Differentially Private Fine-Tuning of Diffusion Models

Yu-Lin Tsai[*]
National Yang Ming Chiao University

Yizhe Li[*]
Xi'an Jiaotong University

Chia-Mu Yu
National Yang Ming Chiao University

Xuebin Ren
Xi'an Jiaotong University

Po-Yu Chen [†]
JPMorganChase

Zekai Chen
Standard Model Biomedicine

Francois Buet-Golfouse
AIML Global Markets, Barclays

Abstract

Generative AI models, particularly diffusion models (DMs), have demonstrated exceptional capabilities in high-quality image synthesis. However, their large memorization capacity raises significant privacy concerns, especially when trained on sensitive datasets. This paper introduces ***DP-LoRA****, a surprisingly simple yet effective framework for differentially private fine-tuning of latent diffusion models (LDMs) using Low-Rank Adaptation (LoRA). By fine-tuning only a small subset of parameters, DP-LoRA achieves state-of-the-art (SoTA) performance in privacy-preserving image generation while significantly improving the privacy-utility trade-off. DP-LoRA leverages pre-trained LDMs and integrates LoRA modules into attention blocks and projection layers, enabling parameter-efficient fine-tuning under Differential Privacy (DP) constraints. Extensive experiments on benchmarks such as CelebA-HQ demonstrate that DP-LoRA outperforms existing methods, achieving competitive Fréchet Inception Distance (FID) scores with strict privacy budgets (e.g.,* $\epsilon \leq 10$*). Additionally, we provide a comprehensive analysis of the impact of LoRA rank, noise multiplicity, and trainable components on model performance. Our results highlight the potential of parameter-efficient techniques to scale privacy-preserving generative models to real-world applications, paving the way for safer deployment of diffusion models in sensitive domains. Our codes are available at* https://github.com/EzzzLi/DP-LORA.

1. Introduction

As generative AI evolves, advanced models raise serious concerns about data privacy [15, 57]. Neural networks can unintentionally reveal training data [6, 59], fueling research on privacy-preserving methods [10] that maintain strong model utility [54]. A solution is *Differential Privacy* (DP) [17, 18], a rigorous framework for protecting individual data points during training. In particular, *Differential Privacy Stochastic Gradient Descent* (DP-SGD) [1] modifies standard SGD by clipping gradients and injecting noise, providing formal privacy guarantees for each training sample.

Recent generative approaches, such as *diffusion models* (DMs) [4, 13, 47, 48, 51], have demonstrated impressive capabilities in synthesizing high-quality images and achieving robust performance on various tasks. However, their large memorization capacity poses heightened privacy risks [6, 15, 27], particularly when trained on sensitive data [3, 7, 46]. Unlike generative adversarial networks (GANs) [22, 54, 55, 58], DMs break down the generation process into iterative steps, which aligns well with DP protocols that rely on incremental noise addition [14].

Nevertheless, the unique denoising architecture of DMs demands tailored strategies [14, 20, 36, 37] to integrate DP without undermining learning capacity. Dockhorn et al. [14] first explored DP-SGD [1] for diffusion models, yielding limited utility on CIFAR10 and CelebA. Later, Ghalebikesabi et al. [20] improved performance via public pretraining

[*] Equal contribution.

[†] **Disclaimer** - The authors' views are their own. The authors' employers past and present make no representation and warranty whatsoever and disclaim all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

and private fine-tuning, while recent work [37] extended the vanilla diffusion scheme [53] to *latent diffusion models* (LDMs) [48], achieving competitive results. However, there remains a gap in optimizing the *privacy-utility trade-off*, especially regarding parameter efficiency and model scalability. Reducing the number of trainable parameters can help maintain DP guarantees by limiting the information learned from sensitive data. In non-private scenarios, *parameter-efficient fine-tuning* (PEFT) techniques (*e.g*. LoRA [28]) have been proposed to address storage and compute constraints [12, 63]. Here, we conduct a comprehensive study to develop a strong, accurate, and parameter-efficient strategy that optimizes the privacy-utility trade-off under DP constraints.

Our contributions are: **1)** We show that our parameter-efficient fine-tuning approach, DP-LoRA, achieves state-of-the-art (SoTA) performance in DP image synthesis, surpassing previous baselines on standard benchmarks. **2)** We thoroughly investigate parameter-efficient training under DP constraints, demonstrating that a small subset of trainable parameters can still provide competitive results. **3)** DP-LoRA enables a modular design where a large pre-trained foundation model can be rapidly adapted to diverse downstream tasks with minimal overhead, facilitating faster and more resource-efficient training of private diffusion models.

2. Preliminary

2.1. Differential Privacy

Differential privacy [17, 18] is a widely adopted defense against membership inference attacks, where adversaries attempt to pinpoint individuals or groups in the training data.

Definition 1 *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if, for any two adjacent inputs $d, d' \in \mathcal{D}$ and any set $S \subset \mathcal{R}$,*

$$\mathbb{P}(\mathcal{M}(d) \in S) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(d') \in S) + \delta. \tag{1}$$

Here, ϵ is privacy budget (higher values imply weaker privacy guarantees) and δ bounds the probability of a privacy breach.

2.2. DPSGD

Neural networks are commonly privatized using Differentially Private Stochastic Gradient Descent (DP-SGD) [1] or variants like DP-Adam [40]. During each training iteration, gradients for each sample are clipped, and Gaussian noise is injected. Formally, let $l_i(f) := L(f, x_i, y_i)$ be the loss function with model parameters $f \in \mathbb{R}^p$, input features x_i, and label y_i. The clipping function $\texttt{clip}_C(v) : v \mapsto \min\left(1, \frac{C}{\|v\|_2}\right) \cdot v$ enforces a maximum ℓ_2-norm of C. For a mini-batch B of size $|B|$, the privatized gradient $\hat{g}$ is computed as

$$\hat{g} = \frac{1}{|B|}\sum_{i \in B} \texttt{clip}_C(\nabla l_i(f)) \; + \; \frac{\sigma C}{|B|}\,\xi,$$

where $\xi \sim \mathcal{N}(0, I_p)$ and $I_p \in \mathbb{R}^{p \times p}$ is the identity matrix. The noise variance σ, batch size $|B|$, and the number of training iterations collectively determine the privacy budget (ϵ, δ). Tuning these hyperparameters is critical for maintaining model accuracy. Moreover, excessive queries to a DP-protected model can still threaten privacy guarantees [16]. In this work, we adopt DP-SGD as the primary optimizer for all experiments.

