# LEMMA: LEARNED MODEL MERGING FOR TEXT-TO-IMAGE GENERATION VIA MULTI-OBJECTIVE OPTIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Merging specialist text-to-image models into a unified generator can consolidate diverse generative capabilities while minimizing deployment overhead. However, existing methods rely on static, global heuristics that ignore layer-specific functionality and fail to balance multiple objectives. To address this, we propose LEMMA, a novel framework that formulates text-to-image model merging as a multi-objective optimization problem. LEMMA adaptively combines specialist task vectors through layer-tailored merging coefficients that are learned directly on the target objectives, while the base and specialist models remain frozen. We demonstrate LEMMA on two foundation architectures (SDXL and SD3) across two multi-objective setups, by merging reward specialists under multi-objective Direct Preference Optimization (DPO), and by merging style specialists under multi-objective Supervised Fine-Tuning (SFT). Extensive experiments show that LEMMA consistently outperforms prior merging baselines across both setups. Importantly, LEMMA often surpasses dedicated single-task specialists on their own target metrics, improving all objectives simultaneously rather than trading one capability for another. These gains are generalized to unobserved evaluation metrics and are corroborated by a user study with human raters, all while introducing no additional inference overhead.

## 1 INTRODUCTION

Pre-training text-to-image models on large datasets yields powerful foundation models (e.g., SDXL (Podell et al., 2024) and SD3 (Esser et al., 2024)), with impressive generative capabilities (Bommasani, 2021). For domain specific applications, these models are fine-tuned on curated datasets (Ruiz et al., 2023; Lee et al., 2023), producing specialist models dedicated for tasks such as artistic style emulation (Sohn et al., 2023) and aesthetic alignment (Lee et al., 2025). While specialization improves performance on target tasks, it introduces significant drawbacks. First, managing a growing library of specialist models incurs substantial storage and operational overhead, as each checkpoint must be served independently (Biggs et al., 2024). Furthermore, isolated fine-tuning prevents models from leveraging the complementary knowledge acquired by other specialists (Yadav et al., 2023).

Model merging has emerged as a promising approach to address these limitations (Wortsman et al., 2022a). By combining parameters from multiple specialists into a single checkpoint, model merging reduces deployment costs while integrating diverse, complementary capabilities. Earlier studies in image classification and natural language processing have demonstrated the efficacy of model merging (Wortsman et al., 2022a; Ilharco et al., 2022; Yang et al., 2024). For instance, Model Soups (Wortsman et al., 2022a) showed that simply averaging the parameters of models fine-tuned on the same task with different hyperparameters can yield a merged model outperforming any individual source model. Another study, Task Arithmetic (Ilharco et al., 2022), introduced task vectors that represent the difference calculated between fine-tuned and original model parameters, which are arithmetically manipulated to add, remove, or combine specific capabilities.

However, model merging techniques for text-to-image generation remain largely unexplored. Notable approaches include Diffusion Soup (Biggs et al., 2024), which computes a weighted average of parameters from diffusion models trained on different data shards, using either uniform or greedily searched merging weights. Another method, Decouple-then-Merge (Ma et al., 2025), first fine-tunes diffusion models on distinct diffusion timestep ranges to handle denoising task variations, then



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

combines these specialist models using task vector arithmetic, with merging weights determined via grid search. More recently, SELMA (Li et al., 2024) introduced a method to combine text-to-image models, by uniformly averaging the parameters of multiple low-rank adaptation modules.

While prior methods offer promising directions, they typically rely on static, depth-agnostic heuristics. Hence, they fail to leverage the heterogeneity of generative architectures, where distinct capabilities are localized across specific layers (e.g., attention, convolutional kernels). Note that forcing a fixed merging across layers can cause destructive interference between parameters, diluting unique specialist skills. Furthermore, the optimal contribution of any given layer from a specialist model is not intrinsic, varying with the specific set of specialist models. For example, the optimal merge weight for layer $l$ of specialist $S_1$ when combined with specialist $S_2$ can differ significantly from its optimal weight when $S_1$ is merged with specialists $S_2$ and $S_3$ due to complex, non-linear interactions and parameter conflicts. While learnable merging has shown efficacy in discriminative and autoregressive domains (Yang et al., 2024; Akiba et al., 2025), formulating a differentiable, layer-tailored framework for multi-objective text-to-image diffusion models remains an open challenge.

In this work, we propose **LE**arned **M**odel **M**erging via multi-objective optimiz**A**tion (**LEMMA**), a novel framework that formulates text-to-image model merging as an end-to-end, differentiable multi-objective optimization problem. Unlike static heuristics that treat all network layers identically, LEMMA synthesizes the merged model as a layer-adaptive combination of specialist task vectors anchored at the pretrained base, driven by a compact merge matrix with one coefficient per specialist per layer. These coefficients are unconstrained real values, so they can reinforce, deactivate, or remove any specialist skill at any layer, while the base and specialist models remain frozen. Importantly, these coefficients are learned via gradient descent directly on target multi-objectives, making LEMMA applicable to any goal. We demonstrate this by merging multiple specialists under a multi-objective Direct Preference Optimization (DPO) using Pareto-improving pairs, and by merging multiple style specialists under Supervised Fine-Tuning (SFT) objectives over pooled data.

Comprehensive experiments on two backbone architectures (SDXL and SD3) show that LEMMA consistently outperforms prior merging baselines, achieving the best score on nearly all reported metrics among competing methods in both DPO and SFT setups. Meanwhile, LEMMA frequently exceeds even the dedicated specialists on their own targeted metrics, improving all competing objectives simultaneously instead of trading one capability for another. Furthermore, these enhancements are generalized to unobserved evaluation metrics in test time and are corroborated by a user study.

Our contributions can be summarized as:

- To our knowledge, LEMMA is the first learnable merging framework for text-to-image generation that casts model merging as a multi-objective optimization problem, with formulations for both DPO and SFT setups.
- LEMMA introduces layer-adaptive merging coefficients that add, neutralize, or remove individual specialist skills at any network depth, learned end-to-end.
- Across two backbone architectures and both setups, LEMMA outperforms prior merging baselines and frequently surpasses dedicated specialists on their own target metrics.

## 2 Related Work

**Static and Heuristic Methods.** Model Soups (Wortsman et al., 2022a), demonstrates that averaging parameters from models fine-tuned with distinct hyperparameters improves accuracy. Task Arithmetic (Ilharco et al., 2022) defines task vectors as the difference between fine-tuned and pre-trained parameters to enable arithmetic operations. To address geometric complexities, WiSE-FT (Wortsman et al., 2022b) linearly interpolates fine-tuned weights with the original model, while SLERP (Jang et al., 2024) utilizes spherical interpolation to better preserve high-dimensional parameter spaces. Parameter interference is mitigated by TIES-Merging (Yadav et al., 2023) via trimming and sign-election mechanisms, and by DARE (Yu et al., 2024) and LEWIS (Chopra et al., 2025) through sparsity.

**Learnable Merging.** To overcome the gaps of static merging, RegMean (Jin et al., 2022) minimizes prediction differences between merged and source models, while evolutionary methods (Akiba et al., 2025) search for optimal recipes. Most relevant to our approach is AdaMerging (Yang et al., 2024), which learns merging coefficients via test-time entropy minimization for discriminative tasks.

In contrast, our method targets the generative text-to-image domain, formulating merging as an end-to-end multi-objective optimization framework, using DPO or SFT.

**Merging in Diffusion Models.** Model merging for diffusion models remains limited but is growing. Diffusion Soup (Biggs et al., 2024) averages models trained on different data shards for text-to-image generation. In the parameter-efficient regime, ZipLoRA (Shah et al., 2024) optimizes merging coefficients via test-time optimization to preserve content and style semantics, while LoRA.rar (Shenaj et al., 2025) amortizes this search by pre-training a hypernetwork. SELMA (Li et al., 2024) explores skill integration by averaging multi-skill LoRAs, and K-LoRA (Ouyang et al., 2025) adopts dynamic top-$k$ pruning. For full-checkpoint merging, DeMe (Ma et al., 2025) decouples training across distinct denoising intervals before merging, and WARP (Ramé et al., 2024) leverages weight-averaged reward models during generative RL fine-tuning.

