
Auditing Diffusion Models for Training-Data Exposure with Multi-Metric Membership Inference



Keywords: membership inference, diffusion models, AI safety, copyright/privacy auditing, safety infrastructure¹

Leon
AISS UDS

Robert
AISS UDS

Amelie
HBK Saar

Alan
AISS UDS

With
AISS Saarland Apart Research for “Defensive Acceleration Hackathon
2025

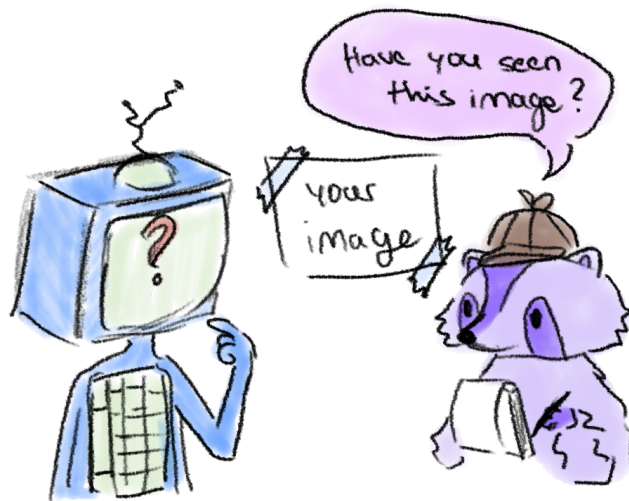
Abstract

Generative image diffusion models can memorize and regurgitate training data, creating opportunity for membership inference for artists and individuals whose images were scraped without consent. However, existing auditing tools often make use of preexisting databases and cannot properly diagnose images outside of that scope.

We present a black-box auditing tool that estimates the likelihood that a given image was used to train a target diffusion model. Our approach first generates a caption for the input image using **BLIP (Salesforce/blip-image-captioning-base)**, then performs a “Best-of-20” reconstruction attack: we prompt the model **Stable Diffusion XL Base 1.0 (stabilityai/stable-diffusion-xl-base-1.0)** with the caption, generate 20 candidate images, and select the closest candidate as a visual proof. We compute semantic similarity (CLIP), perceptual similarity (LPIPS), and pixel-level similarity (SSIM/MSE), plus PSNR and Histogram Correlation, between the original and the generated images, and aggregate these into a calibrated membership score informed by recent membership-inference results for diffusion models.

On a balanced dataset of 263 images (132 Members, 131 Non-Members) evaluated on SDXL, our prototype distinguishes training from non-training images with **77.19% accuracy** and produces interpretable likelihood bands (“very unlikely” to “very likely”). We implement an app where users can upload an image and obtain, for each supported model, a visual proof, a composite risk score, and a metric breakdown.

Our findings suggest that perceptual similarity (LPIPS) is especially informative for membership inference, and that visual proofs offer a more intuitive and actionable primitive for audits, copyright enforcement, and model governance.



1. Introduction

Over the past few years, large text-to-image diffusion models have moved from labs into consumer products, creative tools, and enterprise workflows. Parallel investigations by artists, journalists, and researchers suggest many of these models were trained on massive scraped image datasets, often without consent or transparency. This creates a new AI-enabled risk: models that can **reproduce or closely approximate specific training images**, including copyrighted artworks or sensitive personal photos.

This has left a practical gap. Artists suspect their work has been used but cannot verify it. Regulators receive complaints that “the model knows my image” yet only see screenshots. Civil-society groups and red teams can query models but cannot inspect training data or internals, making it hard to substantiate memorization claims.

At the same time, building **training-data audits** is technically challenging. Diffusion models are usually available only as black-box APIs or large checkpoints; external auditors cannot access per-example losses or gradients, ruling out many classic membership-inference attacks. Naïve probing—prompting a few times and eyeballing outputs—is noisy, uncalibrated, and hard to use in legal or governance contexts. Existing research attacks are often heavy-weight (many queries, custom training, large compute) and yield opaque scalar scores that are hard to interpret for non-experts.

