MITIGATING MEMORIZATION IN TEXT-TO-IMAGE DIFFUSION MODELS WITH REPRESENTATION-SPACE SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models are prone to memorization, i.e., reproducing individual training examples, which raises concerns about data privacy and copyright. Existing mitigation approaches primarily intervene through data preprocessing or specialized sampling procedures, leaving the role of the training objective comparatively underexplored. In this work we study whether changing the space in which diffusion predictions are supervised can mitigate memorization. We posit that direct data-space supervision contributes to memorization by encouraging model predictions to recover the exact training samples. Motivated by this view, we introduce Representation-Space Supervision (RSS), a general training framework that matches model predictions to their targets in the feature spaces of pretrained visual representation models rather than supervising them directly in data space. We evaluate RSS in controlled text-to-image supervised fine-tuning experiments, showing that it consistently and substantially improves the trade-off between generative performance and training-example reproduction relative to standard diffusion training.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021; Karras et al., 2022; Lipman et al., 2023) have become a leading paradigm for high-quality generative modeling, enabling state-of-the-art visual generative models (Esser et al., 2024; Seedream et al., 2025; Black Forest Labs, 2025). Despite their empirical success, diffusion models are known to be prone to *memorization*, i.e., they can reproduce individual examples from their training data (Somepalli et al., 2023a; Carlini et al., 2023; Somepalli et al., 2023b). Such behavior raises data privacy and copyright concerns, motivating methods that reduce memorization while maintaining generative performance.

The generalization and memorization behaviors of diffusion models are mainly shaped by three factors: (i) the training data, (ii) the training method and (iii) the sampling procedure. Existing memorization-mitigation methods have primarily targeted the first and third aspects. Data-side interventions reduce data duplication or modify conditioning information (Carlini et al., 2023; Somepalli et al., 2023a;b), whereas sampling-side approaches detect and steer away from trajectories associated with memorized generations (Wen et al., 2024; Ross et al., 2025; Jain et al., 2025; Chen et al., 2024; 2025a). These approaches provide useful safeguards, but also have inherent limitations. Data-side interventions such as deduplication cannot fully mitigate memorization especially when the dataset size is relatively small as shown in Carlini et al. (2023), which is common in practice such as in supervised fine-tuning (SFT), where a small dedicated, high-quality dataset is used to fine-tune a pretrained model (Dai et al., 2023). Sampling-time defenses rely on specialized inference procedures, which can sacrifice sampling flexibility, and more importantly, do not directly prevent the model from memorizing the training data. In comparison, the role of the training objective itself in controlling memorization remains comparatively underexplored.

In this work, we investigate whether memorization can be mitigated by changing the space in which the diffusion models are supervised during training. Our motivation is intuitive: we posit that memorization can be a natural consequence of data-space supervision. Most diffusion-based training objectives such as denoising score matching (DSM) (Vincent, 2011; Song et al., 2021) or flow

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

matching (FM) (Lipman et al., 2023) are equivalent to training denoisers that take noisy data as input and predict the corresponding clean counterpart (Li & He, 2026). Standard diffusion training therefore encourages the denoisers' predictions to match the training data as closely as possible, which can make them prone to memorization. Indeed, numerous works show that, in finite-data settings, optimal denoisers can retrieve the exact training samples at low noise levels. Sampling with such denoisers can therefore reproduce training samples (Zhang et al., 2024; Gu et al., 2025).

Motivated by this view, we posit that direct data-space supervision should not be applied to training samples that we seek to protect from memorization. At the same time, these samples contain useful information that should still contribute to training. We therefore seek an alternative supervision signal that extracts useful information from the raw data without requiring direct instance-level matching. To this end, we leverage pretrained visual representation models such as DINOv2 (Oquab et al., 2023) and Inception-V3 (Szegedy et al., 2016) whose feature spaces retain high-level perceptual and semantic information of the data while reducing the specific instance-level details that can enable memorization. More specifically, given a target training dataset, we map both model predictions and training targets to the feature space of a representation model, and compute their discrepancy in this representation space rather than directly in the data space. We term this framework **R**epresentation-**S**pace **S**upervision (RSS). In our instantiations, RSS is combined with the standard diffusion objective: samples designated for protection are supervised in representation space, while the remaining training samples receive standard data-space supervision.

We evaluate RSS in the supervised fine-tuning stage of text-to-image diffusion models. Although RSS is not specific to fine-tuning, this setting provides a practically relevant and controlled testbed: the training set is known and relatively small, resulting in repeated exposure that makes memorization prominent and measurable. Our experiments show that standard fine-tuning progressively increases training-example reproduction and can eventually degrade generative performance. In contrast, RSS substantially improves the Pareto frontier of the performance–memorization trade-off relative to standard full-data training.

Summary of Contributions. Our main contributions are as follows:

- We introduce Representation-Space Supervision (RSS), a training-time framework for mitigating memorization by matching the diffusion model's predictions to training targets in the feature spaces of pretrained visual representation models rather than through direct data-space matching.
- Across extensive experiments, we show that RSS substantially improves the trade-off between generative capability and memorization in text-to-image diffusion models.

2 Preliminaries

2.1 Basics of Diffusion Models

Let $p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c})$ denote the ground truth conditional data distribution. Diffusion models gradually transform p_{data} into an isotropic Gaussian distribution through a forward noising process, which can be expressed by the following stochastic differential equation (SDE):

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{w}, \tag{1}$$

where $\boldsymbol{f}(\cdot, t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and $\boldsymbol{w}$ denotes standard Brownian motion. Let $p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})$ denote the transition distribution from $\boldsymbol{x}$ to $\boldsymbol{x}_t$ under eq. (1), and $p_t(\boldsymbol{x}_t|\boldsymbol{c}) = \int_{\boldsymbol{x}} p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}) p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$ be the marginal distribution of $\boldsymbol{x}_t$. For sufficiently large T, $p_T(\boldsymbol{x}|\boldsymbol{c})$ will be indistinguishable from an isotropic Gaussian distribution, denoted by $\pi(\boldsymbol{x})$. Importantly, the forward SDE eq. (1) can be reversed with a reverse-time probability-flow ODE or SDE (Anderson, 1982; Song et al., 2021), hence one can generate samples by first sampling from $\pi(\boldsymbol{x})$ and then running the reverse-time ODE or SDE. Specifically, the reverse-time ODE takes the following form:

$$\mathrm{d}\boldsymbol{x} = \big[\boldsymbol{f}(\boldsymbol{x}, t) - \frac{1}{2} g^2(t) \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}|\boldsymbol{c})\big]\mathrm{d}t. \tag{2}$$

Sampling from this reverse-time ODE requires the conditional score function $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c})$, which can be parameterized through a denoiser as $\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t|\boldsymbol{c}) \approx \big(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c}) - \boldsymbol{x}_t\big)/\sigma_t^2$,

where σ_t is the standard deviation of the additive noise in $\boldsymbol{x}_t$, and $\mathcal{D}_{\boldsymbol{\theta}}$ is a deep network-based denoiser with parameter $\boldsymbol{\theta}$ trained with the denoising score matching objective (Vincent, 2011):

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{DSM}}(\boldsymbol{\theta}) := \mathbb{E}_{t\sim p(t),\boldsymbol{c},\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\left[w(t)||\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{x})||_2^2\right], \tag{3}$$

where $p(t)$ is the training-time noise distribution, $w(t)$ is a time-adaptive weighting function and $\boldsymbol{c}$ specifies the condition such as a class label or text prompt. Importantly, flow matching (Lipman et al., 2023) can likewise be expressed through an equivalent denoising formulation, since velocity and score parameterizations are equivalent up to a linear transformation (Holderrieth & Erives, 2025; Li & He, 2026) under commonly used Gaussian probability paths.

Latent-Space Diffusion Models. Latent diffusion models (LDMs) perform diffusion in the latent space of a pretrained autoencoder (Rombach et al., 2022). Given an image $\boldsymbol{x}$, the encoder maps it to a clean latent $\boldsymbol{z} = \text{Enc}(\boldsymbol{x})$, and the diffusion model is trained to recover $\boldsymbol{z}$ from its noisy counterpart $\boldsymbol{z}_t$. During generation, sampled latents are mapped back to image space through the decoder $\text{Dec}(\cdot)$. We provide the corresponding training objective in Section A.1.

2.2 Memorization in Diffusion Models

The population-optimal solution to the DSM objective is the posterior mean denoiser under the data distribution. In practice, however, training uses a finite dataset $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$, hence the DSM is optimized over the empirical distribution $p_{\text{emp}}(\boldsymbol{x}|\boldsymbol{c}) = \frac{1}{n}\sum_{i=1}^n \delta(\boldsymbol{x}-\boldsymbol{x}_i)$, where $\delta(\cdot)$ denotes the Dirac delta function. Without loss of generality, we assume $p_{0t}(\boldsymbol{x}_t|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}_t;\boldsymbol{x},\sigma_t^2\boldsymbol{I})$. The optimal denoiser under p_{emp} then takes the form:

$$\mathcal{D}_{emp}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c}) = \mathbb{E}_{p_{\text{emp}}(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}[\boldsymbol{x}] = \frac{\sum_{i=1}^n \mathcal{N}(\boldsymbol{x}_t;\boldsymbol{x}_i,\sigma_t^2\boldsymbol{I})\boldsymbol{x}_i}{\sum_{i=1}^n \mathcal{N}(\boldsymbol{x}_t;\boldsymbol{x}_i,\sigma_t^2\boldsymbol{I})} = \frac{\sum_{i=1}^n e^{-\frac{||\boldsymbol{x}_t-\boldsymbol{x}_i||_2^2}{2\sigma_t^2}}\boldsymbol{x}_i}{\sum_{i=1}^n e^{-\frac{||\boldsymbol{x}_t-\boldsymbol{x}_i||_2^2}{2\sigma_t^2}}}. \tag{4}$$

Thus, the predictions of the empirical optimal denoiser are essentially the softmax-weighted combinations of finite training examples. As σ_t approaches zero, the softmax weights will concentrate on a single training sample, and sampling with the empirical denoiser will therefore retrieve the exact replicas from $\boldsymbol{X}$ (Gu et al., 2025). While memorization in practical deep network-based diffusion models is more complex, related training-example reproduction has been widely observed.

2.3 Visual Representation Models

Pretrained visual representation models map images to features useful for downstream tasks such as classification, detection, and segmentation. We consider representations from Inception-V3 (Szegedy et al., 2016), DINOv2 (Oquab et al., 2023), and MAE (He et al., 2022). Compared with raw image space, the feature space of a visual representation model provides a higher-level abstraction of the input. It preserves perceptual and semantic information useful for downstream visual tasks without necessarily retaining all instance-specific information required to reconstruct the original image. We therefore consider such feature spaces as candidates for instantiating RSS, as described in the next section.

3 Motivation and Method

3.1 Problem Setup

Throughout this paper, we consider training a diffusion model on a dataset $\mathcal{S} = \mathcal{S}_{\text{public}} \cup \mathcal{S}_{\text{private}}$, where the two subsets are disjoint. $\mathcal{S}_{\text{public}}$ contains samples for which memorization is not a concern, such as data licensed for the intended use. In contrast, $\mathcal{S}_{\text{private}}$ contains examples we wish to use to improve the model while suppressing their memorization by the model. Our goal is to improve generative performance beyond training on $\mathcal{S}_{\text{public}}$ alone while suppressing memorization of $\mathcal{S}_{\text{private}}$.

3.2 Method: From Data-Space to Transformed-Space Supervision

We posit that one important source of memorization is the strong instance-level supervision imposed by standard diffusion training. As shown in eq. (3), standard diffusion training directly encourages

the model's prediction $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c})$ to match the clean training target $\boldsymbol{x}$, or its latent encoding in an LDM shown in eq. (10), as closely as possible. Throughout this paper, we refer to objectives that directly match model predictions to clean targets in the model's native diffusion space as *data-space supervision*. While effective for learning the data distribution, data-space supervision can facilitate memorization when the training set is finite, as discussed in Section 2.2.

However, direct data-space supervision may not be necessary for leveraging useful information from the training data. For example, suppose our goal is to adapt a model to the color statistics of a private dataset $\mathcal{S}_{\text{private}}$. The aggregate color statistics may be useful, whereas the spatial arrangement and instance-specific structure of individual images need not be directly supervised. More broadly, a training image contains both task-relevant information useful for improving generative capabilities and instance-specific details that need not be directly recovered. Data-space supervision does not distinguish between these types of information and instead encourages the model to recover the complete training sample. This motivates replacing it with supervision in a space that preserves useful information while reducing sensitivity to exact instance-level details.

3.2.1 Transformed-Space Diffusion Training

Motivated by the discussion above, we propose to relax standard data-space supervision by computing the training loss in a transformed space. Let $\Phi(\cdot)$ denote a fixed transformation applied to both the model prediction and the clean training target. For pixel-space diffusion models, we define the transformed-space diffusion supervision as:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\Phi}(\boldsymbol{\theta}) := \mathbb{E}_{t,\boldsymbol{c},p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}),p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\big[w(t)\,||\Phi\big(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})\big) - \Phi(\boldsymbol{x})||_2^2\big]. \tag{5}$$

Unlike standard data-space supervision in eq. (3), transformed-space supervision does not directly require the denoiser prediction to match the clean target $\boldsymbol{x}$. Instead, it requires the model prediction and target to match only after applying $\Phi(\cdot)$. Consequently, the model is no longer explicitly supervised to reproduce every detail of each training image, but only the information retained by the transformation $\Phi(\cdot)$. An appropriately chosen transformation can therefore control which information from the training sample is imposed through the training objective.