## 3 PRELIMINARIES

**Latent Diffusion Models.** These models operate in the lower-dimensional latent space of a pre-trained Variational Autoencoder (VAE) (Rombach et al., 2022; Diederik & Max, 2019). Given an image $x$, an encoder $\mathcal{E}$ produces a latent representation $z_0 = \mathcal{E}(x)$. A forward diffusion process $q(z_t \mid z_0)$ progressively perturbs $z_0$ by adding Gaussian noise over discrete timesteps $t \in [1, T]$ according to a predefined variance schedule $\beta_t$:

$$q(z_t \mid z_0) = \mathcal{N}\left(z_t; \sqrt{\bar{\alpha}_t} z_0, (1 - \bar{\alpha}_t)\mathbf{I}\right), \tag{1}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. The reverse denoising process is parameterized by a neural network $\epsilon_\theta(z_t, t, c)$ conditioned on text prompt $c$, trained to minimize the simplified variational score-matching objective:

$$\mathcal{L}_{\text{LDM}}(\theta) = \mathbb{E}_{z_0, c, \epsilon \sim \mathcal{N}(0,\mathbf{I}), t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2\right]. \tag{2}$$

During inference, starting from pure Gaussian noise $z_T \sim \mathcal{N}(0, \mathbf{I})$, the model iteratively denoises the latent to synthesize a sample $\hat{z}_0$, which is mapped back to pixel space via the VAE decoder to produce the generated image $\hat{x} = \mathcal{D}(\hat{z}_0)$.

**Direct Preference Optimization (DPO).** To align diffusion models with human preferences without the instability of training explicit reward models, DPO for diffusion models (Rafailov et al., 2023; Wallace et al., 2024) directly optimizes a policy over a pairwise preference dataset $\mathcal{D}_{\text{pref}} = \{(c, x_w, x_l)\}$, where $x_w \succ x_l$ indicates that image $x_w$ is preferred over $x_l$ for a given prompt $c$. By approximating the intractable likelihood via the evidence lower bound, the implicit reward difference is expressed through the denoising error differential:

$$\delta_\theta(x_w, x_l) = \mathbb{E}_{t,\epsilon}\left[\|\epsilon - \epsilon_\theta(z_t(x_w), t, c)\|_2^2 - \|\epsilon - \epsilon_\theta(z_t(x_l), t, c)\|_2^2\right]. \tag{3}$$

The specialist model $\theta_{\text{spec}}$ is optimized by minimizing the negative log-likelihood of the Bradley-Terry preference model relative to a frozen reference model $\theta_{\text{ref}}$ (typically initialized as $\theta_{\text{base}}$):

$$\mathcal{L}_{\text{DPO}}(\theta_{\text{spec}}) = -\mathbb{E}_{(c,x_w,x_l)\sim\mathcal{D}_{\text{pref}}}\left[\log\sigma\left(\beta\left(\delta_{\theta_{\text{ref}}}(x_w, x_l) - \delta_{\theta_{\text{spec}}}(x_w, x_l)\right)\right)\right], \tag{4}$$

where $\beta$ is a temperature parameter governing the regularization strength and $\sigma(\cdot)$ is sigmoid.

**Supervised Fine-Tuning (SFT).** To fine-tune a pretrained base model $\theta_{\text{base}}$ to a specialized domain, SFT optimizes specialist parameters $\theta_{\text{spec}}$ over a curated dataset $\mathcal{D}_{\text{spec}} = \{(x, c)\}$ of image-text pairs. Under the rectified flow-matching framework (e.g., SD3), initialized from $\theta_{\text{base}}$, the specialist model is trained to predict the vector field velocity by minimizing:

$$\mathcal{L}_{\text{SFT}}(\theta_{\text{spec}}) = \mathbb{E}_{(x,c)\sim\mathcal{D}_{\text{spec}}, \epsilon\sim\mathcal{N}(0,\mathbf{I}), t\sim\mathcal{U}[0,1]}\left[\left\|v_{\theta_{\text{spec}}}(z_t, t, c) - (\epsilon - z_0)\right\|_2^2\right], \tag{5}$$

where $z_t = (1 - t)z_0 + t\epsilon$ with target velocity $\epsilon - z_0$.

**Model Merging.** Given a collection of $N$ specialist models $\{\theta_n\}_{n=1}^N$ trained or fine-tuned for distinct downstream tasks or objectives, model merging aims to combine their parameters into a single unified model $\theta_{\text{merged}}$. Formally, this is defined as a parameter merging function $\mathcal{M}$:

$$\theta_{\text{merged}} = \mathcal{M}(\theta_1, \theta_2, \ldots, \theta_N), \tag{6}$$

such that the merged model $\theta_{\text{merged}}$ retains and integrates the specialized capabilities of $\{\theta_i\}_{i=1}^N$.

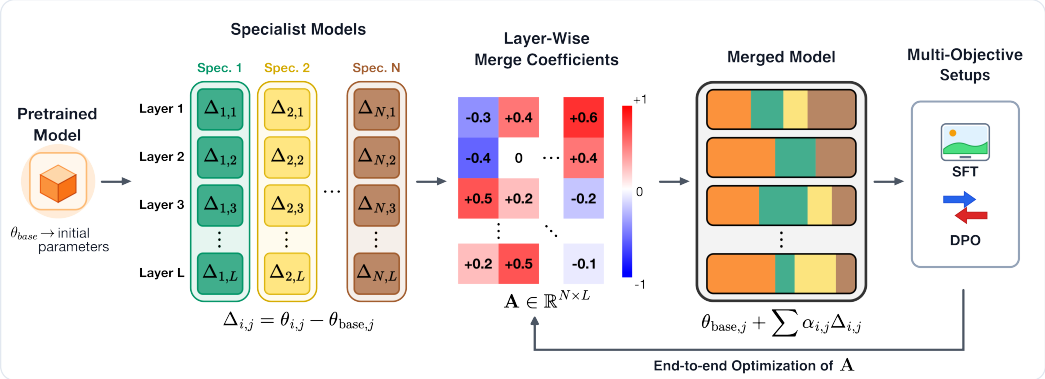



Figure 1: **Overview of LEMMA.** Specialist task vectors are combined via learnable, layer-specific scalar weights $\mathbf{A} = [\alpha_{i,j}] \in \mathbb{R}^{N \times L}$, optimized in multi-objective setups (e.g., SFT, DPO).

## 4 Learned Model Merging

We formulate model merging as an end-to-end multi-objective optimization problem. As illustrated in Fig. 1, our framework, LEMMA, modulates each specialist's contribution across network depth through learnable scalar weights. The base model and all specialist backbones remain frozen, and only a compact set of merging scalars is optimized directly against the target multi-objectives.

### 4.1 Layer-Tailored Merge Parametrization

Consider a pretrained base diffusion model $\theta_{\text{base}}$ and $N$ specialist models $\{\theta_i\}_{i=1}^N$ fine-tuned from $\theta_{\text{base}}$ on distinct target objectives. All models share an identical architecture of $L$ layers (e.g., individual attention projections, convolutional kernels, and biases):

$$\theta_{\text{base}} = \{\theta_{\text{base},j}\}_{j=1}^L, \qquad \theta_i = \{\theta_{i,j}\}_{j=1}^L, \quad \forall i \in \{1, \ldots, N\}. \tag{7}$$

For each specialist $i$ and layer $j$, we define the localized task vector as:

$$\Delta_{i,j} = \theta_{i,j} - \theta_{\text{base},j}. \tag{8}$$

To capture heterogeneity across network depth, LEMMA estimates each merged parameter tensor $\theta_{\text{merged},j}$ as a layer-specific linear combination anchored at the base parameter:

$$\theta_{\text{merged},j}(\mathbf{A}) = \theta_{\text{base},j} + \sum_{i=1}^{N} \alpha_{i,j} \Delta_{i,j}, \quad \forall j \in \{1, \ldots, L\}, \tag{9}$$

where $\alpha_{i,j} \in \mathbb{R}$ is a continuous, learnable scalar governing specialist $i$ at layer $j$. Collecting the coefficients yields the learnable merge matrix $\mathbf{A} \in \mathbb{R}^{N \times L}$:

$$\mathbf{A} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,L} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N,1} & \alpha_{N,2} & \cdots & \alpha_{N,L} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_1^\top \\ \vdots \\ \boldsymbol{\alpha}_N^\top \end{bmatrix}. \tag{10}$$

As illustrated in Fig. 1 by the merge coefficients for layers, $\alpha_{i,j} \in \mathbb{R}$ can take both positive and negative values, supporting constructive reinforcement ($\alpha > 0$), selective dormancy ($\alpha \approx 0$), and task subtraction ($\alpha < 0$). Importantly, because only $\mathbf{A}$ is optimized, the number of learnable parameters ($N \times L$ scalars) is orders of magnitude smaller than the original model.

