# Mix-Opt: Mixed Optimization for Memory-Efficient Personalization of Text-to-Image Diffusion Models

**Seokeon Choi*** **Sunghyun Park*** **Hyoungwoo Park** **Jeongho Kim** **Sungrack Yun**

Qualcomm AI Research†

{seokchoi, sunpar, hwoopark, jeonghok, sungrack}@qti.qualcomm.com

## Abstract

Memory-efficient personalization is essential for adapting text-to-image diffusion models while preserving user privacy and operating within the limited computational resources of edge devices. To this end, we propose Mix-Opt, a novel mixed optimization framework that adaptively chooses between backpropagation on low-resolution images (BP-low) and zeroth-order optimization on high-resolution images (ZO-high), guided by the characteristics of the diffusion process. As observed in our experiments, BP-low efficiently adapts the model to target-specific features but suffers from structural distortions due to resolution mismatch. Conversely, ZO-high refines high-resolution details with minimal memory overhead but faces slow convergence when applied without prior adaptation. By complementing both methods, our framework leverages BP-low for effective personalization while using ZO-high to maintain structural consistency, achieving memory-efficient and high-quality fine-tuning. To maximize the efficacy of both BP-low and ZO-high, we introduce a timestep-aware probabilistic function that dynamically selects the appropriate optimization strategy during training. This function mitigates overfitting from BP-low at high timesteps, where structural information is critical, while ensuring that ZO-high is applied more effectively as training progresses. Experimental results across both U-Net- and DiT-based diffusion models demonstrate that Mix-Opt achieves competitive personalization performance while significantly reducing memory consumption. Mix-Opt enables scalable, high-quality personalization without modifying the inference process or increasing inference latency.

## 1 Introduction

The increasing popularity of text-to-image diffusion models has driven demand for personalized content generation, enabling applications such as creating avatars [34, 12] and customizing favorite objects in novel contexts [2]. To meet this demand, training-free personalization methods [31, 17, 28, 35] have emerged as efficient solutions, eliminating the need for explicit fine-tuning for each novel concept. These methods bypass test-time fine-tuning by streamlining the personalization process during inference. However, they often incur high inference latency, memory overhead, or heavy pre-training requirements, limiting practical deployment under resource constraints.

In contrast, fine-tuning-based methods such as DreamBooth [25], Textual Inversion [9], and Custom Diffusion [15] update model weights using user-provided images, achieving high-fidelity personalization with fine-grained details. Due to their high computational demands, these methods are

* These authors contributed equally.

† Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

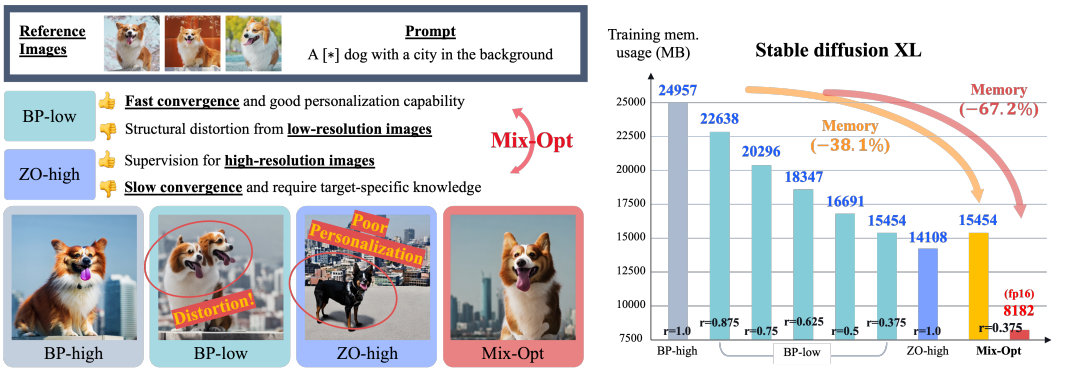



Figure 1: Overview of Mix-Opt. While BP-low and ZO-high offer memory efficiency, each has its limitations. Mix-Opt dynamically combines both methods to achieve high-quality personalization while significantly reducing training memory consumption compared to backpropagation on high-resolution inputs (BP-high).

typically executed on servers. However, growing privacy concerns have increased interest in on-device personalization, which allows users to fine-tune models locally without transmitting sensitive data. While this approach enhances privacy, the fine-tuning process remains computationally and memory-intensive, posing challenges for deployment on resource-constrained devices.

To address this issue, parameter-efficient fine-tuning techniques such as LoRA [11] have been explored, reducing the number of trainable parameters through low-rank adaptation. Although effective, LoRA still relies on full backpropagation, which requires storing activations and gradients and therefore incurs significant memory overhead. To further reduce this overhead, the Memory-Efficient Zeroth-Order Optimizer (MeZO) [18] has been explored. MeZO eliminates the need for backpropagation by estimating gradients using only forward passes, substantially reducing memory consumption and making it suitable for resource-constrained personalization.

However, applying MeZO directly to a pretrained model leads to inaccurate gradient estimates and slow convergence due to the absence of target-specific cues. Similar limitations were reported in ZOODiP [27], which required over 30,000 iterations for personalization using zeroth-order optimization alone. Importantly, we also observe that MeZO becomes more effective when preceded by partial fine-tuning that introduces target-specific information (**Observation 1**), as shown in Fig. 2 (a). High-resolution backpropagation (BP-high) would be ideal for introducing such information, but it is highly memory-intensive. A practical alternative is to use backpropagation on low-resolution images (BP-low), which reduces memory consumption by decreasing activation sizes. However, BP-low often produces distorted outputs during generation due to the mismatch between the training and inference resolutions (**Observation 2**). This issue is particularly severe when training at higher timesteps (750–1000), where diffusion models primarily recover global structures [4]. Repeated optimization in this range can bias the model toward low-resolution structural statistics, making BP-low gradients ineffective for maintaining global consistency at the default inference resolution. T-LoRA [29] also reported structural overfitting at higher timesteps even with full-resolution inputs, supporting our interpretation that BP-low can be particularly risky in this timestep range.

To overcome these limitations, we propose Mix-Opt, a novel memory-efficient framework that combines two complementary strategies: 1) backpropagation on low-resolution images (BP-low) for lightweight target-specific adaptation, and 2) zeroth-order optimization on high-resolution images (ZO-high) for structural refinement, as illustrated in Fig. 1. Based on our observations, BP-low effectively introduces target-specific features, whereas ZO-high maintains structural details using high-resolution inputs. However, BP-low can distort global structures at higher timesteps, while ZO-high has limited impact during the early stage of training because of inaccurate gradient estimates. To address these trade-offs, we design a timestep-aware probabilistic function that dynamically selects between the two optimization strategies according to the diffusion timestep and training progress. Specifically, the function favors BP-low at lower timesteps to capture target-specific cues, while increasingly applying ZO-high during the later stages of training to refine structural fidelity. This approach enables memory-efficient personalization without modifying the sampling process. We evaluate Mix-Opt on both U-Net- and DiT-based diffusion models, including Stable Diffusion, SANA, and FLUX, to examine its applicability across different backbone architectures.

Our contributions are summarized as follows:

- We propose Mix-Opt, a novel mixed optimization framework that combines BP-low and ZO-high for memory-efficient personalization, providing an effective strategy for leveraging zeroth-order optimization in diffusion model personalization.
- Based on observations of the complementary behaviors of BP-low and ZO-high, we introduce a timestep-aware probabilistic function that dynamically selects the optimization strategy according to the diffusion timestep and training progress.
- We validate Mix-Opt across both U-Net- and DiT-based diffusion models, including Stable Diffusion, SANA, and FLUX. Mix-Opt achieves competitive personalization performance while significantly reducing peak optimization memory, including reductions to 2.08 GB on SD V2.1 and 8.18 GB on SDXL, without modifying the inference process or increasing inference latency.

## 2 Related Work

**Text-to-Image Diffusion Models** Text-to-image diffusion models have become a cornerstone for generating high-quality images. Stable Diffusion [23] introduced diffusion in compressed latent spaces, reducing computational costs while maintaining image quality. It refined the UNet [24] backbone and text-conditioning mechanisms, enhancing output fidelity. Recent advancements, such as SDXL [21] and SSD-1B [10], expanded model capacity and training data, improving generalization and image quality. More recently, DiT [19]-based architectures, such as Stable Diffusion 3 [6], SANA [33], and FLUX [16], adopted transformer-based backbones, offering improved scalability at the cost of increased memory usage. While these advancements enhance performance, they introduce memory overhead, making efficient fine-tuning essential for deployment on resource-constrained environments.