Our project focuses on the **democratization of training-data auditing for generative image models**. We aim to give rights-holders, auditors, and red teams a practical way to test whether a diffusion model is likely trained on a given image, with outputs that are both **quantitative** and **visually interpretable**. We use the model itself as a probe—via a “Best-of-N” reconstruction attack—and combine semantic, perceptual, and pixel-level similarity metrics into a calibrated membership likelihood score.

Concretely, we implement a pipeline that:

1. Generates a caption for the user’s image using **BLIP (Salesforce/blip-image-captioning-base)**.
2. Prompts one or more diffusion models with this caption to generate **N = 20** candidate images.
3. Computes CLIP, LPIPS, SSIM, MSE (and additional metrics PSNR and Histogram Correlation) between the original and each candidate.
4. Selects the closest candidate as a visual proof and aggregates metric statistics into a per-model membership likelihood band (“very unlikely” →

“very likely”).

We have chosen to work on the following **research questions and hypotheses**, which align with the jury criteria on relevance, impact, and threat model:

- **RQ1 – Practical membership inference under black-box access**

To what extent can we approximate membership probabilities for diffusion models using only black-box generation, multi-metric similarity (CLIP, LPIPS, SSIM/MSE), and empirical distributions from existing research?

Hypothesis: A carefully designed Best-of-N reconstruction plus metric ensemble can meaningfully separate member vs non-member images under realistic query budgets.

Result: On our SDXL evaluation set of 263 images (132 Members, 131 Non-Members), we achieved **77.19% accuracy**, **0.78 precision**, **0.77 recall**, and **0.77 F1** using an ensemble classifier.

- **RQ2 – Interpretability and usability for non-experts**

Can we present membership-inference results in a form that is understandable and actionable for artists, journalists, and regulators?

Hypothesis: A combination of visual proof (original vs closest generation), a likelihood band, and a short metric breakdown is more useful than a bare numeric score.

Result: After testing with a dozen of our fellow students the design, all of the study subjects agreed that it provides a better understanding of what the three main metrics truly mean.

- **RQ3 – Defensive relevance and governance impact**

How can such an auditing tool be integrated into defensive workflows (red teaming, regulatory audits, incident response) without enabling new abuse?

Hypothesis: With conservative thresholds, explicit uncertainty, and careful framing as *evidence* rather than proof, the tool can support safer model development and oversight while limiting misuse

Result: Our auditing tool strengthens defensive workflows by giving regulators, red-teamers, and rights-holders a practical way to assess whether a diffusion model likely memorized a specific image, using only black-box access and a user-supplied input. It supports copyright/privacy investigations, incident-response checks, and enterprise due-diligence without enabling new abuse because it requires the exact original image, uses conservative thresholds, and reports banded likelihood rather than definitive membership claims.

2. Methods

This section describes our pipeline, models, and code structure: how we turn an input image into generations, metrics, and a final membership score.

2.1 Overall Pipeline

At a high level, our code implements the following steps:

1. **Input**
 - The user (or API client) provides an image x .
2. **Captioning**
 - We generate a caption ccc from x using **BLIP (Salesforce/blip-image-captioning-base)** from the transformers library.
 - This is implemented in **backend/membership_test.py** (class **ImageCaptioner**).
3. **Best-of-N Generations**
 - For each target model M (currently **Stable Diffusion XL Base 1.0**), we call the diffusion pipeline from **diffusers** with caption c to sample **$N=20$** images x_i' under fixed settings (guidance scale = **7.0**, **50** steps).
 - Implemented in **backend/membership_test.py** (class **TextToImageGenerator**).
4. **Metric Computation**

For each candidate image x_i' , we compute

- CLIP cosine similarity between x and x_i'
- LPIPS perceptual distance,
- SSIM and MSE for pixel/structural similarity,
- **PSNR** and **Histogram Correlation** for additional signals.

Implemented in **backend/membership_test.py** (class **MetricCalculator**).