2.3. Differentially Private Latent Diffusion Models

DP-LDM [37] leverages the efficiency of latent diffusion models (LDMs) [49] and integrate DP by fine-tuning a small subset of model parameters. In particular, DP-LDM first pre-trains an LDM on public data and then fine-tune it on private data using DP-SGD, thereby ensuring strong privacy guarantees while maintaining high synthesis quality.

More specifically, an LDM consists of a pre-trained autoencoder, Enc$(\cdot)$ and Dec$(\cdot)$, that maps high-dimensional images $x \in \mathbb{R}^{H \times V \times 3}$ into a lower-dimensional latent representation $z = \text{Enc}(x)$. A diffusion model, typically a modified UNet, is then trained in the latent space to predict the noise component of z at various time steps t and conditional input c. The training loss is defined as

$$L_{\text{ldm}}(\theta) = \mathbb{E}_{z_t, t, c}\left[\|\tau - \tau_\theta(z_t, t, c)\|_2^2\right],$$

where τ_θ is the function approximator parameterized by $\theta = [\theta_U, \theta_{\text{Attn}}, \theta_{\text{Cn}}]$, with θ_U corresponding to the UNet backbone parameters (excluding attention layers), θ_{Attn} the attention module parameters, and θ_{Cn} the conditioning embedder (for conditional generation).

Rather than fine-tuning all parameters, DP-LDM only updates the attention modules (and the conditioning embedder in the conditional case). The selective fine-tuning reduces the number of trainable parameters by approximately 90% compared to full fine-tuning, which not only improves the privacy-utility trade-off but also decreases the computational burden. For the private fine-tuning step, DP-LDM adopts the DP-SGD on the parameters $\{\theta_{\text{Attn}}, \theta_{\text{Cn}}\}$. At each iteration, the gradients are computed on a mini-batch and then clipped to a norm C. This procedure ensures that the fine-tuned model satisfies DP with respect to the private dataset.

2.4. Low-Rank Adaptation (LoRA)

LoRA [28] is a parameter-efficient fine-tuning method that adapts a pre-trained model by introducing low-rank updates to its weight matrices rather than updating all parameters. Specifically, given a pre-trained weight matrix $W \in \mathbb{R}^{m \times h}$, LoRA decomposes the update as

$$W' = W + \Delta W, \quad \Delta W = AB,$$

where $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times h}$ are low-rank matrices and $r \ll \min(m, h)$ is the rank of the decomposition. This

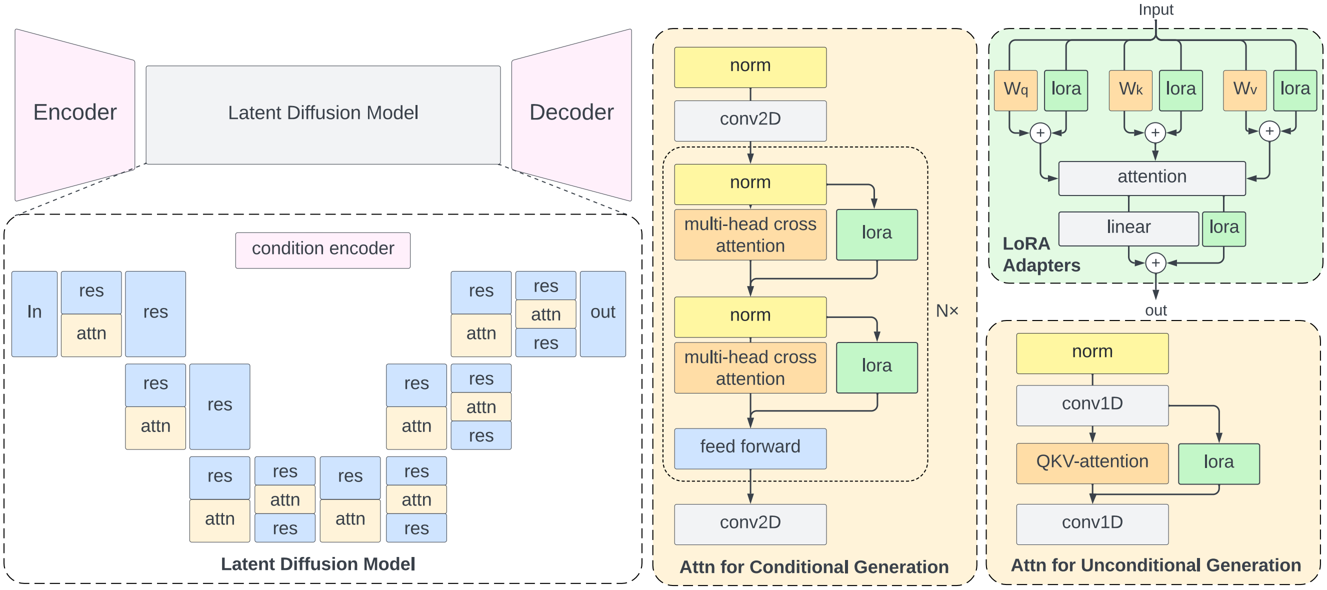


Figure 1. Overview of DP-LoRA. After pre-training the autoencoder and the latent diffusion model (LDM), we fine-tune the LDM by inserting LoRA [28] into each attention block. We apply LoRA not only to the QKV-attention matrices but also to the output projection layer for improved performance.

low-rank parameterization reduces the number of trainable parameters from $m \times h$ to $r \times (m + h)$.

In practice, during the fine-tuning process, the original weights W are kept fixed, and only the matrices A and B are updated via gradient-based optimization. During the forward pass, the modified weight W' is computed by adding the low-rank update ΔW to the pre-trained weight W. This allows the model to adapt to new tasks while largely retaining the pre-trained knowledge.

In our work, we employ LoRA to efficiently adapt the model to the target task under differential privacy constraints, leveraging its advantages in both computational efficiency and retention of pre-trained representations.