**Personalization of Diffusion Models** Personalization in diffusion models can be categorized into training-based and training-free methods. Training-based approaches, such as Textual Inversion [9], DreamBooth [25], and Custom Diffusion [15], fine-tune model weights using user-provided data, enabling high-fidelity image generation. Despite their effectiveness, these methods incur high memory and computational costs, limiting deployment on edge devices. In contrast, training-free methods like ELITE [31], BLIP-Diffusion [17], InstantBooth [28], and JeDi [35], inject personalization cues at inference time without modifying model weights. While these methods avoid fine-tuning, they often increase inference latency and memory usage. Our approach enhances training-based methods by providing a memory-efficient fine-tuning strategy that minimizes resource consumption while maintaining personalization fidelity.

**Memory-Efficient Fine-Tuning** Reducing memory consumption during fine-tuning is critical for enabling on-device personalization of large-scale diffusion models. LoRA [11] reduces trainable parameters by focusing on low-rank subspaces of model weights, but still relies on full backpropagation, leading to significant memory overhead. HollowedNet [3] further optimizes memory usage by removing LoRA modules from non-essential UNet layers, but remains coupled to backpropagation and the UNet architecture. Memory-efficient Zeroth-order Optimizer (MeZO) [18] eliminates backpropagation by estimating gradients through forward passes, reducing memory consumption to inference-level requirements. While promising for large language models, its direct application to diffusion models struggles with personalization quality due to the lack of target-specific knowledge. ZOODiP [27] further illustrated this limitation, requiring over 30,000 iterations for personalization using zeroth-order optimization alone. Building on these foundations, we introduce Mix-Opt, a hybrid optimization framework that leverages the memory efficiency of MeZO while incorporating backpropagation at low resolution (BP-low) for efficient personalization, representing a meaningful advance toward effective ZO utilization.

## 3 Proposed Method

We propose Mix-Opt, a memory-efficient optimization framework for personalization of text-to-image diffusion models. It combines low-resolution backpropagation (BP-low) and high-resolution zeroth-order optimization (ZO-high) using a timestep-aware selection strategy. This hybrid approach enables high-quality fine-tuning with low memory usage.

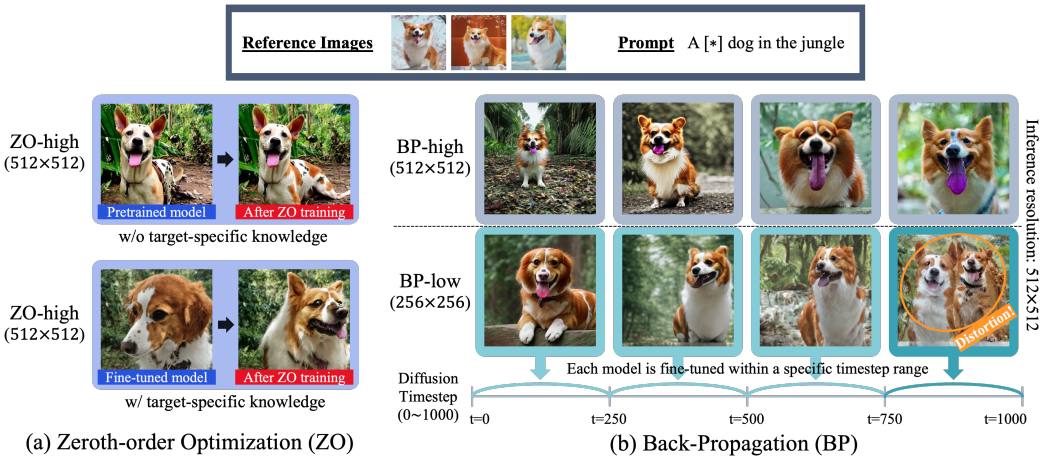



Figure 2: Observations for both optimization strategies. (a) MeZO requires target-specific knowledge; applying it from scratch fails to personalize the model, while using it after partial fine-tuning improves personalization. (b) Fine-tuning results for BP-high and BP-low trained at different timestep ranges. BP-high retains personalization features across all timesteps, whereas BP-low struggles to generate consistent outputs due to resolution mismatch, with severe structural distortions at high timesteps.

### 3.1 Overview of the Optimization Framework

Given images $\mathcal{X}$ and a prompt $c$ (*e.g.*, "a scs dog"), the goal is to fine-tune a pretrained diffusion model $\theta$ to generate images that reflect user-specific content. At each training step $i$, an image $x$ is sampled from $\mathcal{X}$, and a diffusion timestep $t$ is drawn from a uniform distribution $t \sim \mathcal{U}(0, T)$. Based on $t$ and $i$, a probabilistic function selects either BP-low or ZO-high. Importantly, Mix-Opt does not alter the standard timestep sampling; it only uses $(i, t)$ to choose the optimization mode.

BP-low is used at lower timesteps to efficiently inject target-specific features, while ZO-high helps preserve global structure and reduce overfitting, especially in the later phase of training. These design choices are motivated by two empirical observations: (1) MeZO is more effective when preceded by partial fine-tuning that introduces target-specific information, and (2) BP-low tends to distort global structure at high timesteps due to resolution mismatch.

### 3.2 Backpropagation on Low-Resolution Images

Backpropagation is applied to low-resolution images $x_{\text{low}}$ (*e.g.*, $256 \times 256$) to reduce memory usage while capturing coarse personalization features. The optimization follows the standard gradient descent update:

$$\theta_{i+1} = \theta_i - \eta \nabla_\theta \mathcal{L}_{\text{bp}}(\theta_i; x_{\text{low}}, c, t), \tag{1}$$

where $\theta_i$ denotes the parameters at training step $i$, $\eta$ is the learning rate, and $\mathcal{L}_{\text{bp}}$ is the diffusion reconstruction loss at timestep $t$.

### 3.3 MeZO on High-Resolution Images

MeZO [18] estimates gradients using only forward passes by perturbing parameters and contrasting the resulting losses, following the Simultaneous Perturbation Stochastic Approximation (SPSA) [30]. To improve stability and reduce noise in gradient estimation, we adopt an n-SPSA strategy that averages multiple perturbations, enabling more reliable updates without increasing memory overhead.

For each original image $x$ (*e.g.*, $512 \times 512$), MeZO computes the accumulated gradient. For each perturbation $n \in \{1, 2, \ldots, N\}$, we sample a random perturbation vector $\boldsymbol{z}^{(n)} \sim \mathcal{N}(0, I_d)$ from a standard normal distribution and compute the gradient estimate:

$$\hat{g}_i^{(n)} = \frac{\mathcal{L}(\theta_i + \epsilon \boldsymbol{z}^{(n)}; x, c, t) - \mathcal{L}(\theta_i - \epsilon \boldsymbol{z}^{(n)}; x, c, t)}{2\epsilon} \boldsymbol{z}^{(n)}. \tag{2}$$

**Algorithm 1** Training Algorithm

**Require:** Pretrained diffusion model $\theta_0$, personalization images $\mathcal{X}$, prompt $c$, total number of training steps $i_{\max}$, maximum timestep $t_{\max}$, number of perturbations $N$

**for** training step $i = 1$ to $i_{\max}$ **do**
  Sample image $x \in \mathcal{X}$
  Sample diffusion timestep $t \sim \mathcal{U}(0, t_{\max})$
  Compute $p^{zo}_{i,t}$ using Eq. 5
  **if** random() $> p^{zo}_{i,t}$ **then** ▷ BP-low
    Downsample $x$ to $x_{\text{low}}$
    Compute gradient $\nabla_\theta \mathcal{L}_{\text{bp}}(\theta_i; x_{\text{low}}, c, t)$
    Update $\theta_{i+1} = \theta_i - \eta \nabla_\theta \mathcal{L}_{\text{bp}}$
  **else** ▷ ZO-high
    Use high-resolution image $x$
    Initialize accumulated gradient $\hat{g}_i = 0$
    **for** $n = 1$ to $N$ **do**
      Sample perturbation $\boldsymbol{z}^{(n)}$
      Compute $\hat{g}_i^{(n)}$ using Eq. 2
      Accumulate $\hat{g}_i = \hat{g}_i + w_n \hat{g}_i^{(n)}$
    **end for**
    Update $\theta_{i+1} = \theta_i - \alpha \hat{g}_i$
  **end if**
**end for**

The final gradient estimate $\hat{g}_i$ is obtained by a weighted sum of the $N$ estimates, $\hat{g}_i = \sum_{n=1}^{N} w_n \hat{g}_i^{(n)}$, where $w_n = \frac{1}{N}$. The parameter update is given by $\theta_{i+1} = \theta_i - \alpha \hat{g}_i$, with $\alpha$ as the learning rate for MeZO updates.

### 3.4 Timestep-Aware Probabilistic Function

To reduce overfitting and structural artifacts, our probabilistic function selects the optimization strategy (BP-low or ZO-high) at each step according to the diffusion timestep $t$ and training step $i$.

#### 3.4.1 Timestep-Aware Probability

Motivated by **Observation 2**, BP-low tends to distort global structure at high timesteps ($t \approx t_{\max}$), where structural information is critical [4]. Accordingly, we define the timestep-aware probability (TAP) of selecting ZO-high as:

$$p_t^{zo} = \frac{1}{1 + e^{-k(t - t_{\text{mid}})}}, \tag{3}$$

where $k > 0$ controls the steepness, and $t_{\text{mid}}$ is the midpoint of the transition. This ensures ZO-high is more likely to be selected at higher timesteps, while BP-low becomes less frequent.

