Diff2Flow: Training Flow Matching Models via Diffusion Model Alignment

Johannes Schusterbauer* Ming Gui* Frank Fundel Björn Ommer
CompVis @ LMU Munich, MCML

Abstract

Diffusion models have revolutionized generative tasks through high-fidelity outputs, yet flow matching (FM) offers faster inference and empirical performance gains. However, current foundation FM models are computationally prohibitive for finetuning, while diffusion models like Stable Diffusion benefit from efficient architectures and ecosystem support. This work addresses the critical challenge of efficiently transferring knowledge from pre-trained diffusion models to flow matching. We propose Diff2Flow, a novel framework that systematically bridges diffusion and FM paradigms by rescaling timesteps, aligning interpolants, and deriving FM-compatible velocity fields from diffusion predictions. This alignment enables direct and efficient FM finetuning of diffusion priors with no extra computation overhead. Our experiments demonstrate that Diff2Flow outperforms naïve FM and diffusion finetuning particularly under parameter-efficient constraints, while achieving superior or competitive performance across diverse downstream tasks compared to state-of-the-art methods. We will release our code at https://github.com/CompVis/diff2flow.

1. Introduction

Recently, diffusion models [19, 57] have gained substantial popularity due to their exceptional generative capabilities, which have redefined the boundaries of image generation [42, 43, 46, 48]. Among these, foundation models such as Stable Diffusion [46] stand out, not only for their high-fidelity outputs but also for their useful representations and adaptability to downstream tasks, including depth estimation [14, 24], surface normal prediction [7], segmentation [63], and semantic correspondences [11, 60].

Meanwhile, *Flow Matching* (FM) [1, 30, 32] has emerged as a promising alternative, empirically offering faster inference and improved performance [8, 37]. While state-of-the-art foundational models based on flow matching, such as *Flux* [26] or *SDv3* [8], show remarkable generative capabilities, their large size ($>$ 8B parameters) requires high-end hardware for both training and inference, making them particularly computationally expensive. This makes fine-tuning impractical, especially in resource-constrained environments and significantly limits their practical adoption. In contrast, Stable Diffusion's [46] efficient architecture and widespread ecosystem make it a pragmatic choice. This raises an important question: Can the knowledge captured by existing foundational diffusion models be efficiently transferred to a flow matching model? How can we bridge the gap between diffusion and flow matching with minimal additional training, leveraging both the pre-trained prior and the advantageous properties of flow matching?

This work explores the relationship between the diffusion and flow matching paradigms. Although both methods can be generalized under a common framework [1, 25, 37], the actual implementations differ in several key aspects, including the definition of interpolants between the Gaussian noise prior and data samples, timestep scaling, and training objectives. These differences make it difficult to directly use a pre-trained diffusion model as a starting prior for flow matching training, as the two paradigms are not inherently well aligned.

To address this challenge, we propose a novel framework that effectively "warps" diffusion into flow matching, enabling seamless knowledge transfer between the two paradigms. This requires re-scaling their respective timesteps, aligning their differing interpolant formulations, and deriving the velocity field required for flow matching from the diffusion model's predictions, based on its parameterization. By systematically establishing these correspondences, we enable a smooth transition between diffusion and flow matching. To this end, we introduce a training methodology termed *Diff2Flow*, which initializes the flow matching model with a pretrained diffusion prior and directly finetunes it using the flow matching objective. Our analysis reveals that directly applying the FM loss to a diffusion model without incorporating our proposed adjustments significantly slows convergence and degrades overall model performance, whereas Diff2Flow provides a flexible and efficient approach with minimal finetuning overhead.

The benefits of this alignment become particularly evident when finetuning only a small subset of parameters.

*Equal Contribution

Under such constrained computational budgets, the performance gap between naïve FM finetuning and our alignment-aware approach becomes more pronounced. Specifically, we show that directly applying the FM objective with parameter efficient finetuning (PEFT) leads to very suboptimal performance. Leveraging *Diff2Flow* with its alignment-informed training strategy, PEFT enhances training efficiency and minimizes memory consumption while maintaining high performance.