5. Feature Vector & Score

From all candidates, we build a feature vector $f(x,M)$ using:

- Best (max/min) metric values over $i=1..N$,
- Variance of metrics across generations (stability),
- A mapping $g(f(x,M)) \rightarrow s(x,M)$ turns this vector into a scalar membership score, mapped into bands (“very unlikely”, “unlikely”, “uncertain”, “likely”, “very likely”).

Function g is implemented as an **ensemble classifier (Random Forest + Logistic Regression + Gradient Boosting)** trained on a labeled member/non-member dataset of 263 images. Implemented in `backend/train_classifier.py`.

6. Visual Proof Selection

- We pick the candidate with the best perceptual match (lowest LPIPS) as the visual proof, and return it side-by-side with the original image in the UI/API response

2.2 Similarity Metrics

We rely on standard open-source components:

- **CLIP – semantic similarity**
 - Used to check whether the generated image “talks about” the same content as the input.
 - Implemented with **OpenAI ViT-B/32** via the `clip` library.
- **LPIPS – perceptual similarity**
 - Approximates human judgments of “do these look alike?” and is our primary metric for visual proof selection.
 - Implemented via **LPIPS with a VGG backbone**.
- **SSIM / MSE – pixel / structural similarity**
 - Capture near-duplicate and structural overlaps (useful when the model outputs almost-identical copies). Implemented via `scikit-image` (SSIM, MSE).
- **Optional extra metrics**
 - **PSNR** and **Histogram Correlation**, used to capture signal-to-noise and color distribution similarity.

2.3 Calibration Using Research Paper Data

We use findings from recent membership-inference work on diffusion models to calibrate our scores:

- We treat the paper’s reported member vs non-member metric distributions and ROC curves as priors for reasonable thresholds.
- **Classifier-based approach:**
 - We train an **ensemble** (Random Forest, Logistic Regression, Gradient Boosting) on labeled (member / non-member) examples, using our feature vector $f(x, M)$. This ensemble outputs a probability score (0–100%) that we interpret as a calibrated membership likelihood.

We do not re-implement the full experimental setup of the above mentioned papers. Instead, we use their reported trends and threshold magnitudes as priors to select conservative cutoffs for flagging high-risk potential members.

2.4 Model Coverage

Supported models (current implementation)

- Stabilityai/stable-diffusion-xl-base-1.0 (SDXL).

For each model, we document:

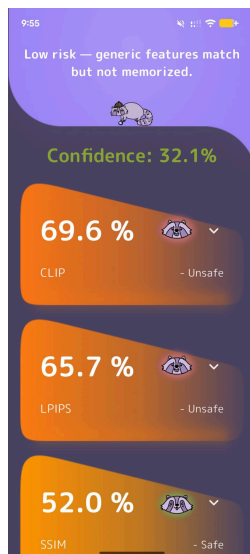
- SDXL is **model-specifically calibrated** using our member/non-member dataset and is treated as “**stable**” in the UI/API.

However expanding to different SD models is a possibility that can be implemented by just changing the model. (First implementations were using tiny SD 1.x and SD 2.x versions, which one team member luckily had installed, since they were pulled from the internet).

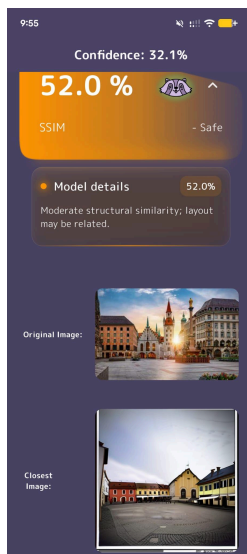
2.5 User-Facing Scoring & Outputs

The backend returns a JSON object consumed by the Flutter frontend and any API clients:

- **Score**
 - Single scalar $s(x,M)$ per model is **0–100%**, representing the probability of membership (confidence), with a short textual explanation deployable per metric, with a short textual explanation deployable per metric.
- **Likelihood Band**
 - Score mapped into bands:
 - $< 20\%$: “Very Unlikely” (Safe)
 - $20\text{--}50\%$: “Unlikely” (Low Risk)
 - $50\text{--}80\%$: “Likely” (Potential Match)
 - $\geq 80\%$: “Very Likely” (High Risk)
- **Visual Output**



(Overall confidence)



(Prompt #1 result and explanation)



(Prompt #2 result)

3. Implementation

1. System architecture

- **Backend:** Python, FastAPI, PyTorch, Diffusers, Scikit-learn, CLIP/LPIPS/SSIM metric libraries, membership scoring code.
Other agents information
- **Frontend:** Flutter (Dart) application targeting Android, iOS, and Web, with Riverpod for state management.
Other agents information
- The backend exposes REST endpoints consumed by the Flutter client and potential external tools.