### 4.2 End-to-End Multi-Objective Optimization

LEMMA learns the merge matrix $\mathbf{A}$ directly against a multi-objective function $\mathcal{L}_{\text{multi}}$. Each training step rebuilds the merged model $\theta_{\text{merged}}(\mathbf{A})$ from Eq. 9, evaluates it under $\mathcal{L}_{\text{multi}}$, and back-propagates the gradient through the merge construction into $\mathbf{A}$, with $\theta_{\text{base}}$ and $\{\theta_i\}_{i=1}^N$ held frozen. The formulation accommodates any objective differentiable in the model parameters, and LEMMA instantiates it in two distinct setups: (i) multi-objective preference optimization over $M$ reward models via DPO, and (ii) multi-objective style generation over $K$ style specialists via SFT.

**Model Merging via Multi-Objective DPO.** LEMMA merges specialists trained on distinct reward axes such as prompt adherence and visual aesthetics by optimizing $\mathbf{A}$ over multiple reward preferences $\mathcal{D}_{\text{pref}}$ scored by $M$ reward models $\mathbf{R} = [r_1, \ldots, r_M]^\top$. Following CaPO (Lee et al., 2025), the raw score $r_m(x_i, c)$ of candidate $x_i$ among $G$ generations for prompt $c$ is calibrated into a pairwise win-rate that places heterogenous rewards scales on common footing,

$$v_m(x_i, c) = \frac{1}{G-1} \sum_{j \neq i} \sigma \left( r_m(x_i, c) - r_m(x_j, c) \right), \tag{11}$$

where $\sigma(\cdot)$ is the logistic function and each summand is the probability that $x_i$ is preferred over competitor $x_j$ under reward $m$. Collecting win-rates across all $M$ rewards yields a normalized preference vector $\mathbf{v}(x_i, c) \in [0, 1]^M$. Preference pairs $(x_w, x_l)$ are drawn along the empirical Pareto frontier, $\mathbf{v}(x_w, c) \succeq \mathbf{v}(x_l, c)$, via Frontier-based Rejection Sampling (Lee et al., 2025), so the winner dominates the loser on every reward and no capability is traded for another. LEMMA then optimizes $\mathbf{A}$ under the resulting preference objective with the frozen base model as the reference $\theta_{\text{ref}} = \theta_{\text{base}}$,

$$\min_{\mathbf{A}} \ \mathcal{L}_{\text{DPO}}(\mathbf{A}) = -\mathbb{E}_{(c, x_w, x_l) \sim \mathcal{D}_{\text{pref}}} \left[ \log \sigma \left( \beta \cdot w(t) \left( \delta_{\theta_{\text{base}}}(x_w, x_l) - \delta_{\theta_{\text{merged}}(\mathbf{A})}(x_w, x_l) \right) \right) \right], \tag{12}$$

where $\delta_\theta$ is the denoising error differential between the winning and losing samples and $w(t)$ is an SNR-dependent weighting term. Since every sampled pair is Pareto-improving, the gradient reaching $\mathbf{A}$ reinforces all $M$ rewards jointly.

**Model Merging via Multi-Objective SFT.** LEMMA uses a pooled data $\mathcal{D}_{\text{pool}} = \bigcup_{k=1}^{K} \mathcal{D}_k$ from $K$ specialist domains, where every caption $c$ carries the style tag (Appendix A), so a sample drawn from the pool selects exactly one domain. Writing $\mathbf{1}_k(z_0, c)$ for the indicator that a sample originates from $\mathcal{D}_k$, LEMMA trains $\mathbf{A}$ on the gated sum of the per-domain score-matching objectives of Eq. 5,

$$\min_{\mathbf{A}} \ \mathcal{L}_{\text{SFT}}(\mathbf{A}) = \mathbb{E} \left[ \sum_{k=1}^{K} \mathbf{1}_k(z_0, c)\, w(t) \left\| v_{\theta_{\text{merged}}(\mathbf{A})}(z_t, t, c) - (\epsilon - z_0) \right\|_2^2 \right], \tag{13}$$

where the expectation is taken over $(z_0, c) \sim \mathcal{D}_{\text{pool}}$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim \mathcal{U}[0, 1]$, with $z_t = (1-t)z_0 + t\epsilon$ and $w(t)$ an SNR-dependent weighting term. The indicator activates a single domain term per draw, and taking the expectation recovers $\mathcal{L}_{\text{SFT}}(\mathbf{A}) = \sum_{k=1}^{K} \pi_k \mathcal{L}_k(\mathbf{A})$ with mixture weights $\pi_k = |\mathcal{D}_k| / |\mathcal{D}_{\text{pool}}|$, so one set of coefficients is optimized against all $K$ domain objectives at once and $\mathbf{A}$ settles where the domains reinforce one another across depth.

Algorithm 1 summarizes the procedure for both setups.

## 5 Experimental Details

**Setups.** We evaluate LEMMA across two setups representing distinct multi-objective optimization problems. In each setup, specialists are individually fine-tuned on a single objective, and then merged by optimizing the corresponding multi-objective problem. **Multi-Objective DPO:** Specialists are fine-tuned via single-objective DPO on individual rewards (general quality, aesthetics, and prompt alignment). LEMMA learns the merge weights via multi-objective DPO to assess balanced improvement across all rewards. **Multi-Objective SFT:** Specialists are fine-tuned via single-objective SFT on distinct art styles. LEMMA merges them via multi-objective SFT to unify stylistic capabilities.

**Algorithm 1** LEMMA

**Require:** Base model $\theta_{\text{base}}$, $N$ specialists $\{\theta_i\}_{i=1}^{N}$, merging dataset $\mathcal{D}$, learning rate $\eta$, target multi-objective function $\mathcal{L}_{\text{multi}} \in \{\mathcal{L}_{\text{DPO}}, \mathcal{L}_{\text{SFT}}\}$

1: **Precompute:** $\Delta_{i,j} = \theta_{i,j} - \theta_{\text{base},j}$ for all specialists $i$ and layers $j$
2: **Initialize:** $\mathbf{A} = [\alpha_{i,j}] \leftarrow \frac{1}{N}$
3: **while** not converged **do**
4: Sample batch $\mathcal{B} \sim \mathcal{D}$
5: Estimate parameters: $\theta_{\text{merged},j} \leftarrow \theta_{\text{base},j} + \sum_{i=1}^{N} \alpha_{i,j} \Delta_{i,j}$
6: Evaluate loss $\mathcal{L}_{\text{multi}}$ and update: $\mathbf{A} \leftarrow \mathbf{A} - \eta \nabla_{\mathbf{A}} \mathcal{L}_{\text{multi}}$
7: **end while**
8: **return** $\theta^*_{\text{merged}}$

Table 1: **Quantitative comparison for DPO via the SDXL backbone on PartiPrompts dataset.** Underline denotes target objective for specialists. **Bold** indicates the best performance across merging methods. † denotes statistically significant improvement over the strongest baseline ($p < 0.01$).

| | Target Metrics | | | | Held-Out Metrics | |
|---|---|---|---|---|---|---|
| **Method** | MPS ↑ | VILA ↑ | VQAScore ↑ | HZ-WS ↑ | HPS ↑ | PickScore ↑ |
| Pretrained | 10.87 | 5.92 | 0.8001 | 0.00 | 28.10 | 22.35 |
| Specialist (MPS) | <u>10.99</u> | 5.98 | 0.8053 | 3.13 | 28.47 | 22.41 |
| Specialist (VILA) | 10.95 | <u>6.21</u> | 0.8035 | 2.27 | 28.66 | 22.38 |
| Specialist (VQAScore) | 10.95 | 5.95 | <u>0.8050</u> | 2.36 | 28.26 | 22.40 |
| Mixture | 10.95 | 6.09 | 0.8054 | 2.81 | 28.80 | 22.44 |
| Dif. Soup [Biggs et al.] | 11.01 | 6.06 | 0.8067 | 4.11 | 28.57 | 22.41 |
| SLERP [Jang et al.] | 11.02 | 6.06 | 0.8054 | 3.72 | 28.58 | 22.41 |
| Task Arith.[Ilharco et al.] | 11.04 | 6.12 | 0.8077 | 4.85 | 28.72 | 22.43 |
| TIES [Yadav et al.] | 11.04 | 6.10 | 0.8080 | 4.86 | 28.67 | 22.42 |
| **LEMMA (Ours)** | **11.24**† | **6.13**† | **0.8148**† | **9.37**† | **29.54**† | **22.58**† |

**Datasets.** We conduct experiments on three datasets. In the **multi-objective DPO setup**, we construct synthetic preference pairs following (Lee et al., 2025) by sampling 100K prompts from **DiffusionDB** (Wang et al., 2023) and generating $G$=16 candidates per prompt using the pretrained model. Each candidate is scored by $M$=3 reward models (MPS, VILA, and VQAScore). Specialists are trained on pairs ranked by their single target reward, while Mixture and LEMMA use Pareto-improving pairs across all three. For testing, we evaluate on **PartiPrompts** (Yu et al., 2022), a dataset of 1.6K diverse prompts. In the **multi-objective SFT setup**, we use digitized paintings from **WikiArt**[1], focusing on $K$=4 artistic movements: Baroque, Expressionism, Impressionism, and Realism (∼35K images). Because raw entries contain only images, we use Gemini 2.5 Flash (Comanici et al., 2025) to generate captions. Details on data splits and captioning are provided in Appendix A.