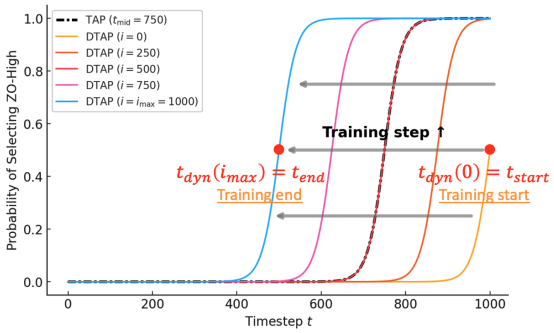



Figure 3: Comparison of TAP and DTAP over training steps ($t_{\max}, i_{\max} = 1000, k = 0.05, t_{\text{mid}} = 750$). The black dashed line shows TAP with a fixed $t_{\text{mid}}$, while colored curves show DTAP adapting over training. At $i = i_{\text{mid}}$, DTAP coincides with TAP.

#### 3.4.2 Dynamic Timestep-Aware Probability

To reflect **Observation 1** that MeZO becomes more effective as training progresses, we make the midpoint dynamic in the training step $i$, allowing $t_{\text{mid}}$ to shift from $t_{\text{start}}$ to $t_{\text{end}}$:

$$t_{\text{dyn}}(i) = t_{\text{start}} + \frac{i}{i_{\max}}(t_{\text{end}} - t_{\text{start}}). \tag{4}$$

To ensure dynamic timestep-aware probability (DTAP) aligns with TAP such that $t_{\text{dyn}}(i_{\text{mid}}) = t_{\text{mid}}$, we control the transition using $t_{\text{mid}}$ by setting $t_{\text{start}} = t_{\max}$ and $t_{\text{end}} = 2t_{\text{mid}} - t_{\max}$. The final probability

| | Method (SD V1.5) | Metrics DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|---|
| TI | TI | 0.5590 | 0.7780 | 0.2850 | 6.75 |
| | GF-TI | 0.0110 | 0.5400 | 0.2530 | 2.37 |
| | ZooDiP | 0.5580 | 0.7720 | 0.2870 | 2.37 |
| DB | Full fine-tuning | 0.7149 | 0.8153 | 0.2887 | 17.32 |
| | PEQA | 0.6040 | 0.7910 | 0.2750 | 6.31 |
| | TuneQDM | 0.5550 | 0.7880 | 0.2890 | 8.96 |
| | LoRA-BP-High | 0.6434 | **0.7926** | 0.2890 | 5.15 |
| | LoRA-MeZO | 0.2218 | 0.6046 | 0.2645 | 3.63 |
| | **Ours (r=0.750)** | 0.6403 | 0.7921 | 0.2904 | 4.44 |
| | **Ours (r=0.500)** | 0.6437 | 0.7885 | 0.2904 | 3.91 |
| | **Ours (r=0.375)** | 0.6114 | 0.7796 | 0.2847 | 3.73 |
| | **Ours† (r=0.750)** | 0.6424 | 0.7918 | **0.2913** | 2.44 |
| | **Ours† (r=0.500)** | **0.6448** | 0.7891 | 0.2893 | 2.07 |
| | **Ours† (r=0.375)** | 0.6059 | 0.7764 | 0.2851 | **1.94** |

Table 1: Comparison of personalization methods on SD V1.5. † denotes results obtained with mixed precision (fp16).

| | Method (SD V2.1) | Metrics DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|---|
| DB | Full fine-tuning | 0.6889 | 0.7989 | 0.3039 | 17.47 |
| | HollowedNet (H-30) | 0.6690 | 0.8050 | 0.3030 | 4.18 |
| | HollowedNet (H-39) | 0.6600 | 0.8050 | 0.3000 | 3.88 |
| | HollowedNet (H-57) | 0.6180 | 0.7820 | 0.3020 | 3.16 |
| | LoRA-BP-High | 0.6762 | **0.8075** | 0.2961 | 5.09 |
| | LoRA-MeZO | 0.3688 | 0.6664 | **0.3217** | 3.66 |
| | **Ours (r=0.750)** | 0.6758 | 0.8058 | 0.2969 | 4.42 |
| | **Ours (r=0.500)** | **0.6833** | 0.8067 | 0.2938 | 3.92 |
| | **Ours (r=0.375)** | 0.6532 | 0.7893 | 0.2950 | 3.75 |
| | **Ours† (r=0.750)** | 0.6789 | 0.8065 | 0.2965 | 2.44 |
| | **Ours† (r=0.500)** | 0.6776 | 0.8069 | 0.2926 | 2.08 |
| | **Ours† (r=0.375)** | 0.6526 | 0.7922 | 0.2947 | **1.82** |

Table 2: Comparison of personalization methods on SD V2.1. † denotes results obtained with mixed precision (fp16).

becomes:

$$p_{i,t}^{zo} = \frac{1}{1 + e^{-k(t - t_{\text{dyn}}(i))}}. \tag{5}$$

During the early stage of training, DTAP favors BP-low over a broad range of timesteps so that the model can efficiently acquire target-specific features. As training progresses, the transition boundary shifts toward lower timesteps, increasing the frequency of ZO-high updates and improving structural consistency at the default resolution.

# 4 Experiments

## 4.1 Experimental Setup

**Implementation Details:** We implemented Mix-Opt on top of the DreamBooth [25] LoRA module using Stable Diffusion V2.1 (SD V2.1) [23], which served as the primary backbone for most experiments, covering ablation studies, user studies, and hyperparameter tuning (with ratio $r = 0.5$). To evaluate the scalability of Mix-Opt, we extended our analysis to models suitable for edge-device adaptation, including Stable Diffusion V1.5 (SD V1.5) [23], SDXL [21], SSD-1B [10], SANA [33], and FLUX [16]. All experiments were conducted on NVIDIA A100 GPUs. For each training sample, we precomputed low- and default-resolution image latents for BP-low and ZO-high, respectively, together with the corresponding text embeddings. The frozen VAE and text encoder were then offloaded from GPU memory, and the cached representations were reused throughout optimization. For all experiments, we set the total number of training steps to $i_{\max} = 1000$, the LoRA rank to $R = 4$, the maximum diffusion timestep to $t_{\max} = 1000$, and the learning rate to $\eta = 1 \times 10^{-4}$. The batch size was set to 1, and the resolution matched each model's default setting: $512 \times 512$ for SD V1.5 and SD V2.1, and $1024 \times 1024$ for SDXL and SSD-1B. For Mix-Opt, DTAP was used as the default selection strategy, with key parameters set as follows: $t_{\text{mid}} = 750$, $k = 0.05$, MeZO learning rate $\alpha = 1 \times 10^{-2}$, and gradient accumulation steps $N = 4$.

**Dataset:** All experiments were conducted using the DreamBooth dataset [25], which contains 30 subjects, each paired with 25 unique prompts. Each subject consists of 4 to 6 images, with a total of 9 live subjects and 21 objects. For each subject-prompt pair, four images were generated using different random seeds, and the final scores were computed by averaging the results across all generations. In total, this setup yields 3,000 generated images, providing a comprehensive basis for evaluating fine-tuning performance.

**Evaluation Metrics:** For quantitative evaluation, we used CLIP-I, CLIP-T [22], and DINO [1] scores. CLIP-I measures similarity to reference subject images, CLIP-T evaluates text-image alignment, and DINO assesses subject consistency. Training memory usage was reported in each experiment to assess resource efficiency. VAE and text encoder components were precomputed and excluded from training, and peak memory was measured using *torch.cuda.max_memory_allocated*.

| | Method (Base: SDXL) | Performance DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|---|
| DB | LoRA-BP-High | **0.7397** | 0.8145 | 0.3005 | 24.96 |
| | LoRA-MeZO | 0.4349 | 0.6670 | **0.3261** | 14.11 |
| | **Ours (r=0.750)** | 0.7310 | 0.8093 | 0.3012 | 20.30 |
| | **Ours (r=0.500)** | 0.7350 | 0.8137 | 0.2948 | 16.69 |
| | **Ours (r=0.375)** | 0.7302 | 0.8121 | 0.2913 | 15.45 |
| | **Ours† (r=0.750)** | 0.7269 | 0.8084 | 0.3003 | 11.49 |
| | **Ours† (r=0.500)** | 0.7336 | 0.8135 | 0.2941 | 9.05 |
| | **Ours† (r=0.375)** | 0.7322 | **0.8146** | 0.2898 | **8.18** |

Table 3: Comparison of personalization methods on SDXL. † denotes results obtained with mixed precision (fp16).