3. Parameter-efficient Differentially Private Latent Diffusion Models

In this section, we introduce our method, DP-LoRA, which follows a two-stage process: **1) pre-train** an LDM on a *large* public dataset to ensure generative quality, and **2) fine-tune** the LDM on a *small* private dataset with limited privacy budgets (ϵ) using Low-Rank Adaptation (LoRA) [28].

3.1. Fine-tuning LDM via Low-Rank Adaptation

Let $f(W_{\texttt{PT}}; x)$ be a pre-trained model with parameters $W_{\texttt{PT}}$ and input x. We can augment this function with additional parameters θ, where $\dim(\theta) \ll \dim(W_{\texttt{PT}})$, to enable fine-tuning. These new parameters are initialized to θ_0 such that

$$f_{\texttt{FT}}(W_{\texttt{PT}}, \theta_0; x) = f(W_{\texttt{PT}}; x). \tag{2}$$

LoRA then prescribes an additive update:

$$f_{\texttt{FT}}(W_{\texttt{PT}}, \theta; x) = f\big(W_{\texttt{PT}} + \lambda(\theta); x\big), \tag{3}$$

where the correction $\lambda(\theta)$ is parameterized by θ. Here $W_{\texttt{FT}} = W_{\texttt{PT}} + \lambda(\theta)$ lies on a low-dimensional manifold of dimension $\dim(\theta) \ll \dim(W_{\texttt{PT}})$. Consequently, even if θ is corrupted by significant DP noise, the combined weights $W_{\texttt{FT}}$ remain near $W_{\texttt{PT}}$, preserving image generation quality.

To integrate DP using Eq. (3), we propose DP-LoRA and employ LDMs for enhanced generation quality at higher resolutions [37]. An overview of DP-LoRA is shown in Figure 1. The training process consists of two main steps:

Pre-training. First, we train an auto-encoder on public data using SGD to map high-resolution images into a reduced latent space [48]. This step simplifies subsequent training by operating on lower-dimensional representations. We then train the LDM from scratch (i.e., all parameters $W_{\texttt{PT}}$) without LoRA modules. This establishes a solid foundation for high-fidelity image generation in the latent space.

Fine-tuning. Next, we encode private data into the learned latent space using the pre-trained auto-encoder. We fine-tune the LDM on this private data with LoRA modules via DP-SGD, incorporating the *noise multiplicity* strategy from DPDM [14]. Specifically, we inject LoRA into the self-attention blocks (for unconditional image generation) and the cross-attention blocks (for text-conditional generation). LoRA adapters are placed not only in the QKV-attention modules but also in the linear output projection layers that restore dimensionality.

3.2. Discussions

Limitations of fully fine-tuning. In differentially private training, there is an inherent trade-off between utility and privacy. As noted in Section 2.2, DP-SGD [1] clips per-sample gradients proportionally to network size, causing large models to suffer disproportionately under private training [39]. Moreover, diffusion models demand high computational budgets, often requiring many more iterations than standard classifiers [26]. The combined noise from diffusion training and DP-SGD makes private training especially challenging.

Parameter-efficiency benefits private fine-tuning. Minimizing the number of trainable parameters is crucial for improving the privacy-utility trade-off [39]. From the intrinsic dimensionality hypothesis [32] and observations in large language models [2], *the intrinsic dimension required for effective training can be much smaller than the total parameter count*. Low-rank decomposition [28] thus enables efficient adaptation of large models with significantly fewer trainable parameters. For instance, DP-LoRA achieves a FID of 8.4 ($\epsilon = 10$) on CelebA-64 using just 3.6% of the parameters, exceeding prior SoTA by over 50%. Additionally, our lightweight modification can be easily integrated into any public pre-trained model.

Which modules are most worthwhile to optimize? Recent research [24, 62] shows that fine-tuning *attention layers* in DMs is highly effective for tasks like image editing and text-to-image generation. Liu et al. [37] further advocate tuning attention blocks and conditioning embedders for private domain shifts. In DP-LoRA, we also adjust the output projection layer, ensuring that adapted features align with the new distribution. As shown in Table 1, fine-tuning both components (attention and projection) yields a FID of 7.71 on CelebA-64, compared to 9.82 when excluding the projection layer. Conversely, tuning other parts such as *ResBlocks* may degrade performance in private settings [37].

FT Modules	FID	Δ #Params
+ QKV & Project	**7.71 / 8.41**	718K / 479K
- QKV	11.78 / 12.89	249K / 159K
- Project Layer	9.82 / 11.24	479K / 319K

Table 1. Tuning different components on CelebA-32 (left) / CelebA-64 (right). QKV plus projection layers jointly achieve optimal FID.

4. Experiments

Datasets. We evaluate DP-LoRA on datasets of varying complexity: the widely recognized MNIST (resized to 32×32) [31], CIFAR-10 (32×32) [30], CelebA (64×64) [38], and the high-resolution CelebA-HQ (256×256) [29]. Note that we resize MNIST by following the setting of DP-LDM [37] (one of our baselines, see below). For class-conditional tasks, we use MNIST, CIFAR-10, and CelebA-HQ; for unconditional tasks, we test on CelebA at different resolutions. Public datasets include EMNIST [11] for MNIST and scaled ImageNet [50] for CIFAR-10, CelebA, and CelebA-HQ.

We emphasize that generating high-resolution DP images is more challenging than generating low-resolution images; thus, more attention must be paid to the quality of the DP-synthesized images. Although non-private image synthesis has advanced to photo-realistic high-resolution outputs up to 512×512 [37], the DP noise introduced during training degrades image quality. As shown in Table 2, previous work still considers CelebA-32 (32×32) and CelebA-64 (64×64) for unconditional generation, and MNIST and CIFAR-10 for conditional generation, which justifies our dataset selection.

Methods	**Uncondition**	**Condition**
PrivImage (USENIX'24)	CelebA-32, CelebA-64	CIFAR10
DP-LDM (TMLR'24)	CelebA-32, CelebA-64	CIFAR10, MNIST, Camelyon17-WILDS, CelebAHQ (Gender), CelebA-64 (Gender)
DP-MEPF (TMLR'23)	CelebA-32, CelebA-64	MNIST, Fashion-MNIST, CIFAR10
dp-promise (USENIX'24)	CelebA-32, CelebA-64, CIFAR10	MNIST, Fashion-MNIST
DP-SAD (ECCV'24)	CelebA-64	MNIST, Fashion-MNIST, CelebA-64 (Gender), CelebA-64 (Hair)
DPDM (TMLR'23)	CelebA-64	MNIST, Fashion-MNIST
DP-Diffusion (DeepMind'23)	-	MNIST, CIFAR-10, Camelyon17

Table 2. Datasets used by prior work.