We show the efficiency of our method on a diverse set of downstream tasks. ① We show that when finetuning on a high-aesthetics dataset [6] on the text-to-image task, we outperform diffusion-based finetuning while converging significantly faster than naïvely training with the flow matching objective. This empirically demonstrates that smartly aligning timestep scaling and objective significantly facilitate learning, and additionally mitigate the zero-terminal SNR issue [28] common in diffusion models, where full black or white images can't be generated. ② When repurposing the model to generate images at resolutions different from the pre-trained sweet-spot resolution (similar to [17]), our method achieves superior results compared to standard diffusion and flow matching fine-tuning. ③ We show that we can apply *Reflow* [32] using *Diff2Flow* on a base diffusion model, a method that straightens sampling trajectories, resulting in faster inference speed. By rectifying the sampling trajectories of Stable Diffusion v1.5 we can generate images with as few as 2 sampling steps without consistency distillation [58]. ④ Finally, we demonstrate the effectiveness of our method on domain adaptation, leveraging the Stable Diffusion prior to predicting affine-invariant depth maps similar to Marigold [24] and DepthFM [14], achieving state-of-the-art results with reduced training time.

2. Related Work

Diffusion and Flow Matching Models Diffusion models [19, 57] have demonstrated wide-ranging capabilities in data synthesis, extending from image [10, 44, 46, 48] and video [3, 15, 20] to audio generation [31] and beyond. While these models excel at producing high-fidelity outputs, they often require extensive sampling time, necessitating techniques like distillation [39, 49, 50, 58], noise schedule optimization [23], or training-free sampling [34, 56] to achieve faster generation. Following diffusion models, flow matching models [1, 30, 32] have gained attention due to their benefit of straighter probability paths. These models have been shown to perform competitively or surpass diffusion models in terms of both speed and quality [8, 37, 54].

The relationship between diffusion and flow matching models has been explored in Lee et al. [27], demonstrating that finetuning a flow matching model based on a pre-trained diffusion model is feasible. Our work extends this by explicitly formulating a method to map discrete diffusion trajectory to a continuous flow matching trajectory. Furthermore, we empirically demonstrate that our strategy works for various diffusion parameterization objectives and present results that go beyond reflow [32], achieving competitive performance on several benchmarks.

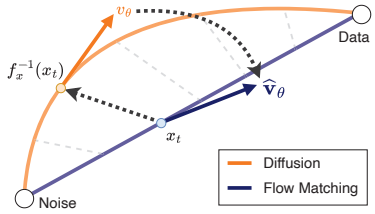


Figure 1. We introduce a novel finetuning technique to traverse between flow matching and diffusion, which enables effectively aligning the two processes with minimal additional training. The interpolant on the flow matching trajectory is computed as a function f of the diffusion timestep t, the sample x, and the associated diffusion coefficients. Our approach further enables a velocity prediction $\hat{\mathbf{v}}$ regardless of the diffusion model's parameterization.

Parameter-efficient Finetuning As foundational models expand in size and complexity, parameter-efficient finetuning (PEFT) techniques such as Low-Rank Adaptation (LoRA) [21] offer viable alternatives to full model finetuning. Originally developed for large language models, LoRA and other PEFT techniques have since been applied effectively to diffusion models [47], proving valuable across a broad range of tasks. These include domain-specific finetuning [67], additional image conditioning [59], image editing [12], distillation [36] and among others. By updating only a low-rank decomposition of the weight matrix, LoRA reduces memory requirements and alleviate catastrophic forgetting [2], making it highly adaptable for targeted adjustments in large foundational models.

Knowledge in Generative Models Following the rise of text-to-image diffusion models [41, 43, 44, 46, 48], numerous studies have explored methods to extract information from diffusion priors. Many works have effectively utilized the diffusion prior for tasks such as monocular depth estimation [9, 14, 16, 24, 38, 51], surface normal prediction [7], and semantic correspondences [11, 35, 60]. While these approaches predominantly focus on distilling information into another diffusion model, our work explores the transfer of this knowledge to flow matching models and hereby leverages the inherent advantages of flow matching, including more efficient inference and enhanced task performance.

3. Method

In this section, we explain our algorithm and the process of aligning the trajectories. We begin by discussing the definition of diffusion and flow matching models. Following this, we describe how we traverse between the two trajectories as depicted in Fig. 1 and use the diffusion model to generate a reasonable estimate of the flow velocity.

