Post-Training Strains Diffusion Models: A Stability Analysis of Distillation, Robustness, and Unlearning

Anirudh Bharadwaj
University of Pennsylvania
anirudh2@seas.upenn.edu

Joy Liu
University of Pennsylvania
joyjliu@seas.upenn.edu

Angelina Zheng
University of Pennsylvania
a7zheng@seas.upenn.edu

Abstract

Diffusion models are increasingly modified after pretraining for deployment constraints, robustness requirements, and data-governance objectives, yet the stability of their learned generative behavior under such interventions remains poorly understood. We present a controlled empirical study of DDPM stability along three common post-training axes: (i) progressive distillation to reduce sampling steps, (ii) robustness fine-tuning under realistic, model-independent corruptions, and (iii) targeted unlearning to remove designated subsets of training data. Collectively, these interventions probe complementary aspects of learned generative structure, including trajectory compressibility, robustness to distributional perturbation, and the localization of memorized information. We compare low-resolution (CIFAR-10, 32×32*) and high-resolution (CelebA-HQ,* 256×256*) settings and evaluate complementary notions of stability using Fréchet Inception Distance (FID), Inception Score (IS), and negative log-likelihood (NLL) computed via the DDPM variational bound. Across interventions, we observe clear dataset-dependent trade-offs. Distillation largely preserves fidelity on CIFAR-10 but can collapse sample quality on CelebA-HQ when the step budget is aggressively reduced. Robustness fine-tuning improves tolerance to distributional shift but tends to erode clean-sample quality as training continues, while likelihood- and perceptual-based metrics can diverge substantially under adaptation. Targeted unlearning reliably increases forget-set NLL, indicating reduced likelihood assigned to the forgotten subset, but can also substantially degrade global sample quality, especially at higher resolution. Together, these results suggest that post-training interventions expose distinct failure modes in learned generative representations and highlight the need for evaluation frameworks that distinguish robustness, memorization, and generative generalization under model adaptation.*

1. Introduction

Diffusion models have rapidly become a dominant paradigm in generative modeling, producing state-of-the-art samples across images, video, and audio. At the same time, a central question in modern deep generative modeling remains unresolved: *what do diffusion models actually learn, and which aspects of learned generative structure remain stable under post-training adaptation?* As diffusion models are increasingly adapted, compressed, or modified for real-world deployment, understanding whether these interventions preserve robust generative structure, disrupt probabilistic consistency, or expose brittle memorization-like behavior has become increasingly important. Despite growing practical interest in robustness training, model editing, and distillation techniques, the effects of such transformations on the underlying generative process remain poorly understood. These questions are especially relevant for computer vision because diffusion models are increasingly used not only as image generators, but also as components in vision-facing pipelines such as data augmentation, robustness evaluation, restoration, purification, and model editing. In these settings, preserving visual fidelity alone is insufficient: post-training interventions may also alter semantic diversity, likelihood calibration, and the stability of visually meaningful structure.

In this work, we provide a systematic empirical study of diffusion model stability along three widely used intervention axes:

1. **Progressive distillation**: reducing the number of denoising steps to accelerate sampling, which may alter the learned noise prior and the geometry of the generative trajectory.
2. **Robustness fine-tuning**: continuing training under structured, model-independent input corruptions to improve stability under distributional shift, potentially affecting likelihood calibration and sample quality.
3. **Targeted unlearning**: selectively removing subsets of training data, an increasingly important requirement for data privacy, content moderation, and legal compliance.

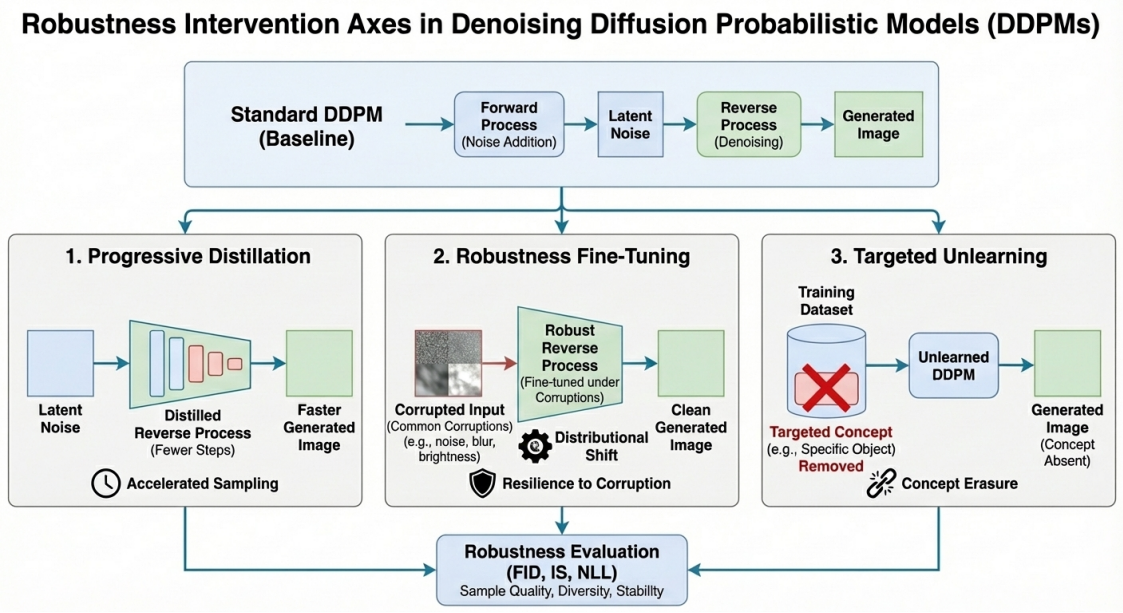

Figure 1. Overview of our evaluation pipeline. Starting from a pretrained DDPM, we apply three classes of post-training intervention—(i) progressive distillation, (ii) robustness fine-tuning, and (iii) targeted unlearning—and evaluate their effects using FID, IS, and NLL.