Target	CelebA-32	CelebA-64	MNIST	CIFAR-10
pretrain-dataset	ImageNet	ImageNet	EMNIST (Letters)	ImageNet
Input size	32	64	32	32
Latent size	16	32	4	16
f	2	2	8	2
z-shape	16×16×3	16×16×3	4×4×3	32×32×3
Channels	128	192	128	128
Channel multiplier	[1,2]	[1,2]	[1,2,3,5]	[1,2]
Attention resolutions	[16,8]	[16,8]	[32,16,8]	[16,8]
Batch size	16	16	50	16
Epochs	4	10	50	4

Table 3. Parameter settings for pretraining autoencoders.

Evaluation metric. We demonstrate the performance of DP-LoRA by assessing: *(1) image generation quality* and *(2) downstream classification accuracy*. For image quality, we adopt the Fréchet Inception Distance (FID) [25], a standard measure in DP image generation [14]. For downstream utility in class-conditional tasks, we train a classifier (using CNN, Wide Residual Network (WRN) [61], or ResNet-9 [23]) on synthetic data and test on the real test set.

Baselines. We compare DP-LoRA with state-of-the-art approaches: DPDM [14], DP-MEPF [22], DP-Diffusion [20], DP-LDMs [37], PrivImage [33], and dp-promise [56].

Targte	CelebA-32	CelebA-64	MNIST	CIFAR-10
pretrain-dataset	ImageNet	ImageNet	EMNIST (Letters)	ImageNet
model channels	192	192	64	128
channel multiplier	[1,2,4]	[1,2,4]	[1,2]	[1,2,2,4]
attention resolutions	[1,2,4]	[1,2,4]	[1,2]	[1,2,4]
num res blocks	2	2	1	2
num heads	-	8	2	8
num head channels	32	-	-	-
Batch size	384	256	512	512
Epochs	40	40	120	40
use spatial transformer	False	False	True	True
cond stage key	-	-	class label	class label
conditioning key	-	-	crossattn	crossattn
num classes	-	-	26	1000
embedding dim	-	-	5	512
transformer depth	-	-	1	1

Table 4. Parameter settings for pretraining latent diffusion models.

Implementations. All experiments are conducted using PyTorch [43] with Opacus [60] for DP-SGD training and privacy accounting. Following standard practice [5], we set $\delta = 10^{-5}$ for MNIST and CIFAR-10, and $\delta = 10^{-6}$ for CelebA, ensuring that δ is smaller than the reciprocal of the dataset size.

DP is implemented via Opacus. Parameter-Efficient Fine-Tuning (PEFT) and LoRA use the PEFT library from Huggingface, with added support for Conv1D. Textual inversion is implemented with Huggingface's Diffusers library. The codebase builds on the Latent Diffusion paper and the DP-LDMs code. GPU devices used include RTX 4090/3090.

Detailed parameter settings for the auto-encoder pre-training are provided in Table 3, and those for pre-training the latent diffusion models are listed in Table 4.

4.1. Conditional Generations

Classification results. Table 5 compares classification accuracy across multiple datasets and privacy budgets (ϵ). On MNIST, our CNN-based classifier achieves 96.4% at $\epsilon = 1$, peaking at 97.9% for $\epsilon = 10$. The WRN variant also performs competitively. DP-Diffusion [20] with WRN obtains a top accuracy of 98.6% when privacy constraints are relaxed ($\epsilon = 10$). For CIFAR-10, our ResNet9-based method consistently outperforms others with the same architecture, reaching 73.98% at $\epsilon = 10$. Overall, we observe that in the realm of low-resolution images, DP-LoRA works very well but still has competitors.

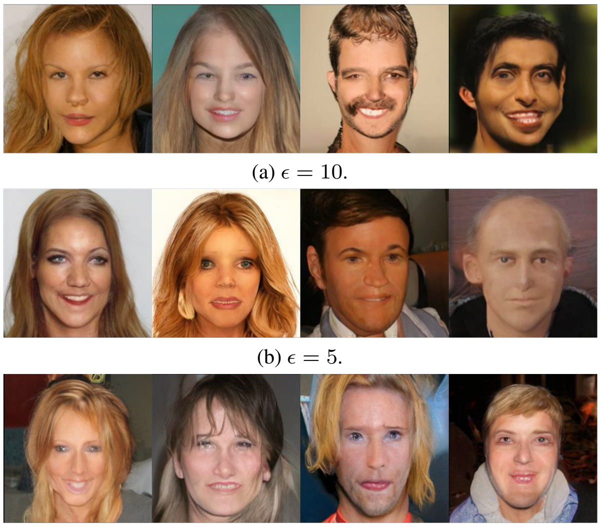
Generation on high-resolution images. Table 6 presents FID scores for gender-conditional generation on CelebA-HQ (256 × 256). DP-LoRA outperforms other methods across all ϵ values, with more than a 20% improvement at stricter privacy levels. We focus on gender-conditional generation for CelebA-HQ because DP-LDM also considers this task (see Table 2). Although no other prior work reports results on CelebA-HQ, we include additional experiments with DP-MEPF [22] for comparison; DP-MEPF exhibits significantly higher FIDs, indicating lower image fidelity under DP constraints.

4.2. Unconditional Generations

Table 7 compares FID scores on CelebA-64. DP-LoRA significantly outperforms other methods at all privacy levels, achieving FIDs of 12.0, 9.5, and 8.4 for $\epsilon = 1, 5, 10$, respectively. DP-LDMs [37] show moderate improvement as ϵ increases, but remain behind our results. dp-promise [56] and PrivImage [33] also lag considerably, suggesting challenges when training in pixel space or compressing latent representations too aggressively.

Table 8 and 9 show the complete results on unconditional generation on CelebA across different resolutions.

Interestingly, we observe that the generation quality on female images is much better than the generation quality of male images (see Figure 2 which consistently across different ϵ). Because of the highly imbalance between female and male images (the number of female images is way more than the male images), the generation quality of female images are generally better than male images.