3.1. Diffusion and Flow Matching

Diffusion models (DM) [19, 56, 57] gradually diffuse data by typically adding noise to real data samples $x_0 \sim p_0(x_0)$ in a predefined discrete forward process, which can be characterized by

$$p(x_t|x_0) = \mathcal{N}(x_t; \alpha_t x_0, \sigma_t^2 \mathbf{I}), \tag{1}$$

where α_t and σ_t are the predefined noise schedules, t is the discrete timestep, and x_t is the resulting noisy sample, often referred to as the interpolant. In the setting of variance preserving schedules $\sigma_t = \sqrt{1-\alpha_t^2}$, and the interpolant x_t is then $x_t = \alpha_t x_0 + \sqrt{1-\alpha_t^2}\epsilon$ with $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. A neural network, parameterized by θ, is learned to reverse the forward process by gradually removing noise from x_t. Ho et al. [19] propose the simplified loss term

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\epsilon,x_0\sim p(x_0)}||\epsilon_t - \epsilon_\theta(x_t, t)||^2, \tag{2}$$

where predicting ϵ instead of x_0 is observed to yield better results and enhance convergence. There also exists the v-parameterization [49], defined as

$$v_t = \alpha_t \epsilon - \sigma_t x_0. \tag{3}$$

Flow matching (FM) models [1, 30, 32], another flexible class of generative models, utilizes the same idea of gradually deteriorating data samples and then synthesizing new data by reversing the process. We adopt the setting from Lipman et al. [30] where x_0 represents noise, and x_1 corresponds to data. The interpolant on the continuous timesteps $t \in [0, 1]$ is defined as

$$x_t = t x_1 + (1-t)x_0, \tag{4}$$

where x_0 is typically a Gaussian noise sample. The model $\mathbf{v}_\theta$ is trained to regress a vector field along the trajectory, following the linear path that points from x_0 to x_1 using the following objective:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{t,x_0\sim\mathcal{N}(0,\mathbf{I}),x_1\sim p(x_1)}||(x_1 - x_0) - \mathbf{v}_\theta(x_t, t)||^2. \tag{5}$$

Sampling from a flow matching model is accomplished by integrating over the trajectory of the learned ODE, e.g. using the forward Euler method with the update rule

$$x_{t+t_\Delta} = x_t + t_\Delta \mathbf{v}_\theta(x_t, t) \qquad \forall t \in [0, 1), \tag{6}$$

Figure 2. Although non-integer-shifted DDIM timesteps are neither defined nor trained, generated images remain of high quality.

with $t_\delta = 1/N$ and N being the number of function evaluations (NFE). We generate a discrete sequence that approximates the path by advancing in small increments along the ODE. The trajectory's smoothness, or "straightness," is critical, as it determines the number of steps required to achieve accurate results, trading off computational cost and fidelity. Straighter paths allow smaller N, whereas paths with higher curvature require more steps for simulating the ODE.

Reflow In FM training, we randomly sample from a Gaussian distribution and form pairs with the data points to create the interpolant x_t. This random pairing leads to curved sampling trajectories, increasing the curvature of the ODE trajectories during inference [61]. To straighten these trajectories, Liu et al. [32] propose *Reflow*, which iteratively trains on samples obtained from a pre-trained flow matching model. Given an already trained ODE model $\mathbf{v}_\Phi$, Reflow first samples $x_0 \sim \mathcal{N}(0, \mathbf{I})$ and integrates along the ODE trajectory using Eq. (6) to obtain a corresponding data sample x_1. This process generates image-noise pairings. Training on this paired data has been found to reduce inference cost by straightening the sampling trajectories [32, 33, 64].

3.2. Diffusion Prior for Flow Matching Model

Flow matching methods are known for straighter trajectories and more efficient inference [14, 61]. Hence, transforming the diffusion prior into a flow matching equivalent offers the potential for improvements in inference speed and performance [8, 37]. While previous research has successfully adapted diffusion priors for other downstream diffusion tasks, our approach combines these advantages with those of flow matching models.

Importantly, recent works on utilizing flow matching models to inherit a diffusion prior for downstream applications, such as DepthFM [14], have shown significant promise in accelerating inference compared to diffusion models. While this approach provides faster sampling than its diffusion model's counterpart, this inheritance method has limitations: DepthFM finetunes from a v-parameterized text-to-image diffusion model [46] directly. This approach requires the flow matching model to "warp" the v-parameterization to predict velocity, a different parameterization scheme. This introduces misalignments in the training objective, the interpolant, and the timesteps.

To address these limitations, we propose a novel strategy that enables a seamless transition between diffusion and flow matching trajectories. Our method minimizes misalignments and yields more accurate velocity predictions. In addition to monocular depth estimation, we demonstrate that our approach generalizes to other downstream tasks and offers a versatile and efficient solution for leveraging prior information from diffusion model.

3.2.1 Traversing Between Trajectories

Let us transform a foundation diffusion model's trajectory x_t by applying two invertible operations, one f_t that reparameterizes $t \in [0, T]$ and one f_x that transforms x to an alternate trajectory, defined by the following equation:

$$\bar{x}(f_t) = f_x(x_{f_t}). \tag{7}$$

The boundary conditions need to satisfy $\bar{x}(f_0) = x_0$ and $\bar{x}(f_T) = x_T$ to ensure that the starting and target samples are the same. This is different to Shaul et al. [55] since our approach rectifies the trajectories, which requires the exact alignment of the starting and the target distribution.

