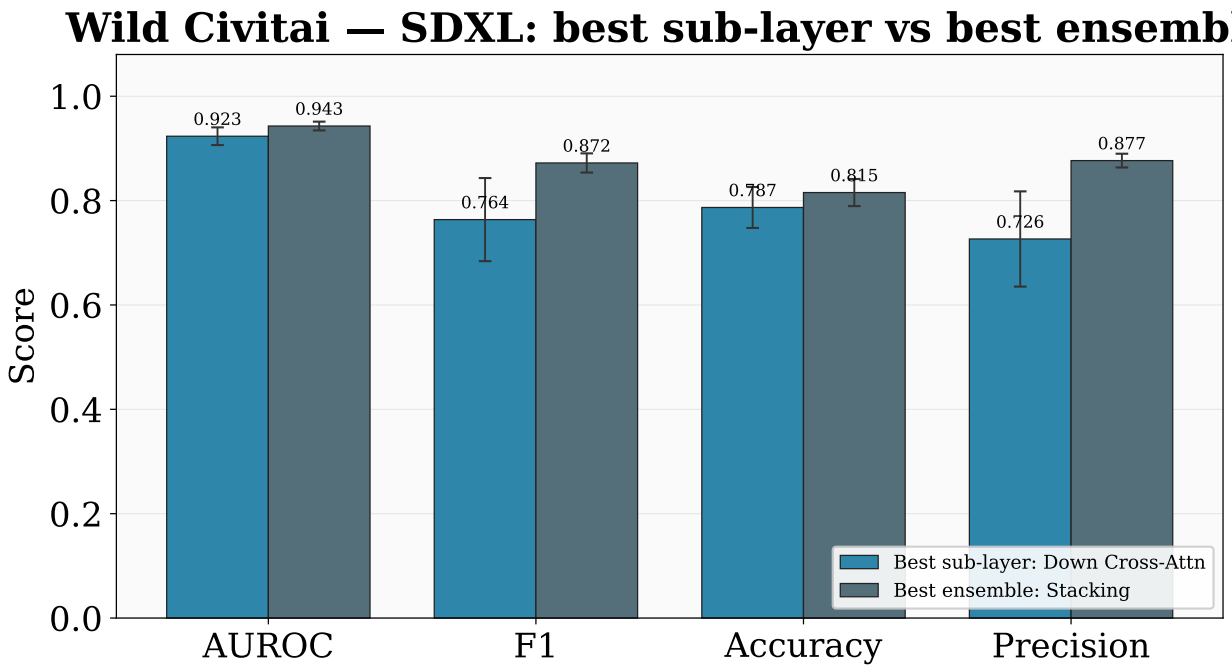


Figure 23. Best single-layer probe vs. best ensemble on wild CivitAI SDXL 1.0. Ensembling probes across layers (stacking) improves over the best single layer (down cross-attn) on all metrics.