(a) $\epsilon = 10$.

(b) $\epsilon = 5$.

(c) $\epsilon = 1$.

Figure 2. Examples generated from CelebA-HQ with $\epsilon = 1, 5, 10$.

4.3. Textual-Inversion

We also evaluate textual inversion [19], which learns semantic embeddings for text prompts. As shown in Table 11 and Figure 3, we do not observe notable improvements over standard training. Generic text prompts can struggle to represent diverse or detailed images, diminishing their effectiveness in privacy-preserving settings.

4.4. Ablation Study

We conduct additional experiments to investigate the impact of key hyperparameters in our pipeline, focusing on noise

Dataset	Method	Classifier	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = \infty$
MNIST	**Ours**	CNN	**96.4**	-	97.9	98.35
	Ours	WRN	94.8	-	97.8	98.16
	DP-LDM (TMLR'24)	CNN	95.9±0.1	-	97.4±0.1	-
	DP-LDM (TMLR'24)	WRN	-	-	97.5±0.0	-
	DPDM (TMLR'23)	CNN	95.2	-	98.1	-
	DP-Diffusion (DeepMind'23)	WRN	-	-	**98.6**	-
	dp-promise (USENIX'24)	CNN	95.8	-	98.2	-
	DP-SAD (ECCV'24)	CNN	91.2	-	94.1	-
	DP-FETA (S&P'25)	CNN	94.2	-	97.6	-
	DP-FETA (S&P'25)	WRN	95.0	-	98.2	-
CIFAR-10	**Ours**	ResNet9	**67.76**	**72.97**	**73.98**	79.85
	Ours	CNN	62.81	67.59	69.87	72.01
	DP-LDM (TMLR'24)	ResNet9	51.3±0.1	59.1±0.2	65.3±0.3	-
	DP-LDM (TMLR'24)	WRN	-	-	78.6±0.3	-
	DP-MEPF (TMLR'23)	ResNet9	28.9	47.9	48.9	-
	DP-Diffusion (DeepMind'23)	WRN	-	-	75.6	-
	PrivImage+G (USENIX'24)	CNN	47.5	39.2	44.3	-
	PrivImage+D (USENIX'24)	CNN	66.2	69.4	68.8	-
	DP-FETA (S&P'25)	ResNet9	31.3	37.9	44.6	-
	DP-FETA (S&P'25)	WRN	33.6	40.2	46.6	-
CelebA-64 (Gender)	**Ours**	ResNet9	**94.3**	-	**95.8**	-
	DP-MEPF (TMLR'23)	ResNet9	82.9	-	93.8	-
	DP-SAD (ECCV'24)	ResNet9	82.6	-	84.1	-
	DP-FETA (S&P'25)	ResNet9	84.7	-	85.5	-

Table 5. Classification accuracy (%) with class-conditional generations under various privacy levels. Higher is better; best in **bold**, second best underlined.

Method	$\epsilon = 10$	$\epsilon = 5$	$\epsilon = 1$
Ours	**17.2**	**18.2**	**20.0**
DP-LDM (TMLR'24)	19.0±0.0	20.5±0.1	25.6±0.1
DP-MEPF (TMLR'23)	43.5	79.4	117.1

Table 6. FID scores (lower is better) for gender-conditional generation on CelebA-HQ. DP-LoRA significantly outperforms other methods at all privacy levels.

Method	CelebA-64		
	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$
Ours	**12.0**	**9.5**	**8.4**
DP-LDMs (TMLR'24)	21.1	16.1	14.3
DP-SAD (ECCV'24)	24.7	20.7	17.9
DP-MEPF (ϕ_1, ϕ_2) (TMLR'24)	19.0	19.1	18.5
DP-MEPF (ϕ_1) (TMLR'24)	18.4	16.5	17.4
PrivImage+G (USENIX'24)	45.1	45.2	38.2
PrivImage+D (USENIX'24)	71.4	52.9	49.3
dp-promise (USENIX'24)	29.1	26.2	25.3

Table 7. Unconditional generation results on CelebA-64. Our method consistently achieves the lowest FID.

multiplicity and LoRA [28] rank settings.

Method	$\epsilon = 10$	$\epsilon = 5$	$\epsilon = 1$
DP-LDM (average case) (TMLR'24)	14.3±0.1	16.1±0.2	21.1±0.2
DP-LDM (best case) (TMLR'24)	14.2	15.8	21.0
DP-MEPF (TMLR'23)	17.4	16.5	20.4
dp-promise (USENIX'24)	25.3	26.2	29.1
PrivImage (USENIX'24)	49.3	52.9	71.4
Ours (r=8, k=4, n=10,000, epoch=5)	14.8621	17.2584	21.4400
Ours (r=8, k=4, n=60,000, epoch=5)	14.0125	16.3800	20.1930
Ours (r=8, k=4, n=10,000, epoch=15)	/	/	16.7637
Ours (r=16, k=4, n=10,000, epoch=15)	/	/	15.8495
Ours (r=16, k=4, n=60,000, epoch=15)	/	/	15.5615
Ours (r=16, k=4, n=60,000, epoch=40)	11.2422	11.3459	14.2692
Ours (r=16, k=4, n=60,000, epoch=40, project=True)	**8.4098**	**9.5134**	**12.0592**

Table 8. Results with different number, n, of training samples and pre-training epochs (k is the number of noise samples per data sample).

Algorithm	CelebA-32			CelebA-64			CelebA-HQ		
	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$
Ours (k=4)	12.5	11.9	7.7	**12.0**	**9.5**	**8.4**	**23.1**	**20.3**	**19.9**
DP-LDMs (TMLR'24)	25.8	16.8	16.2	21.1	16.1	14.3	29.7	26.0	24.3
DP-MEPF (ϕ_1, ϕ_2) (TMLR'23)	19.0	17.5	17.4	19.0	19.1	18.5	42.7	79.1	98.4
DP-MEPF (ϕ_1) (TMLR'23)	17.2	16.9	16.3	18.4	16.5	17.4	45.8	81.3	102.1
PrivImage+G (USENIX'24)	31.8	19.8	18.9	45.1	45.2	38.2	179.5	142.9	98.3
PrivImage+D (USENIX'24)	26.0	20.1	19.1	71.4	52.9	49.3	194.2	167.7	112.5
dp-promise (USENIX'24)	**9.0**	**6.5**	**6.0**	29.1	26.2	25.3	200.1	138.7	101.2

Table 9. FID with unconditioned generations on CelebA across different image resolutions.