**Backbone Models.** While LEMMA is model-agnostic, we demonstrate it on two foundation models. **Stable Diffusion XL (SDXL)** (Podell et al., 2024) is a latent diffusion U-Net with dual text encoders, sampled with a 50-step DDIM scheduler. **Stable Diffusion 3 (SD3)** (Esser et al., 2024) is a Multimodal Diffusion Transformer (MMDiT) trained with rectified flow matching and three text encoders, sampled with a 50-step Flow DPM-Solver. Both generate at $1024 \times 1024$ resolution.

**Implementation Details.** For each setup, we first fine-tune single-objective specialist models from the pretrained base: three DPO specialists (MPS, VILA, VQAScore), and four SFT specialists across WikiArt styles. In LEMMA, all base and specialist models remain frozen, learning only the layer-wise merge scalars $\mathbf{A} = [\alpha_{i,j}]$ initialized uniformly at $1/N$. We optimize $\mathbf{A}$ using AdamW, supervised by the DPO loss with $\theta_{\text{ref}} = \theta_{\text{base}}$ in the former setup and the flow-matching reconstruction loss in the latter. All hyperparameters are selected based on validation set performance (Appendix D for details).

**Evaluation Metrics.** We assess performance with dedicated metrics for each setup. In **multi-objective DPO**, we evaluate models on the three training reward objectives: **MPS** (Zhang et al., 2024) for overall visual quality, **VILA** (Ke et al., 2023) for aesthetic appeal, and **VQAScore** (Lin et al., 2024) for text-image alignment. To test robust out-of-distribution transfer, we also evaluate generalization on two held-out metrics: **PickScore** (Kirstain et al., 2023) and **HPS** (Wu et al., 2023). To summarize multi-objective performance into a single metric that penalizes one-sided trade-offs, we introduce the **Harmonic Z-Score (HZ-WS)**. We first standardize each metric $k \in \mathcal{K} = \{\text{MPS}, \text{VILA}, \text{VQAScore}\}$ for an evaluated model $\theta$ relative to the pretrained backbone $\theta_{\text{base}}$ as $Z_{\theta,k} = (\bar{s}_{\theta,k} - \mu_{\text{base},k})/\sigma_{\text{base},k}$, where $\bar{s}_{\theta,k}$ is the mean score of $\theta$ on metric $k$, and $\mu_{\text{base},k}$ and $\sigma_{\text{base},k}$ are the mean and standard deviation of the per-prompt scores of $\theta_{\text{base}}$, aggregating as:

$$\text{HZ-WS}_\theta = \begin{cases} 100K \Big/ \sum_{k\in\mathcal{K}} Z_{\theta,k}^{-1}, & \text{if } Z_{\theta,k} > 0 \ \forall k, \\ 100 \cdot \min_k Z_{\theta,k}, & \text{otherwise.} \end{cases} \tag{14}$$

Dominated by its lowest component, the harmonic mean rewards balanced multi-objective gains, while any regression relative to the pretrained backbone defaults to the worst negative $Z$-score.

In **multi-objective SFT**, reward models fail to capture specific artistic movement, so we instead measure how closely generations match the target style distribution via reference-based metrics. We evaluate generated samples against paired ground-truth paintings using **LPIPS** (Zhang et al., 2018) (↓) for fine-grained perceptual distance and **image-to-image CLIP similarity** (Radford et al., 2021) (↑) for high-level stylistic and semantic fidelity.

[1] https://www.wikiart.org/

Table 2: **Quantitative comparison for DPO via the SD-3 backbone on the PartiPrompts dataset.** Underline denotes target objective for specialists. **Bold** indicates the best performance across merging methods. † denotes statistically significant improvement over the strongest baseline ($p < 0.01$).

| | Target Metrics | | | | Held-Out Metrics | |
|---|---|---|---|---|---|---|
| **Method** | MPS ↑ | VILA ↑ | VQAScore ↑ | HZ-WS ↑ | HPS ↑ | PickScore ↑ |
| Pretrained | 11.49 | 5.84 | 0.8406 | 0.00 | 29.82 | 22.58 |
| Specialist (MPS) | 11.70 | 5.95 | 0.8423 | 1.89 | 30.30 | 22.70 |
| Specialist (VILA) | 11.55 | 5.98 | 0.8389 | −0.79 | 29.83 | 22.53 |
| Specialist (VQAScore) | 11.52 | 5.85 | 0.8460 | 0.99 | 29.70 | 22.61 |
| Mixture | 11.53 | 5.89 | 0.8407 | 0.13 | 29.93 | 22.59 |
| Dif. Soup [Biggs et al.] | 11.66 | 5.95 | 0.8424 | 1.92 | 30.30 | 22.67 |
| SLERP [Jang et al.] | 11.68 | 5.95 | 0.8427 | 2.24 | 30.31 | 22.67 |
| Task Arith.[Ilharco et al.] | 11.70 | 5.96 | 0.8449 | 3.85 | 30.40 | 22.69 |
| TIES [Yadav et al.] | 11.71 | 5.97 | **0.8453** | 4.07 | 30.39 | **22.70** |
| **LEMMA (Ours)** | **11.75**† | **5.99**† | **0.8453** | **4.35** | **30.46**† | 22.67 |

**Competing Methods.** We benchmark LEMMA against the pretrained base model, single-objective **Specialists**, and **Mixture**, which fine-tunes the base model on all objectives jointly in the corresponding setup. As merging baselines, we compare against **Diffusion Soup** (Biggs et al., 2024), a uniform parameter average, **Task Arithmetic** (Ilharco et al., 2022), a globally scaled sum of task vectors anchored at the base model, **SLERP** (Jang et al., 2024), spherical interpolation preserving parameter geometry, and **TIES-Merging** (Yadav et al., 2023) resolving parameter interference by trimming redundant weights and enforcing sign consensus.

**User Study.** We conduct a user study comparing images generated by LEMMA against the pretrained model and strongest merging baseline, with 12 raters evaluating 100 prompts per comparison (protocol in Appendix E.3).

# 6 Results

## 6.1 Multi-Objective DPO

**Quantitative Comparison.** Tables 1 and 2 report results on SDXL and SD-3 using the PartiPrompts dataset, respectively. As expected, single-objective specialists improve their target metric but degrade on untargeted objectives. For example, on SD-3, the VILA specialist improves VILA from 5.84 to 5.98, but its VQAScore drops from 0.8406 to 0.8389 (compared to pretrained model), resulting in a negative overall score (HZ-WS$= -0.79$). Mixture also struggles to balance the objectives, yielding a HZ-WS score of only 2.81 on SDXL and 0.13 on SD-3. While merging baselines like Task Arithmetic and TIES perform better than individual specialists, their gains are limited. In contrast, **LEMMA** achieves the best overall performance on both backbones. On SDXL, it reaches an HZ-WS of 9.37, outperforming the strongest baseline (TIES at 4.86, $p < 0.01$) and surpassing dedicated specialists on their own target metrics (11.24 vs. 10.99 on MPS, and 0.8148 vs. 0.8050 on VQAScore). On SD-3, LEMMA similarly attains the highest overall score (HZ-WS=4.35 vs. 4.07 for TIES), exceeding the individual MPS (11.75 vs. 11.70) and VILA (5.99 vs. 5.98) specialists on their targets. Evaluating metrics on metrics excluded from training objectives (HPS and PickScore) confirms that these gains reflect generalizable image quality: LEMMA achieves the top HPS score on

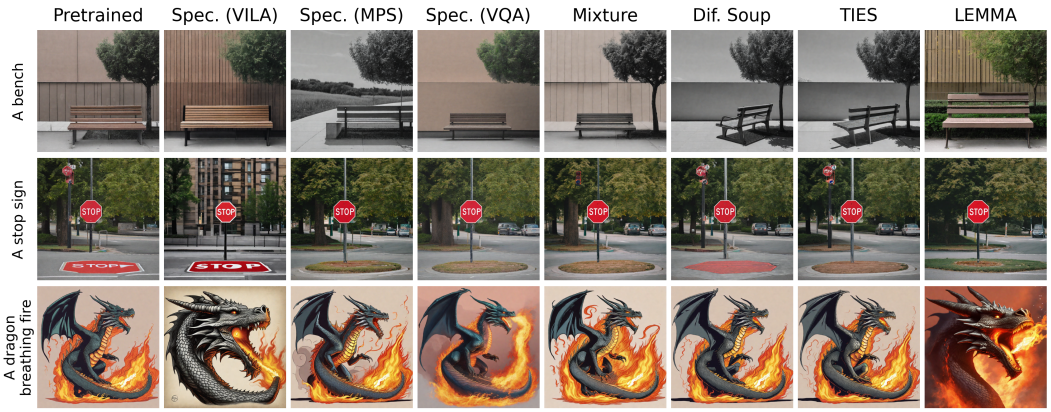



Figure 2: **Qualitative comparison of the competing methods for DPO on the SDXL backbone.**