To illustrate how transformed-space supervision relaxes the constraints imposed by standard diffusion training, we consider a simplified setting where the transformation is linear. Let $\Phi(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{A} \in \mathbb{R}^{d\times N}$ has rank $d < N$ and N is the data dimension. In this case, eq. (5) becomes:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t,\boldsymbol{c},p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}),p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\big[w(t)\,||\boldsymbol{A}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c}) - \boldsymbol{A}\boldsymbol{x}||_2^2\big]. \tag{6}$$

The following proposition characterizes the optimal predictions when the denoiser is treated as an unrestricted function.

Proposition 1 (Linear transformed-space supervision)**.** *Let $\boldsymbol{V} \in \mathbb{R}^{N\times d}$ contain the right singular vectors of $\boldsymbol{A}$ associated with its nonzero singular values, and let $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^T$ denote the orthogonal projection onto the row space of $\boldsymbol{A}$. For fixed $(\boldsymbol{x}_t, \sigma_t, \boldsymbol{c})$, define*

$$\boldsymbol{m}(\boldsymbol{x}_t, \boldsymbol{c}) := \mathbb{E}_{p_{data}(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}[\boldsymbol{x}].$$

When the denoiser is treated as an unrestricted function, the pointwise minimizers of eq. (6) *are*

$$\mathcal{D}^*(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c}) = \boldsymbol{P}\boldsymbol{m}(\boldsymbol{x}_t,\boldsymbol{c}) + (\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v}, \tag{7}$$

where $\boldsymbol{v} \in \mathbb{R}^N$ is an arbitrary vector.

Moreover, if the denoiser prediction at this input is treated as an independent optimization variable, gradient descent with a sufficiently small constant step size and initialization $\mathcal{D}_{\text{init}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})$ converges to

$$\mathcal{D}^*(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c}) = \boldsymbol{P}\boldsymbol{m}(\boldsymbol{x}_t,\boldsymbol{c}) + (\boldsymbol{I}-\boldsymbol{P})\mathcal{D}_{\text{init}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c}). \tag{8}$$

The proof is provided in Section E. Under standard data-space supervision, corresponding to $\boldsymbol{A} = \boldsymbol{I}$, the unique pointwise optimum is the posterior mean $\boldsymbol{m}(\boldsymbol{x}_t, \boldsymbol{c})$. In contrast, a rank-deficient transformation fixes only the component of the denoiser prediction in the row space of $\boldsymbol{A}$, while leaving its null-space component unconstrained. The unconstrained component instead depends on factors such as model initialization and the optimization algorithm, as illustrated by eq. (8).

When training with finite samples, p_{data} is replaced by the empirical distribution p_{emp}, and the posterior mean becomes the empirical optimal denoiser in eq. (4), which leads to memorization. If the initialized null-space component differs from that of the empirical optimal denoiser, the gradient descent solution in Proposition 1 differs as well. Thus, transformed-space supervision need not constrain every component of the prediction to match the empirical posterior mean.

The linear example shows how a transformation can relax supervision, but its usefulness ultimately depends on the information retained by $\Phi(\cdot)$. A suitable transformation should preserve information that benefits the desired generative capabilities while reducing sensitivity to instance-specific details whose direct supervision can facilitate memorization. More specifically, $\Phi(\cdot)$ should satisfy the following two properties:

- **Useful Information Preservation.** The transformation $\Phi(\cdot)$ should preserve information from the training data that is beneficial for improving the desired generative capabilities.
- **Memorization Suppression.** The transformation $\Phi(\cdot)$ should suppress the abundant information of the training data whose direct supervision can result in memorization.

Extension to Latent Diffusion Models. As discussed in Section 2.1, latent diffusion models perform diffusion in the latent space of a pretrained autoencoder. Because the standard LDM objective directly matches the denoiser prediction to the clean latent target $\boldsymbol{z} = \mathrm{Enc}(\boldsymbol{x})$, it remains a form of data-space supervision in the model's native diffusion space. Consequently, the finite-data memorization issue discussed in Section 2.2 applies analogously to the empirical distribution of encoded training samples. To apply transformed-space supervision using an image-space transformation $\Phi(\cdot)$, we decode both the predicted clean latent $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_t; \sigma_t, \boldsymbol{c})$ and the target latent $\boldsymbol{z}$, and compare $\Phi(\mathrm{Dec}(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_t; \sigma_t, \boldsymbol{c})))$ with $\Phi(\mathrm{Dec}(\boldsymbol{z}))$. Thus, latent diffusion changes the space in which the generative process is defined, whereas transformed-space supervision changes the space in which the training discrepancy is measured. The complete LDM objectives are provided in Section A.2.

3.2.2 Representation-Space Supervision

Although it is difficult to construct a transformation that provably satisfies the two desired properties introduced in the previous section, pretrained visual representations provide a natural choice. As discussed in Section 2.3, their feature spaces retain high-level perceptual and semantic information while tending to be less sensitive to exact pixel-level details. We therefore instantiate $\Phi(\cdot)$ using features extracted from a pretrained visual representation model and evaluate this choice empirically.

In our main experiments, we use the [CLS] feature of DINOv2. We further evaluate DINOv2 patch-token features and their pooled variants, as well as Inception-V3 and MAE features, in the ablation study. We term this instantiation of transformed-space supervision **R**epresentation-**S**pace **S**upervision (RSS).

In preliminary experiments, however, training with RSS alone can produce unwanted visual artifacts, indicating that the model can exploit invariances of the chosen representation space and that representation-space supervision alone does not preserve all information necessary for high-quality generation. We therefore complement RSS with standard data-space supervision as an anchor. Specifically, under the dataset partition introduced in Section 3.1, we apply standard data-space supervision to $\mathcal{S}_{\text{public}}$ and RSS to $\mathcal{S}_{\text{private}}$. For pixel-space diffusion models, the resulting objective is:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{t,\boldsymbol{c},\boldsymbol{x}\sim\mathcal{S}_{\text{public}},\boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\Big[w(t)\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{x}\|_2^2\Big] \\ & +\lambda\,\mathbb{E}_{t,\boldsymbol{c},\boldsymbol{x}\sim\mathcal{S}_{\text{private}},\boldsymbol{x}_t\sim p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\Big[w(t)\|\Phi\big(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})\big)-\Phi(\boldsymbol{x})\|_2^2\Big].\end{aligned} \tag{9}$$

which can be adapted to LDMs as shown in eq. (12). Here λ controls the relative strength of RSS.

4 Related Work

Memorization in Diffusion Models. A large body of work has shown that diffusion models can memorize and reproduce individual training examples (Somepalli et al., 2023a; Carlini et al., 2023; Somepalli et al., 2023b; Yoon et al., 2023; Gu et al., 2025; Zhang et al., 2024; Wang & Vastola, 2024; Li et al., 2024; Wen et al., 2024; Ross et al., 2025; Jain et al., 2025; Zhang et al., 2026). Prior

studies associate memorization with factors such as data duplication and text conditioning, motivating data-side interventions including deduplication and condition randomization (Carlini et al., 2023; Somepalli et al., 2023a;b). Another line of work mitigates memorization during sampling by exploiting its connection to classifier-free guidance (Wen et al., 2024; Ross et al., 2025; Jain et al., 2025; Chen et al., 2024; 2025a). These methods provide inference-time safeguards, whereas our work controls memorization through the supervision imposed during training.

Learning from Restricted Training Signals. A closely related line of research restricts the information exposed by individual training examples. Some approaches train diffusion models from corrupted or partial observations, including masked or blurred images (Daras et al., 2023; 2026; Zhao et al., 2026b), or images perturbed with additional Gaussian noise (Daras et al., 2024; Shah et al., 2025). Differentially private diffusion methods instead use mechanisms such as gradient clipping and noise injection to provide formal privacy guarantees (Dockhorn et al., 2023). Although not all of these methods were designed specifically to mitigate memorization, they demonstrate that diffusion models can learn from restricted training signals. RSS shares this general motivation, but uses pretrained visual representations to determine which information is retained in the supervision signal. Unlike generic corruption operators, these representations are designed to preserve high-level perceptual and semantic information, thereby providing a better candidate for balancing useful-information preservation and memorization suppression.

Representation-Based Supervision for Diffusion Models. Pretrained visual representations have previously been incorporated into diffusion training to improve optimization efficiency, generation quality, or sampling efficiency. REPA (Yu et al., 2025) aligns intermediate diffusion-model representations with features from a pretrained visual model, while Self-Perceptual Training (Lin & Yang, 2023), PixelGen (Ma et al., 2026), and Perceptual Flow Matching (Zhao et al., 2026a) incorporate perceptual or representation-based objectives more directly into generative training. Whereas these methods primarily use visual representations to improve generative performance or efficiency, RSS applies representation-space supervision selectively to examples whose direct reproduction should be suppressed and studies the resulting generative capability–memorization trade-off.

5 Experimental Results

Although the proposed RSS framework is not specific to a particular training stage, we evaluate it empirically in the supervised fine-tuning (SFT) stage of text-to-image (T2I) diffusion models. Modern T2I diffusion models are typically trained in multiple stages. Large-scale pretraining on web-scale datasets containing hundreds of millions of image–text pairs equips a model with broad visual knowledge and the ability to generate diverse visual concepts. However, pretrained models often remain limited in prompt-following ability, visual fidelity, or aesthetic quality. It is therefore common to fine-tune them on smaller, curated datasets targeting high-quality, domain-specific, or personalized generation. This additional training stage is commonly referred to as SFT. Despite its practical importance, SFT can be particularly susceptible to memorization because its datasets are often orders of magnitude smaller than those used for pretraining (Dai et al., 2023; Startsev et al., 2026; Li et al., 2025; Tong et al., 2026; Wang et al., 2026). The smaller scale of SFT datasets increases repeated exposure to individual examples and can make memorization especially pronounced. At the same time, the relatively small and known fine-tuning set provides a controlled test bed in which training-example reproduction can be measured more reliably than during web-scale pretraining. These properties make SFT both practically relevant and experimentally tractable for studying memorization.

5.1 Experimental Setup

Our experiments examine whether RSS improves the performance–memorization trade-off across different public/private partitions, visual representations, and diffusion-model formulations.

Data and Models. We pretrain DiT-style text-to-image diffusion models with approximately 1B parameters from scratch on a subset of GPIC (Chandrasegaran et al., 2026) containing approximately 10M diverse internet images. All main SFT experiments start from the same pretrained checkpoint.

Following Wang et al. (2026), we perform SFT on 120K-Mix, a high-quality synthetic dataset containing 120K image–text pairs collected from BLIP3o-60K (Chen et al., 2025b), the LAION DALL-

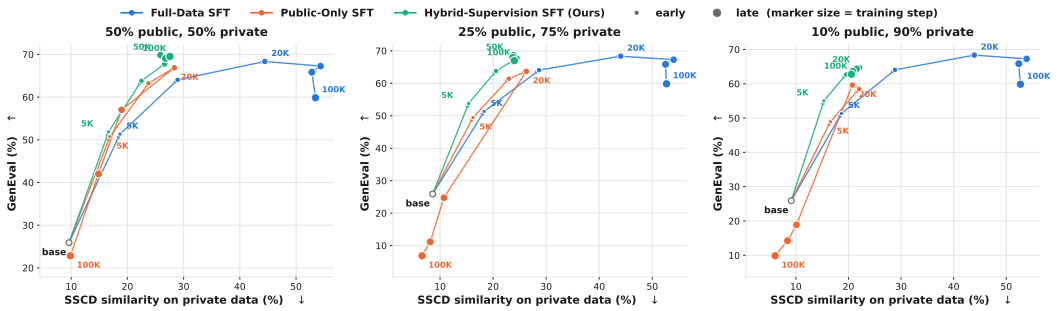


Figure 1: **Performance–memorization trade-offs.** Each panel shows the trade-off between GenEval performance and SSCD-based memorization on the private subset under a different public/private split. We compare *Full-Data SFT*, *Public-Only SFT*, and *Hybrid-Supervision SFT* (ours) over 100K training iterations, with larger markers indicating later training steps. Across all splits, Hybrid-Supervision SFT substantially improves the performance–memorization trade-off, approaching the generative performance of Full-Data SFT while exhibiting markedly less memorization of the private data.

E 3 Discord dataset (OpenDatasets, 2023), and ShareGPT-4o-Image (Chen et al., 2025c). Our main experiments use a latent diffusion model, while the results for pixel-space diffusion model are presented in Section B.3.

Training Protocol. For each experiment, we randomly partition 120K-Mix into $\mathcal{S}_{\text{public}}$ and $\mathcal{S}_{\text{private}}$ using public/private ratios of 50/50, 25/75, and 10/90. Here, "public" and "private" indicate whether direct data-space supervision is permitted under our experimental protocol. For each ratio, the same fixed partition is used by all compared methods. We compare three SFT strategies as follows.