Number of noise multiplicity steps. Table 10 shows how increasing noise multiplicity (k) improves image quality but

Metric	CelebA-32				CelebA-64			
	$k = 1$	$k = 2$	$k = 4$	$k = 8$	$k = 1$	$k = 2$	$k = 4$	$k = 8$
FID	13.75	10.91	7.71	**7.32**	12.29	11.03	**8.41**	8.94
training time/epoch	6 min	9 min	15 min	28 min	22 min	37 min	70 min	140 min

Table 10. Ablation on the number of noise multiplicity steps (k). Higher k yields better FID but increases training time.

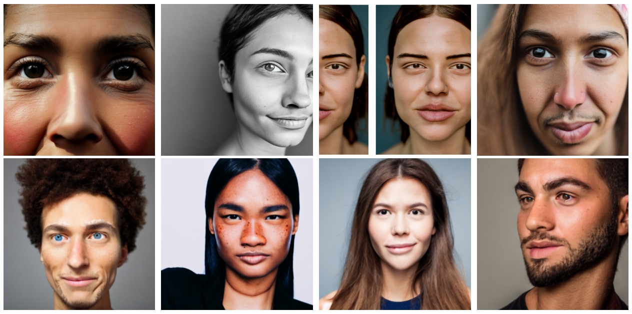


Figure 3. Examples generated using the prompt "`a good and full photo of <HF>`" on Stable-Diffusion-v1.5 with $\epsilon = 10$.

	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$	$\epsilon = \infty$
Ours	72.6	64.9	48.9	41.1

Table 11. FID scores for textual-inversion on MM-CelebA-HQ (512×512)(30,000 images from Stable-Diffusion-v1.5).

significantly prolongs training. On CelebA-64, performance peaks at $k = 4$ with an FID of 8.41, but training time rises accordingly. In practice, $k = 4-8$ offers a good balance between quality and efficiency.

Lower rank still achieves competitive results. Table 12 examines the effect of LoRA rank on FID. A moderate rank ($r = 16$) gives the best FID of 7.71 (CelebA-32) and 8.41 (CelebA-64), outperforming both lower ($r = 8$) and higher ($r = 64$) ranks. This suggests an optimal rank range where efficiency meets performance.

Rank	FID	Δ #Params
$r = 8$	8.09 / 10.01	359K / 239K
$r = 16$	**7.71 / 8.41**	718K / 479K
$r = 32$	7.83 / 9.14	1.4M / 958K
$r = 64$	10.16 / 9.03	2.9M / 1.9M

Table 12. Effect of LoRA rank on FID for CelebA-32 (left) / CelebA-64 (right).

Which modules matter most? Table 1 compares the impact of tuning different attention components. Incorporating both QKV matrices and the projection layer yields the best performance (FID of 7.71 on CelebA-32, 8.41 on CelebA-64). Omitting either component substantially degrades image quality, confirming that both are critical for effective private fine-tuning of diffusion models.

Different pretraining dataset on DP-LoRA. We pretrained models on CIFAR-10 and CIFAR-100, and evaluated both conditional and unconditional settings on CelebA32. The results are summarized in Table 13, where DP-LoRA still constantly outperforms competitors.

Condition	Training Dataset	Method	$\varepsilon = 1$	$\varepsilon = 10$
Unconditional	CIFAR-10 (FID)	DP-LoRA	**54.55**	**39.31**
		PrivImage	64.34	50.94
		DP-LDM	68.58	49.30
Unconditional	CIFAR-100 (FID)	DP-LoRA	**59.98**	**42.81**
		PrivImage	69.94	58.74
		DP-LDM	63.57	59.10
Conditional	CIFAR-10 (Acc)	DP-LoRA	**90.19%**	**93.32%**
		PrivImage	81.38%	87.92%
		DP-LDM	85.85%	89.47%

Table 13. Unconditional FID and gender downstream classification Acc on CelebA32 with different training datasets.

4.5. Discussion

Comparison to Private Evolution (PE). We also consider PE [34]. However, the 89.13% accuracy reported by PE on CIFAR-10 comes from training an ensemble of five classifiers on 1M generated images, which differs significantly from our setup. So, we instead compare PE and DP-LoRA under the same setting by excerpting results from PrivImage [33] and DPImageBench [21]. As Tables 14 and 15 show, DP-LoRA outperforms PE in nearly all cases. This discrepancy arises because PE is highly sensitive to the similarity between the pretraining and private datasets. For example, in the case of a large domain gap (e.g., ImageNet→CelebA), PE's performance drops drastically. Another source of discrepancy is usually-unreported hyperparameters (e.g., random seed and initial weights) used in different papers. A concurrent work, Sim-PE [35], also benchmarks DP-LoRA as a competitor. Table 16 shows DP-LoRA (excerpted from Sim-PE) and Sim-PE perform comparably, each excelling in different settings; notably, DP-LoRA (reported from our paper) surpasses Sim-PE in every case.

Dataset	ε	PE (from PrivImage)	DP-LoRA (from our paper)
CIFAR-10 (Acc)	1	47.1%	**61.81%**
	5	46.1%	**67.59%**
	10	47.9%	**69.87%**
CelebA32 (FID)	1	37.9	**12.5**
	5	33.8	**11.9**
	10	23.8	**7.7**
CelebA64 (FID)	1	54.9	**12.0**
	5	49.4	**9.5**
	10	49.0	**8.4**

Table 14. Performance comparison with PE in PrivImage.

Dataset	ε	PE (from DPImageBench)	DP-LoRA (from DPImageBench)	DP-LoRA (from our paper)
MNIST (Acc)	1	33.7%	76.6%	**96.4%**
	10	32.3%	97.4%	**97.9%**
CIFAR10 (Acc)	1	67.3%	64.1%	**67.8%**
	10	73.9%	**77.7%**	74.0%
CelebA32 (Acc)	1	69.8%	88.5%	**91.2%**
	10	75.8%	92.5%	**93.9%**
Camelyon (Acc)	1	61.2%	**85.7%**	/
	10	62.4%	**87.1%**	/

Table 15. Performance comparison with PE in DPImageBench.