Collectively, these interventions probe complementary aspects of learned generative structure, including trajectory compressibility, robustness to distributional perturbation, and the extent to which memorized information can be removed without globally disrupting generation. While each intervention has been studied in isolation, their collective effects—and the extent to which diffusion models preserve generative fidelity, diversity, and probabilistic consistency under such modifications—remain largely uncharacterized. We address this gap through a controlled empirical analysis comparing low-resolution (32×32) and high-resolution (256×256) diffusion models subjected to each intervention.

Our evaluation spans complementary notions of model quality and stability, including sample fidelity, semantic diversity, and probabilistic consistency. We measure these properties using Fréchet Inception Distance (FID), Inception Score (IS), and negative log-likelihood (NLL) computed via the DDPM variational bound. Importantly, these metrics capture distinct aspects of generative behavior, allowing us to study when perceptual quality, likelihood fidelity, and robustness remain aligned under intervention—and when they diverge.

Our results show that diffusion models exhibit distinct and sometimes competing failure modes across intervention axes, revealing nontrivial trade-offs between efficiency, robustness, and data dependence. Distillation largely preserves fidelity on low-resolution datasets but can collapse sample quality at higher resolution under aggressive trajectory compression. Robustness fine-tuning can improve tolerance to distributional shift while simultaneously degrading clean-sample fidelity, and targeted unlearning can substantially increase forget-set NLL while disrupting global generative quality. Together, these findings provide a unified lens for understanding how common post-training modifications interact with learned generative structure and highlight open challenges in maintaining reliable generalization under model adaptation.

Beyond reporting empirical trends, we interpret these failure modes through the lens of diffusion dynamics. In particular, we view post-training interventions as perturbations to the learned score field and generative trajectory. Under this perspective, stability depends not only on model capacity but also on how sensitive the denoising process is to changes in trajectory length, input distribution, and objective structure. This viewpoint provides a unifying explanation for why high-resolution models exhibit greater fragility and why likelihood-based and perceptual metrics can diverge under post-training modification.

2. Methods

2.1. Distillation for Accelerated Sampling

We study *diffusion model distillation* as a post-training intervention aimed at accelerating sampling while preserving the generative behavior of a pretrained Denoising Diffusion Probabilistic Model (DDPM). From a foundational perspective, distillation probes the extent to which learned denoising trajectories can be compressed while preserving generative fidelity and probabilistic consistency. Our formulation follows the *Simple and Fast Distillation* (SFD) paradigm

of Zhou et al. [9], which casts distillation as a trajectory-matching problem between a high-fidelity teacher sampler and a fine-tuned student denoiser.

Given a pretrained DDPM denoising network ϵ_θ (teacher), the goal of distillation is to obtain a student denoiser ϵ_ψ, initialized from θ, that produces samples of comparable quality using a reduced number of denoising steps. The diffusion forward process, variance schedule, and training parameterization are held fixed; distillation only updates the denoiser parameters. From a geometric perspective, this intervention preserves the model's learned data distribution while altering the path taken through latent space during generation.

Progressive distillation is performed by comparing complete denoising trajectories generated by the teacher and student from a shared noise realization. Rather than matching individual denoising transitions in isolation, SFD minimizes a cumulative discrepancy between trajectories, encouraging the student to approximate the teacher's overall denoising behavior using fewer steps. Distillation proceeds in stages, approximately halving the number of sampling steps at each stage (e.g., $1000 \to 500 \to 250$).

This substantially improves sampling efficiency but may alter the learned noise prior and the geometry of the generative trajectory. In this work, we treat progressive distillation as a controlled post-training intervention and evaluate its effect on generative stability under trajectory compression. Full mathematical details of the trajectory-matching objective, numerical solvers, and training procedure are provided in Appendix A.

2.2. Robustness Fine-Tuning

We study *robustness fine-tuning* as a post-training intervention aimed at improving the stability of a pretrained DDPM under distributional shift. Robustness is defined here as consistent denoising behavior when inputs are subjected to common, model-independent corruptions, rather than worst-case or adversarial perturbations. This framing follows the average-case robustness perspective of Hendrycks and Dietterich [3]. From a broader perspective, robustness fine-tuning probes whether learned score fields remain coherent under structured perturbations to the input distribution.

Robustness fine-tuning is implemented by continuing training from a clean pretrained DDPM while injecting stochastic corruptions into a subset of training images prior to the diffusion process. These corruptions correspond to semantically meaningful degradations such as noise, blur, and photometric distortions, and are independent of model parameters. The diffusion forward process, noise schedule, and denoising objective remain unchanged; only the input distribution is perturbed.