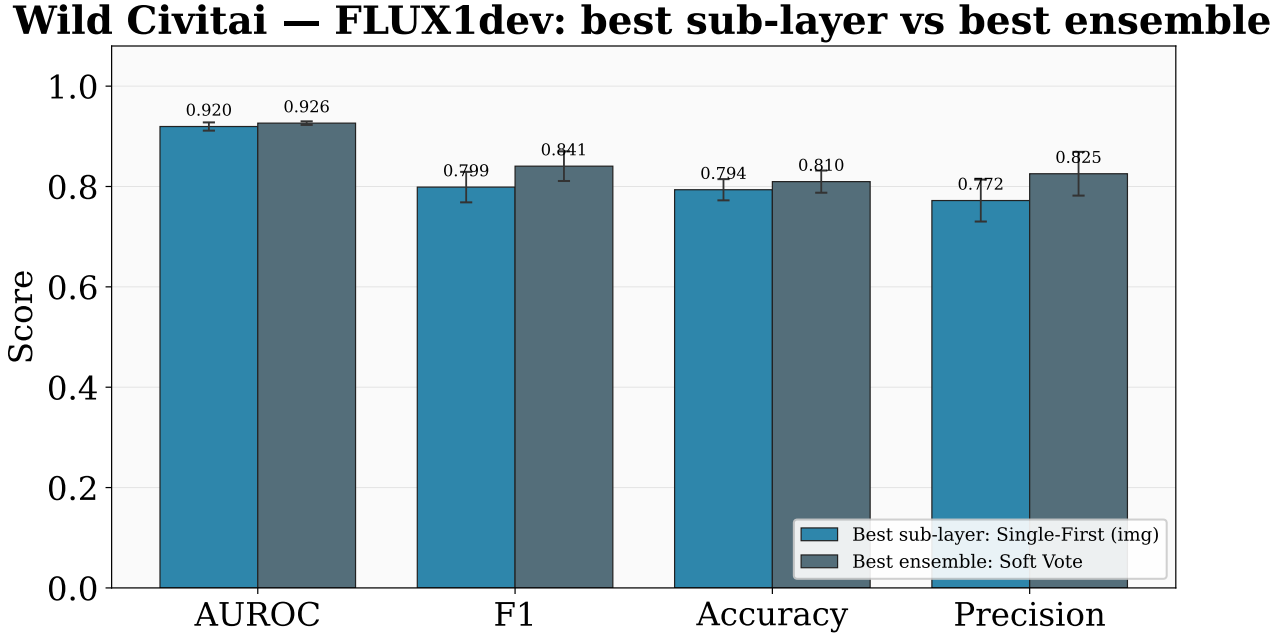


Figure 24. Best single-layer probe vs. best ensemble on wild CivitAI FLUX.1-dev. Ensembling probes across layers (soft vote) improves over the best single layer (single first image) on all metrics.

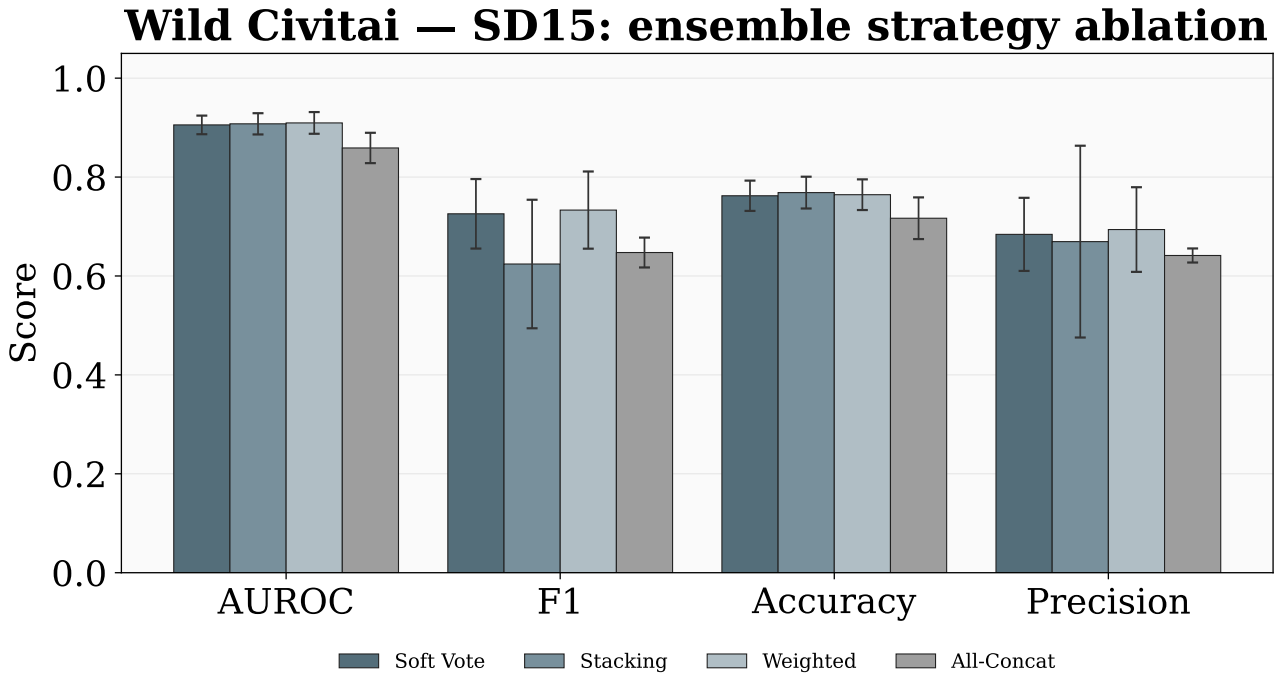


Figure 25. Comparing performance of different ensembling strategies on wild CivitAI SD 1.5. Weighted soft vote seems to provide the best performance across the metrics.