Dataset	ε	Sim-PE	DP-LoRA (from Sim-PE)	DP-LoRA (from our paper)
MNIST (Acc)	1	89.1%	82.2%	**96.4%**
	10	93.6%	97.1%	**97.9%**
CelebA32 (Acc)	1	80.0%	87.0%	**91.2%**
	10	82.5%	92.0%	**93.9%**
CelebA32 (FID)	1	24.7	53.3	**12.5**
	10	20.8	32.2	**7.7**

Table 16. Performance Comparison with Sim-PE

Theoretical insights and difference to DP-LDM. The LoRA in DP-LoRA confines learning to an r-dimensional subspace ($r \ll |W|$), reducing trainable parameters from $\approx 100\%$ to $\approx 0.5\%$, which brings two benefits. First, fewer parameters incur less clipping-induced information loss during DP-SGD. Second, a smaller parameter count allows a lower clipping bound C, so DP-SGD injects much less Gaussian noise. Consequently, noise corrupts only the small adapter $\lambda(\theta)$, leaving the public weights $W_{\texttt{PT}}$ intact and keeping the denoising trajectory close to the pretrained manifold even under strict privacy budgets. Moreover, placing adapters in QKV and projection layers, where diffusion models concentrate their expressive power, enables this compact, less-noisy parameter set to recapture task-specific information more efficiently than prior baselines.

DP-LDM [37] fine-tunes attention blocks via DP-SGD. DP-LoRA has two key designs: applying LoRA to these blocks to drastically reduce trainable parameters (see Table 17) and using noise multiplicity k to lower noise scale.

Dataset	DP-LoRA	DP-LDM
MNIST	57K / 107M (0.05%)	4M / 107M (3.74%)
CIFAR-10	433K / 105M (0.41%)	90M / 105M (85.71%)
CelebA32	718K / 195M (0.37%)	162M / 194M (83.51%)
CelebA64	479K / 120M (0.40%)	72M / 119M (60.50%)

Table 17. Different numbers of trainable / total parameters.

Discrepancy to DPImageBench results. DPImageBench [21] differs from our implementation in terms of network architecture, pretraining data, and hyperparameters. Thus, DP-LDM (PrivImage) exhibits worse (better) performance in DPImageBench, compared to their originally reported results (see Table 18). In fact, DPImageBench reports highly mismatched results across methods (from ↓ 41.7 to ↑ 10.0).

Dataset	ε	DP-LoRA	DP-LDM	DPDM	PDP-Diffusion	DP-FETA	PrivImage
MNIST	1	76.6 (↓ 19.8)	54.2 (↓ 41.7)	90.0 (↓ 5.3)	/	96.2 (↓ 0.3)	/
	10	97.4 (↓ 0.5)	94.3 (↓ 3.1)	98.0 (↓ 0.1)	97.7 (↓ 0.9)	98.3 (↓ 0.2)	/
CIFAR-10	1	64.1 (↓ 3.6)	43.9 (↓ 7.4)	/	/	/	75.0 (↑ 8.8)
	10	77.7 (↑ 3.7)	63.1 (↓ 2.2)	/	69.4 (↓ 18.6)	/	78.8 (↑ 10.0)

Table 18. Performance mismatch between DPImageBench and original results. Arrows indicate performance change compared to the experimental results in original paper. "/" denotes that the result was not reported in the original paper.

5. Related Work

Diffusion models. Diffusion models have recently attracted considerable interest for their ability to generate high-quality synthetic data by iteratively denoising samples drawn from simple distributions [13, 26, 52]. Recent efforts to improve their efficiency [41, 42, 51] and scalability [44, 48] include techniques for faster sampling, latent-space representations, and more stable training.

Differentially private image generation. Applying differential privacy to diffusion models is an emerging research direction, aimed at producing high-fidelity synthetic data while protecting individual privacy [14]. Previous work on privacy-preserving generative modeling has mostly focused on applying DP-SGD [1] to GANs [8, 55, 58] and VAEs [45]. With the advent of diffusion models [48], recent studies have explored DP-SGD in this context. Dockhorn et al. [14] first investigated DP for diffusion, followed by Ghalebikesabi et al. [20], who showed that pretraining on public data and fine-tuning on private data achieves state-of-the-art performance. Liu et al. [37] proposed DP-LDM, a latent diffusion model with significantly fewer parameters to fine-tune than pixel-space diffusion. Meanwhile, custom architectures have been explored in works such as DP-MEPF [22] (which privatizes feature-embedding means), DPGEN [9] (energy-based modeling with random responses), PrivImage [33] (semantic query functions using public data), and DP-Promise [56] (DP noise added in early forward steps). Nonetheless, the substantial size of modern DMs still makes fine-tuning computationally expensive, limiting their practical utility.

6. Conclusion

We explored the integration of Differential Privacy (DP) with diffusion models (DMs), addressing the substantial privacy risks posed by the memorization capabilities of these models. Our study focused on optimizing the privacy-utility trade-off through a parameter-efficient fine-tuning strategy that minimizes the number of trainable parameters, thus enhancing the model's privacy while maintaining high utility. We empirically demonstrated that DP-LoRA achieves state-of-the-art performance in DP synthesis, significantly surpassing previous benchmarks with a small privacy budget. This work highlights the potential of parameter-efficient techniques in advancing privacy-preserving generative models.