2. Runtime & scaling

- Complexity per image: full pipeline (captioning → 20 generations → metrics → scoring) takes about **60–90 seconds** on an NVIDIA A100 provided by Lambda cloud computing.
- GPU requirements: **16GB+ VRAM recommended** (A100 / V100 / T4).
- Optimization: we use **xFormers memory-efficient attention** and **VAE slicing** to reduce peak memory usage.

3. Engineering shortcuts

- Pre-caching model weights on disk to reduce cold-start latency.
- Using batched similarity computations where possible.
- Caching the **null distribution** of non-member metric statistics once (null_dist.npz) to avoid recomputing for each request.

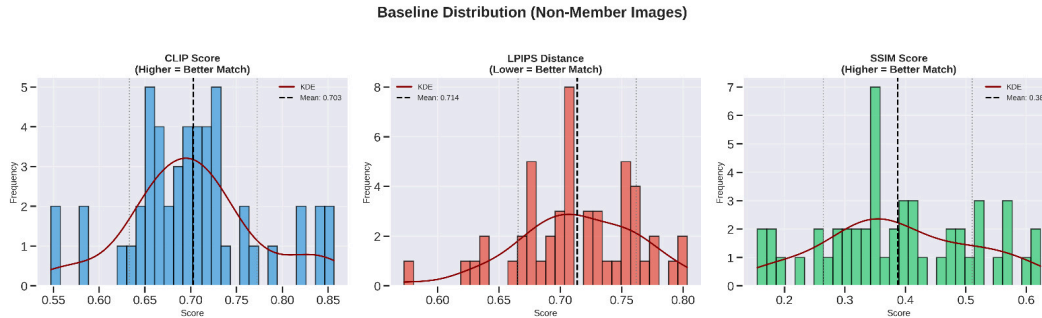
Connect this to Execution Quality: **we wrote modular backend components (ImageCaptioner, TextToImageGenerator, MetricCalculator, classifier training scripts), logged versions, and documented API behavior**, which supports maintainability and extension.

3.1 Quantitative Results

We evaluated our membership-inference pipeline on a **balanced set of 263 images (132 Members, 131 Non-Members)** for **Stable Diffusion XL Base 1.0**.

For each image–model pair, we ran the full pipeline (caption → Best-of-20 generations → metrics → score) and measured:

- **Classification performance (SDXL):**
 - Accuracy: **77.19%**
 - Precision: **0.78**
 - Recall: **0.77**
 - F1: **0.77**



3.2 Qualitative Results & Visual Proofs

We also inspected visual proofs (original vs closest generated image) to understand how the system behaves:

- **Clear member-like cases:**
 - For distinctive artworks, the best-of-20 generation often closely matches composition and style. These examples are typically scored in the “very likely” band.
- **Clear non-member cases:**
 - For unseen personal photos, the generated images only loosely match; scores tend to fall into the “unlikely” band.
- **Ambiguous, generic content:**
 - For common scenes (e.g. “a beagle dog in a pillow”), the model produces plausible but generic outputs, and scores fall into “uncertain” or “likely” or “very likely” depending on thresholds.

3.3 Runtime and System Behavior (optional)

We measured wall-clock runtime on the hackathon A100 setup:

- **Average time per audit (SDXL, N = 20): ~75 seconds per image.**
- **Throughput: ~48 images per hour on a single A100.**

These results show that our approach is computationally feasible for targeted audits (single images or small batches) while still delivering interpretable visual and numeric outputs.