Because the DDPM objective decomposes into a sequence of conditional denoising problems across noise levels, exposure to corrupted inputs alters the distribution of denoising tasks encountered during training. Intuitively, this encourages the learned score field to remain stable under structured, non-Gaussian perturbations, promoting smooth interpolation between clean and corrupted data manifolds. Robustness fine-tuning does not introduce additional regularization terms, auxiliary losses, or architectural modifications, isolating corruption-induced distributional shift as the sole intervention.

We restrict attention to a small set of representative corruptions spanning distinct corruption families, reflecting the empirical observation that robustness gains arise primarily from exposure to diverse corruption modes rather than specialization to individual perturbations. The full mathematical formulation of the diffusion objective and corruption placement is provided in Appendix B, with implementation details in Appendix D.

2.3. Targeted Unlearning

We study *targeted unlearning* in pretrained denoising diffusion probabilistic models. Given a pretrained model and a designated *forget subset* $\mathcal{D}_f$ of the training distribution, the goal is to update parameters such that the model's fit to $\mathcal{D}_f$ degrades while preserving utility on the complementary *keep subset* $\mathcal{D}_k$. From a foundational perspective, targeted unlearning probes whether memorized information can be selectively removed without globally disrupting learned generative structure.

We evaluate targeted unlearning on two datasets: (i) CelebA-HQ, where $\mathcal{D}_f$ is defined by an attribute-based filter (Smiling), and (ii) CIFAR-10, where $\mathcal{D}_f$ corresponds to a single semantic class (airplane). In both cases, we fine-tune only the U-Net denoiser of a pretrained DDPM, keeping the diffusion schedule and variance parameters fixed.

Unlearning is implemented using a SISS-style negative deletion objective [1], which combines standard denoising on $\mathcal{D}_k$ with a negative denoising term on $\mathcal{D}_f$. This update explicitly trades off retention and deletion, encouraging the model to remain accurate on $\mathcal{D}_k$ while becoming inaccurate on $\mathcal{D}_f$. We hold the unlearning strength fixed within each experiment and train for a limited number of fine-tuning steps.

To evaluate unlearning, we measure both forgetting and retained utility. Forgetting is quantified using likelihood-based metrics, computing the DDPM variational negative log-likelihood bound on forget-set examples before and after unlearning. Retained utility is assessed using standard sample-quality metrics, including Fréchet Inception Distance (FID) and Inception Score (IS), computed under a fixed sampling budget. Localized forgetting would suggest partially modular or disentangled generative representations, while severe global degradation under deletion may

indicate strongly entangled learned structure. Full details of the unlearning objective and likelihood computation are provided in Appendix C.

3. Experimental Setup

Pretrained Models. All experiments start from clean DDPM checkpoints released with the original DDPM work [4], pretrained separately on CIFAR-10 and CelebA-HQ. Specifically, we use the publicly available CIFAR-10 (32×32) and CelebA-HQ (256×256) pretrained DDPM models provided by Google.[1]

3.1. Progressive Distillation

Setup. We apply progressive distillation to accelerate sampling from pretrained DDPMs by iteratively reducing the number of diffusion timesteps while preserving sample quality. Starting from a 1000-step teacher model, we train a sequence of student models with progressively fewer steps.

Training Protocol. Distillation proceeds in sequential stages that halve the timestep count ($1000 \to 500 \to 250$). At each stage, the student is trained to reproduce in a single denoising step the effect of two teacher steps, following a trajectory-matching objective. Each student is initialized from the teacher's weights and fine-tuned using the same architecture and noise schedule.

Evaluation. We evaluate distilled models by comparing sample quality across timestep reductions using fixed random seeds and standard generative metrics. Full objective definitions and implementation details are provided in Appendix A.

3.2. Robustness Fine-Tuning

Setup. We study robustness by continuing training from a clean pretrained DDPM while injecting stochastic image corruptions on-the-fly during data loading, inducing a controlled distributional shift without modifying the diffusion objective or architecture.

Training Protocol. During fine-tuning, each training image is independently corrupted with fixed probability using a representative corruption mix spanning noise, blur, weather, and digital artifacts adapted from CIFAR-10-C [3]. Fine-tuning is performed for a limited number of gradient steps to probe robustness without extensive retraining.

[1] https://huggingface.co/google/ddpm-cifar10-32, https://huggingface.co/google/ddpm-celebahq-256

Evaluation. We evaluate robustness by comparing sample quality metrics before and after fine-tuning. Full objective and implementation details are provided in Appendix B and Appendix D.

3.3. Targeted Unlearning

Setup. We evaluate targeted unlearning on pretrained DDPMs for both CelebA-HQ and CIFAR-10 by defining dataset-specific keep and forget subsets. For CelebA-HQ, the forget set consists of images annotated with the *Smiling* attribute, with all remaining images forming the keep set. For CIFAR-10, the forget set is the *airplane* class, with all other classes retained.

Training Protocol. Unlearning is performed by fine-tuning the denoising network using paired minibatches from the keep and forget sets. Each update applies a SISS-style deletion objective that increases loss on the forget set while preserving performance on the keep set. The unlearning strength is held fixed within each experiment.