Table 3: **Quantitative comparison for multi-style SFT on the SD-3 backbone.** Underlined values denote each specialist's targeted style, whereas **bold** indicates best overall performance. † denotes statistically significant improvement over the strongest baseline ($p < 0.01$, paired $t$-test).

| | Baroque | | Expressionism | | Impressionism | | Realism | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ |
| Pretrained | 0.7029 | 0.7770 | 0.7288 | 0.7780 | 0.7085 | 0.8102 | 0.6962 | 0.7878 | 0.7091 | 0.7883 |
| Specialist | 0.6615 | 0.8037 | 0.7053 | 0.7873 | 0.6844 | 0.8265 | 0.6680 | 0.8031 | 0.6798 | 0.8052 |
| Mixture | 0.6753 | 0.7962 | 0.7131 | 0.7867 | 0.6929 | 0.8218 | 0.6776 | 0.8002 | 0.6897 | 0.8012 |
| Dif. Soup [Biggs et al.] | 0.6694 | 0.7948 | 0.7108 | 0.7868 | 0.6896 | 0.8247 | 0.6729 | 0.8033 | 0.6857 | 0.8024 |
| SLERP [Jang et al.] | 0.6697 | 0.7960 | 0.7099 | 0.7863 | 0.6880 | 0.8256 | 0.6730 | 0.8029 | 0.6852 | 0.8027 |
| Task Arith.[Ilharco et al.] | 0.6688 | 0.8018 | 0.7152 | 0.7891 | 0.6938 | 0.8278 | 0.6759 | 0.8057 | 0.6884 | 0.8061 |
| TIES [Yadav et al.] | 0.6688 | 0.8024 | 0.7154 | 0.7889 | 0.6938 | 0.8276 | 0.6761 | 0.8061 | 0.6885 | 0.8063 |
| **LEMMA (Ours)** | **0.6593**† | **0.8071** | **0.6989**† | **0.7962**† | **0.6814**† | **0.8313**† | **0.6664**† | **0.8116**† | **0.6765**† | **0.8116**† |

both backbones (29.54 on SDXL and 30.46 on SD-3) and the highest PickScore on SDXL (22.58), demonstrating robust zero-shot transfer.

**Visual Quality.** Qualitative comparisons on SDXL (Fig.2) and SD-3 (Appendix Fig.A.2) show that LEMMA produces richer textures, fewer artifacts, and better prompt adherence than competing methods. Where individual specialists and merging baselines often exhibit visual distortions or flat renderings, LEMMA reliably generates photorealistic, coherent images.

**Representation Analysis.** To understand how the merged model combines specialist behaviors, Fig. 3 projects the visual features of generated images into a 2D space. The specialists separate into distinct clusters, confirming that they capture different properties (quality, aesthetics, and prompt faithfulness). When projected into this same space, LEMMA lies directly in the center between all three clusters, showing that it successfully combines their strengths into a balanced model rather than collapsing toward any single objective (see Appendix E for further details).

### 6.2 Multi-Objective SFT

**Quantitative Comparison.** Table 3 reports LPIPS and CLIP similarity scores across the four WikiArt styles. While specialists improve upon the pretrained base in their targeted style, Mixture method suffers from cross-style interference and falls behind individual specialists (average LPIPS of 0.6897 vs. 0.6798). Prior merging methods improve over Mixture by combining specialist weights post-hoc, but their uniform, layer-agnostic heuristics cannot resolve conflicting parameter updates across styles, causing them to stay behind dedicated specialists on perceptual distance (e.g., SLERP average LPIPS of 0.6852 vs. 0.6798 compared to specialists). In contrast, by learning layer-tailored merging weights, LEMMA consistently outperforms competing methods across styles ($p < 0.01$). Notably, LEMMA surpasses the dedicated specialists on their own target styles, achieving the best overall average scores of 0.6765 LPIPS and 0.8116 CLIP.

**Visual Quality.** Fig. 4 presents qualitative comparisons alongside ground-truth paintings. LEMMA reproduces target artistic styles more faithfully while maintaining accurate prompt alignment. For example, in the Realism example (first row), it accurately renders the texture of the ruined yellow and pink wall with the arched opening. In the Expressionism example (second row), LEMMA captures the distinct painterly strokes for the figures aiming a bow with noticeably better visual coherence.

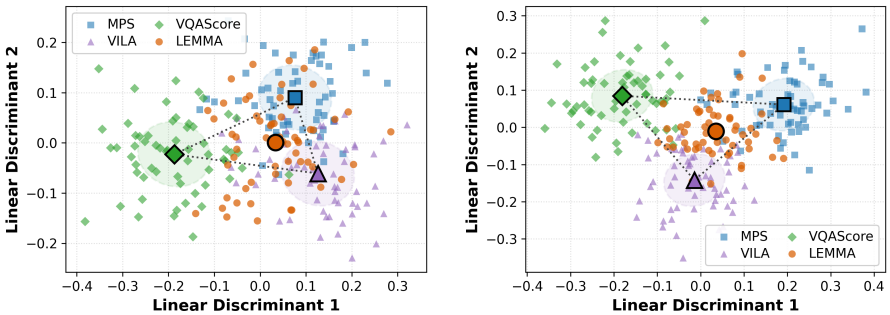



Figure 3: **Feature space analysis** on images from test prompts for SDXL (left) and SD-3 (right). Dashed ellipses denote cluster spread for each specialist. LEMMA lands centrally between the three specialists (LEMMA images were excluded when fitting the projection).

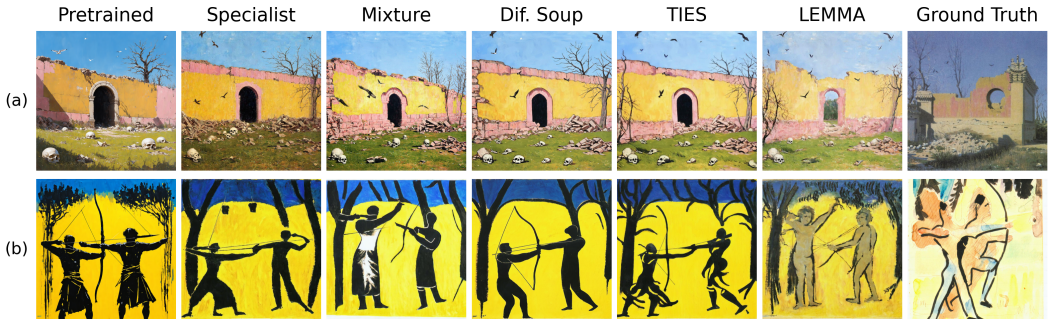



Figure 4: **Qualitative results for SFT on the SD-3 backbone.** (a) *"...ruined yellow and pink wall with an arched opening, ... with bare trees and flying birds. Skulls and rubble are scattered in the grassy foreground in realism style."*, (b) *"A stylized aiming a bow, against a yellow and blue background with dark tree-like shapes in expressionism style."*

Table 4: **Blind pairwise user study.** Tie-split preference scores for LEMMA, averaged over 12 raters, each judged 100 prompts per task (mean $\pm$ std). $^\dagger$ denotes statistical significance ($p < 0.01$).

| Comparison | SDXL (DPO) | SD3 (SFT) |
|---|---|---|
| LEMMA vs. Pretrained | $\mathbf{59.3 \pm 5.6}^\dagger$ | $\mathbf{83.3 \pm 9.2}^\dagger$ |
| LEMMA vs. TIES | $\mathbf{59.4 \pm 3.7}^\dagger$ | $\mathbf{60.7 \pm 7.7}^\dagger$ |

**Representation Analysis.** Appendix Fig. A.1 projects test images into a per-style 2D space fit on pretrained SD-3 and the corresponding specialists, together with ground-truth images (LEMMA excluded from the fit). For every style, the pretrained model sits far from the ground truth cluster and the specialist shifts toward it, yet LEMMA lands closer to the ground truth than the specialist itself, indicating that merging captures multiple artistic distributions without degradation (see Appendix E.2).