References

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. B. McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016. 1, 2, 4, 8

[2] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *ACL*, abs/2012.13255, 2020. 4

[3] Hazrat Ali, Shafaq Murad, and Zubair Shah. Spot the fake lungs: Generating synthetic medical images using neural diffusion models. In *Irish Conference on Artificial Intelligence and Cognitive Science*, 2022. 1

[4] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, Tero Karras, and Ming-Yu Liu. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers. *ArXiv*, abs/2211.01324, 2022. 1

[5] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. *NeurIPS*, abs/2111.01177, 2021. 5

[6] Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. *USENIX Security Symposium*, abs/2301.13188, 2023. 1

[7] Pierre Chambon, Christian Blüthgen, Jean-Benoit Delbrouck, Rogier van der Sluijs, Malgorzata Polacin, Juan Manuel Zambrano Chaves, T. Abraham, Shivanshu Purohit, Curt P. Langlotz, and Akshay Chaudhari. Roentgen: Vision-language foundation model for chest x-ray generation. *ArXiv*, abs/2211.12737, 2022. 1

[8] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *NeurIPS*, abs/2006.08265, 2020. 8

[9] Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, Tzai-Wei Pang, and Chun-Shien Lu. Dpgen: Differentially private generative energy-guided network for natural image synthesis. In *CVPR*, 2022. 8

[10] Richard J. Chen, Ming Y. Lu, Tiffany Y. Chen, Drew F. K. Williamson, and Faisal Mahmood. Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5:493 – 497, 2021. 1

[11] Gregory Cohen, Saeed Afshar, Jonathan C. Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. *IJCNN*, pages 2921–2926, 2017. 4

[12] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *NeurIPS*, abs/2305.14314, 2023. 2

[13] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, abs/2105.05233, 2021. 1, 8

[14] Tim Dockhorn, Tianshi Cao, Arash Vahdat, and Karsten Kreis. Differentially private diffusion models. *TMLR*, abs/2210.09929, 2022. 1, 3, 4, 8

[15] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? *ArXiv*, abs/2302.01316, 2023. 1

[16] Cynthia Dwork. Differential privacy: A survey of results. In *TAMC*. Springer, 2008. 2

[17] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9:211–407, 2014. 1, 2

[18] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography Conference*, 7:17–51, 2006. 1, 2

[19] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ICLR*, abs/2208.01618, 2022. 5

[20] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L. Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *ArXiv*, abs/2302.13861, 2023. 1, 4, 5, 8

[21] Chen Gong, Kecen Li, Zinan Lin, and Tianhao Wang. Dpimagebench: A unified benchmark for differentially private image synthesis. In *ACM CCS*, 2025. 7, 8

[22] Fredrik Harder, Milad Jalali Asadabadi, Danica J Sutherland, and Mijung Park. Pre-trained perceptual features improve differentially private image generation. *TMLR*, 2022. 1, 4, 5, 8

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE CVPR*, 2016. 4

[24] Amir Hertz, Ron Mokady, Jay M. Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *ICLR*, abs/2208.01626, 2022. 4

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 2017. 4

[26] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, abs/2006.11239, 2020. 4, 8

[27] Hailong Hu and Jun Pang. Membership inference of diffusion models. *ArXiv*, abs/2301.09956, 2023. 1

[28] J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ICLR*, abs/2106.09685, 2021. 2, 3, 4, 6

[29] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, abs/1710.10196, 2017. 4

[30] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 4

[31] Yann LeCun and Corinna Cortes. The mnist database of handwritten digits. 2005. 4

[32] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *ICLR*, abs/1804.08838, 2018. 4

[33] Kecen Li, Chen Gong, Zhixiang Li, Yuzhong Zhao, Xinwen Hou, and Tianhao Wang. Privimage: Differentially private synthetic image generation using diffusion models with semantic-aware pretraining, 2024. 4, 5, 7, 8

[34] Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model APIs 1: Images. In *ICLR*, 2024. 7

[35] Zinan Lin, Tadas Baltrusaitis, Wenyu Wang, and Sergey Yekhanin. Differentially private synthetic data via apis 3: Using simulators instead of foundation model. In *ICLR Workshop on Synthetic Data*, 2025. 7

[36] Zi-Han Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. Differentially private synthetic data via foundation model apis 1: Images. *ICLR*, abs/2305.15560, 2024. 1

[37] Michael F Liu, Saiyue Lyu, Margarita Vinaroz, and Mijung Park. Differentially private latent diffusion models. *Transactions on Machine Learning Research*, 2024. 1, 2, 3, 4, 5, 8

[38] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 4

[39] Zelun Luo, Daniel J. Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. *CVPR*, pages 5057–5066, 2021. 4

[40] H. B. McMahan and Galen Andrew. A general approach to adding differential privacy to iterative training procedures. *ArXiv*, abs/1812.06210, 2018. 2

[41] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *ICML*, abs/2102.09672, 2021. 8

[42] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2021. 8

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, abs/1912.01703, 2019. 5

[44] William S. Peebles and Saining Xie. Scalable diffusion models with transformers. *ICCV*, pages 4172–4182, 2022. 8

[45] Bjarne Pfitzner and Bert Arnrich. Dpd-fvae: Synthetic data generation using federated variational autoencoders with differentially-private decoder. *ArXiv*, abs/2211.11591, 2022. 8

[46] Walter Hugo Lopez Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F. Da Costa, Virginia Fernandez, Parashkev Nachev, Sébastien Ourselin, and Manuel Jorge Cardoso. Brain imaging generation with latent diffusion models. *MICCAI Workshop*, abs/2209.07162, 2022. 1

[47] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1

[48] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CVPR*, pages 10674–10685, 2021. 1, 2, 3, 8

[49] 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*, pages 10684–10695, 2022. 2

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115:211 – 252, 2014. 4

[51] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, abs/2205.11487, 2022. 1, 8

[52] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *ICLR*, abs/2010.02502, 2020. 8

[53] Yang Song, Jascha Narain Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, abs/2011.13456, 2020. 2

[54] Amirsina Torfi, Edward A. Fox, and Chandan K. Reddy. Differentially private synthetic medical data generation using convolutional gans. *Information Sciences*, 586:485–500, 2020. 1

[55] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. *CVPR Workshop*, pages 98–104, 2019. 1, 8

[56] Haichen Wang, Shuchao Pang, Zhigang Lu, Yihang Rao, Yongbin Zhou, and Minhui Xue. dp-promise: Differentially private diffusion probabilistic models for image synthesis. *USENIX*, 2024. 4, 5, 8

[57] Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against text-to-image generation models. *ArXiv*, abs/2210.00968, 2022. 1

[58] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *ArXiv*, abs/1802.06739, 2018. 1, 8

[59] Hongxu Yin, Arun Mallya, Arash Vahdat, José Manuel Álvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. *CVPR*, pages 16332–16341, 2021. 1

[60] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*, 2021. 5

[61] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *BMVC*, abs/1605.07146, 2016. 4

[62] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *ICCV*, pages 3813–3824, 2023. 4

[63] Qingru Zhang, Minshuo Chen, Alexander W. Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. *ICLR*, abs/2303.10512, 2023. 2