| | Method (Base: SSD-1B) | Performance DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|---|
| DB | LoRA-BP-High | 0.6841 | 0.7971 | 0.3070 | 15.41 |
| | LoRA-MeZO | 0.3900 | 0.6539 | **0.3109** | 9.10 |
| | **Ours (r=0.750)** | 0.6964 | 0.8027 | 0.3047 | 12.57 |
| | **Ours (r=0.500)** | **0.7145** | **0.8106** | 0.3001 | 10.48 |
| | **Ours (r=0.375)** | 0.7057 | 0.8048 | 0.2994 | 9.75 |
| | **Ours† (r=0.750)** | 0.6913 | 0.7985 | 0.3048 | 7.14 |
| | **Ours† (r=0.500)** | 0.7080 | 0.8075 | 0.2996 | 5.67 |
| | **Ours† (r=0.375)** | 0.7041 | 0.8040 | 0.2989 | **5.14** |

Table 4: Comparison of personalization methods on SSD-1B. † denotes results obtained with mixed precision (fp16).

| Backbone | Method | DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|---|
| SANA-0.6B | LoRA-BP-High | **0.7239** | **0.8043** | 0.3057 | 3.97 |
| | **Ours** | 0.7076 | 0.7975 | **0.3076** | **1.75** |
| SANA-1.6B | LoRA-BP-High | **0.7125** | **0.7942** | **0.3103** | 6.22 |
| | **Ours** | 0.7039 | 0.7924 | 0.3073 | **3.43** |
| FLUX.1-dev | LoRA-BP-High | **0.6717** | **0.7991** | 0.2667 | 55.61 |
| | **Ours** ($r = 0.5$) | 0.6716 | 0.7911 | **0.2922** | **33.51** |

Table 5: Comparison of LoRA-BP-High and Mix-Opt on DiT-based models. The results are obtained with mixed precision (bf16).

### 4.2 Scalability across Diffusion Architectures

**Stable Diffusion V1.5 in Table 1:** TI [9]-based methods such as GF-TI [7] and ZooDiP [27] show low memory usage but limited personalization fidelity. In particular, ZooDiP combines quantization and zeroth-order optimization, achieving 2.37 GB memory usage, but its reliance on textual inversion leads to lower CLIP-I and DINO scores. ZooDiP also requires over 30,000 steps to converge, while Mix-Opt reaches better performance in just 1,000 steps, despite also utilizing zeroth-order optimization. Among DreamBooth-based methods, full fine-tuning achieves high performance but requires excessive memory, limiting its applicability on edge devices. Quantization-based approaches (PEQA [14], TuneQDM [26]) reduce memory relative to full fine-tuning, but incur notable fidelity loss and remain less competitive than Mix-Opt in both fidelity and memory efficiency. Furthermore, MeZO [18] alone fails to achieve robust personalization in DreamBooth settings, highlighting the limitations of relying solely on zeroth-order optimization. Mix-Opt addresses this by combining ZO with BP-based updates in a dynamic framework. We report Mix-Opt results for $r \in \{0.75, 0.5, 0.375\}$, and find that $r = 0.5$ offers the best trade-off between fidelity and memory usage in SD V1.5. When $r = 0.5$, Mix-Opt achieves performance comparable to LoRA-BP-High while using **23.0%** less memory. When mixed precision (†) with fp16 is applied, Mix-Opt reduces the memory requirement by **59.8%** relative to LoRA-BP-High while maintaining comparable personalization performance.

**Stable Diffusion V2.1 in Table 2:** Compared to SD V1.5, SD V2.1 yields higher overall fidelity scores. We compare Mix-Opt with HollowedNet [3], which reports three variants (H-30, H-39, H-57) based on pruning extent. H-57 uses the least memory (3.16 GB) but shows severe performance drop. Under comparable memory budgets, Mix-Opt consistently outperforms HollowedNet in DINO and CLIP-I scores, demonstrating its robustness without requiring architecture-specific structural pruning. Unlike HollowedNet, which must be manually redesigned for each backbone, Mix-Opt is model-agnostic and supports flexible resolution control, making it more practical. Moreover, HollowedNet requires additional precomputing for intermediate activations and doubles the inference steps, whereas Mix-Opt operates without additional precomputing and leaves the inference process entirely unchanged, further highlighting its efficiency and practicality.

**SDXL in Table 3 and SSD-1B in Table 4:** For high-resolution models (1024×1024), Mix-Opt maintains fidelity while significantly reducing memory compared to LoRA-BP-High. The higher resolution allows us to safely lower the resize ratio to $r = 0.375$, enabling further memory savings. With mixed precision, training requires only **8.18 GB** on SDXL and **5.14 GB** on SSD-1B. This demonstrates that Mix-Opt not only scales to large backbones but also supports high-resolution personalization.

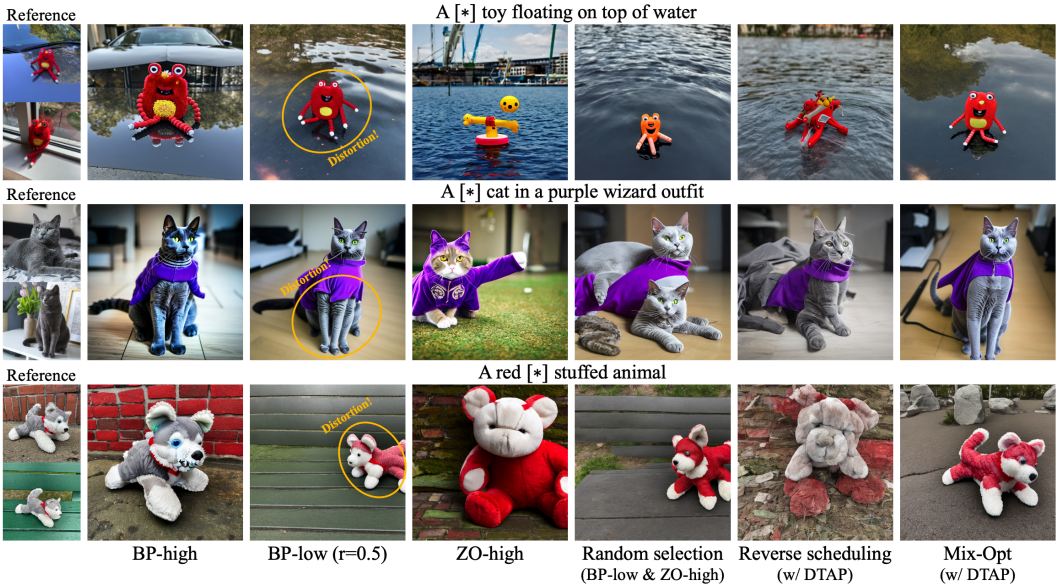
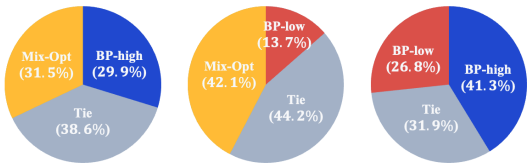



Figure 4: Qualitative comparison of different optimization methods, including BP-high, BP-low, ZO-high, Random Selection, Reverse Scheduling, and Mix-Opt (DTAP), conducted with Stable Diffusion v2.1 at low resolution ratio = 0.5.

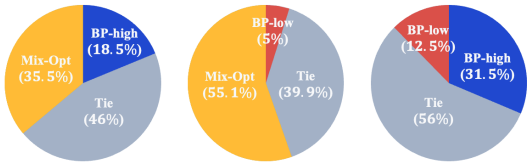
**DiT-based Models in Table 5:** To investigate whether Mix-Opt can be applied beyond U-Net-based diffusion models, we evaluate it on SANA [33] and FLUX [16], which represent two different diffusion transformer designs. SANA uses a no positional encoding (NoPE) design with local spatial processing in its Mix-FFN, while FLUX uses a rotary positional encoding (RoPE). Mix-Opt reduces training memory by **55.9%** on SANA-0.6B and **44.9%** on SANA-1.6B while retaining competitive DINO, CLIP-I, and CLIP-T scores. On FLUX, it reduces memory by **39.7%**, with nearly unchanged DINO, lower CLIP-I, and improved CLIP-T relative to LoRA-BP-High. Together, these results show that Mix-Opt extends beyond U-Net and NoPE-based DiT architectures to the evaluated PE-based DiT model, supporting its applicability across structurally different diffusion backbones.