### 6.3 User Study

As reported in Table 4, human raters prefer LEMMA to the base models, with tie-split preference scores of 59.3 on SDXL (DPO) and 83.3 on SD-3 (SFT). LEMMA is also preferred over TIES, scoring 59.4 and 60.7, respectively (details in Appendix E.3).

### 6.4 Ablation Studies

We evaluate the impact of merge granularity in LEMMA. Coarse uses one scalar per specialist, Intermediate one scalar per architectural block, and Fine-Grained one scalar per layer, which is our proposed LEMMA. As shown in Table 5 (SDXL DPO) and Table 6 (SD3 SFT), performance improves consistently with granularity in both setups. This confirms that layer-wise merging in LEMMA is necessary to resolve parameter conflicts that are localized in depth, and it is efficient: fewer than $5.1 \times 10^3$ added parameters ($< 0.0002\%$ of the backbone), see Appendix D.4.

Table 5: Ablation results on merge granularity for multi-objective DPO (SDXL).

| | Target Metrics | | | | Held-Out Metrics | |
|---|---|---|---|---|---|---|
| **Merge Granularity** | MPS ↑ | VILA ↑ | VQAScore ↑ | HZ-WS ↑ | HPS ↑ | PickScore ↑ |
| LEMMA (Coarse) | 10.96 | 6.14 | 0.8015 | 1.38 | 28.53 | 22.38 |
| LEMMA (Intermediate) | 11.13 | **6.15** | 0.8066 | 5.19 | 28.81 | 22.48 |
| **LEMMA (Fine-Grained)** | **11.24** | 6.13 | **0.8148** | **9.37** | **29.54** | **22.58** |

Table 6: Ablation results on merge granularity for multi-objective SFT (SD-3).

| | Baroque | | Expressionism | | Impressionism | | Realism | | Aggregate | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Merge Granularity** | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ | LPIPS ↓ | CLIP ↑ |
| LEMMA (Coarse) | 0.6694 | 0.7927 | 0.7070 | 0.7930 | 0.6875 | 0.8290 | 0.6730 | 0.8061 | 0.6842 | 0.8052 |
| LEMMA (Intermediate) | 0.6627 | 0.7958 | 0.7054 | 0.7918 | 0.6875 | 0.8266 | 0.6714 | 0.8014 | 0.6818 | 0.8039 |
| **LEMMA (Fine-Grained)** | **0.6593** | **0.8071** | **0.6989** | **0.7962** | **0.6814** | **0.8313** | **0.6664** | **0.8116** | **0.6765** | **0.8116** |

## 7 Conclusion

In this work, we proposed LEMMA, a model merging framework for text-to-image generation that learns layer-tailored merging coefficients while keeping the base and specialist backbones frozen. While prior learnable merging methods target discriminative or autoregressive models, LEMMA addresses text-to-image generation and formulates merging as an end-to-end multi-objective problem, optimizing the merge matrix directly on multi-objective DPO for reward specialists and multi-objective SFT for style specialists. Because the coefficients act per layer, specialist skills can be reinforced, neutralized, or removed at any depth, allowing LEMMA to outperform prior baselines and often the specialists themselves on their own metrics, at no additional inference cost.

## AI STATEMENT

We used Large Language Models (LLMs) for two purposes. First, for synthetic data preparation on the WikiArt dataset. Since the raw entries contain only images, we used an LLM to generate captions, yielding the paired data required for text-to-image training and inference. Second, for polishing the writing. LLMs were used to polish author written text for grammatical correctness and clarity, and to assist with minor formatting.

## REFERENCES

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2):195–204, 2025.

Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 257–274. Springer, 2024.

Rishi Bommasani. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Hetarth Chopra, Vidhi Rambhia, and Vikram Adve. Lewis (layer wise sparsity)–a training free guided model merging approach. *arXiv preprint arXiv:2503.03874*, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

P Kingma Diederik and Welling Max. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

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.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Young Kyun Jang, Dat Huynh, Ashish Shah, Wen-Kai Chen, and Ser-Nam Lim. Spherical linear interpolation and text-anchoring for zero-shot composed image retrieval. In *European Conference on Computer Vision*, pp. 239–254. Springer, 2024.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

Junjie Ke, Keren Ye, Jiahui Yu, Yonghui Wu, Peyman Milanfar, and Feng Yang. Vila: Learning image aesthetics from user comments with vision-language pretraining. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10041–10051. IEEE, 2023.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.

Kyungmin Lee, Xiahong Li, Qifei Wang, Junfeng He, Junjie Ke, Ming-Hsuan Yang, Irfan Essa, Jinwoo Shin, Feng Yang, and Yinxiao Li. Calibrated multi-preference optimization for aligning diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18465–18475, 2025.

Jialu Li, Jaemin Cho, Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. Selma: Learning and merging skill-specific text-to-image experts with auto-generated data. *Advances in Neural Information Processing Systems*, 37:38530–38558, 2024.

Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pp. 366–384. Springer, 2024.

Qianli Ma, Xuefei Ning, Dongrui Liu, Li Niu, and Linfeng Zhang. Decouple-then-merge: Finetune diffusion models as multi-task learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 23281–23291, 2025.

Ziheng Ouyang, Zhen Li, and Qibin Hou. K-lora: Unlocking training-free fusion of any subject and style loras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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. In *International Conference on Learning Representations*, volume 2024, pp. 1862–1874, 2024.

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*, pp. 8748–8763. PmLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Alexandre Ramé, Johan Ferret, Nino Vieillard, Robert Dadashi, Léonard Hussenot, Pierre-Louis Cedoz, Pier Giuseppe Sessa, Sertan Girgin, Arthur Douillard, and Olivier Bachem. Warp: On the benefits of weight averaged rewarded policies. *arXiv preprint arXiv:2406.16768*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

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*, pp. 22500–22510, 2023.

Viraj Shah, Nataniel Ruiz, Forrester Cole, Erika Lu, Svetlana Lazebnik, Yuanzhen Li, and Varun Jampani. Ziplora: Any subject in any style by effectively merging loras. In *European Conference on Computer Vision*, pp. 422–438. Springer, 2024.

Donald Shenaj, Ondrej Bohdal, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. Lora. rar: Learning to merge loras via hypernetworks for subject-style conditioned image generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16132–16142. IEEE, 2025.

Kihyuk Sohn, Lu Jiang, Jarred Barber, Kimin Lee, Nataniel Ruiz, Dilip Krishnan, Huiwen Chang, Yuanzhen Li, Irfan Essa, Michael Rubinstein, et al. Styledrop: Text-to-image synthesis of any style. *Advances in Neural Information Processing Systems*, 36:66860–66889, 2023.

Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffusiondb: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 893–911, 2023.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022a.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022b.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis, 2023. URL https://arxiv.org/abs/2306.09341.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36:7093–7115, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *International Conference on Learning Representations*, volume 2024, pp. 22743–22763, 2024.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

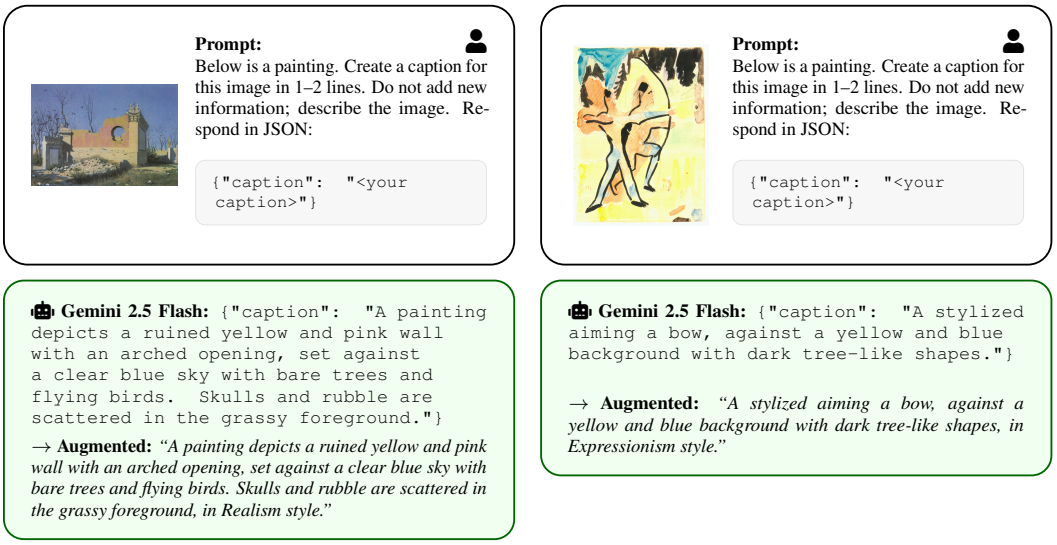
Sixian Zhang, Bohan Wang, Junqiang Wu, Yan Li, Tingting Gao, Di Zhang, and Zhongyuan Wang. Learning multi-dimensional human preference for text-to-image generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8018–8027. IEEE, 2024.