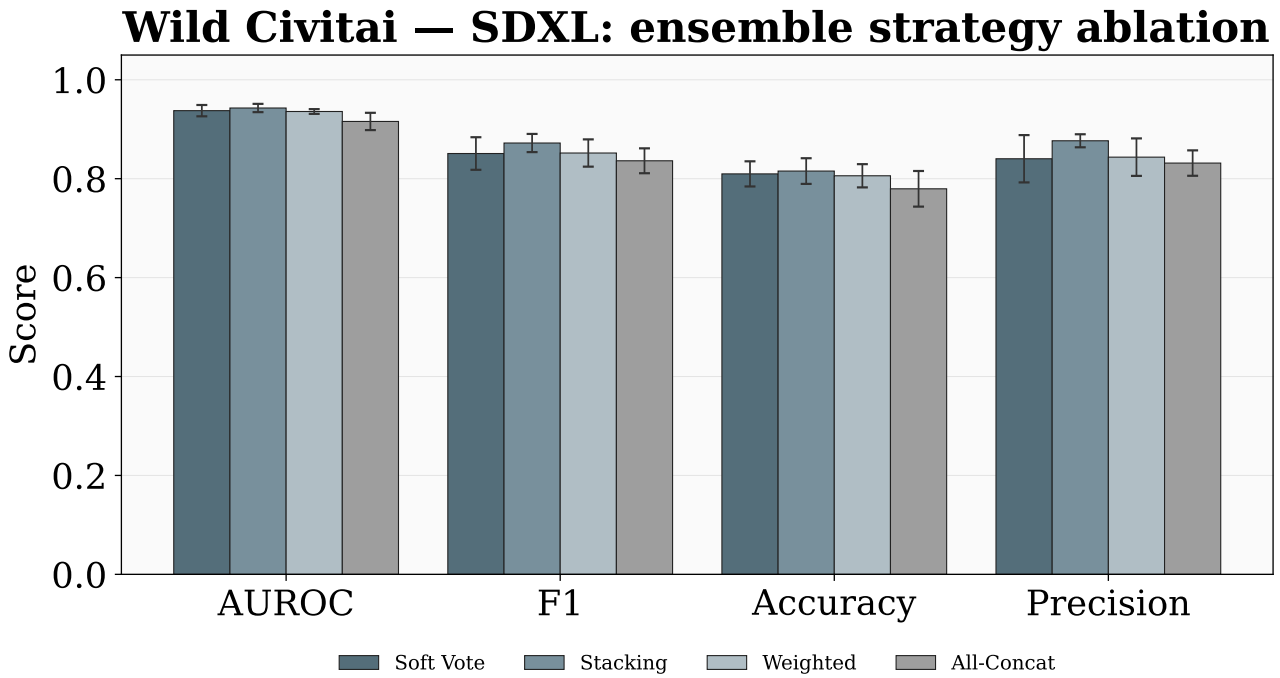


Figure 26. Comparing performance of different ensembling strategies on wild CivitAI SDXL 1.0. Stacking seems to provide the best performance across the metrics.

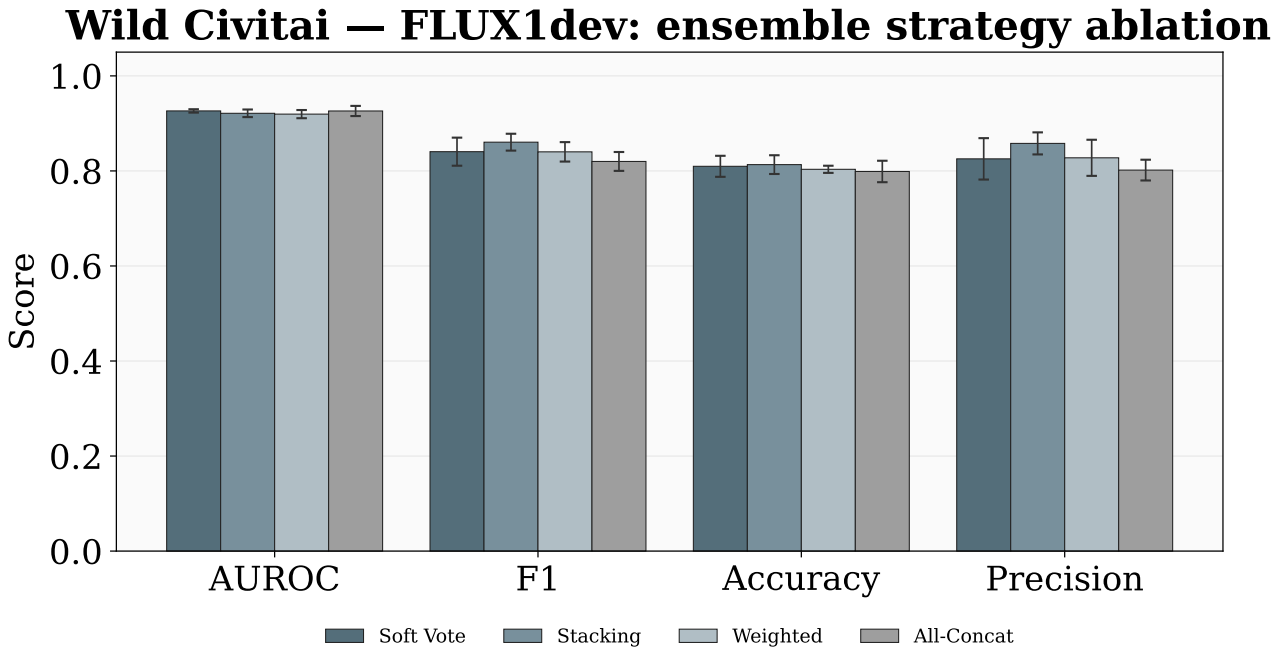


Figure 27. Comparing performance of different ensembling strategies on wild CivitAI FLUX.1-dev. Standard soft vote seems to provide the best performance across the metrics.