### 4.3 Qualitative and User Study Analysis

Qualitative comparisons against various optimization strategies are presented in Fig. 4. **BP-high** occasionally suffers from overfitting and saturation-like artifacts, hindering text alignment. **BP-low** exhibits structural distortions, particularly at high timesteps, leading to object deformations and inconsistent subject representation. Conversely, **ZO-high** suffers from slow convergence, failing to learn target-specific features. **Random Selection** lacks a stable optimization path, resulting in suboptimal subject preservation and frequent structural distortions. **Reverse Scheduling** refers to a variant where we reversed the selection probability of BP-low and ZO-high ($k = -0.05$). This modification directly contradicts our key observations by prioritizing ZO-high in the early stages and BP-low at high timesteps. This approach resulted in severe structural distortions and failed personalization, confirming that MeZO is ineffective when applied too early and that BP-low exacerbates artifacts when applied at high timesteps. In contrast, **Mix-Opt** successfully balances




(a) Subject Fidelity:
*"Select the image that is most similar to the training image."*




(b) Text Fidelity:
*"Select the image that best matches the text prompt used for generation."*

Figure 5: User study results on SD V2.1, comparing Mix-Opt with BP-high and BP-low in terms of Subject Fidelity and Text Fidelity. In BP-Low and Mix-Opt, the low-resolution ratio is set to 0.5.

| Method | DINO | CLIP-I | CLIP-T |
|---|---|---|---|
| BP-low | 0.6816 | 0.8099 | 0.2893 |
| ZO-high | 0.3688 | 0.6664 | 0.3217 |
| Random selection | 0.6530 | 0.7844 | 0.3050 |
| Ours (w/ TAP) | 0.6681 | 0.7992 | 0.2948 |
| Ours (w/ DTAP) | 0.6833 | 0.8067 | 0.2938 |

Table 6: Ablation study on selection strategies conducted using SD V2.1, with a low-resolution ratio fixed at 0.5.

| | Method | DINO | CLIP-I | CLIP-T |
|---|---|---|---|---|
| **TAP** | $t_{mid} = 250$ | 0.5151 | 0.7210 | 0.3165 |
| | $t_{mid} = 500$ | 0.6267 | 0.7726 | 0.3060 |
| | $t_{mid} = 750$ | 0.6681 | 0.7992 | 0.2948 |
| **DTAP** | $t_{mid} = 250$ | 0.5811 | 0.7551 | 0.3112 |
| | $t_{mid} = 500$ | 0.6327 | 0.7789 | 0.3038 |
| | $t_{mid} = 750$ | 0.6833 | 0.8067 | 0.2938 |

Table 7: Impact of $t_{\text{mid}}$ in TAP and DTAP conducted using SD V2.1 with a low-resolution ratio fixed at 0.5.

BP-low and ZO-high through our DTAP, ensuring structural consistency while preserving fine-grained subject details. In some cases, Mix-Opt even surpasses BP-high in text alignment, demonstrating its ability to alleviate overfitting issues in back propagation.

These findings are validated by a user study with 24 participants (Fig. 5) using A/B comparisons across all DreamBooth subjects. Mix-Opt consistently outperforms both BP-high and BP-low in subject fidelity and text alignment. Compared to BP-high, Mix-Opt improves text alignment and mitigates instance-level overfitting while maintaining fidelity. Against BP-low, it shows significant gains in both metrics, addressing inherent structural distortions. Notably, BP-low's poor performance relative to BP-high underscores the risks of relying solely on low-resolution optimization for memory efficiency.

### 4.4 Ablation Study

**Ablation Study on Selection Strategies:** To evaluate the impact of our DTAP, we compared different selection strategies in Table 6. Interestingly, BP-low achieves competitive DINO and CLIP-I scores but suffers from structural distortions that are not fully captured by these metrics. ZO-high fails to maintain subject fidelity, highlighting its limitations in personalization when applied without adaptation. Random Selection performs between BP-low and ZO-high, underscoring its instability. In contrast, Mix-Opt (w/ & w/o DTAP) consistently outperforms Random Selection, demonstrating the effectiveness of our DTAP.

We further analyze the effect of $t_{\text{mid}}$ in Table 7. Lower $t_{\text{mid}}$ values increase the frequency of MeZO updates while reducing the role of BP, leading to weaker personalization, as indicated by lower CLIP-I and DINO scores. This confirms the importance of BP in establishing target-specific adaptations. Additionally, DTAP consistently surpasses TAP across all $t_{\text{mid}}$ values, demonstrating that dynamically increasing MeZO in later training stages achieves a better balance between personalization and text alignment. Notably, DTAP at $t_{\text{mid}} = 750$ provides the best trade-off, ensuring structural consistency while mitigating overfitting effects.

**Hyperparameter Selection for Mix-Opt:** We conducted an ablation study on Mix-Opt hyperparameters: the learning rate of MeZO ($\alpha$), the gradient accumulation steps ($N$), and the steepness of the DTAP ($k$). The first row of Fig. 6 examines the impact of $\alpha$. A low learning rate prevents ZO-high from effectively refining structural distortions in BP-low, while excessively high values cause optimization instability, leading to gradient exploding or under-personalization. We select $\alpha = 1 \times 10^{-2}$ as the optimal balance. The second row analyzes the effect of $N$. When $N = 1$, the less stable gradient estimation may sometimes lead to slightly reduced personalization performance. Increasing $N$ improves detail refinement, but excessive accumulation significantly slows training without clear performance gains. We find $N = 4$ to provide the best trade-off between stability and efficiency. The third row evaluates $k$, which controls the transition sharpness between BP-low and ZO-high. A high $k$ disrupts structural consistency by introducing abrupt optimization shifts, while a low $k$ flattens the slope of the function, increasing the probability of selecting MeZO too early in training and leading to insufficient personalization. We determine that $k = 0.05$ provides the most balanced optimization strategy, ensuring a smooth transition while maintaining stability. Although Mix-Opt shows robustness to hyperparameter variations overall, the examples in Fig. 6 highlight extreme cases to illustrate the underlying trends more clearly. In practice, performance remains relatively stable across a reasonable range of values, which makes Mix-Opt a scalable and practical optimization framework for diverse personalization tasks.

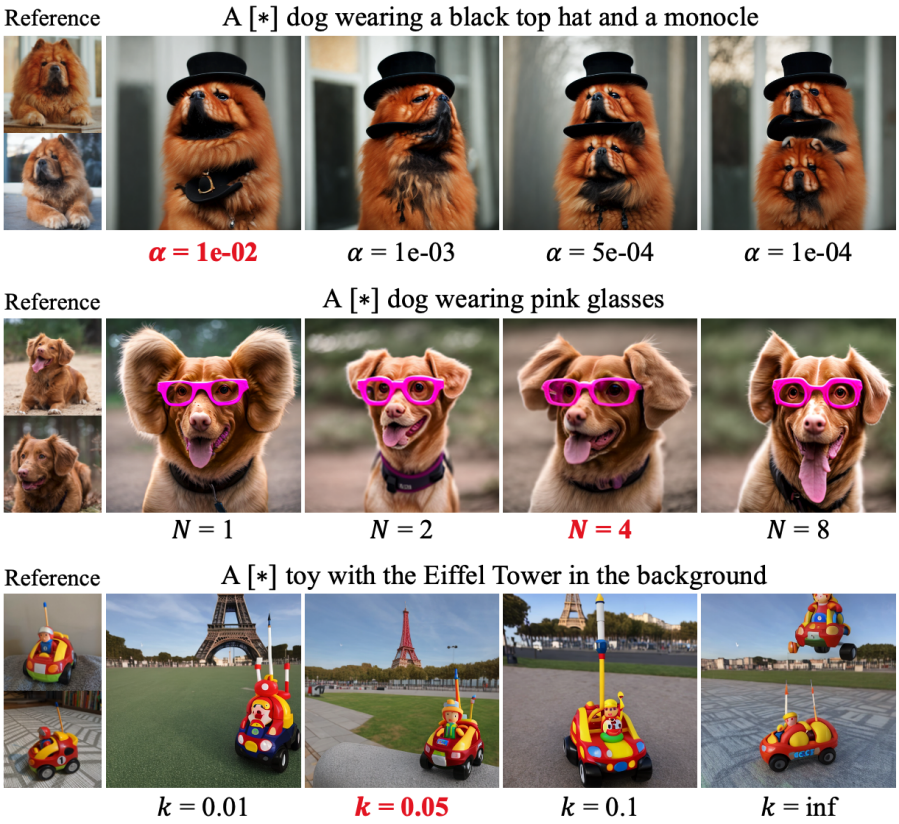



Figure 6: Hyperparameter selection for Mix-Opt in SD V2.1, with the low-resolution ratio set to 0.5. We analyze the effects of different choices for the MeZO learning rate $\alpha$, gradient accumulation steps $N$, and the steepness parameter $k$.

| Method | DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|
| LoRA-BP-High | 0.6762 | **0.8075** | 0.2961 | 5.09 |
| LoRA-BP-High† + 4-bit | 0.6575 | 0.7991 | **0.2964** | 2.58 |
| **Ours**† ($r = 0.5$) | **0.6776** | 0.8069 | 0.2926 | 2.08 |
| **Ours**† ($r = 0.5$) + 4-bit | 0.6742 | 0.7998 | 0.2922 | **1.67** |

Table 8: Results with 4-bit weight quantization on SD V2.1. † denotes mixed precision with fp16.