## A WIKIART DATASET

In the WikiArt dataset, we use four styles: Baroque (4,225 images), Expressionism (6,730), Impressionism (12,827), and Realism (10,720). Each style is split 90/5/5 into train, validation, and test sets. Each specialist is trained on the full training split of its own style. To keep the styles balanced, Mixture and LEMMA use at most 3,802 training images per style, which is the size of the Baroque training split, the smallest of the four.

Because raw WikiArt[2] images lack detailed text descriptions, we caption all training and testing images using Gemini 2.5 Flash Comanici et al. (2025) via the Vertex AI API (temperature $= 0.5$, top-$p$ $= 1.0$, fixed seed). The model is prompted to produce a concise 1–2 line summary of the visual content, strictly grounded in the image. This is then augmented with the Style tag of the corresponding image as shown below.

*"⟨Description⟩, in ⟨Style⟩ style"*




## B FRONTIER-BASED REJECTION SAMPLING (FRS)

For multi-objective preference pair construction we employ Frontier-based Rejection Sampling (FRS) (Lee et al., 2025), which constructs preference pairs via Pareto dominance across all reward dimensions. Given $M$ reward models and $G$ candidate generations per prompt, FRS applies non-dominated sorting to identify the upper and lower Pareto frontiers. A sample $x$ dominates $x'$ if and only if $R^{(j)}(x, c) \geq R^{(j)}(x', c)$ for all $j = 1, \ldots, M$. The set of non-dominated points forms the upper frontier $\mathcal{X}^+(c)$, while the set of dominated points forms the lower frontier $\mathcal{X}^-(c)$. Preference pairs are then constructed by sampling $x^+ \sim \mathcal{X}^+(c)$ and $x^- \sim \mathcal{X}^-(c)$, so that the DPO signal simultaneously pushes away from the lower frontier and pulls toward the upper frontier across all reward dimensions, without requiring manual reward weighting.

## C KL DIVERGENCE REGULARIZATION

In the DPO-based merge optimization, the implicit KL divergence regularization is controlled by the DPO $\beta$ parameter. The reference model $\pi_{\text{ref}}$ for the KL term is set to the pretrained base model, ensuring that the merged model stays close to the pre-trained distribution while improving along the specialist reward directions.

[2] https://www.wikiart.org/

# D Implementation Details

Hyperparameters are determined based on validation data performance.

## D.1 Specialists

**Specialists (DPO with the SDXL backbone):** We fine-tune the SDXL UNet with DPO on a synthetic preference dataset with $G = 16$ candidate generations per prompt. All three specialists (MPS, VILA, VQAScore) share the same hyperparameters of lr $= 1 \times 10^{-5}$, $\beta = 100$, and are trained for 5,000 steps, using a batch size of 256.

**Specialists (DPO with the SD3 backbone):** We fine-tune the SD3-Medium MMDiT with DPO using per-reward hyperparameters, where we use lr $= 2 \times 10^{-5}$, $\beta = 50$, 3,000 steps for MPS specialists, lr $= 1 \times 10^{-5}$, $\beta = 150$, 4,000 steps for VILA specialists, and lr $= 1.5 \times 10^{-5}$, $\beta = 30$, 5,000 steps for VQAScore specialist, using a batch size of 256.

**Specialists (SFT with the SD3 backbone):** We fine-tune four SFT specialists on the WikiArt dataset, with $lr = 3 \times 10^{-5}$ for 10,000 steps on Baroque, $lr = 5 \times 10^{-5}$ for 20,000 on Expressionism, $lr = 3 \times 10^{-5}$ for 20,000 on Impressionism, and $lr = 4 \times 10^{-5}$ for 5,000 on Realism, using a batch size of 128.

## D.2 The Mixture Baseline

The Mixture baseline fine-tunes a single model on all objectives concurrently, serving as the non-merging multi-objective reference:

**Mixture (DPO with the SDXL backbone):** As in the specialists, we fine-tune the SDXL UNet with DPO on the synthetic preference dataset with $G = 16$ candidates per prompts, using lr $= 1 \times 10^{-5}$, $\beta = 100$ for 5,000 steps with the FRS pair selection, using batch size of 256.

**Mixture (DPO with the SD3 backbone):** Similar to the SDXL backbone, we fine-tune the SD3-Medium MMDiT with DPO using FRS pair selection, using lr $= 5 \times 10^{-6}$, $\beta = 50$ for 5,000 steps, using a batch size of 256.

**SD3-Medium SFT:** Joint four-style SFT on the pooled WikiArt dataset using lr $= 1.5 \times 10^{-5}$ for 50,000 steps, using a batch size of 128.

## D.3 LEMMA Merge Optimization

With all specialist models frozen, LEMMA learns per-layer merging coefficients $\{\alpha_{i,j}\}$. All merge weights are initialized uniformly at $\alpha_{i,j} = 1/N$ (where $N$ is the number of specialists). We use AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$) with gradient clipping at max norm 1.0. All training uses bfloat16 precision.

**DPO with the SDXL backbone:** The merge weights span 1,680 leaf modules per specialist across the UNet in SDXL. We use the DPO loss with FRS pair selection, no weight decay, and a linear warmup of 100 steps. The selected configuration uses lr $= 0.1$, $\beta = 3{,}000$ for 20,000 steps with a batch size of 32.

**DPO with the SD3 backbone:** The merge weights span 490 parameters per specialist across the 24-block MMDiT architecture. The selected configuration uses lr $= 6 \times 10^{-4}$, $\beta = 200$ for 10,000 steps with a batch size of 32.

**SFT with the SD3 backbone:** The merge weights spanning 490 parameters are optimized over four style specialists using the flow-matching MSE reconstruction loss with signal-to-noise timestep weighting (loss bias $= -1.0$), for 5,000 steps, lr $= 5 \times 10^{-2}$, and a batch size of 8.

### D.4 MERGE GRANULARITY

We compare three levels of merge granularity:

**Coarse:** Assigns a single global scalar coefficient $\alpha_i \in \mathbb{R}$ per specialist model.

**Intermediate:** Assigns one scalar per top-level architectural block, where 12 paramaters in SDXL UNET (`conv_in`, `time_embedding`, `add_embedding`, `down_blocks_0..2`, `mid_block`, `up_blocks_0..2`, `conv_norm_out`, `conv_out`), and 28 blocks in SD3 (`DiTBlock_0..23` blocks, 4 `embedding/head` modules).

**Fine-Grained:** Assigns an independent scalar coefficient $\alpha_{i,j}$ to every leaf parameter tensor $j$ (1,680 leaf tensors per specialist in SDXL UNet; 490 leaf tensors per specialist in SD3 MMDiT).

Table A.1 summarizes the total learnable merge parameters across multi-objective optimization setups

Table A.1: **Total learnable merge parameters ($\alpha$) across merge granularities.**

| Architecture & Task | Granularity Level | Total Learnable Params |
|---|---|---|
| **SDXL + DPO** ($N = 3$ specialists) | Coarse | 3 |
| | Intermediate | 36 |
| | Fine-Grained | **5,040** |
| **SD3 + SFT** ($N = 4$ specialists) | Coarse | 4 |
| | Intermediate | 112 |
| | Fine-Grained | **1,960** |

## E REPRESENTATION MANIFOLD ANALYSIS DETAILS

Representation manifold analysis is performed for multi-objective DPO and multi-objective SFT generated images. In both cases, we fit Linear Discriminant Analysis (LDA) on training prompts and evaluate on held-out test prompts, with LEMMA excluded during manifold fitting.

### E.1 MULTI-OBJECTIVE DPO

**Feature Extraction:** We extract 2048-dimensional feature vectors using CLIP.

**Prompt Split:** We partition the 200 randomly selected PartiPrompts into 140 training prompts (70%) and 60 held-out test prompts (30%).

**Manifold Fitting and Projection:** Linear Discriminant Analysis (LDA) is fitted strictly on the images generated from the 140 training prompts using the three specialists. LEMMA is completely excluded during fitting. We then project images generated from the 60 held-out test prompts into this fixed 2D space.

### E.2 MULTI-OBJECTIVE SFT

**Feature Extraction:** Similar to the multi-objective DPO analysis, we extract 2048-dimensional feature vectors using CLIP.

**Prompt Split:** We evaluate four artistic styles from WikiArt: Baroque, Expressionism, Impressionism, and Realism. For each style, we select 200 prompts and split them into 140 training prompts (70%) and 60 held-out test prompts (30%). For Ground Truth, we use the corresponding WikiArt artwork for each prompt.