- ***Full-Data SFT.*** This setting applies the standard diffusion objective to both $\mathcal{S}_{\text{public}}$ and $\mathcal{S}_{\text{private}}$, representing conventional SFT using the entire dataset with no memorization protection.
- ***Public-Only SFT.*** This setting discards $\mathcal{S}_{\text{private}}$ and applies standard data-space supervision only to $\mathcal{S}_{\text{public}}$, providing a reference that never accesses the private examples during fine-tuning.
- ***Hybrid-Supervision SFT.*** This setting applies standard data-space supervision to $\mathcal{S}_{\text{public}}$ and RSS to $\mathcal{S}_{\text{private}}$, following eqs. (9) and (12).

These baselines define the desired performance–memorization trade-off. Relative to Public-Only SFT, Hybrid-Supervision SFT should ideally achieve stronger generative performance, indicating that it benefits from the information in $\mathcal{S}_{\text{private}}$. Relative to Full-Data SFT, it should substantially reduce training-example similarity on $\mathcal{S}_{\text{private}}$ and approach the level attained by Public-Only SFT.

Unless otherwise specified, all methods are trained for 100K optimization steps using the same initialization, optimizer, learning-rate, batch size, and noise-sampling distribution. In the main RSS experiments, $\Phi(\cdot)$ is instantiated using the [CLS] feature of a frozen DINOv2 encoder (Oquab et al., 2023), with RSS weight $\lambda = 0.1$.

Evaluation. Because a severely degraded model can obtain low training-set similarity by generating poor or meaningless images, we evaluate training-example reproduction (memorization) jointly with generative capability. We measure generative capability using GenEval (Ghosh et al., 2023) and DPG (Hu et al., 2024), which evaluate object-level compositional generation and alignment with complex, long-form prompts, respectively. To evaluate training-example reproduction, we construct fixed probe sets by sampling 2,000 image–text pairs separately from $\mathcal{S}_{\text{public}}$ and $\mathcal{S}_{\text{private}}$. For each pair, we generate four images from independent initial noises conditioned on the training prompt, and compare them with the corresponding training image. We measure their similarity in both pixel space and the SSCD feature space (Pizzi et al., 2022), which is designed for image copy detection. We average the similarity over the four generations and then over the probe set, yielding pixel- and SSCD-based training-example similarity. All methods use the same sampling configuration and are evaluated at multiple checkpoints throughout the 100K-step SFT trajectory.

5.2 RSS Improves the Performance–Memorization Trade-off

Standard SFT Exhibits a Performance–Memorization Trade-off. As shown in Figures 1 to 3, *Full-Data SFT* initially improves GenEval and DPG performance, but the private-set similarity continues to increase throughout fine-tuning, and the generative performance eventually degrades. *Public-Only SFT* avoids direct supervision on the private subset $\mathcal{S}_{\text{private}}$ and consequently maintains substantially lower private-set similarity. However, removing the private data reduces the available supervision signal, resulting in lower peak performance and faster degradation, particularly when the public subset size is relatively small. These two baselines expose the central trade-off: using all available data provides stronger generative performance but increases training-data reproduction (memorization), whereas discarding private data reduces memorization at the cost of useful training signal.

RSS Improves the Performance–Memorization Frontier. Across all three partitions, *Hybrid-Supervision SFT* achieves a more favorable trade-off than both baselines (Figure 1). Relative to *Full-Data SFT*, it substantially reduces SSCD similarity on the private subset while retaining comparable generative capability. Relative to *Public-Only SFT*, it achieves stronger GenEval performance especially under the 25/75 and 10/90 partitions where *Public-Only SFT* discards most of the SFT data, while maintaining private-set similarity close to that of *Public-Only SFT* at its peak performance. For example, under the 50/50 partition at 100K steps, *Hybrid-Supervision SFT* achieves a GenEval score of 69.5 with a private-set SSCD similarity of 27.6, compared with 59.9 and 53.5, respectively, for *Full-Data SFT*. Thus, RSS maintains stronger generative performance under prolonged training while substantially reducing private-data reproduction. The early-stopping comparison in Table 1 leads to the same conclusion. Under the 50/50 partition, at each method's highest-GenEval checkpoint, *Hybrid-Supervision SFT* achieves a GenEval score of 69.8 with a private-set SSCD similarity of 25.9, compared with 68.3/44.5 for *Full-Data SFT* and 66.9/28.3 for *Public-Only SFT*.

RSS Remains Effective Throughout the Evaluated Training Horizon. Low training-set similarity alone does not establish memorization mitigation, as illustrated by *Public-Only SFT*: after 20K steps, its private-set similarity decreases together with substantial deterioration in GenEval, DPG, and visual quality. In contrast, as shown in Figure 3, *Hybrid-Supervision SFT* maintains strong generative performance while its private-set similarity largely plateaus after approximately 20K steps, rather than continuing to increase as under *Full-Data SFT*. The initial increase under RSS need not indicate memorization, since a comparable increase occurs under *Public-Only SFT* and coincides with improved prompt following. Qualitatively, Figure 2 shows that, on private training prompts, RSS preserves the prompt semantics without reproducing the corresponding training images, while retaining coherent generations on held-out prompts.

5.3 Further Analysis and Ablations

Alternative Representation Spaces. Our main experiments instantiate RSS using the DINOv2 [CLS] representation. To examine whether its behavior is specific to this choice, we additionally evaluate DINOv2 patch representations with different pooling strategies, Inception-V3 features, and the MAE [CLS] representation. Under a shared RSS weight, all evaluated alternatives reduce private-set training-example similarity relative to *Full-Data SFT*, although their ability to preserve generative performance varies (Figure 4). Because we do not tune λ separately for each representation, these results should not be interpreted as a ranking of their best attainable performance. Complete training dynamics, trade-off curves, and qualitative comparisons are provided in Section B.2.

Pixel-Space Diffusion Models. While our main experiments use LDMs, we further evaluate RSS on a pixel-space diffusion model under the 50/50 public/private split. RSS achieves comparable GenEval and stronger DPG performance than *Full-Data SFT*, while maintaining substantially lower private-set similarity (Figure 10). These results show that RSS can be instantiated for both latent- and pixel-space diffusion models. The complete experimental results are provided in Section B.3.

Additional Analyses and Baselines. We also provide additional experiments that examine RSS and alternative approaches in greater detail. Mixing pretraining data into standard SFT slows but does not halt the growth of training-example similarity (Figure 11 in Section B.4). Applying RSS together with data-space supervision to the entire SFT dataset improves and stabilizes generative performance, but similarity continues to increase when protected examples still receive direct data-space supervision (Section B.5). We further compare RSS with masked-pixel supervision (Zhao

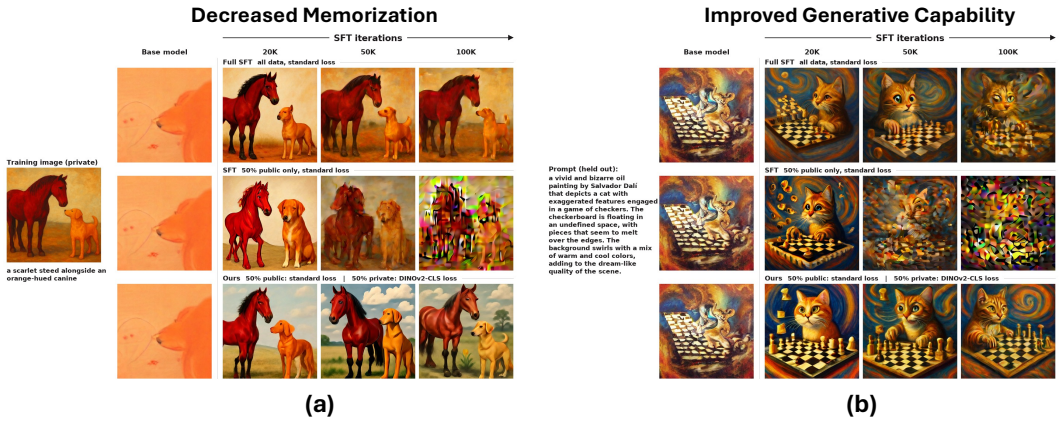


Figure 2: **Qualitative results on memorization suppression and generative capability preservation.** (a) Hybrid-Supervision SFT substantially reduces memorization of private training images compared with Full-Data SFT. (b) At the same time, it maintains stronger generative capability than Public-Only SFT, especially at later training iterations.

et al., 2026b) and DP-SGD (Dockhorn et al., 2023). As discussed in Section 4, these approaches share with RSS the high-level idea of constraining the information from individual training examples exposed to the model. However, they impose this constraint differently: masked-pixel supervision removes part of the data-space signal, whereas DP-SGD clips and perturbs per-example gradients. Our results show that masked-pixel supervision does not attain the same performance–memorization trade-off, while DP-SGD achieves low training-example similarity but substantially weaker generative performance. These results suggest that simply limiting the exposure of training data is insufficient: the semantic information in the data must also be preserved to provide a useful learning signal. RSS achieves a better balance by suppressing instance-specific details while retaining high-level perceptual and semantic information.

6 Conclusion and Discussion

We introduced Representation-Space Supervision (RSS), a training-time framework for mitigating memorization in diffusion models. Instead of directly matching protected training examples in data space, RSS supervises the model through pretrained visual representations that retain useful perceptual and semantic information while reducing instance-specific details. Across different public/private splits and training checkpoints, RSS substantially improves the performance–memorization trade-off over standard data-space supervision. Our additional experiments further show that the framework is compatible with multiple representation spaces and both latent- and pixel-space diffusion models.

Limitations. We note that RSS has several limitations. RSS is an empirical memorization-mitigation method rather than a formal privacy mechanism. Its effectiveness depends on the choice of representation and the relative weight of representation-space supervision, and our evaluation focuses primarily on supervised fine-tuning and similarity-based memorization metrics. Evaluating RSS against stronger extraction attacks, extending it to larger-scale pretraining, and developing principled methods for selecting or learning the representation space are important directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to implement and debug parts of the experimental, evaluation, and plotting code, to reformat experimental results into tables, and to assist with translation during writing. We have not used generative AI tools to develop the theoretical model or conceptual framework, formulate mathematical claims, provide ingredients for or assist in writing proofs, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, or interpret results. Additionally, we used generative AI tools to edit the manuscript for readability, draft and revise figure and table captions, and search for relevant literature. We have reviewed all AI-assisted work: AI-generated code was checked and tested by the authors, every figure and table was verified against the underlying data, and all AI-edited text was read and corrected by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI

ETHICS STATEMENT

This work studies training-example memorization in diffusion models and proposes an empirical method for reducing such behavior. Our use of the terms "public" and "private" refers to whether direct data-space supervision is permitted in our experimental protocol; it does not indicate that the evaluated datasets contain sensitive personal information. RSS may help reduce unintended reproduction of training examples, but it does not provide a formal differential-privacy guarantee and should not be treated as a substitute for appropriate data licensing, consent, access control, or privacy auditing. Conversely, techniques for measuring memorization could potentially be used to identify or extract training examples. We therefore present these evaluations for auditing and mitigation purposes. All experiments use existing research datasets described in the paper, and the authors are responsible for ensuring compliance with their applicable licenses and terms of use.

REPRODUCIBILITY STATEMENT

Code and complete implementation details will be released upon publication.

REFERENCES

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Black Forest Labs. FLUX.2: Analyzing and enhancing the latent space of FLUX – representation comparison, 2025. URL https://bfl.ai/research/representation-comparison.

Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Keshigeyan Chandrasegaran, Kyle Sargent, Suchir Agarwal, Michael Jang, Michael Poli, Juan Carlos Niebles, Justin Johnson, Jiajun Wu, and Li Fei-Fei. Gpic: A giant permissive image corpus for visual generation. *arXiv preprint arXiv:2605.30341*, 2026.

Chen Chen, Daochang Liu, and Chang Xu. Towards memorization-free diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8425–8434, 2024.

Chen Chen, Daochang Liu, Mubarak Shah, and Chang Xu. Enhancing privacy-utility trade-offs to mitigate memorization in diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8182–8191, 2025a.

Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025b.

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation. *arXiv preprint arXiv:2506.18095*, 2025c.

Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023.

Giannis Daras, Kulin Shah, Yuval Dagan, Aravind Gollakota, Alex Dimakis, and Adam Klivans. Ambient diffusion: Learning clean distributions from corrupted data. *Advances in Neural Information Processing Systems*, 36:288–313, 2023.

Giannis Daras, Alex Dimakis, and Constantinos Costis Daskalakis. Consistent diffusion meets tweedie: Training exact ambient diffusion models with noisy data. In *International Conference on Machine Learning*, pp. 10091–10108. PMLR, 2024.

Giannis Daras, Adrian Rodriguez-Munoz, Adam Klivans, Antonio Torralba, and Constantinos Daskalakis. Ambient diffusion omni: Training good models with bad data. *Advances in Neural Information Processing Systems*, 38:38064–38101, 2026.

Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *Transactions on Machine Learning Research*, 2025.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Peter Holderrieth and Ezra Erives. An introduction to flow matching and diffusion models. *arXiv preprint arXiv:2506.02070*, 2025.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

Anubhav Jain, Yuya Kobayashi, Takashi Shibuya, Yuhta Takida, Nasir Memon, Julian Togelius, and Yuki Mitsufuji. Classifier-free guidance inside the attraction basin may cause memorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12871–12879, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Jialuo Li, Wenhao Chai, Xingyu Fu, Haiyang Xu, and Saining Xie. Science-t2i: Addressing scientific illusions in image synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2734–2744. IEEE, 2025.

Tianhong Li and Kaiming He. Back to basics: Let denoising generative models denoise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36115–36125, 2026.

Xiang Li, Yixiang Dai, and Qing Qu. Understanding generalizability of diffusion models requires rethinking the hidden gaussian structure. *Advances in Neural Information Processing Systems*, 37:57499–57538, 2024.

Shanchuan Lin and Xiao Yang. Diffusion model with perceptual loss. *arXiv preprint arXiv:2401.00110*, 2023.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Zehong Ma, Ruihan Xu, and Shiliang Zhang. Pixelgen: Improving pixel diffusion with perceptual supervision. *arXiv preprint arXiv:2602.02493*, 2026.

OpenDatasets. LAION DALL-E 3 Discord Dataset. `https://huggingface.co/datasets/OpenDatasets/dalle-3-dataset`, 2023. Hugging Face dataset.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14512–14522. IEEE, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 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.

Brendan Ross, Hamidreza Kamkari, Tongzi Wu, Rasa Hosseinzadeh, Zhaoyan Liu, George Stein, Jesse Cresswell, and Gabriel Loaiza-Ganem. A geometric framework for understanding memorization in generative models. In *International Conference on Learning Representations*, volume 2025, pp. 34773–34805, 2025.

Team Seedream, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, et al. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Kulin Shah, Alkis Kalavasis, Adam Klivans, and Giannis Daras. Does generation require memorization? creative diffusion models using ambient diffusion. In *International Conference on Machine Learning*, pp. 54143–54166. PMLR, 2025.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6048–6058, 2023a.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Valerii Startsev, Alexander Ustyuzhanin, Alexey Kirillov, Dmitry Baranchuk, and Sergey Kastryulin. Alchemist: Turning public text-to-image data into generative gold. *Advances in Neural Information Processing Systems*, 38, 2026.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Shengbang Tong, Boyang Zheng, Ziteng Wang, Bingda Tang, Nanye Ma, Ellis Brown, Jihan Yang, Rob Fergus, Yann LeCun, and Saining Xie. Scaling text-to-image diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2601.16208*, 2026.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Binxu Wang and John Vastola. The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications. *Transactions on Machine Learning Research*, 2024.

Xianbang Wang, Hanhong Zhao, Yiyang Lu, Kangyang Zhou, Linrui Ma, and Kaiming He. Minit2i: A minimalist baseline for text-to-image synthesis, 2026. URL `https://peppaking8.github.io/#/post/minit2i`.

Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML 2023 workshop on structured probabilistic inference {\&} generative modeling*, 2023.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025.

Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Peng Wang, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first international conference on machine learning*, 2024.

Zekai Zhang, Xiao Li, Xiang Li, Lianghe Shi, Meng Wu, Molei Tao, and Qing Qu. Generalization of diffusion models arises with a balanced representation space. In *International Conference on Learning Representations*, volume 2026, pp. 101444–101475, 2026.

Chuyang Zhao, Yifei Song, Hongfa Wang, Jianlong Yuan, Yuan Zhang, Siming Fu, Zhineng Chen, Huilin Deng, Haoyang Huang, and Nan Duan. Perceptual flow matching for few-step generative modeling. *arXiv preprint arXiv:2607.03524*, 2026a.

Wenshuai Zhao, Zhiyuan Li, Yi Zhao, Mohammad Hassan Vali, Martin Trapp, Joni Pajarinen, Juho Kannala, and Arno Solin. Sparsely supervised diffusion. *arXiv preprint arXiv:2602.02699*, 2026b.

A RSS FOR LATENT SPACE DIFFUSION MODELS

A.1 LATENT-SPACE DIFFUSION MODELS.

Training and sampling diffusion models directly in data space with DSM can be computationally expensive, as real-world data such as images are typically high-dimensional. For improved efficiency, latent diffusion models (LDMs) (Rombach et al., 2022) therefore perform diffusion in a lower-dimensional latent space obtained through a pretrained encoder $\mathrm{Enc}(\cdot)$. In this case, eq. (3) becomes:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{LDM}}(\boldsymbol{\theta}) := \mathbb{E}_{t\sim p(t),\boldsymbol{c},\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{z}=\mathrm{Enc}(\boldsymbol{x}),\boldsymbol{z}_t\sim p_{0t}(\boldsymbol{z}_t|\boldsymbol{z})}\big[w(t)||\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{z})||_2^2\big], \quad (10)$$

where $\boldsymbol{z}$ is typically of lower dimension compared to $\boldsymbol{x}$ and $p_{0t}(\boldsymbol{z}_t|\boldsymbol{z})$ is the noising process in latent space. After sampling a latent $\boldsymbol{z}$, a decoder $\mathrm{Dec}(\cdot)$ maps it back into the original data space. The encoder and decoder are trained such that $\mathrm{Dec}(\mathrm{Enc}(\boldsymbol{x})) \approx \boldsymbol{x}$

A.2 RSS FOR LDMS

To apply transformed-space supervision to an LDM, we assume that $\Phi(\cdot)$ operates in image space. We decode both the predicted clean latent and the target latent back to the original data space, and then compute their discrepancy after applying $\Phi(\cdot)$. The resulting objective is:

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{LDM}\text{–}\Phi}(\boldsymbol{\theta}) := \mathbb{E}_{t,\boldsymbol{c},p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}),\boldsymbol{z}=\mathrm{Enc}(\boldsymbol{x}),p_{0t}(\boldsymbol{z}_t|\boldsymbol{z})}\big[w(t)\,||\Phi\big(\mathrm{Dec}(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_t;\sigma_t,\boldsymbol{c}))\big)-\Phi\big(\mathrm{Dec}(\boldsymbol{z})\big)||_2^2\big]. \quad (11)$$

In practice, we apply RSS on private dataset $\mathcal{S}_{\text{private}}$ while applying standard diffusion training on public dataset $\mathcal{S}_{\text{public}}$:

$$\begin{aligned}\min_{\boldsymbol{\theta}} \quad & \mathbb{E}_{t,\boldsymbol{c},\boldsymbol{x}\sim\mathcal{S}_{\text{public}},\boldsymbol{z}=\mathrm{Enc}(\boldsymbol{x}),\boldsymbol{z}_t\sim p_{0t}(\boldsymbol{z}_t|\boldsymbol{z})}\Big[w(t)\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{z}\|_2^2\Big] \\ & + \lambda\,\mathbb{E}_{t,\boldsymbol{c},\boldsymbol{x}\sim\mathcal{S}_{\text{private}},\boldsymbol{z}=\mathrm{Enc}(\boldsymbol{x}),\boldsymbol{z}_t\sim p_{0t}(\boldsymbol{z}_t|\boldsymbol{z})}\Big[w(t)\|\Phi\big(\mathrm{Dec}(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{z}_t;\sigma_t,\boldsymbol{c}))\big)-\Phi\big(\mathrm{Dec}(\boldsymbol{z})\big)\|_2^2\Big],\end{aligned} \quad (12)$$

B ADDITIONAL EXPERIMENTAL RESULTS

B.1 COMPLETE TRAINING DYNAMICS AND QUANTITATIVE RESULTS

We provide complete results supporting the main comparison in Section 5.2. Figure 3 reports the evolution of generative performance and training-example similarity throughout the 100K-step fine-tuning trajectory under all three public/private partitions. Across these settings, *Hybrid-Supervision SFT* maintains stable generative performance while its private-set similarity largely plateaus, whereas *Full-Data SFT* exhibits progressively increasing similarity and *Public-Only SFT* undergoes substantial late-stage degradation. Table 1 further reports the corresponding results after 100K steps and at each method's highest-GenEval checkpoint.

B.2 RSS WITH ALTERNATIVE REPRESENTATION SPACES

Our main experiments in Section 5.2 use the DINOv2 [CLS] token as the representation space for RSS. To examine whether the observed behavior of RSS is specific to this choice, we additionally consider DINOv2 patch representations with different pooling strategies, Inception-V3 features, and the MAE [CLS] token. We use the same RSS weight $\lambda = 0.1$ and training protocol for all representations, without representation-specific hyperparameter tuning. As shown in Figure 4, the evaluated alternative representation spaces also reduce private-set training-example similarity relative to *Full-Data SFT*, indicating that the memorization-mitigation effect of RSS is not specific to DINOv2 [CLS]. However, their ability to preserve generative capability varies under the shared configuration. In particular, the runs using DINOv2 Channel Pool and MAE [CLS] exhibit generative-performance degradation at rates comparable to *Full-Data SFT*, as measured by GenEval and DPG, whereas the

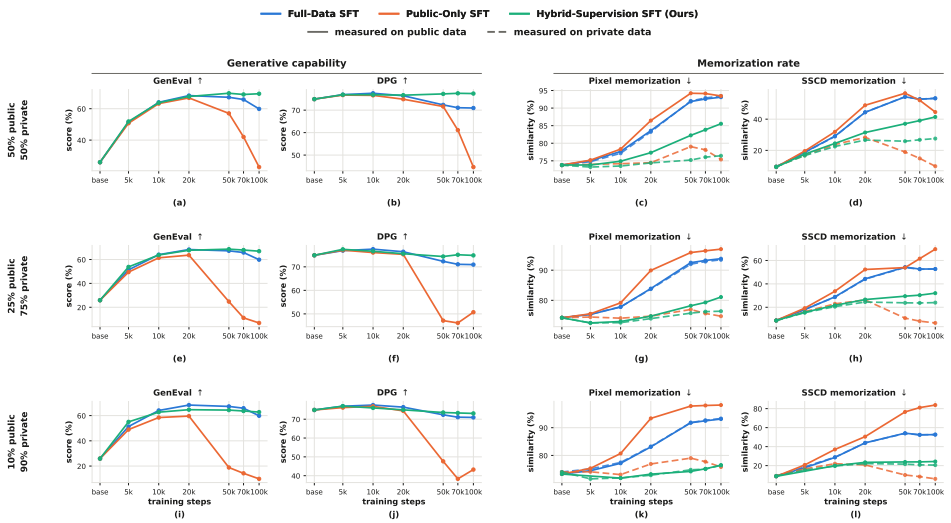


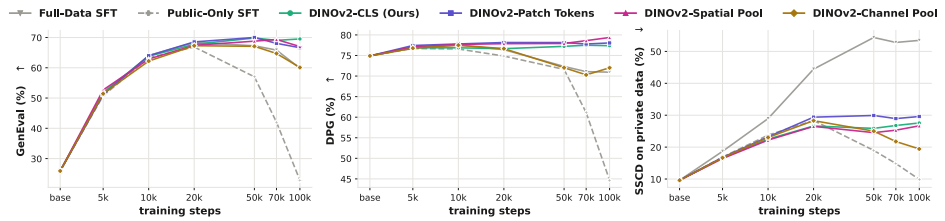
Figure 3: **Evolution of generative performance and memorization during SFT.** Each row shows the training dynamics under a different public/private split. We report GenEval and DPG for generative capability, together with pixel- and SSCD-based memorization measured separately on the public and private subsets. We compare Full-Data SFT, Public-Only SFT, and Hybrid-Supervision SFT (ours) over 100K training iterations. Full-Data SFT exhibits increasing private-set memorization over training, accompanied by degradation in generative performance at later iterations. In contrast, Hybrid-Supervision SFT maintains substantially lower private-set memorization while preserving stable generative performance across different public/private splits.

Table 1: **Quantitative comparison across different public/private splits.** We report generative capability and memorization after 100K SFT steps and at the checkpoint achieving the best GenEval score for each method. Memorization is evaluated separately on the public and private subsets. For generative metrics, the best and second-best results among the three SFT methods are highlighted in **bold** and underline, respectively. For memorization metrics, the pretrained base model is additionally included in the ranking, with public- and private-set memorization evaluated independently.