**Compatibility with 4-bit Weight Quantization.** We further evaluate whether Mix-Opt can be combined with weight quantization on SD V2.1. Following QLoRA [5], we quantize the frozen base-model weights to 4-bit precision using `bitsandbytes`, while retaining the trainable LoRA parameters, activations, and gradients in fp16. As shown in Table 8, 4-bit quantization reduces the memory usage of LoRA-BP-High from 5.09 GB to 2.58 GB. When combined with Mix-Opt, the memory requirement is further reduced to 1.67 GB while maintaining competitive personalization performance. These results show that Mix-Opt is complementary to base-weight quantization.

## 5 Conclusion

In this work, we introduced Mix-Opt, a memory-efficient framework for personalization of text-to-image diffusion models. By integrating BP-low and ZO-high through a timestep-aware probabilistic function, Mix-Opt balances personalization quality and structural consistency while significantly reducing memory consumption. Extensive experiments across multiple diffusion models demonstrated its scalability, generalization, and efficiency, achieving performance comparable to BP-high with substantially lower memory requirements. While our study evaluated memory-efficient training on GPUs, an important future direction is deploying Mix-Opt in real on-device environments, optimizing for hardware constraints, and refining inference efficiency.

## References


[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[2] Daewon Chae, Nokyung Park, Jinkyu Kim, and Kimin Lee. Instructbooth: Instruction-following personalized text-to-image generation. *arXiv preprint arXiv:2312.03011*, 2023.

[3] Wonguk Cho, Seokeon Choi, Debasmit Das, Matthias Reisser, Taesup Kim, Sungrack Yun, and Fatih Porikli. Hollowed net for on-device personalization of text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 37:43058–43079, 2025.

[4] Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11472–11481, 2022.

[5] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023.

[6] 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.

[7] Zhengcong Fei, Mingyuan Fan, and Junshi Huang. Gradient-free textual inversion. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1364–1373, 2023.

[8] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *arXiv preprint arXiv:2306.09344*, 2023.

[9] 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. *arXiv preprint arXiv:2208.01618*, 2022.

[10] Yatharth Gupta, Vishnu V Jaddipal, Harish Prabhala, Sayak Paul, and Patrick Von Platen. Progressive knowledge distillation of stable diffusion xl using layer level loss. *arXiv preprint arXiv:2401.02677*, 2024.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[12] Liangwei Jiang, Ruida Li, Zhifeng Zhang, Shuo Fang, and Chenguang Ma. Emojidiff: Advanced facial expression control with high identity preservation in portrait generation. *arXiv preprint arXiv:2412.01254*, 2024.

[13] Maxwell Jones, Sheng-Yu Wang, Nupur Kumari, David Bau, and Jun-Yan Zhu. Customizing text-to-image models with a single image pair. In *SIGGRAPH Asia 2024 conference papers*, pages 1–13, 2024.

[14] Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. *Advances in Neural Information Processing Systems*, 36:36187–36207, 2023.

[15] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1931–1941, 2023.

[16] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[17] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems*, 36:30146–30166, 2023.

[18] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

[19] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[20] Yuqi Peng, Lingtao Zheng, Yufeng Yang, Yi Huang, Mingfu Yan, Jianzhuang Liu, and Shifeng Chen. Tara: Token-aware lora for composable personalization in diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 8385–8393, 2026.

[21] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[23] 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.

[24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[25] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[26] Hyogon Ryu, Seohyun Lim, and Hyunjung Shim. Memory-efficient fine-tuning for quantized diffusion model. In *European Conference on Computer Vision*, pages 356–372. Springer, 2024.

[27] Hoigi Seo, Wongi Jeong, Kyungryeol Lee, and Se Young Chun. Efficient personalization of quantized diffusion model without backpropagation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7717–7727, 2025.

[28] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8543–8552, 2024.

[29] Vera Soboleva, Aibek Alanov, Andrey Kuznetsov, and Konstantin Sobolev. T-lora: Single image diffusion model customization without overfitting. *arXiv preprint arXiv:2507.05964*, 2025.

[30] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE transactions on automatic control*, 37(3):332–341, 2002.

[31] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15943–15953, 2023.

[32] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models. *arXiv preprint arXiv:2110.11323*, 2021.

[33] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, et al. Sana: Efficient high-resolution image synthesis with linear diffusion transformers. *arXiv preprint arXiv:2410.10629*, 2024.

[34] Yifei Zeng, Yuanxun Lu, Xinya Ji, Yao Yao, Hao Zhu, and Xun Cao. Avatarbooth: High-quality and customizable 3d human avatar generation. *arXiv preprint arXiv:2306.09864*, 2023.

[35] Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. Jedi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6786–6795, 2024.

# A Reproducibility

We provide implementation details and pseudocode in the main paper to ensure reproducibility. All experiments were repeated four times, and the results were averaged. Experiments were conducted using `diffusers 0.32.0` on top of the DreamBooth LoRA module.

# B Theoretical Interpretation of DTAP

We provide a relative-risk interpretation of DTAP by considering the errors of the BP-low and ZO-high gradient estimators with respect to the desired high-resolution gradient $g_H(i,t)$. We define their risks as

$$R_B(t) = \mathbb{E}\left[\|\hat{g}_B(i,t) - g_H(i,t)\|_2^2\right], \tag{6}$$

and

$$R_Z(i,t) = \mathbb{E}\left[\|\hat{g}_Z(i,t) - g_H(i,t)\|_2^2\right]. \tag{7}$$

Here, $R_B(t)$ reflects the bias introduced by optimizing a low-resolution surrogate objective, whereas $R_Z(i,t)$ includes the smoothing bias and perturbation variance of the zeroth-order estimator. The relative risks of the two estimators vary with the diffusion timestep and training progress. In particular, BP-low becomes less reliable at high timesteps because low-resolution optimization can reinforce inaccurate global structures, whereas ZO-high becomes more effective after the model has acquired target-specific information through partial adaptation.

Using an entropy-regularized soft selection between the two estimators, the probability of selecting ZO-high can be expressed as

$$p^*_{\mathrm{ZO}}(i,t) = \mathrm{sigmoid}\left(\frac{R_B(t) - R_Z(i,t)}{\tau}\right), \tag{8}$$

where $\tau$ controls the smoothness of the selection. Thus, ZO-high is selected more frequently when the relative risk of BP-low is higher.

Around a point at which the two estimators have similar risks, we locally approximate their relative risk as

$$R_B(t) - R_Z(i,t) \approx a(t - t_0) + b(i - i_0), \tag{9}$$

where $a > 0$ reflects the increasing relative risk of BP-low at higher timesteps, and $b > 0$ reflects the increasing effectiveness of ZO-high as training progresses. Substituting this approximation into Eq. 8 yields a sigmoid selection probability with a transition boundary that decreases linearly over the training steps. This corresponds to the form of DTAP used in Eq. 5.

This derivation provides a local relative-risk interpretation of DTAP rather than a global optimality guarantee. The DTAP parameters, including $k$ and $t_{\mathrm{mid}}$, are empirically selected.

# C Additional Experiments

**Multi-Concept Personalization with TARA.** We combine Mix-Opt with TARA [20] to evaluate its applicability to multi-concept personalization. TARA independently trains a concept-specific LoRA module and composes the trained modules during inference. We replace BP-high with Mix-Opt during the training of each module while retaining the original TARA objective and inference procedure. As shown in Table 9, Mix-Opt reduces peak optimization memory while maintaining comparable multi-concept personalization performance. In particular, the $r = 0.5$ setting reduces memory from 2.76 GB to 2.07 GB and achieves a higher CLIP-I score than BP-high.

**Style Personalization with PairCustomization.** We further combine Mix-Opt with PairCustomization [13] to evaluate its applicability to style personalization. We retain the original PairCustomization objective and replace BP-high with Mix-Opt during the training of each style-LoRA. For evaluation, we use text prompts from StyleAlign [32] and generate 800 images for each optimization setting. We report the DreamSim distance [8] between the target-style references and generated images, where a lower value indicates better perceptual style alignment. As shown in Table 10, Mix-Opt reduces peak optimization memory while maintaining comparable style alignment on both SD V1.5 and SD V2.1.

| Method | DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|
| TARA + BP-high† | **0.2744** | 0.6179 | 0.3165 | 2.76 |
| TARA + Ours† ($r = 0.75$) | 0.2653 | 0.6218 | **0.3173** | 2.44 |
| TARA + Ours† ($r = 0.5$) | 0.2658 | **0.6252** | 0.3159 | **2.07** |

Table 9: Multi-concept personalization results with TARA on SD V1.5. † denotes mixed precision with fp16.

| Base Model | Method | DreamSim ↓ | Mem. (GB) ↓ |
|---|---|---|---|
| SD V1.5 | PairCustom + BP-high† | 0.7791 | 2.76 |
| | PairCustom + Ours† ($r = 0.75$) | 0.7816 | 2.44 |
| | PairCustom + Ours† ($r = 0.5$) | **0.7712** | **2.07** |
| SD V2.1 | PairCustom + BP-high† | **0.7901** | 2.77 |
| | PairCustom + Ours† ($r = 0.75$) | 0.8005 | 2.44 |
| | PairCustom + Ours† ($r = 0.5$) | 0.7991 | **2.08** |

Table 10: Style-personalization results with PairCustomization. Lower DreamSim indicates better style alignment, and † denotes mixed precision with fp16.