**Manifold Fitting and Projection:** For each style, we fit a 2D Linear Discriminant Analysis (LDA) manifold strictly on the 140 training prompts of Ground Truth, Pretrained SD3, and the single-style specialist. LEMMA is completely excluded during fitting. We then project representations from the 60 held-out test prompts into this fixed 2D space.

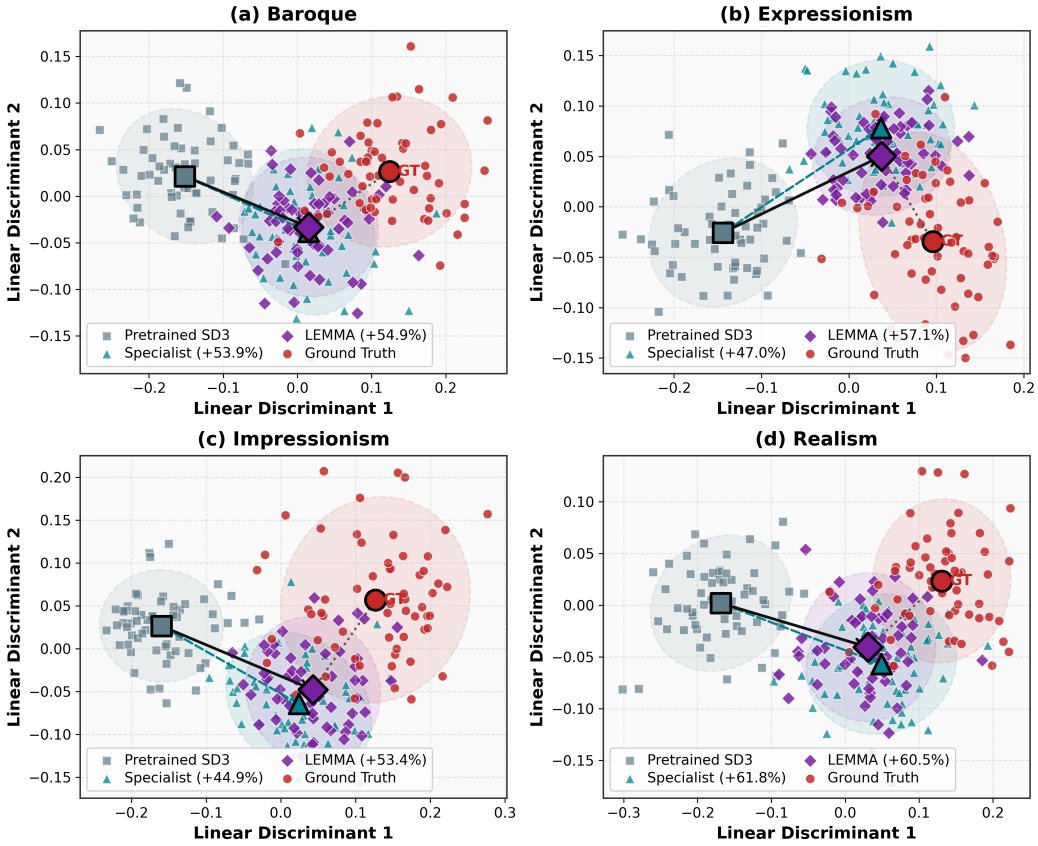



Figure A.1: **Style specialization on held-out test prompts, visualized with 2D Linear Discriminant Analysis (LDA).** Axes LD 1 and LD 2 are the first two linear discriminants. Arrows show the shift from the pretrained model (gray square) toward ground truth (colored circle), and dashed ellipses denote the 80% confidence spread. **LEMMA** (purple diamond) lands closer to the ground truth manifold than the dedicated specialist (teal triangle).

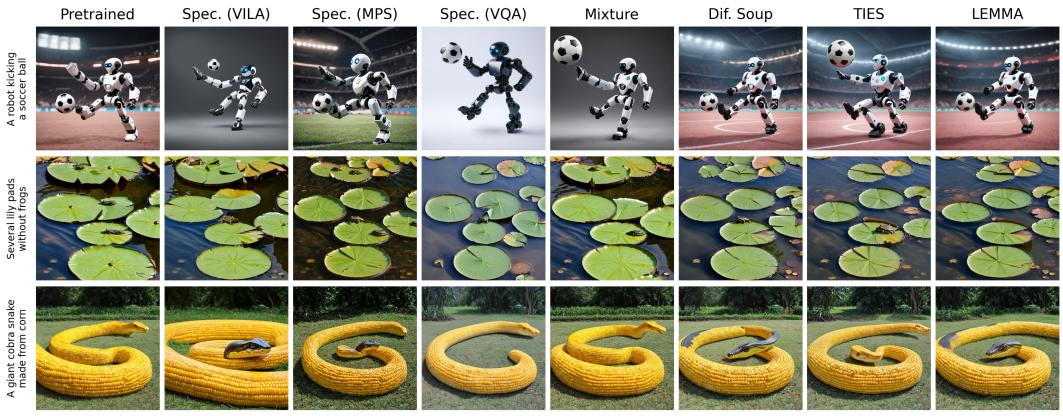



Figure A.2: **Qualitative results for DPO on the SD3 backbone.**

### E.3 User Study Protocol

We conduct human user studies across both setups (multi-objective DPO and multi-objective SFT). In total, we evaluate four pairwise comparisons: LEMMA against the pretrained base model and against the strongest merging baseline (TIES), using the SDXL backbone for DPO and the SD3 backbone for SFT.

**Setup & Anonymization:** For each benchmark, we evaluate 200 test prompts. Images generated by the methods under comparison are anonymized and presented side-by-side with randomized display order to eliminate positional bias. In the multi-objective SFT study, raters are additionally provided with the ground-truth reference image.

**Evaluation Task:** A pool of 12 raters participates in the study, with each rater evaluating 100 prompts per task (a total of 400 prompts across the four tasks). For each prompt, raters select their preferred output from three choices: Method A, Method B, or Tie (equal quality).

**Scoring:** We quantify preferences using the tie-split preference score for each rater $r$:

$$S_r = w_r + \frac{1}{2} t_r, \tag{15}$$

where $w_r$ and $t_r$ denote the percentage of wins and ties for LEMMA by the $r$-th rater, respectively. Ties are credited equally to both methods such that $S_r = 50\%$ indicates no preference. We report the mean and standard deviation of $S_r$ across the 12 raters.

## F Parameters Expressivity

Figure A.3 illustrates the resulting cross-layer expressivity $\sigma(\alpha)$ for each specialist. Moving from intermediate block-level merging to fine-grained leaf-level merging increases cross-layer weight standard deviation by $\sim 2.7\times$ on SDXL + DPO and $\sim 3.9\times$ on SD3 + SFT. This demonstrates that LEMMA actively uses leaf-level flexibility to assign different specialists to different layers, while introducing only 5,040 learnable parameters for SDXL + DPO and 1,960 for SD3 + SFT ($< 0.0002\%$ of total backbone weights). Note that because every layer within specialist $i$ shares the exact same scalar weight, the cross-layer standard deviation is identically zero ($\sigma(\alpha_i) = 0.00$).

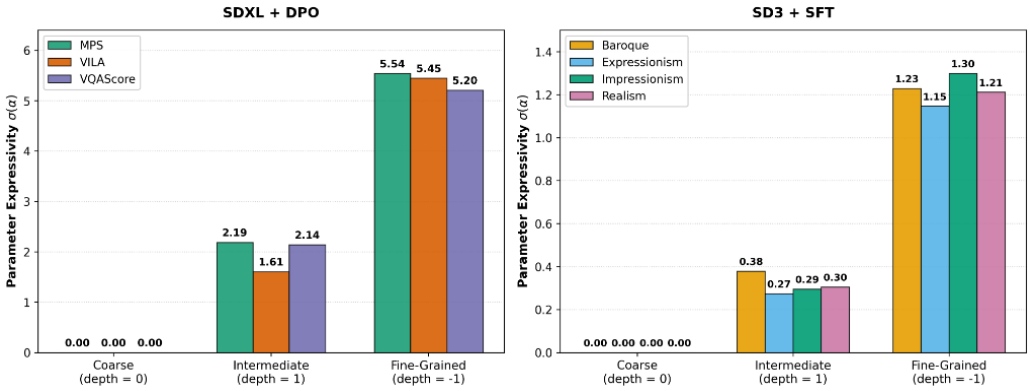



Figure A.3: **Parameter Expressivity $\sigma(\alpha)$ across Merge Granularities.** Standard deviation of learned merging coefficients $\alpha$ across layers for **SDXL + DPO** (left; MPS, VILA, VQAScore) and **SD3 + SFT** (right; Baroque, Expressionism, Impressionism, Realism).