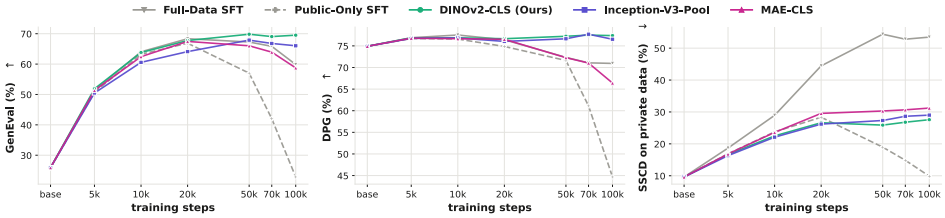
		100K Steps						Best-GenEval Checkpoint						
		Generative Capability		Memorization on Public Data		Memorization on Private Data		Step	Generative Capability		Memorization on Public Data		Memorization on Private Data	
Split	Method	GenEval ↑	DPG ↑	Pixel ↓	SSCD ↓	Pixel ↓	SSCD ↓		GenEval ↑	DPG ↑	Pixel ↓	SSCD ↓	Pixel ↓	SSCD ↓
50/50	*Base model* (no SFT)	25.9	74.9	**73.8**	**9.3**	**73.7**	**9.6**	—	25.9	74.9	**73.8**	**9.3**	**73.7**	**9.6**
	Full-Data SFT	59.9	70.9	93.1	53.3	93.4	53.5	20K	68.3	76.4	83.6	44.4	83.2	44.5
	Public-Only SFT	22.8	44.7	93.5	44.6	75.4	9.9	20K	66.9	74.9	86.5	48.9	74.5	28.3
	Hybrid-Supervision SFT (Ours)	**69.5**	**77.4**	85.5	41.4	76.4	27.6	50K	**69.8**	**77.2**	82.2	37.1	75.2	25.9
25/75	*Base model* (no SFT)	25.9	74.9	**74.2**	**8.8**	**74.2**	8.6	—	25.9	74.9	**74.2**	**8.8**	**74.2**	**8.6**
	Full-Data SFT	59.9	70.9	93.8	52.7	93.5	52.7	20K	68.3	**76.4**	83.9	44.3	83.7	44.1
	Public-Only SFT	6.9	50.7	97.0	69.7	74.7	**6.6**	20K	63.7	75.3	89.9	52.3	74.7	26.3
	Hybrid-Supervision SFT (Ours)	**67.0**	**74.9**	81.1	32.1	76.4	24.0	50K	**68.7**	74.4	78.2	29.5	75.7	23.8
10/90	*Base model* (no SFT)	25.9	74.9	**73.3**	**8.7**	**74.0**	9.1	—	25.9	74.9	**73.3**	**8.7**	74.0	**9.1**
	Full-Data SFT	59.9	70.9	93.1	52.6	93.3	52.7	20K	**68.3**	**76.4**	83.1	43.9	83.2	43.9
	Public-Only SFT	9.9	43.2	98.2	83.8	75.8	**6.1**	20K	59.6	74.4	93.4	50.5	76.9	20.7
	Hybrid-Supervision SFT (Ours)	**62.8**	**73.1**	76.5	24.3	76.4	20.6	20K	64.7	75.0	**73.3**	23.4	**72.9**	22.1

other evaluated representations achieve an overall stable generative performance. These results suggest that RSS is compatible with multiple representation spaces, although its effectiveness depends on the specific representation. We note that because we use a shared λ rather than tuning it separately for each representation, these results should be interpreted as comparisons under a common configuration rather than as a ranking of the best performance attainable with each representation. In Section B.6 we further show that the choice of λ can substantially affect the resulting performance of RSS. Representation-specific hyperparameter tuning may therefore further improve some of these results.

Additional results also support the analysis above. Figure 5 reports the evolution of the same metrics as Figure 4, but under the 25/75 public/private split. Figures 6 and 7 present the complete performance–memorization trade-offs for the evaluated representation spaces under the 25/75 and



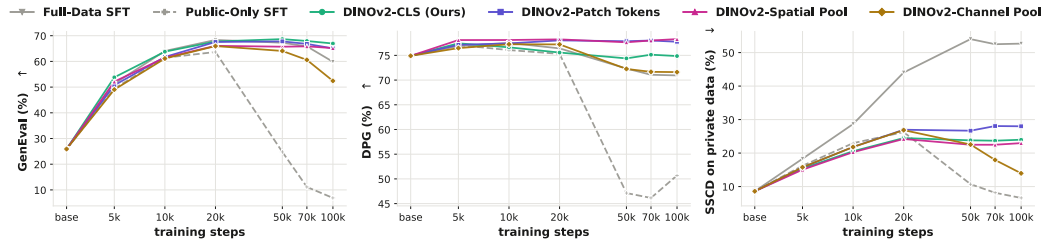
(a) DINOv2 representation variants



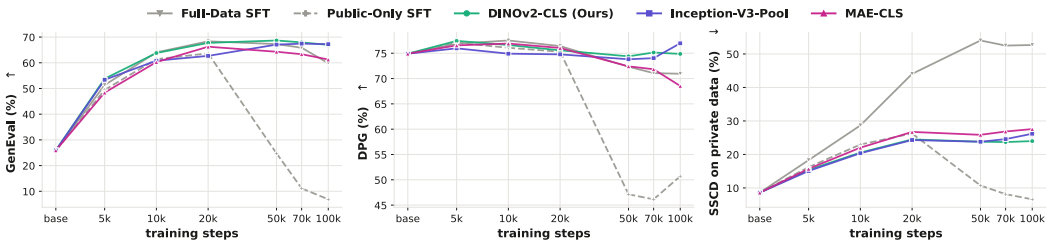
(b) Alternative visual representation models

Figure 4: **RSS with alternative representation spaces.** We compare training dynamics under the 50/50 public/private split using (a) alternative DINOv2 representations and (b) representations from alternative visual representation models including Inception-V3 and MAE.

50/50 splits, respectively. Finally, Figures 8 and 9 provide qualitative comparisons using the corresponding 100K-step checkpoints. Together, these results show that RSS is effective at mitigating memorization across multiple representation spaces, while the preservation of generative capability depends on the chosen representation.



(a) DINOv2 representation variants



(b) Alternative visual representation models

Figure 5: **RSS with alternative representation spaces.** We compare training dynamics under the 25/75 public/private split using (a) alternative DINOv2 representations and (b) representations from alternative visual representation models including Inception-V3 and MAE.

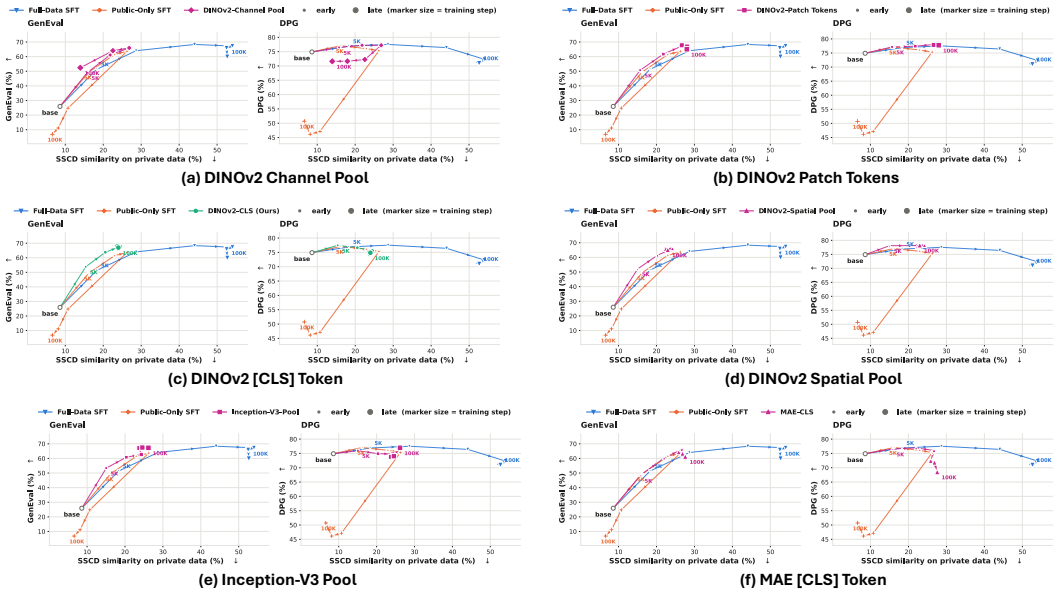


Figure 6: **RSS with alternative representation spaces.** Each panel shows the generative performance-memorization trade-offs under the 25/75 public/private split for a specific type of representation.

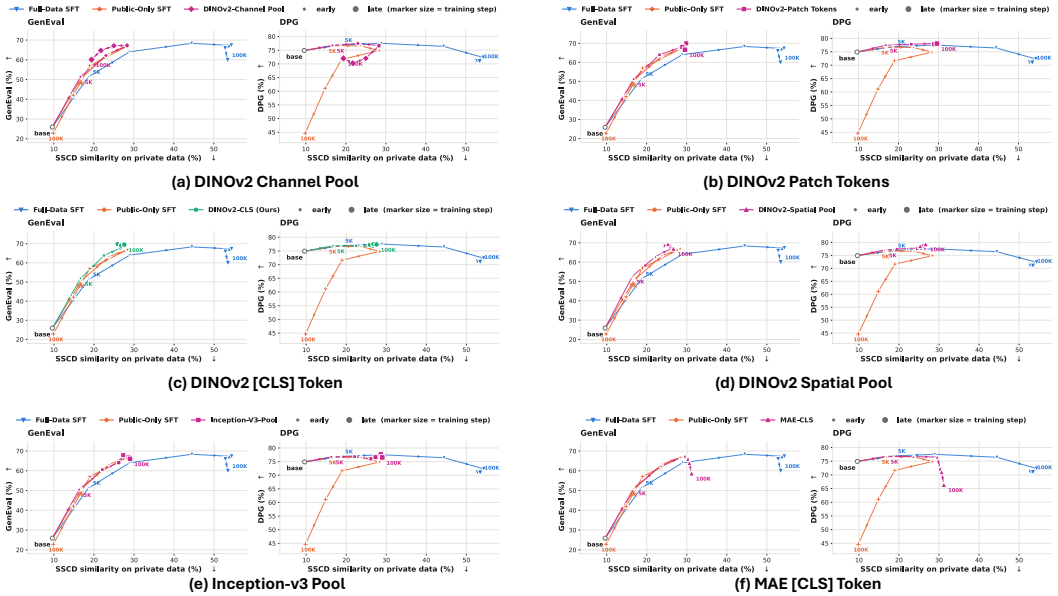


Figure 7: **RSS with alternative representation spaces.** Each panel shows the generative performance-memorization trade-offs under the 50/50 public/private split for a specific type of representation.

B.3 RSS WITH PIXEL-SPACE DIFFUSION MODELS

Our main experiments in Section 5.2 evaluate RSS using LDMs. To examine whether its effectiveness extends beyond latent-space diffusion, we additionally evaluate RSS on a pixel-space diffusion model pretrained on the same dataset with a configuration matched to that of the LDM used in our

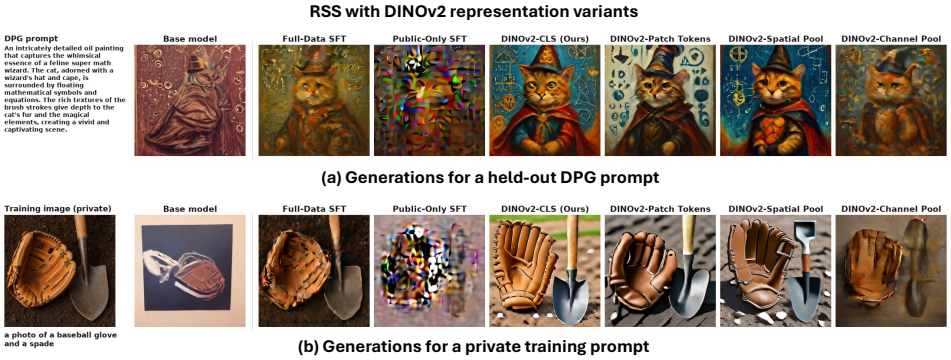


Figure 8: **Qualitative comparison of RSS using DINOv2 representation variants under the 25/75 public/private split.** (a) Generations for a held-out DPG prompt illustrate the prompt-following capability. (b) Generations conditioned on the prompt of a private training example illustrate different degrees of training-example reproduction. The private training image is shown in the leftmost column. All generations are obtained from the corresponding 100K-step checkpoints.

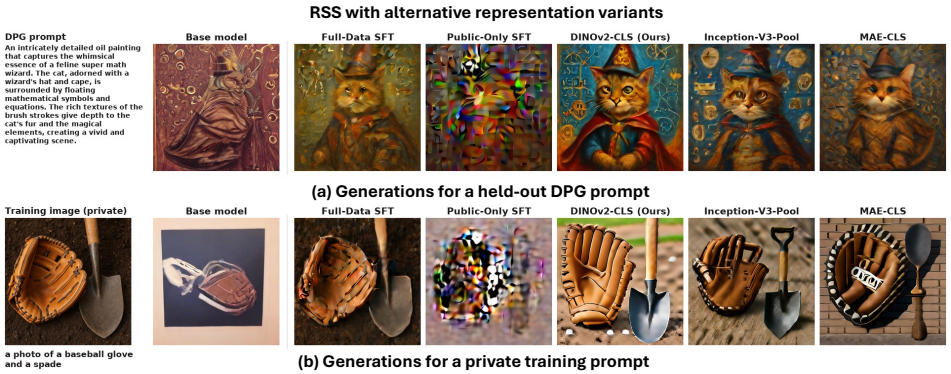


Figure 9: **Qualitative comparison of RSS using alternative representation models under the 25/75 public/private split.** (a) Generations for a held-out DPG prompt illustrate the prompt-following capability. (b) Generations conditioned on the prompt of a private training example illustrate different degrees of training-example reproduction. The private training image is shown in the leftmost column. All generations are obtained from the corresponding 100K-step checkpoints.

main experiments. We again compare *Full-Data SFT*, *Public-Only SFT*, and *Hybrid-Supervision SFT* under the 50/50 public/private split. For *Hybrid-Supervision SFT*, we use $\lambda = 10^{-4}$.