**Ablation Study.** As shown in Table 11, we evaluate Mix-Opt under four DreamBooth configurations by varying LoRA rank ($R \in \{4, 128\}$) and enabling or disabling prior preservation loss to assess model-agnostic performance. Across all settings, Mix-Opt remains stable and effective. At high rank ($R = 128$), LoRA-BP-High occasionally produces saturation-like artifacts, so the number of iterations was reduced to 500. This demonstrates that Mix-Opt can robustly adapt its optimization strategy even when the iteration count changes, dynamically selecting methods as needed. Moreover, with prior preservation loss enabled, Mix-Opt further enhances text alignment and shows robustness even when richer loss functions are incorporated, while still saving memory.

| | Method | Performance Metrics DINO | CLIP-I | CLIP-T | Mem. (GB) |
|---|---|---|---|---|---|
| **R-4** | LoRA-BP-High | 0.6762 | **0.8075** | 0.2961 | 5.09 |
| | Ours (r=0.750) | 0.6758 | 0.8058 | **0.2969** | 4.42 |
| | Ours (r=0.625) | 0.6787 | 0.8043 | 0.2955 | 4.14 |
| | Ours (r=0.500) | **0.6833** | 0.8067 | 0.2938 | 3.92 |
| **R-4(pp)** | LoRA-BP-High | **0.6325** | **0.7803** | 0.3084 | 6.65 |
| | Ours (r=0.750) | 0.6313 | 0.7772 | 0.3103 | 5.31 |
| | Ours (r=0.625) | 0.6280 | 0.7723 | **0.3109** | 4.45 |
| | Ours (r=0.500) | 0.6126 | 0.7658 | 0.3103 | **4.32** |
| **R-128** | LoRA-BP-High | 0.7020 | 0.8167 | **0.2905** | 5.49 |
| | Ours (r=0.750) | 0.7095 | 0.8170 | 0.2893 | 4.78 |
| | Ours (r=0.625) | **0.7121** | **0.8201** | 0.2886 | 4.48 |
| | Ours (r=0.500) | 0.6913 | 0.8132 | 0.2845 | **4.25** |
| **R-128(pp)** | LoRA-BP-High | 0.6520 | 0.7981 | 0.2855 | 7.13 |
| | Ours (r=0.750) | 0.6793 | **0.8006** | 0.2897 | 5.72 |
| | Ours (r=0.625) | **0.6811** | 0.7977 | **0.2918** | 5.15 |
| | Ours (r=0.500) | 0.6776 | 0.7998 | 0.2909 | **4.67** |

Table 11: Impact of LoRA rank (R) and prior preservation loss (pp) on performance and memory consumption in SD V2.1. Rank-128 is trained for only 500 iterations due to artifacts in BP-High.

# D Analysis on DTAP Function

Dynamic Timestep-Aware Probability (DTAP) introduced in the main paper defines the probability of selecting ZO-high at step $(i, t)$ as:

$$p_{i,t}^{zo} = \frac{1}{1 + e^{-k(t - t_{\text{dyn}}(i))}},$$

where $t_{\text{dyn}}(i)$ is modeled as a linear function of $i$:

$$t_{\text{dyn}}(i) = t_{\text{start}} + \frac{i}{i_{\text{max}}}(t_{\text{end}} - t_{\text{start}}) = t_{\text{max}} + \frac{2i}{i_{\text{max}}}(t_{\text{mid}} - t_{\text{max}}),$$

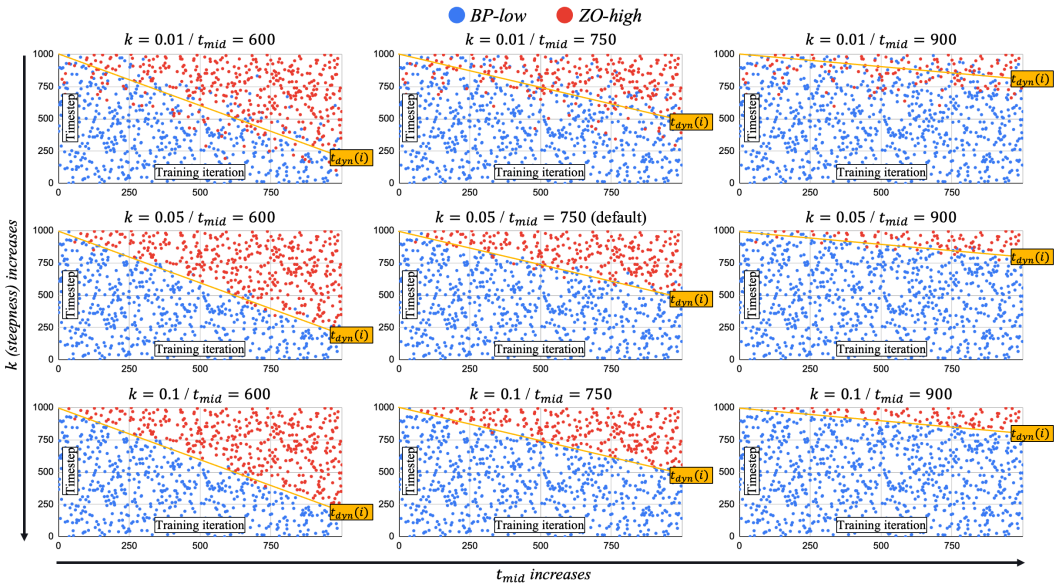



Figure 7: DTAP selection patterns across training iterations (x-axis) and timesteps (y-axis). Blue points represent BP-low and red points represent ZO-high. The boundary is defined as a linear function of $i$, given by $t_{\text{dyn}}(i)$, illustrating how $k$ and $t_{\text{mid}}$ influence the transition.

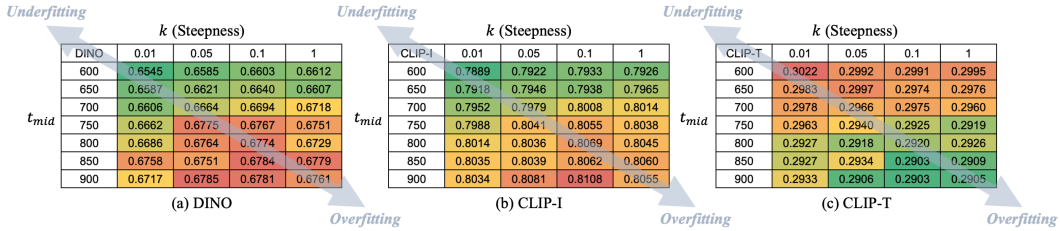



Figure 8: Effect of $k$ and $t_{\text{mid}}$ on Mix-Opt performance for SD V2.1 ($r = 0.5$). Heatmaps use red for higher scores and green for lower scores. Higher $t_{\text{mid}}$ (lower ZO-high frequency) increases DINO and CLIP-I but reduces CLIP-T, indicating overfitting risks.

where $t_{\text{start}} = t_{\max}$ and $t_{\text{end}} = 2t_{\text{mid}} - t_{\max}$.

This design ensures BP-low dominates early iterations while ZO-high becomes more frequent later, mitigating overfitting and structural distortions. Fig. 7 illustrates how DTAP partitions the $(i, t)$ plane under different $(k, t_{\text{mid}})$ settings. The x-axis represents training iteration and the y-axis represents timestep. Blue points indicate BP-low and red points indicate ZO-high. In the experiments, timesteps are uniformly sampled within the range of 0 to 1000. The boundary separating BP and ZO is formed linearly with $i$ according to $t_{\text{dyn}}(i)$, showing how $k$ and $t_{\text{mid}}$ control the transition. Because $t_{\text{start}}$ is set to $t_{\max}$, early iterations behave similarly across settings, but as $t_{\text{mid}}$ decreases, ZO-high is selected more frequently in later training stages. At $i_{\text{mid}}$, the boundary aligns with $t_{\text{mid}}$, confirming the intended scheduling.

We also analyze the frequency of ZO-high selections. Under $k = 0.05$ and $i_{\max} = t_{\max} = 1000$, the average number of ZO-high selections decreases as $t_{\text{mid}}$ increases: 400 times at $t_{\text{mid}} = 600$, 246 times at $t_{\text{mid}} = 750$, and 101 times at $t_{\text{mid}} = 900$. This reflects the desired scheduling: higher $t_{\text{mid}}$ biases toward BP-low for longer. The parameter $k$ controls the sharpness of the BP-low/ZO-high transition. Larger $k$ yields a sharper boundary and more deterministic selection near $t \approx t_{\text{dyn}}(i)$; smaller $k$ increases stochastic mixing. In the limit, $k \to 0$ approaches random selection, whereas $k \to \infty$ approximates a hard threshold similar to a step function. Both extremes are undesirable; we find $k \in [0.01, 0.1]$ robust, with $k = 0.05$ used by default.