Evaluation. We evaluate unlearning along two axes: (i) *forgetting*, measured by changes in the DDPM variational NLL bound on forget-set examples, and (ii) *utility*, measured by FID and Inception Score on generated samples. Effective unlearning corresponds to increased forget-set NLL with minimal degradation in global sample quality. Full objective and implementation details are provided in Appendix C.

Together, these metrics characterize the unlearning trade-off: effective targeted unlearning manifests as increased forget-set NLL with minimal degradation in global sample quality.

4. Results and Discussion

4.1. Progressive Distillation

We train and evaluate progressive distillation on CIFAR-10 and CelebA-HQ by iteratively halving the number of diffusion steps from the original 1000-step model (Stage 0) to 500 steps (Stage 1) and 250 steps (Stage 2). Performance is assessed using FID to measure sample quality and negative log-likelihood (NLL) to quantify likelihood fidelity. Lower values are better for both metrics. Table 1 reports the FID results, and Table 2 reports the corresponding NLL results.

On CIFAR-10, progressive distillation yields a small but consistent improvement in FID as the number of steps decreases, suggesting that the distilled models preserve sample quality despite reduced sampling depth. The improvements are modest, however, and are best interpreted as preservation of image quality rather than meaningful improvement.

Dataset	Steps (Distillation Stage)		
	1000 (S0)	**500** (S1)	**250** (S2)
CIFAR-10	134.95	134.27	133.57
CelebA-HQ	92.83	106.47	297.62

Table 1. FID scores for progressive distillation on CIFAR-10 and CelebA-HQ. We examine FID scores for both models at Stages 0, 1, and 2, halving the number of sampling steps each time from 1000 steps.

Dataset	1000 (S0)	500 (S1)	250 (S2)
CIFAR-10	52921.0	**52730.6**	52867.6
CelebA-HQ	**3209782.8**	3284714.8	3422308.0

Table 2. Negative log-likelihood (NLL) for progressive distillation on CIFAR-10 and CelebA-HQ. Lower is better.

Dataset	Fine-Tuning Steps			
	500 / 50	**1000 / 100**	**1500 / 150**	**2000 / 200**
CIFAR-10	**265.73**	294.54	312.08	324.54
CelebA-HQ	**87.78**	135.94	120.81	108.09

Table 3. FID scores after robustness fine-tuning. Columns indicate fine-tuning steps for CIFAR-10 / CelebA-HQ respectively. Lower is better.

Dataset	Fine-Tuning Steps			
	500 / 50	**1000 / 100**	**1500 / 150**	**2000 / 200**
CIFAR-10	**52998.8**	53023.3	53087.0	53155.6
CelebA-HQ	3264617.0	3226634.8	**3111223.0**	3144088.0

Table 4. Negative log-likelihood (NLL) after robustness fine-tuning. Lower is better.

In contrast, CelebA-HQ exhibits substantially greater fragility. While the 500-step model incurs only moderate degradation in FID relative to the base model, the 250-step model suffers a severe collapse in sample quality; qualitatively, generated images become largely unrecognizable at this stage. This sharp increase in FID indicates that aggressive trajectory compression can substantially degrade high-resolution generation. More broadly, the contrast between CIFAR-10 and CelebA-HQ suggests that low-resolution generative structure may be more compressible, while high-resolution generation depends more sensitively on the full denoising trajectory.

Similarly to FID, NLL is relatively preserved across distillation stages on CIFAR-10. NLL is lowest at Stage 1 (500 steps), with only minor variation across stages, indicating that moderate distillation does not meaningfully harm likelihood modeling in this low-resolution setting.

CelebA-HQ, however, shows a clear degradation in NLL as the number of steps decreases: the base 1000-step model achieves the lowest NLL, and each subsequent distillation stage increases it. This trend aligns with the FID results and suggests that likelihood fidelity deteriorates as the diffusion process is aggressively shortened. The agreement between FID and NLL in this setting indicates that, under severe trajectory compression, perceptual quality and probabilistic consistency can fail together.

4.2. Robustness Fine-Tuning

We evaluate the effects of robustness fine-tuning on CIFAR-10 and CelebA-HQ by continuing training from a clean pretrained DDPM while injecting representative, model-independent input corruptions. Models are fine-tuned for a small number of gradient steps (500–2000 for CIFAR-10 and 50–200 for CelebA-HQ, due to larger computational costs), and performance is again assessed using FID, NLL, and IS. Lower values are better for FID and NLL, while higher values are better for IS. Table 3 reports FID, Table 4 reports NLL, and Table 5 reports IS.

On CIFAR-10, robustness fine-tuning consistently degrades FID as the number of fine-tuning steps increases. The lowest FID is achieved after 500 steps, with progressively worse sample quality observed at longer fine-tuning horizons. This suggests a trade-off between exposure to corrupted inputs during fine-tuning and clean-sample fidelity, with over-fine-tuning leading to noticeable degradation in generative quality.

CelebA-HQ exhibits a similar but less monotonic trend. While the 50-step model achieves the lowest FID overall, moderate fine-tuning (150–200 steps) partially recovers sample quality relative to the 100-step checkpoint. Nonetheless, none of the robustness-fine-tuned models outperform the clean baseline, indicating that even limited corruption exposure can negatively impact high-resolution sample fidelity.