As shown in Figure 10, *Hybrid-Supervision SFT* achieves comparable GenEval and stronger DPG performance than *Full-Data SFT*, while maintaining substantially lower private-set training-example similarity. It also outperforms *Public-Only SFT* at comparable levels of similarity, indicating that RSS effectively utilizes information from the private subset. These results demonstrate that RSS can be effectively instantiated for both latent- and pixel-space diffusion models.

B.4 Effect of Mixing Pretraining Data during SFT

Our main experiments in Section 5.2 fine-tune pretrained models with the SFT dataset alone, without utilizing the pretraining data. While this choice is natural in practice, where the pretraining data is not available to users, a potential approach to mitigate memorization is to mix the SFT dataset with samples from the pretraining distribution, which may regularize fine-tuning and reduce repeated exposure to individual SFT examples. To evaluate this alternative, we fine-tune the same pretrained model using a mixture of the full SFT dataset and the pretraining dataset. Under this setting, each training batch contains approximately equal numbers of pretraining and SFT samples. Both sets of

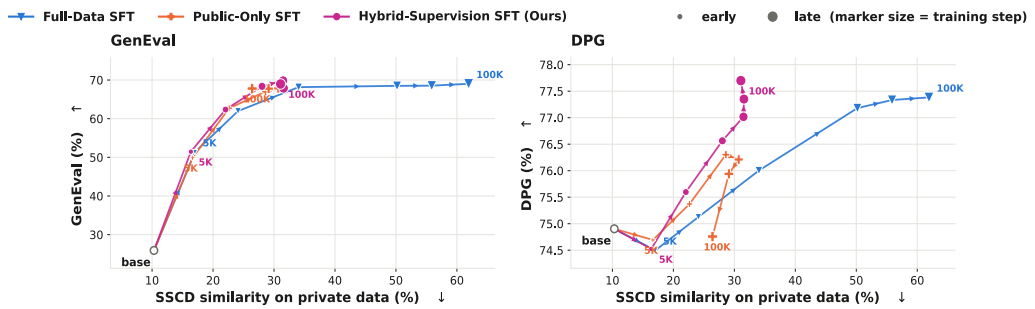


Figure 10: **RSS improves the performance–memorization trade-off on a pixel-space diffusion model.** The left and right panels show GenEval and DPG, respectively, against private-set SSCD similarity under the 50/50 public/private split.

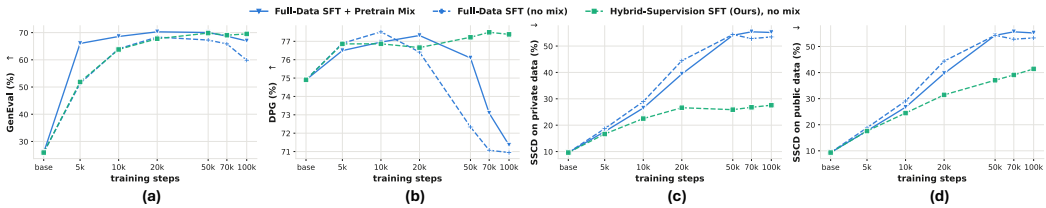


Figure 11: **Mixing pretraining data delays but does not halt the growth of memorization.** We compare *Full-Data SFT* with and without pretraining-data mixing and *Hybrid-Supervision SFT* (RSS) without mixing. In the pretraining-mix setting, each batch contains approximately equal numbers of SFT and pretraining examples, all of which receive data-space supervision. From left to right, the panels report GenEval, DPG, private-set SSCD similarity, and public-set SSCD similarity throughout training. Mixing pretraining data slows the early increase in private-set similarity and delays the degradation in generative performance, but its private-set SSCD similarity becomes comparable to that of *Full-Data SFT* without mixing by 100K steps. In contrast, *Hybrid-Supervision SFT* maintains stronger generative performance while its private-set similarity largely plateaus.

samples receive standard data-space supervision, and RSS is not applied. We refer to this setting as *Full-Data SFT + Pretrain Mix*.

As shown in Figure 11, mixing pretraining data slows both the increase in private-set training-example similarity and the degradation in generative performance relative to *Full-Data SFT*. However, private-set similarity continues to grow, and at the 100K-step checkpoint, *Full-Data SFT + Pretrain Mix* and *Full-Data SFT* exhibit similar SSCD similarity on $\mathcal{S}_{\text{private}}$. By contrast, private-set similarity under *Hybrid-Supervision SFT* largely plateaus after approximately 20K steps. These results indicate that pretraining-data mixing provides useful regularization but does not produce the sustained suppression observed with RSS over the evaluated training horizon.

B.5 Beyond Memorization Mitigation: The Regularizing Effect of RSS

Thus far, we have studied RSS primarily as a means of mitigating memorization: training data in $\mathcal{S}_{\text{private}}$ receives RSS without direct data-space supervision. Under this hybrid-supervision setting, RSS mitigates memorization while alleviating the degradation in generative performance observed after prolonged training. Interestingly, we find that RSS can also improve generative performance when it is combined with, rather than used to replace, data-space supervision.

Specifically, we train the pretrained model on the entire SFT dataset using both RSS and data-space supervision. At each training step, we randomly partition the batch into two equal subsets: one receives standard data-space supervision, while the other receives RSS. Because the partition is resampled throughout training, each example may receive either form of supervision across repeated visits. We refer to this setting as *Full-Data SFT with RSS*. We train it for 200K steps so that its expected number of data-space-supervised example updates matches that of *Full-Data SFT* trained

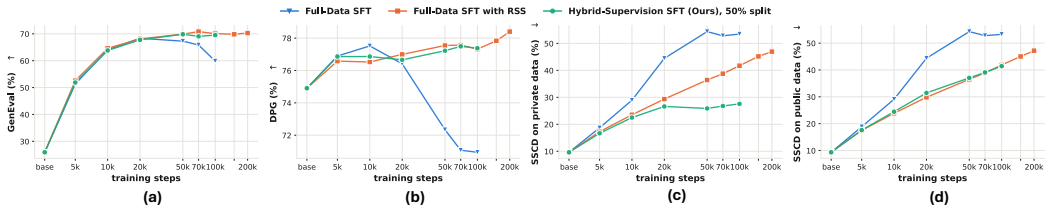


Figure 12: **The regularizing effect of RSS.** Training dynamics under the 50/50 public/private split. Panels (a)–(d) report GenEval, DPG, private-set SSCD similarity, and public-set SSCD similarity, respectively. Combining RSS with data-space supervision stabilizes and improves generative performance and slows, but does not halt, the growth of training-example similarity (memorization).

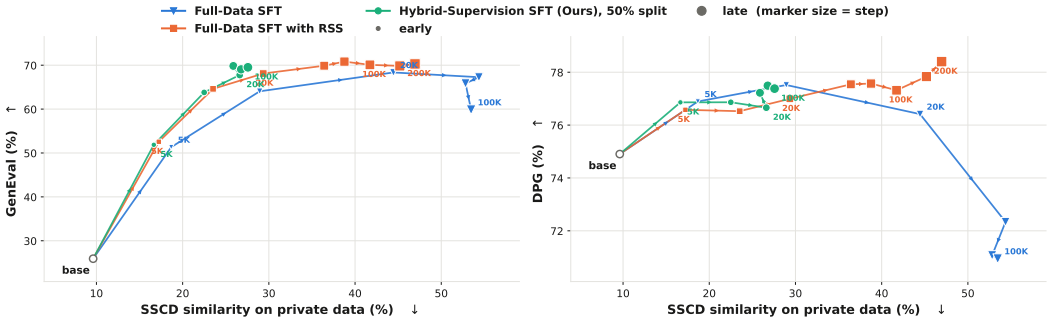


Figure 13: **RSS improves the performance–memorization trade-off when combined with data-space supervision.** The Left and right panels show GenEval and DPG, respectively, against private-set SSCD similarity under the 50/50 public/private split.

for 100K steps. As shown in Figures 12 and 13, combining RSS with data-space supervision slows the growth of training-example similarity relative to using data-space supervision alone. It also improves the best-achievable generative performance and avoids the late-stage performance degradation observed under *Full-Data SFT*, even over the extended 200K-step trajectory. These results suggest that RSS provides a useful regularizing effect when combined with data-space supervision.

Nevertheless, training-example similarity continues to increase under *Full-Data SFT with RSS*. Therefore, combining the two forms of supervision delays but does not halt memorization. This result further supports our design choice of not to impose direct data-space supervision on training samples that are to be protected from memorization.

B.6 Effect of the RSS Weight λ

As discussed in Section 3.2.2, λ controls the relative strength of RSS when it is combined with the standard diffusion objective. To study its effect on the performance of RSS, we evaluate $\lambda \in \{10^{-1}, 10^{-3}, 10^{-4}\}$ on the pixel-space diffusion model under the 50/50 public/private split.

As shown in Figure 14, the choice of λ substantially affects the resulting performance–memorization trade-off. The largest weight, $\lambda = 10^{-1}$, produces the lowest private-set similarity but exhibits considerable late-stage degradation in GenEval and DPG. In contrast, $\lambda = 10^{-3}$ and $\lambda = 10^{-4}$ preserve stronger generative performance while maintaining substantially lower private-set similarity than *Full-Data SFT*. These results illustrate that λ controls the balance between generative performance and memorization mitigation and should therefore be tuned in practical deployment.

Loss normalization. Lastly, we note that in practice, we apply a batch-wise stop-gradient normalization to the RSS loss before scaling it by λ. For a private minibatch containing B examples, let $h(\boldsymbol{x}) := \Phi\big(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c})\big) - \Phi(\boldsymbol{x}) \in \mathbb{R}^d$ for pixel-space diffusion models. We compute the RSS

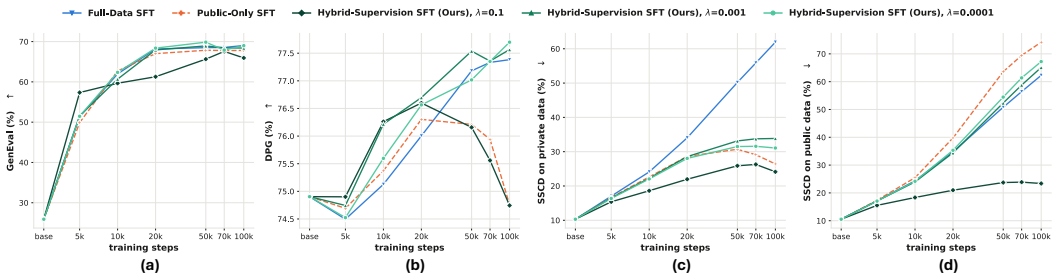


Figure 14: **Effect of RSS Weight λ.** Training dynamics under the 50/50 public/private split, where RSS is tested with different λ. Panels (a)–(d) report GenEval, DPG, private-set SSCD similarity and public-set SSCD similarity, respectively.

loss for the i-th sample $\boldsymbol{x}_i$ as:

$$\ell_i = \frac{1}{d}||h(\boldsymbol{x}_i)||_2^2, \qquad \mathcal{L}_i^{\text{RSS}} = \lambda \cdot w(t_i) \cdot \frac{\ell_i}{\text{sg}\left(\frac{1}{B}\sum_j w(t_j)\ell_j\right) + \varepsilon}, \tag{13}$$

where sg is the stop-gradient operation and ε is a small constant used for numerical stability.

C Comparison with Masked-Pixel Supervision

Experimental setup: masked pixel loss. A natural question is whether our representation-level feature loss can be replaced by simply weakening pixel-level supervision on the private data. To test this, we introduce a *masked pixel loss* baseline that follows the same training setup as our method, but replaces the private-side feature loss with a masked reconstruction loss, i.e., $\Phi(\cdot)$ is instantiated as random masks. All runs fine-tune the same latent diffusion model, initialized from the same GPIC-pretrained checkpoint (1M steps), on a 50% public / 50% private data split, with a learning rate of 10^{-4} and an effective batch size of 512. The public data are always trained with the standard flow-matching loss.

For the private data, the masked baseline decodes the LDM's prediction into pixel space (256×256) and computes the MSE against the training image $\boldsymbol{x}$ over only a random subset of pixels. The mask is sampled independently for each pixel and redrawn at every training step, and the loss is averaged over the unmasked pixels. We consider two masking ratios, removing 40% and 80% of the pixels. For a controlled comparison, we normalize the private-side loss by its stop-gradient batch mean and set its weight to λ=0.1 following eq. (13), matching the relative contribution of the feature loss in our method. Thus, the key difference is the type of supervision provided by the private data: masked pixel values versus visual representations.

We compare against Full-Data SFT (standard loss on all data), Public-Only SFT (public data only), and our Hybrid-Supervision SFT. All methods are trained for up to 100K steps with the same optimizer and evaluated at the same checkpoints using GenEval, DPG-Bench, and SSCD memorization probes constructed separately from the public and private splits.

Results. Masking the pixel loss does not prevent memorization. As shown in Figure 15, private-half SSCD for the masked runs rises to approximately 39 by 30K steps and then plateaus around 36, reaching 36.6 and 36.0 for the 40% and 80% mask ratios at 70K steps, respectively. These values are substantially higher than that of our method (26.8), although lower than Full-Data SFT (52.8). A similar pattern appears on the public half: the masked runs reach SSCD scores of 56.3 and 56.1, close to the pixel-supervised baselines (52.8 for Full-Data SFT and 52.1 for Public-Only SFT) and well above our method (39.1).