Fig. 8 shows performance heatmaps for SD V2.1 with $r = 0.5$ across $k$ and $t_{\text{mid}}$. Colors range from red (high score) to green (low score). Higher $t_{\text{mid}}$ (which corresponds to lower ZO-high frequency)

| Training Cost (TFLOPS) | N=1 | N=2 | N=4 | N=8 |
|---|---|---|---|---|
| Mix-Opt (r=0.750) | 1.5026 | 1.8370 | 2.5057 | 3.8432 |
| Mix-Opt (r=0.625) | 1.2417 | 1.5761 | 2.2448 | 3.5823 |
| Mix-Opt (r=0.500) | 1.0283 | 1.3627 | 2.0314 | 3.3689 |
| Mix-Opt (r=0.375) | 0.8623 | 1.1967 | 1.8654 | 3.2029 |

Table 12: Average training computational cost (lower is better) per iteration under Mix-Opt for different $r$ and $N$. Mix-Opt can achieve up to 58% reduction by lowering $r$ and $N$. BP-high baseline is 2.0622 TFLOPS per iteration. Values lower than BP-high are highlighted.

| Training Time (s) | N=1 | N=2 | N=4 | N=8 |
|---|---|---|---|---|
| Mix-Opt (r=0.750) | 184.5 | 211.9 | 288.1 | 431.2 |
| Mix-Opt (r=0.625) | 189.3 | 231.4 | 294.8 | 431.5 |
| Mix-Opt (r=0.500) | 177.8 | 210.6 | 282.8 | 416.6 |
| Mix-Opt (r=0.375) | 179.1 | 217.7 | 280.1 | 450.1 |

Table 13: Training time (lower is better) over 1000 iterations for different $r$ and $N$. Reducing $N$ provides the largest speedup, while changes in $r$ have a smaller effect. BP-high baseline is 251.4 seconds. Values lower than BP-high are highlighted.

| Subject Fidelity | N=1 | N=2 | N=4 | N=8 |
|---|---|---|---|---|
| Mix-Opt (r=0.750) | 0.7396 | 0.7434 | 0.7408 | 0.7370 |
| Mix-Opt (r=0.625) | 0.7454 | 0.7420 | 0.7415 | 0.7442 |
| Mix-Opt (r=0.500) | 0.7361 | 0.7355 | 0.7450 | 0.7356 |
| Mix-Opt (r=0.375) | 0.7184 | 0.7171 | 0.7213 | 0.7203 |

Table 14: Subject fidelity (higher is better) measured by the average of DINO and CLIP-I scores across $r$ and $N$. Performance remains stable down to $r = 0.5$ with a slight drop at $r = 0.375$, and variation in $N$ has minimal impact. BP-high baseline is 0.7419. Values higher than BP-high are highlighted.

tends to increase DINO and CLIP-I but reduce CLIP-T, indicating overfitting risks. Even when object structures are visually distorted (e.g., overlapping shapes), DINO and CLIP-I remain high if local features match the reference, while CLIP-T drops as text alignment deteriorates. These trends confirm that hyperparameter selection based solely on metrics can be misleading; visual inspection is recommended. We typically select $k \in [0.01, 0.1]$ and $t_{\text{mid}} \in [700, 800]$ for balanced results. When $k = 1$, the transition becomes too abrupt, slightly degrading performance as shown in Fig. 8. For practical adjustment, increase $(k, t_{\text{mid}})$ when underfitting occurs and decrease them when overfitting or structural distortion appears.

## E Training Computation and Time

We analyze the training computational cost in terms of average FLOPs per iteration (TFLOPs) and total training time (s). Two key factors influence efficiency: the resolution ratio $r$ for BP-low and the number of MeZO gradient accumulation steps $N$. Table 12 summarizes average TFLOPS per iteration, Table 13 shows training time over 1000 iterations, and Table 14 reports subject fidelity scores.

Training computation decreases substantially as $r$ and $N$ are reduced, enabling significant efficiency gains compared to BP-high (2.0622 TFLOPS). For example, $(r = 0.5, N = 1)$ averages 1.0283 TFLOPS (about 50% reduction), and $(r = 0.375, N = 1)$ averages 0.8623 TFLOPS (about 58% reduction). These values are averaged over 1000 iterations under DTAP with $k = 0.05$ and $t_{\text{mid}} = 750$, where BP-low is selected approximately 754 times and ZO-high 246 times. From the perspective of training time, reducing $N$ provides the largest speedup because MeZO cost scales with the number of forward passes, while changes in $r$ have a smaller effect. Notably, even when $N$ is set to 1 or 2, the training time remains lower than BP-high, highlighting the efficiency advantage of the proposed approach.

As shown in Table 14, Subject fidelity, measured by average CLIP-I and DINO scores, remains stable down to $r = 0.5$ with only a slight drop at $r = 0.375$, and variation in $N$ has minimal impact

on personalization fidelity. Our default setting ($N = 4, r = 0.5$) offers a strong balance between efficiency and performance, though training time under this setting is slightly higher than BP-high. This can be mitigated by lowering $N$ to 2 without noticeable quality loss, providing a practical trade-off for scenarios where training time is critical.

To summarize, across its different variants, Mix-Opt simultaneously reduces training memory, computational cost, and training time compared to BP-high, while maintaining competitive performance. These improvements have direct implications for on-device deployment: smaller training memory enables personalization of larger models, and lower TFLOPS together with shorter training cycles directly translate to reduced energy consumption and thermal load, making personalization practical on resource-constrained hardware. Importantly, Mix-Opt achieves these gains without modifying the inference process, the standard timestep sampling schedule, or the training loss function.

# F Further Discussions

We highlight additional discussions and insights gained from our experiments.

**Layer-wise ZO Perturbation:** We explored an alternative ZO-High approach by splitting LoRA layers and independently estimating the approximate gradient for each split. This method aimed to improve gradient estimation by allowing finer-grained updates within the model. However, we observed no significant advantage over applying perturbations across the entire LoRA parameter space. The global perturbation strategy remained the most effective in maintaining stable personalization and refinement while keeping memory consumption low.

**Replacing ZO-High with Random Perturbation:** Instead of using ZO-High to approximate gradients, we tested a purely random perturbation approach. When the perturbation scale was too small, there was little effect on training, while larger perturbations led to gradient exploding. Even when the scale was carefully tuned for fairness in comparison with MeZO, personalization often degraded significantly. In contrast, MeZO effectively incorporates target-specific information, improving personalization performance and serving as a structural refinement mechanism for high-resolution supervision.

**Mix-Opt with Full Fine-Tuning:** We attempted to apply Mix-Opt in a full fine-tuning setting instead of LoRA. While Mix-Opt remained functional, several issues emerged. The backpropagation memory cost of low-resolution updates was significantly higher in full fine-tuning compared to LoRA-based optimization, reducing the effectiveness of our memory-efficient approach. Given that Mix-Opt is primarily designed for resource-efficient training, we focused on LoRA-based fine-tuning, which offers a more practical trade-off between memory consumption and personalization performance.

**Downsampling Method:** We experimented with different downsampling strategies for BP-low, including *resize*, *crop*, and *crop & resize*. The cropping method often resulted in excessive magnification of the subject, while crop & resize required significantly more iterations to achieve the same level of personalization as BP-high. Given these trade-offs, we opted for a simple resize operation, which provided a balance between memory efficiency and personalization quality.

**Future Directions:** An important direction is to evaluate Mix-Opt in real on-device environments. Although our experiments demonstrate substantial reductions in peak optimization memory, practical deployment also requires consideration of hardware-specific constraints, training latency, energy consumption, and numerical precision. Extending Mix-Opt beyond LoRA to other parameter-efficient fine-tuning techniques may further broaden its applicability. These directions could improve the practical deployment of memory-efficient personalization across resource-constrained devices and future diffusion architectures.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims are made in the abstract and introduction accurately.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses practical limitations and future directions in the conclusion and appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: We do not have theorems.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method, hyperparameters, dataset, and evaluation details are described in the paper and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While we do not explicitly release the code, all training datasets used in our experiments are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main training and evaluation settings are specified in the paper and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports averaged results but not error bars or significance tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the training memory and all experiments were conducted on NVIDIA A100 GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that our research adheres to all outlined principles.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper does not release any datasets or models that pose a risk of misuse.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release any datasets or models that pose a risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external datasets and codebases used in this work are properly cited.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new released assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper includes a user study and describes the details of the user study. But we did not conduct crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve any research with human subjects and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs were not used in any core methods of this research. They were only used for writing and editing assistance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.