In order to construct f_t and f_x reasonably, we need to probe into the diffusion and flow matching model and investigate how the interpolants are related. Let x^{DM} represent a data sample on the diffusion trajectory and x^{FM} a data sample on the flow matching trajectory. Similarly, t_{DM} is the diffusion timestep and t_{FM} is the flow matching timestep. Following the convention of diffusion and flow matching literature [1, 19, 30, 32, 57], diffusion operates on discrete timesteps $t_{\text{DM}} \in \mathbb{Z}_{\geq 0} \cap [0, T]$, where $x^{\text{DM}}_{t_{\text{DM}}=0}$ represents the data samples and $x^{\text{DM}}_{t_{\text{DM}}=T}$ corresponds to the Gaussian noise. In contrast, flow matching uses continuous timesteps $t_{\text{FM}} \in [0, 1]$ with $x^{\text{FM}}_{t_{\text{FM}}=1}$ representing the data samples and $x^{\text{FM}}_{t_{\text{FM}}=0}$ representing the Gaussian noise. To summarize, the boundary condition enforces $x^{\text{DM}}_{t_{\text{DM}}=0} = x^{\text{FM}}_{t_{\text{FM}}=1}$ and $x^{\text{DM}}_{t_{\text{DM}}=T} = x^{\text{FM}}_{t_{\text{FM}}=0}$, while we seek two invertible mappings $f_t : [0, T] \to [0, 1]$ which maps t_{DM} to t_{FM}, and $f_x : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ which maps x^{DM} to x^{FM}. The diffusion interpolant is

$$x^{\text{DM}}_{t_{\text{DM}}} = \alpha_{t_{\text{DM}}} x^{\text{DM}}_0 + \sigma_{t_{\text{DM}}} x^{\text{DM}}_T, \tag{8}$$

where $\alpha^2_{t_{\text{DM}}} + \sigma^2_{t_{\text{DM}}} = 1$ for the variance-preserving schedule and $\alpha_{t_{\text{DM}}} = 1,\ \forall t_{\text{DM}}$ with $\sigma_{t_{\text{DM}}}$ monotonically increasing over time for the variance-exploding schedule [19, 57].
The flow matching interpolant is defined as

$$x^{\text{FM}}_{t_{\text{FM}}} = t_{\text{FM}} x^{\text{FM}}_1 + (1 - t_{\text{FM}}) x^{\text{FM}}_0. \tag{9}$$

There are multiple ways to define f_x so that the diffusion trajectory can be transformed and warped into a linear equation, one being

$$\begin{aligned} f_x(x^{\text{DM}}_{t_{\text{DM}}}) &= \frac{\alpha_{t_{\text{DM}}}}{\alpha_{t_{\text{DM}}} + \sigma_{t_{\text{DM}}}} x^{\text{DM}}_0 + \frac{\sigma_{t_{\text{DM}}}}{\alpha_{t_{\text{DM}}} + \sigma_{t_{\text{DM}}}} x^{\text{DM}}_T \\ &= \frac{1}{\alpha_{t_{\text{DM}}} + \sigma_{t_{\text{DM}}}} x^{\text{DM}}_{t_{\text{DM}}}, \end{aligned} \tag{10}$$

yielding a scaled version of $x^{\text{DM}}_{t_{\text{DM}}}$.

To design f_t as an invertible mapping between the discrete space t_{DM} and the continuous space t_{FM}, we start by establishing an invertible transformation function that enables continuous interpolation between discrete points in t_{DM} while aligning with f_x. To ensure continuity and invertibility, we define f_t as a piecewise linear function and interpolate between these discrete points. Let $t_{\overline{\text{DM}}} \in [0, T]$ be a continuous interpolated time space of the discrete t_{DM}. The correlation between Eq. (9) and Eq. (10) allows to calculate t_{FM} via the diffusion coefficients:

$$f_t(t_{\text{DM}}) = \frac{\alpha_{t_{\text{DM}}}}{\alpha_{t_{\text{DM}}} + \sigma_{t_{\text{DM}}}}, \tag{11}$$