Likelihood-based evaluation reveals complementary behavior. On CIFAR-10, NLL increases gradually with additional fine-tuning steps, indicating a slow erosion of likelihood fidelity under robustness pressure. While the changes are relatively small in magnitude, the trend mirrors the degradation observed in FID.

In contrast, CelebA-HQ exhibits an improvement in NLL with moderate fine-tuning, reaching its lowest value at 150 steps before slightly increasing at 200 steps. This suggests that limited robustness fine-tuning may improve likelihood calibration on high-resolution data, even as perceptual sample quality, as measured by FID, remains degraded. This divergence between FID and NLL is important: robust-

Dataset	Fine-Tuning Steps			
	500 / 50	1000 / 100	1500 / 150	2000 / 200
CIFAR-10	**1.493**	1.325	1.284	1.249
CelebA-HQ	2.007	**2.240**	2.000	1.779

Table 5. Inception Score (IS) after robustness fine-tuning. Higher is better.

ness adaptation may preserve or improve aspects of density modeling while simultaneously degrading perceptual realism.

Inception Score trends further reinforce these findings. On CIFAR-10, IS steadily decreases with additional fine-tuning, consistent with reduced sample diversity and semantic coherence. CelebA-HQ again exhibits a non-monotonic pattern, with IS peaking at 100 steps before declining, suggesting that limited robustness exposure may transiently improve semantic diversity without preserving overall perceptual quality. Together, these results indicate that robustness fine-tuning does not produce a single monotonic notion of improvement; instead, it can shift the model along competing axes of likelihood fidelity, visual quality, and semantic diversity.

4.3. Targeted Unlearning

We evaluate targeted unlearning on two pretrained DDPMs (CIFAR-10 and CelebA-HQ 256×256) using three complementary metrics: (i) forget-set negative log-likelihood (NLL) to quantify forgetting strength on $\mathcal{D}_f$, and (ii) FID and IS to assess overall sample quality and diversity before vs. after unlearning. Lower FID indicates better sample quality, while higher NLL on $\mathcal{D}_f$ indicates stronger forgetting.

Table 6 reports global sample-quality metrics. On both datasets, unlearning increases FID, indicating a degradation in sample quality under the same sampling budget. The magnitude of degradation differs substantially across datasets: CIFAR-10 exhibits a moderate increase in FID, while CelebA-HQ shows a dramatic increase. Inception Score trends are mixed: IS increases slightly on CIFAR-10 after unlearning, but decreases sharply on CelebA-HQ, consistent with the larger FID deterioration in that setting.

Table 7 reports forget-set NLL before and after unlearning. In both settings, unlearning increases NLL on $\mathcal{D}_f$, indicating that the updated model assigns lower likelihood, or higher negative log-likelihood, to the targeted forget subset. The increase is modest on CIFAR-10 but extremely large on CelebA-HQ, suggesting substantially stronger and more disruptive forgetting in the CelebA-HQ experiment.

Trade-off between forgetting and utility. Across both datasets, the unlearning objective successfully increases forget-set NLL, consistent with targeted forgetting, but with a degradation in global sample quality as measured by FID. The trade-off is mild on CIFAR-10, where forget-set NLL increases by $\approx 48\%$ and FID increases by $\approx 54\%$. In contrast, CelebA-HQ exhibits a much stronger effect on both axes: forget-set NLL increases by more than three orders of magnitude, accompanied by a large deterioration in FID and a sharp drop in IS. These results highlight that unlearning strength and retained generative utility can vary substantially across datasets and model scales, even under the same unlearning procedure and comparable evaluation budgets.

From the perspective of memorization and generalization, the CelebA-HQ result suggests that targeted forgetting may not remain localized in high-resolution diffusion models. Instead, removing an attribute-defined subset can substantially disrupt global generative structure, indicating that the targeted concept may be entangled with broader visual features learned by the model. This contrasts with CIFAR-10, where forgetting induces a milder degradation, suggesting that class-level deletion in lower-resolution settings may be less globally destabilizing.

5. Related Work

Diffusion Model Distillation and Acceleration. A substantial body of recent work has focused on accelerating diffusion-based generative models by reducing the number of sampling steps while preserving sample quality. Early approaches such as DDIM [8] and higher-order ODE solvers [6] reinterpret diffusion sampling as deterministic or semi-deterministic numerical integration, enabling faster inference without retraining. Subsequent methods introduce learned distillation objectives, training a student model to approximate the behavior of a slower teacher sampler. Progressive distillation methods [9] formalize this process as trajectory matching, iteratively halving the number of denoising steps while transferring the teacher's generative dynamics to the student.

While these methods demonstrate impressive speedups, prior work largely evaluates distillation through perceptual metrics such as FID or human preference scores, with limited attention to likelihood fidelity or stability across datasets and resolutions. Our work complements this literature by explicitly examining how aggressive step reduction impacts both perceptual quality and likelihood-based consistency, revealing dataset-dependent failure modes that are not apparent from sample quality alone.

Robustness and Stability in Diffusion Models. Robustness in generative models has traditionally been studied through adversarial purification and defense mechanisms. Diffusion-based purification methods [5] leverage the iterative denoising process to remove adversarial perturbations, with recent analyses highlighting the role of sampling

Dataset	FID (Before)	FID (After)	Δ FID	IS (Before)	IS (After)
CIFAR-10	134.79	208.27	+73.47	1.60 ± 0.05	1.77 ± 0.073
CelebA-HQ (256)	72.07	408.25	+336.18	2.09 ± 0.14	1.35 ± 0.05

Table 6. Sample-quality metrics before vs. after unlearning. Lower FID is better. Inception Score (IS) is reported as mean $\pm$ std over splits.