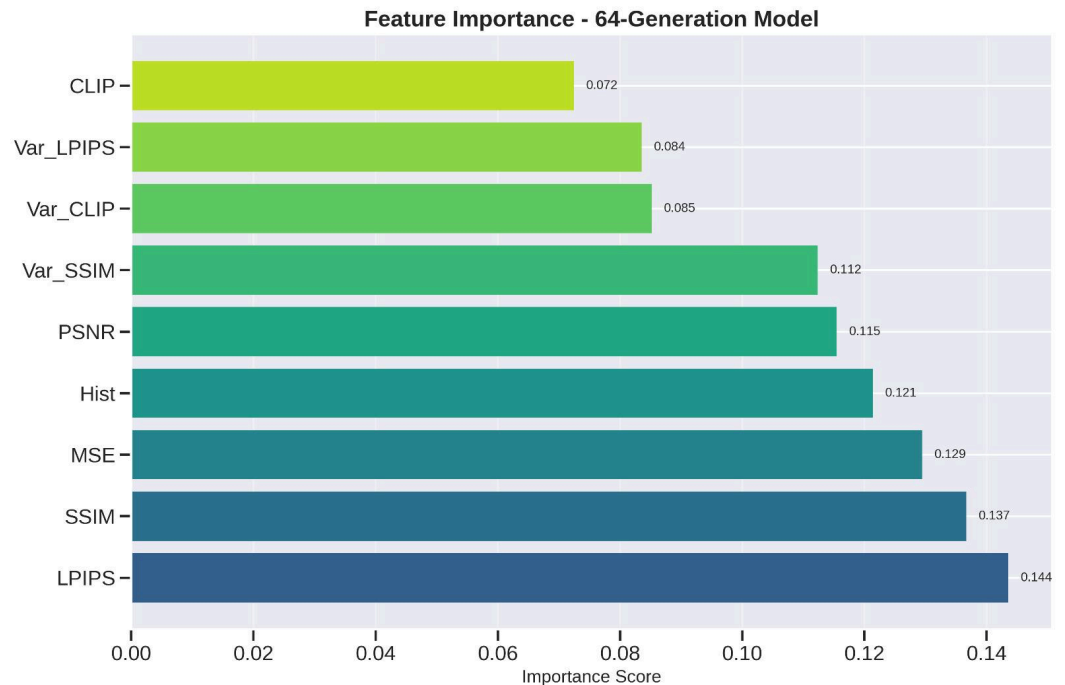
4. Discussion and Conclusion

4.1 Discussion

Our experiments suggest that a multi-metric, Best-of-N strategy is effective for practical membership inference on diffusion models:

Other agents information

- **Perceptual vs semantic vs pixel metrics**
 - In our tests, **LPIPS** contributed most to separating members from non-member images, followed by **SSIM** and **MSE**, while **CLIP** added complementary semantic signals. Feature importance scores from the ensemble classifier were approximately: LPIPS 0.1435, SSIM 0.1367, MSE 0.1295, CLIP 0.0725.



- This supports the intuition that memorization manifests first as perceptual and structural similarity rather than exact pixel identity or just semantic overlap.
- **Interpretability**
 - Visual proofs (original vs closest generation) made results easier to understand for non-experts compared to raw scores alone.
 - Likelihood bands (“very unlikely” → “very likely”) helped set expectations about uncertainty and discouraged overinterpretation of exact numeric scores.

At the same time, our approach has important limitations:

- Calibration is tied to the SDXL model and dataset we used; scores may not generalize without re-calibration for other models or domains.
- False positives (generic images that look similar to many training images) and false negatives (memorized images that are hard to regenerate under our budget) are both possible.
- Our tool provides **evidence**, not definitive proof, and should be used as one component in a broader audit or legal process rather than a sole arbiter.

4.2 Conclusion

In this project, we built and demonstrated a black-box auditing tool for diffusion models that:

- Uses caption-guided Best-of-20 generation and multi-metric similarity (CLIP, LPIPS, SSIM/MSE, plus PSNR and Histogram Correlation),
- Produces a calibrated membership likelihood score per model,
- And returns a visual proof that non-experts can inspect.