Masking also fails to preserve generation quality. At 70K steps, GenEval drops to 59.1/58.9 and DPG to 68.3/69.7 for the 40%/80% mask ratios, compared with 65.8/71.1 for Full-Data SFT and 69.1/77.5 for our method. The masked runs therefore show a similar degradation in generation quality to Full-Data SFT, rather than maintaining quality as our method does.

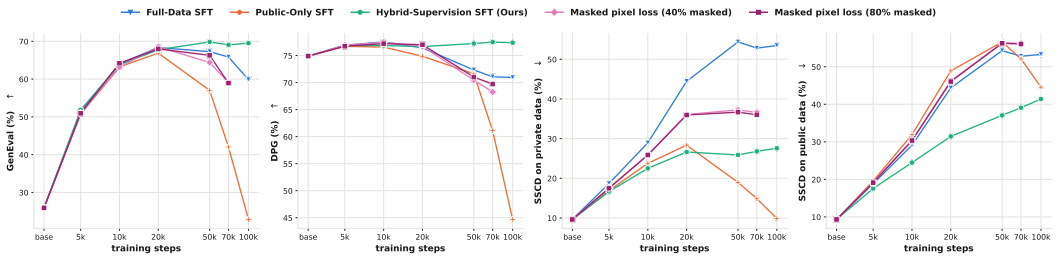


Figure 15: **Masked pixel loss vs. our method over fine-tuning.** GenEval and DPG measure generation quality (higher is better); SSCD measures memorization as the similarity between images generated from training captions and the corresponding training images, on the private and public halves of the 50%/50% split (lower is better). The masked pixel loss (40% or 80% of pixels masked, redrawn every step) does not protect the private data and loses quality as training proceeds, while our method keeps both memorization and quality stable.

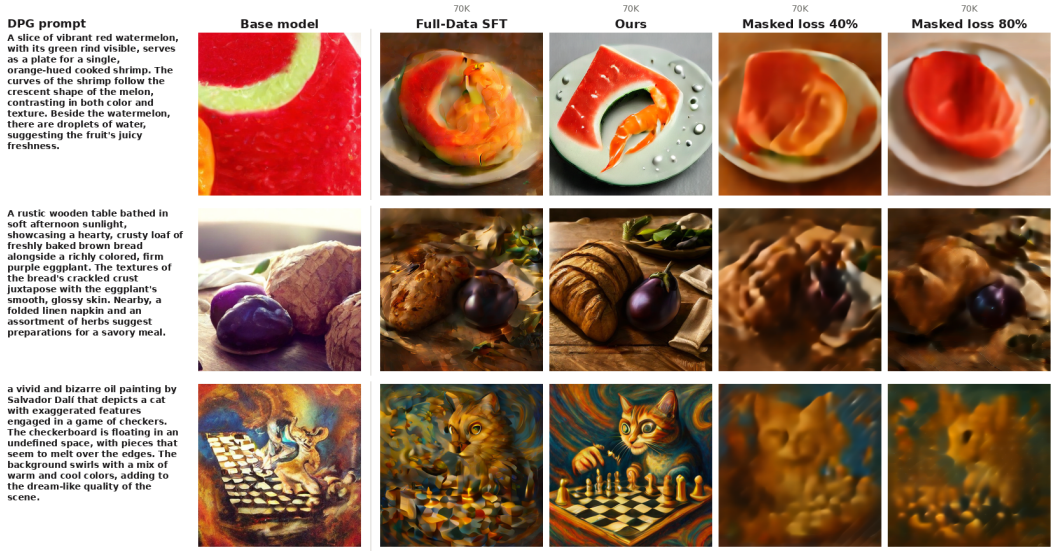


Figure 16: **Qualitative comparison on DPG-Bench prompts at 70K steps.** Both masked-pixel-loss variants produce blurry images and omit prompt-specified objects (e.g., the shrimp, loaf of bread, and checkerboard). Full-Data SFT largely preserves the overall layout but produces smeared textures, whereas our method renders the requested objects with sharper details and better prompt fidelity.

Interestingly, the two mask ratios behave nearly identically throughout training, typically differing by no more than one SSCD point. This is expected because the mask is resampled at every step: with repeated exposure to the same training image, each pixel is eventually supervised many times. Masking therefore reduces the pixel-level supervision available at any single step, but does not remove this information over the course of training.

The qualitative results show the same pattern. On DPG-Bench prompts, the masked models produce blurrier details and more often omit prompt-specified objects than our method (Figure 16). The memorization behavior is even clearer when querying with captions from private training images (Figure 17): like Full-Data SFT, both masked runs reproduce the corresponding private training images nearly verbatim, including their object layouts and placements, whereas our method generates a different image that still follows the same caption.

Overall, simply weakening pixel-level supervision does not reproduce the benefits of our feature loss: masking neither prevents memorization of the private data nor preserves generation quality, whereas our method achieves both.

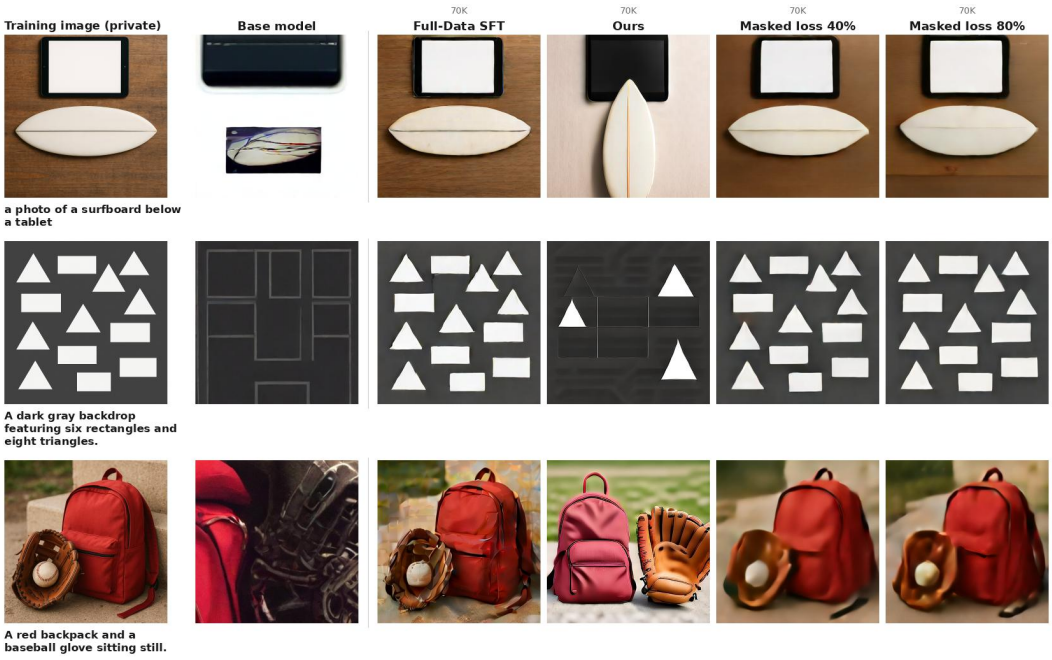


Figure 17: **Querying with private training captions at 70K steps** (50% public / 50% private split). Each row shows a private training image and the images generated from its caption. Full-Data SFT and both masked-pixel-loss variants reproduce the training image almost exactly (the tablet above the surfboard on wood, the arrangement of rectangles and triangles, the red backpack with the baseball glove and ball leaning against it), whereas our method follows the caption with a newly composed image.

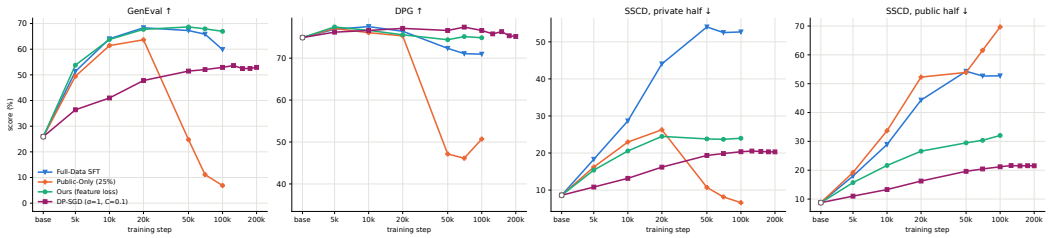


Figure 18: **DP-SGD versus the 25% public-data baselines.** Generation quality (GenEval and DPG; higher is better) and memorization (SSCD on the private 75% and public 25% subsets of the fine-tuning data; lower is better) across fine-tuning steps. DP-SGD is applied to the private subset with σ=1 and C=0.1, corresponding to a nominal privacy budget of $\varepsilon \approx 9.0$ at 100K steps with δ=10^{-5}. Dashed lines indicate the performance of the base model before fine-tuning.

D Comparison with DP-SGD

Experimental setup: DP-SGD. We also compare against differentially private fine-tuning, a standard approach for limiting the influence of any individual private training example Dockhorn et al. (2023). The DP-SGD baseline uses the same latent diffusion model (1.12B trainable parameters), GPIC-pretrained initialization, and fine-tuning data as the other methods, with the 25% public / 75% private split. All samples use the standard flow-matching objective, with DP-SGD applied to the private samples instead of our feature loss.

Each training step contains 128 public and 384 private samples, for an effective batch size of 512. Public samples contribute standard gradients. For each private sample, we clip the gradients of the 122 linear layers separately to $C/\sqrt{122}$, ensuring that the combined per-sample norm is at most

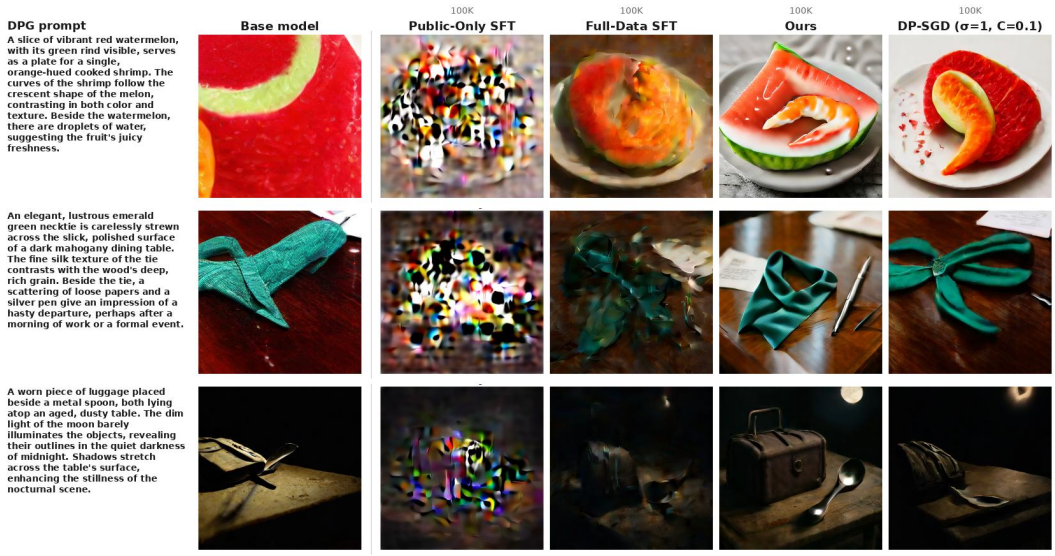


Figure 19: **Qualitative comparison with DP-SGD on DPG-Bench prompts at 100K steps** (25% public / 75% private split). DP-SGD (σ=1, C=0.1) largely preserves the visual quality of the base model but exhibits weaker prompt adherence: it merges the shrimp with the melon, renders the necktie as a bow while omitting the pen, and fails to render recognizable luggage. In contrast, our method faithfully renders all requested objects, while Public-Only SFT has collapsed to noise by 100K steps.

C=0.1. Gaussian noise with standard deviation σC, with σ=1, is then added to the summed private gradient once per step before averaging over the batch. The remaining optimization settings match the other runs: AdamW with a learning rate of 10^{-4}, EMA of 0.9999, and global gradient-norm clipping at 1.0 as post-processing. Normalization parameters and the text positional embedding, together accounting for only 0.05% of the model parameters, are frozen at their pretrained values because their per-sample gradients are not supported by our clipping implementation.

Using an RDP accountant, this configuration gives a nominal privacy budget of $\varepsilon \approx 9.0$ after 100K steps (δ=10^{-5}), increasing to 13.7 at 200K steps. We refer to these values as nominal because the training data are consumed as a shuffled stream rather than using the Poisson sampling assumed by the accountant.

Comparison with DP-SGD. Figure 18 compares our method with DP-SGD under the same 25%/75% public/private split. Fine-tuning on the public split alone severely overfits: GenEval peaks at 63.7 after 20K steps but subsequently collapses to 6.9 by 100K steps, with DPG dropping to 50.7. Meanwhile, memorization of the public subset increases to 69.7. The low SSCD on the private subset at the end of training (6.6) therefore reflects model collapse rather than meaningful privacy protection.