Dataset	Forget NLL (Before)	Forget NLL (After)	Δ NLL	After/Before
CIFAR-10	48,346	71,546	23,200	1.480
CelebA-HQ (256)	267,583	1,263,735,808	1,263,468,224	4,722.2

Table 7. Forget-set negative log-likelihood (NLL) before vs. after unlearning. Higher NLL on $\mathcal{D}_f$ indicates stronger forgetting.

stochasticity in robustness. Parallel work proposes adversarial or robust training objectives for diffusion models by incorporating worst-case perturbations or modifying the denoising objective [7].

In contrast to adversarial robustness, Hendrycks and Dietterich [3] introduce a benchmark for average-case robustness under common, semantically meaningful corruptions, emphasizing distributional shift rather than worst-case attacks. Our robustness fine-tuning protocol adopts this perspective, applying model-independent corruptions during post-training without modifying the diffusion objective or architecture. Unlike prior work that proposes new robust diffusion formulations, we focus on measuring how limited robustness fine-tuning affects generative stability, revealing systematic trade-offs between corruption tolerance, perceptual fidelity, and likelihood calibration.

Machine Unlearning in Generative Models. Machine unlearning has emerged as an important requirement for privacy, copyright compliance, and data governance. Early unlearning methods focus on discriminative models, relying on retraining or influence-based approximations [2]. More recently, unlearning in generative models has gained attention, with approaches ranging from data poisoning reversal to targeted fine-tuning objectives. For diffusion models, Alberti et al. [1] propose Subtracted Importance Sampled Scores (SISS), framing unlearning as a trade-off between retaining utility on a keep set and degrading performance on a designated forget set.

Most existing evaluations of unlearning in diffusion models emphasize perceptual changes or classifier-based membership signals. Our work extends this line by explicitly evaluating unlearning through likelihood-based metrics derived from the DDPM variational bound, enabling a direct measurement of probability mass removal from the forget set. By pairing this with standard sample-quality metrics, we characterize the stability–utility trade-off induced by unlearning across datasets and model scales.

Memorization, Generalization, and Evaluation in Generative Models. A growing line of work studies whether deep generative models learn robust distributional structure or instead memorize and reproduce aspects of their training data. These questions are closely tied to privacy, robustness, and reliable evaluation, since high sample quality alone does not necessarily imply genuine generalization. In diffusion models, memorization and generalization are especially difficult to disentangle because generation proceeds through a long denoising trajectory, and likelihood-based, perceptual, and semantic metrics may capture different aspects of the learned distribution. Our work uses post-training interventions as controlled probes of this behavior: distillation tests whether generative trajectories are compressible, robustness fine-tuning tests whether learned score fields remain stable under distributional perturbation, and targeted unlearning tests whether information associated with a forget subset can be removed without globally disrupting generation.

Contribution of Our Work. While prior work studies distillation, robustness, and unlearning largely in isolation, our contribution is a unified empirical analysis of these interventions as *post-training interventions* to pretrained diffusion models. By evaluating their effects using both perceptual and likelihood-based metrics across low- and high-resolution settings, we provide a comparative perspective on diffusion model stability that complements algorithmic advances in each individual area. More broadly, our results show that post-training interventions can expose differences between fidelity preservation, likelihood consistency, and localized forgetting, offering a practical lens for studying what diffusion models preserve, destabilize, or forget under adaptation.

6. Conclusion and Future Work

In this work, we presented a unified empirical study of diffusion model stability under three common post-training interventions: progressive distillation, robustness fine-tuning

under realistic corruptions, and targeted unlearning. We evaluated across both low-resolution (CIFAR-10) and high-resolution (CelebA-HQ) settings using FID, IS, and DDPM variational-bound NLL. Our results reveal clear, dataset-dependent trade-offs: moderate distillation can preserve fidelity at low resolution but can collapse sample quality at high resolution when pushed aggressively; robustness fine-tuning tends to erode clean-sample fidelity as training continues, while likelihood and perceptual metrics can diverge; and SISS-style unlearning reliably increases forget-set NLL, but can significantly degrade global sample quality—especially on CelebA-HQ, where forgetting appears markedly more disruptive.

More broadly, our findings suggest that post-training interventions can serve as controlled probes of what diffusion models learn, preserve, and destabilize under adaptation. The differing stability profiles across interventions, datasets, and metrics indicate that generative fidelity, likelihood consistency, robustness, and localized forgetting may depend on partially distinct aspects of learned generative structure. In particular, the strong degradation observed in high-resolution settings suggests that some learned concepts may be deeply entangled with global generative behavior rather than cleanly separable from it.

A natural next step is to disentangle selective forgetting from global degradation by adding retain-set likelihood and quality measurements (e.g., NLL on $\mathcal{D}_k$, conditional FID/IS on non-forget subsets) and by sweeping the unlearning strength λ to map the Pareto frontier between forgetting and utility. More broadly, future work should explore intervention interactions (e.g., unlearning on distilled models, or robustness fine-tuning followed by unlearning), and design stabilizing variants—such as regularization toward the pretrained model on $\mathcal{D}_k$, constrained updates, or alternative objectives—to achieve stronger guarantees of reliability when diffusion models are adapted beyond their original training regime. Overall, our findings suggest that post-training interventions should be evaluated as changes to the visual generative system as a whole, rather than only as efficiency, robustness, or deletion mechanisms in isolation.