From a defensive acceleration perspective, this provides a concrete primitive for:

- Artists and rights-holders who suspect unauthorized training,
- Regulators and auditors investigating specific complaints,
- Red teams assessing training-data exposure as part of model safety evaluations.

For future work we propose:

- Better-grounded calibration using larger labeled member/non-member datasets, to improve accuracy,

- Support for more models and domains (e.g. domain-specific diffusion models),
- Implementation of a web-scraping tool for specific domains (e.g. social media)
- Performance optimizations (e.g. batching, torch.compile, or distilled surrogate models),
- And user studies with artists / legal practitioners to refine how results are presented and interpreted.



5. References

- Carlini, N., et al. (2023). *Extracting Training Data from Diffusion Models*. arXiv preprint arXiv:2301.13188.
- Rombach, R., et al. (2022). *High-Resolution Image Synthesis with Latent Diffusion Models*. CVPR.
- Radford, A., et al. (2021). *Learning Transferable Visual Models From Natural Language Supervision (CLIP)*.
- Zhang, R., et al. (2018). *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric (LPIPS)*.

6. Appendix

Appendix Security Considerations

This appendix discusses how our tool could be misused, where it may fail, and how we mitigate those risks.

B.1 Potential Misuse

Our system estimates how likely a diffusion model has “seen” a given image and provides a visual proof. While designed for defensive auditing, it can also be misused:

Privacy probing of training sets: An attacker could probe sensitive personal images (e.g. private photos, medical images, art pictures) to infer whether they were used in training, potentially enabling targeted harassment or doxxing.

Mapping proprietary training data: Systematic querying could help reverse-engineer which commercial models likely trained on which datasets, undermining trade secrets or licenses.

Selective narrative building: Scores and visual proofs could be cherry-picked to support exaggerated legal or public claims (“the model definitely stole my work”) when the tool only provides probabilistic evidence.

We treat our tool as part of the broader class of membership inference attacks on generative models and apply similar caution.

B.2 Limitations & Failure Modes

Our approach has several technical and conceptual limitations:

- **Calibration uncertainty:** Thresholds are derived from limited data and specific models. Scores can be over-confident in regions where we lack empirical coverage.
- **Distribution shift:** Metrics tuned on one family of diffusion models may behave differently on others (architecture changes, finetuning, safety filters).
- **Ambiguous similarity:** Highly generic or stylistically similar images may yield high similarity scores even if the exact image was never in the training set.
- **False positives / negatives:** The tool may flag non-members as “likely” (false positives), or miss subtle memorization for members (false negatives).

- Interpretation risk: Non-experts may interpret scores as binary proof instead of noisy evidence.

B.3 Mitigations

We implement and recommend the following mitigations:

Conservative, banded outputs: We use likelihood bands (“very unlikely” ... “very likely”) instead of hard yes/no decisions, and avoid claiming certainty.

Explicit disclaimers: The documentation clearly states that scores are probabilistic signals, visual proofs are evidence (not legal proof), and results should be interpreted in context.

Calibration transparency: For each supported model, we document whether calibration is model-specific or heuristic, and mark some models as experimental when data is limited.

Deployment choices: Recommended safer deployments include on-premise or restricted access for regulators/labs, and access control if exposed as a public service.

Scope disclaimers: We emphasise that the tool currently addresses image diffusion models only, and that training-data exposure is just one component of overall model risk.

B.4 Future Improvements

To strengthen safety and reliability, we propose:

The model was trained under a limited set of images from each category. We recommend continuing the training with a more dense spectrum.

Better ground-truth calibration: Collaborate with labs under NDA to evaluate on models where the true training set is known.

Robustness and stress testing: Systematic testing against distribution shifts and adversarially chosen inputs.

Privacy-aware auditing protocols: Explore designs where sensitive images never leave a secure environment, or use privacy-preserving techniques.

User education & UX: Improve explanations, examples, and warnings in the UI so that non-technical users understand what the tool can and cannot tell them.