DP-SGD (noise multiplier σ=1, per-sample gradient norm clipping threshold C=0.1) prevents this collapse and maintains DPG near the base-model level (75–77 vs. 74.9), but learns substantially more slowly. GenEval increases from 36.4 at 5K steps to 52.9 at 100K steps and remains near 53 through 200K steps, approximately 14 points below our method (67.0 at 100K steps). DP-SGD also achieves the lowest memorization among the non-collapsed models. However, memorization on its private and public subsets differs by only about one point, suggesting that DP-SGD reduces memorization broadly by suppressing the overall learning rate rather than preferentially reducing memorization of private examples.

In contrast, our method achieves substantially higher generation quality, improving GenEval by approximately 14 points over DP-SGD, while incurring only moderately higher memorization on the private subset (24.0 vs. 20.3). This result highlights a more favorable utility–memorization trade-off:

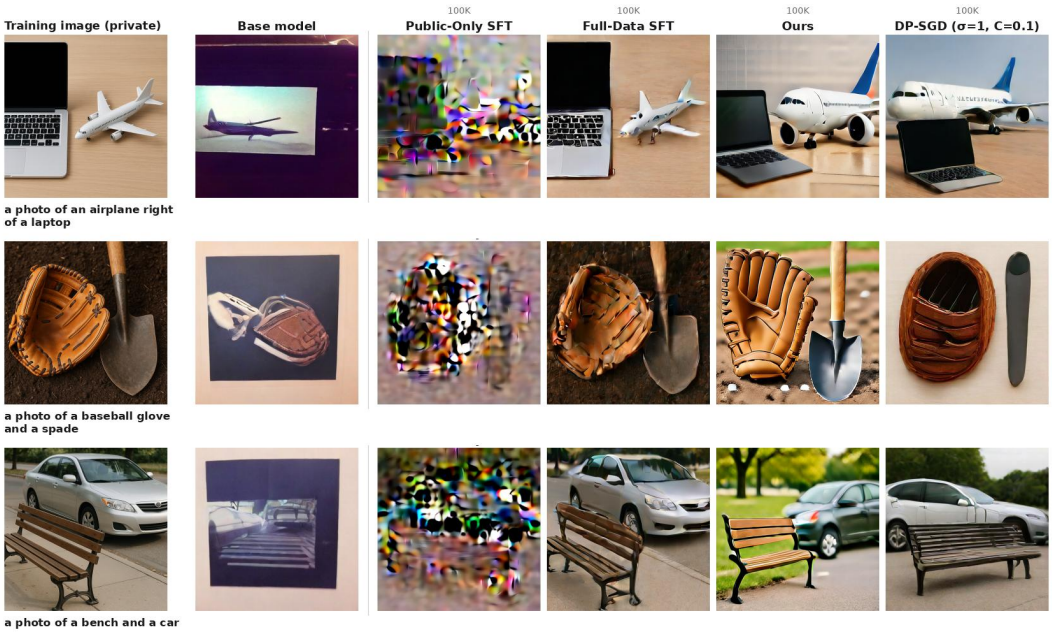


Figure 20: **Querying with private training captions at 100K steps** (25% public / 75% private split). Each row shows a private training image and the images generated from its caption. Full-Data SFT reproduces the training photo almost exactly (the laptop and toy airplane on wood, the glove and spade on soil, the bench in front of the silver sedan), whereas our method and DP-SGD follow the caption with newly composed images; Public-Only SFT has collapsed.

our method preserves substantially more learning capacity while keeping private-data memorization close to the level achieved by DP-SGD.

Qualitative comparison with DP-SGD. Figure 19 compares generations on DPG-Bench prompts after 100K fine-tuning steps. DP-SGD preserves the visual quality of the base model but often omits or merges prompt-specified objects, suggesting that the injected noise limits its ability to learn new compositional capabilities from the private data. In contrast, our method renders the requested objects more faithfully while maintaining low memorization of the private data. Public-Only SFT severely overfits and collapses to noise by 100K steps.

Querying with captions from private training images (Figure 20) reveals the other side of this trade-off. Full-Data SFT reproduces the private training images nearly verbatim, whereas both DP-SGD and our method generate novel images that follow the captions without copying the training examples. However, DP-SGD avoids memorization at the cost of learning little from the private data, while our method avoids memorization while still acquiring the compositional capabilities enabled by fine-tuning.

E Theoretical Analysis

In this section we provide the proofs for Proposition 1.

Proof of Proposition 1. We first simplify the linear transformed-space loss as follows:

$$\mathbb{E}_{t,\boldsymbol{c},p_{\text{data}}(\boldsymbol{x}|\boldsymbol{c}),p_{0t}(\boldsymbol{x}_t|\boldsymbol{x})}\left[w(t)\,||\boldsymbol{A}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{A}\boldsymbol{x}||_2^2\right] \quad (14)$$

$$=\mathbb{E}_{t,\boldsymbol{c},p_t(\boldsymbol{x}_t|\boldsymbol{c}),p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}\left[w(t)\,||\boldsymbol{A}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{A}\boldsymbol{x}||_2^2\right] \quad (15)$$

$$=\mathbb{E}_{t,\boldsymbol{c},p_t(\boldsymbol{x}_t|\boldsymbol{c}),p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}\left[w(t)\,(||\boldsymbol{A}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})||_2^2-2(\boldsymbol{A}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c}))^T\boldsymbol{A}\boldsymbol{x}+||\boldsymbol{A}\boldsymbol{x}||_2^2)\right] \quad (16)$$

$$=\mathbb{E}_{t,\boldsymbol{c},p_t(\boldsymbol{x}_t|\boldsymbol{c})}\left[w(t)\,(||\boldsymbol{A}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t;\sigma_t,\boldsymbol{c})-\boldsymbol{A}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t,\boldsymbol{c})}[\boldsymbol{x}]||_2^2)\right]+C, \quad (17)$$

where C is a constant independent of $\boldsymbol{\theta}$. Note the equivalent loss eq. (17) can be minimized for each fixed $(\boldsymbol{x}_t, \sigma_t, \boldsymbol{c})$, and the corresponding pointwise minimizers must satisfy:

$$\boldsymbol{A}(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c}) - \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}]) = 0 \tag{18}$$

$$\Leftrightarrow \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c}) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}] + (\boldsymbol{I} - \boldsymbol{P})\tilde{\boldsymbol{v}}, \tag{19}$$

where $\boldsymbol{P} = \boldsymbol{V}\boldsymbol{V}^T$ is the projection operator onto the row space of $\boldsymbol{A}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times d}$ contains the right singular vectors of $\boldsymbol{A}$, and $\tilde{\boldsymbol{v}}$ is an arbitrary vector. Note that eq. (19) can be further simplified as:

$$\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}] + (\boldsymbol{I} - \boldsymbol{P})\tilde{\boldsymbol{v}} = \boldsymbol{P}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}] + (\boldsymbol{I} - \boldsymbol{P})(\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}] + \tilde{\boldsymbol{v}}) \tag{20}$$

$$= \boldsymbol{P}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}] + (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{v}, \tag{21}$$

where $\boldsymbol{v}$ is an arbitrary vector. Next, we analyze gradient descent by fixing $(\boldsymbol{x}_t, \sigma_t, \boldsymbol{c})$ and treating the denoiser prediction at this input as an independent optimization variable.

Let $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c})$ be an independent optimization variable and suppose it has initialization $\mathcal{D}_{\text{init}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c})$, we consider solving the following optimization using gradient descent with a constant learning rate η:

$$\mathcal{L}(\boldsymbol{m}) := ||\boldsymbol{A}\boldsymbol{m} - \boldsymbol{A}\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}]||_2^2 \tag{22}$$

$$= ||\boldsymbol{A}\boldsymbol{m} - \boldsymbol{A}\boldsymbol{e}||_2^2, \tag{23}$$

where we let $\boldsymbol{m}$ and $\boldsymbol{e}$ denote $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c})$ and $\mathbb{E}_{p_{\text{data}}(\boldsymbol{x}|\boldsymbol{x}_t, \boldsymbol{c})}[\boldsymbol{x}]$ respectively for simplicity, and the positive scalar $w(t)$ is absorbed into the learning rate. Note $\nabla_{\boldsymbol{m}}\mathcal{L}(\boldsymbol{m}) = 2\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{m} - \boldsymbol{e})$, therefore the gradient descent update is:

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_k - 2\eta\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{m}_k - \boldsymbol{e}) \tag{24}$$

$$\Rightarrow \boldsymbol{m}_{k+1} = (\boldsymbol{I} - 2\eta\boldsymbol{A}^T\boldsymbol{A})^{k+1}\boldsymbol{m}_{\text{init}} + \sum_{i=0}^{k}(\boldsymbol{I} - 2\eta\boldsymbol{A}^T\boldsymbol{A})^i 2\eta\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{e}, \tag{25}$$

where $\boldsymbol{m}_{\text{init}} := \mathcal{D}_{\text{init}}(\boldsymbol{x}_t; \sigma_t, \boldsymbol{c})$ and k is the index of gradient descent steps. Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}_{\text{full}}^T$ be the *full* singular value decomposition, where $\boldsymbol{V}_{\text{full}} = [\boldsymbol{V}|\boldsymbol{V}_\perp]$, where $\boldsymbol{V} \in \mathbb{R}^{N \times d}$ contains the right singular vectors of $\boldsymbol{A}$ associated with nonzero singular values and the columns of $\boldsymbol{V}_\perp \in \mathbb{R}^{N \times (N-d)}$ form an orthonormal basis for $\text{span}(\boldsymbol{V}_\perp) = \ker(\boldsymbol{A})$. Then we have:

$$(\boldsymbol{I} - 2\eta\boldsymbol{A}^T\boldsymbol{A})^{k+1} = (\boldsymbol{V}_{\text{full}}(\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})\boldsymbol{V}_{\text{full}}^T)^{k+1} \tag{26}$$

$$= \boldsymbol{V}_{\text{full}}(\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^{k+1}\boldsymbol{V}_{\text{full}}^T. \tag{27}$$

Let λ_{max} be the largest singular value of $\boldsymbol{A}$, suppose $0 < \eta < \frac{1}{2\lambda_{\text{max}}^2}$, then we have:

$$\lim_{k\to\infty} \boldsymbol{V}_{\text{full}}(\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^{k+1}\boldsymbol{V}_{\text{full}}^T = \boldsymbol{V}_\perp\boldsymbol{V}_\perp^T \tag{28}$$

$$= \boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^T. \tag{29}$$

Similarly, we have:

$$\lim_{k\to\infty} \sum_{i=0}^{k}(\boldsymbol{I} - 2\eta\boldsymbol{A}^T\boldsymbol{A})^i 2\eta\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{e} \tag{30}$$

$$= \boldsymbol{V}_{\text{full}}\left[\lim_{k\to\infty} \sum_{i=0}^{k}(\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})^i 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}\right]\boldsymbol{V}_{\text{full}}^T\boldsymbol{e}. \tag{31}$$

Since $\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}$ is a diagonal matrix, we can analyze its diagonal entries independently. Let λ_p be the p-th *nonzero* diagonal entry of $\boldsymbol{\Lambda}$, i.e., the p-th *nonzero* singular value of $\boldsymbol{A}$, then the p-th diagonal entry of $\sum_{i=0}^{k}(\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})$ is:

$$\sum_{i=0}^{k}(\boldsymbol{I} - 2\eta\boldsymbol{\Lambda}^T\boldsymbol{\Lambda})_{pp}^i = \frac{1 - (1 - 2\eta\lambda_p^2)^{k+1}}{1 - (1 - 2\eta\lambda_p^2)} \tag{32}$$

$$= \frac{1 - (1 - 2\eta\lambda_p^2)^{k+1}}{2\eta\lambda_p^2}. \tag{33}$$

Since $0 < \eta < \frac{1}{2\lambda_{\max}^2}$, we have $0 < 1 - 2\eta\lambda_p^2 < 1$, which implies:

$$\lim_{k\to\infty} \sum_{i=0}^{k} (\boldsymbol{I} - 2\eta \boldsymbol{\Lambda}^T \boldsymbol{\Lambda})_{pp}^i = \frac{1}{2\eta\lambda_p^2}, \tag{34}$$

and the p-th nonzero diagonal entry of the bracketed matrix in eq. (31) converges to 1. For directions corresponding to zero singular values, the same entry is zero for every k. Therefore, we have:

$$\lim_{k\to\infty} \sum_{i=0}^{k} (\boldsymbol{I} - 2\eta \boldsymbol{A}^T \boldsymbol{A})^i 2\eta \boldsymbol{A}^T \boldsymbol{A}\boldsymbol{e} = \boldsymbol{V}\boldsymbol{V}^T \boldsymbol{e}. \tag{35}$$

Lastly, substitute eqs. (29) and (35) back to eq. (25), we have:

$$\lim_{k\to\infty} \boldsymbol{m}_k = (\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^T)\boldsymbol{m}_{\text{init}} + \boldsymbol{V}\boldsymbol{V}^T \boldsymbol{e} \tag{36}$$

$$= (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{m}_{\text{init}} + \boldsymbol{P}\boldsymbol{e}. \tag{37}$$

This completes the proof. □