References

[1] Silas Alberti, Kenan Hasanaliyev, Manav Shah, and Stefano Ermon. Data unlearning in diffusion models. *arXiv preprint arXiv:2503.01034*, 2025. Preprint. 3, 7, 9

[2] Antonio Ginart, Melody Guan, Gregory Valiant, and James Zou. Making ai forget you: Data deletion in machine learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 7

[3] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 3, 4, 7, 10

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. 4

[5] Yiming Liu, Kezhao Liu, Yao Xiao, Ziyi Dong, Xiaogang Xu, Pengxu Wei, and Liang Lin. Towards understanding the robustness of diffusion-based purification: A stochastic perspective. *arXiv preprint arXiv:2404.14309*, 2025. 6

[6] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022. 6

[7] Mehrdad Moradi and Kamran Paynabar. Rddpm: Robust denoising diffusion probabilistic model for unsupervised anomaly segmentation. *arXiv preprint arXiv:2508.02903*, 2025. 7

[8] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 6

[9] Zhenyu Zhou, Defang Chen, Can Wang, Chun Chen, and Siwei Lyu. Simple and fast distillation of diffusion models. *arXiv preprint arXiv:2409.19681*, 2024. Preprint. 3, 6, 8

A. Progressive Distillation: Objective and Training Details

A.1. Trajectory Matching Objective

Progressive distillation follows the Simple and Fast Distillation (SFD) framework of Zhou et al. [9], which formulates distillation as a global trajectory-matching problem between a high-fidelity teacher sampler and a student denoiser.

Let ϵ_θ denote the denoising network of a pretrained DDPM (teacher), and ϵ_ψ denote the student network initialized from θ. Starting from a shared noise realization $x_N \sim \mathcal{N}(0, I)$, the teacher generates a reference denoising trajectory

$$\{\tilde{x}_N, \tilde{x}_{N-1}, \ldots, \tilde{x}_0\},$$

while the student generates its own trajectory

$$\{x_N, x_{N-1}, \ldots, x_0\}.$$

The student is trained to minimize the cumulative discrepancy between its trajectory and the teacher's:

$$\mathcal{L}_{\text{SFD}}(\psi) = \sum_{n=0}^{N-1} d\big(x_n^\psi, \tilde{x}_n\big), \tag{1}$$

where $d(\cdot, \cdot)$ denotes a distance in data space (we use L_1). This objective encourages the student to approximate the teacher's full denoising trajectory rather than individual transitions in isolation.

A.2. Numerical Solvers

Teacher trajectories are generated using high-order numerical solvers such as DDIM or Heun samplers applied to

ϵ_θ. Student trajectories are generated using efficient solvers such as DPM-Solver. Although both teacher and student traverse the same sequence of timesteps, they differ in how the denoising vector field is evaluated and integrated, leading to potentially divergent trajectories that are aligned through distillation.

A.3. Training Procedure

Distillation is performed progressively by halving the number of sampling steps at each stage (1000 $\rightarrow$ 500 $\rightarrow$ 250). Each student is initialized from the teacher's weights and fine-tuned using AdamW optimization with gradient norm clipping (maximum norm 1.0) and an exponential moving average (EMA) update schedule.

After each denoising step during training, the intermediate state is detached from the computation graph before proceeding to the next timestep. This prevents backpropagation through time and ensures stable optimization as the student learns to correct accumulated trajectory errors.

B. Robustness Fine-Tuning: Objective and Training Details

This appendix describes the diffusion objective used during robustness fine-tuning. The corruption operators and stochastic corruption protocol are detailed separately in Appendix D.

B.1. Forward Diffusion Process

Let ϵ_θ denote the denoising network of a pretrained DDPM. For a clean image x_0 and timestep t sampled uniformly from $\{0, \dots, T-1\}$, the forward diffusion process is given by

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1-\alpha_t}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{2}$$

where $\{\alpha_t\}$ is derived from the fixed variance schedule $\{\beta_t\}$.

B.2. Denoising Objective

Training minimizes the standard DDPM denoising objective

$$\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E}_{x_0,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right]. \tag{3}$$

During robustness fine-tuning, corruptions are applied to the input image x_0 *prior* to the diffusion process. The diffusion forward process, noise schedule, and objective remain unchanged; only the input distribution is modified. This alters the distribution of conditional denoising tasks encountered during training and induces robustness pressure without introducing additional losses or architectural modifications.

C. Targeted Unlearning: Objective and Evaluation Details

C.1. SISS-Style Negative Deletion Loss

Targeted unlearning is implemented using the Subtracted Importance Sampled Scores (SISS) family of objectives [1]. We adopt the No-IS variant, which combines standard DDPM denoising on the keep set $\mathcal{D}_k$ with a negative denoising term on the forget set $\mathcal{D}_f$.

For a batch of images x, the standard DDPM loss is

$$\mathcal{L}_{\text{DDPM}}(x) = \mathbb{E}_{t\sim\text{Unif}\{0,\dots,T-1\},\,\epsilon\sim\mathcal{N}(0,I)}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right]. \tag{4}$$

Given paired minibatches $x_k \sim \mathcal{D}_k$ and $x_f \sim \mathcal{D}_f$, we optimize

$$\mathcal{L}_{\text{unlearn}} = \mathcal{L}_{\text{DDPM}}(x_k) - \lambda\,\mathcal{L}_{\text{DDPM}}(x_f), \tag{5}$$

where $\lambda > 0$ controls unlearning strength. The same timestep vector and noise tensor are used for both batches to isolate the effect of deletion.

C.2. Variational NLL Computation

Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. The forward diffusion distribution is

$$q(x_t \mid x_0) = \mathcal{N}\Big(\sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t) I\Big). \tag{6}$$

The true posterior admits the closed form

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(\tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I), \tag{7}$$

with

$$\tilde{\beta}_t = \beta_t \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}. \tag{8}$$

We use the standard reverse-process parameterization

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t), \tilde{\beta}_t I), \tag{9}$$

where

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\epsilon_\theta(x_t, t)\right). \tag{10}$$

C.3. Discretized Likelihood Term

The variational bound on the negative log-likelihood decomposes as

$$\begin{aligned} -\log p_\theta(x_0) \;\le\; & \mathrm{KL}(q(x_T \mid x_0)\,\|\,p(x_T)) \\ & + \sum_{t=2}^{T} \mathrm{KL}(q(x_{t-1} \mid x_t, x_0)\,\|\,p_\theta(x_{t-1} \mid x_t)) \\ & - \log p_\theta(x_0 \mid x_1), \end{aligned} \tag{11}$$

with $p(x_T) = \mathcal{N}(0, I)$. For $t \geq 2$, each KL term is computed between Gaussians with equal variance $\tilde{\beta}_t I$, differing only in the posterior and model means.

The $t = 1$ term is computed using a discretized Gaussian likelihood for 8-bit images mapped to $[-1, 1]$, following the Improved DDPM formulation. We report mean and standard deviation of NLL on the forget set before and after unlearning. Successful unlearning corresponds to a selective increase in NLL on $\mathcal{D}_f$.

D. Corruption Implementation Details

This appendix provides the exact implementation details for the corruption-based robustness fine-tuning procedure described in Section 2.2. All corruptions are applied on-the-fly during data loading and are independent of model parameters.

D.1. Representative Corruptions and Parameters

Following the corruption families introduced by Hendrycks and Dietterich [3], we select one representative corruption from each semantic family used during training. Table 8 summarizes the corruptions and parameters used at severity level 3.

Table 8. Representative corruptions and parameters at severity level 3.

Corruption	Family	Parameters (Severity 3)
Gaussian noise	Noise	$\sigma = 0.18$
Defocus blur	Blur	$\sigma = 3.0$ pixels
Fog	Weather	$\alpha = 0.3$
JPEG compression	Digital	$\beta = 0.2$

D.2. Corruption Operators

All corruptions operate on images represented in the $[0, 1]$ range and are applied channel-wise.

Gaussian Noise.

$$x' = \text{clip}\left(x + \epsilon,\, 0,\, 1\right), \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{12}$$

Defocus Blur (Proxy Implementation).

$$x' = x * G_\sigma, \quad G_\sigma(i, j) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2}\right) \tag{13}$$

Fog (Proxy Implementation).

$$x' = \text{clip}(x + \alpha,\, 0,\, 1) \tag{14}$$

JPEG Compression (Proxy Implementation).

$$x' = \text{clip}\left(\beta(x - \mu) + \mu,\, 0,\, 1\right), \tag{15}$$

where μ denotes the mean pixel value of the image.

D.3. Stochastic Corruption Protocol

For each training image, corruption is applied independently according to the following procedure:

1. Sample $u \sim \text{Uniform}(0, 1)$.
2. If $u < 0.2$, return the clean image.
3. Otherwise, sample a corruption uniformly from the set

   $$\{\text{Gaussian noise},\ \text{Defocus blur},\ \text{Fog},\ \text{JPEG compression}\}.$$

4. Apply the sampled corruption at severity level 3.

D.4. Dataset-Specific Image Handling

CIFAR-10. Images are loaded in the $[0, 1]$ range and corrupted directly. Outputs are clipped to $[0, 1]$ prior to input to the diffusion forward process.

CelebA-HQ. Images are stored and trained in the $[-1, 1]$ range. Prior to corruption, images are mapped to $[0, 1]$ via

$$x_{[0,1]} = \frac{x_{[-1,1]} + 1}{2}. \tag{16}$$

After corruption, images are mapped back to $[-1, 1]$:

$$x'_{[-1,1]} = 2x'_{[0,1]} - 1. \tag{17}$$

D.5. Implementation Proxies

For computational efficiency, several corruptions are implemented using proxy operators that preserve the semantic characteristics of their respective corruption families:

- **Defocus blur**: implemented using Gaussian blur with $\sigma = 3.0$ pixels.
- **Fog**: implemented using brightness adjustment with $\alpha = 0.3$.
- **JPEG compression**: implemented using contrast adjustment with $\beta = 0.2$.

These proxies enable efficient on-the-fly corruption during training while maintaining consistency with the intended corruption families.