where in both variance-preserving and variance-exploding diffusion schedules $f_t(t_{\text{DM}})$ is monotonic. While the diffusion coefficients α_t and σ_t are not defined for all $t_{\overline{\text{DM}}}$, we interpolate between the corresponding nearest neighbors. Let t_{DM_1} and t_{DM_2} be the nearest neighbors of $t_{\overline{\text{DM}}}$, and we design f_t to be just as a linear interpolation in-between. Interestingly, we find that although the continuous values of $t_{\overline{\text{DM}}}$ were not trained during the foundation model's training, direct inference with these values still produces reasonable results, as shown in Fig. 2. Specifically, instead of performing DDIM inference at discrete timesteps $t_{\text{DDIM}} \in \mathbb{Z}_{\geq 0} \cap [0, 1000]$, we do inference at $t_{\text{DDIM}} + 0.5$ with linearly interpolated α_t and σ_t. We hypothesize that timesteps represented as sinusoidal embeddings create a well-defined continuous time space. This observation lays the foundation for interpolating timestep embeddings and traversing trajectories.

With $t_{\overline{\text{DM}}}$ defined on the continuous domain, the inverse $f_t^{-1}(t_{\text{FM}})$ can be defined as follows. First, we find t_{DM_1} and t_{DM_2} in the discrete timesteps such that $f_t(t_{\text{DM}_1})$ and $f_t(t_{\text{DM}_2})$ are the two nearest neighbors of t_{FM} with $f_t(t_{\text{DM}_1}) \leqslant t_{\text{FM}} \leqslant f_t(t_{\text{DM}_2})$. Next, we perform a linear interpolation between the two neighbors to reverse the mapping from t_{FM} to $t_{\overline{\text{DM}}}$:

$$f_t^{-1}(t_{\text{FM}}) = t_{\text{DM}_1} + \frac{t_{\text{FM}} - f_t(t_{\text{DM}_1})}{f_t(t_{\text{DM}_2}) - f_t(t_{\text{DM}_1})}(t_{\text{DM}_2} - t_{\text{DM}_1}). \tag{12}$$

With $f_t(\cdot)$ bidirectionally defined as in Eq. (11) and Eq. (12), we deduce the bidirectional mapping from the FM trajectory x^{FM} back to the DM trajectory x^{DM}:

$$f_x^{-1}(x^{\text{FM}}_{t_{\text{FM}}}) = (\alpha_{f_t^{-1}(t_{\text{FM}})} + \sigma_{f_t^{-1}(t_{\text{FM}})}) x^{\text{FM}}_{t_{\text{FM}}}. \tag{13}$$

3.2.2 Unifying the Objectives

While traversing between the diffusion and flow matching trajectories is explained in the previous chapter, it is still essential to unify the training and inference objectives. Prior works [14, 64] fine-tuned the model to adapt the diffusion prior, originally predicting ϵ or v, to directly predict velocity. However, these approaches force the model to transition between different parameterizations, therefore requiring longer convergence times and also ultimately impacting model performance. In contrast, our method unites the objectives by leveraging the diffusion prior and utilizing the relationships between the parameterizations. We term this technique *Objective change* as we incorporate both trajectory traversal and a principled objective realignment to facilitate velocity prediction based on the diffusion prior. This approach is broadly applicable across different parameterizations; here, we demonstrate its effectiveness using v-parameterization.

Let $v_\theta(x^{\text{DM}}, t_{\text{DM}})$ be the v-prediction (Eq. (3)) from a pre-trained diffusion model. Using the notation of Eq. (8)

$$v_\theta(x^{\text{DM}}, t_{\text{DM}}) = \alpha_{t_{\text{DM}}} x_T^{\text{DM}} - \sigma_{t_{\text{DM}}} x_0^{\text{DM}} \tag{14}$$

where x_0 and the corresponding x_T can be estimated as

$$\begin{aligned} \widehat{x_0^{\text{DM}}} &= \alpha_{t_{\text{DM}}} x_{t_{\text{DM}}}^{\text{DM}} - \sigma_{t_{\text{DM}}} v_\theta(x_{t_{\text{DM}}}^{\text{DM}}, t_{\text{DM}}) \\ \widehat{x_T^{\text{DM}}} &= \alpha_{t_{\text{DM}}} v_\theta(x_{t_{\text{DM}}}^{\text{DM}}, t_{\text{DM}}) - \sigma_{t_{\text{DM}}} x_{t_{\text{DM}}}^{\text{DM}}, \end{aligned} \tag{15}$$

with the wide hat symbol indicating an estimate predicted by the model. An approximation for the velocity $\mathbf{v}$ at the corresponding FM data point x^{FM} is then formulated with respect to the boundary conditions defined in Eq. (7):

$$\begin{aligned} \mathbf{v}_\theta(x^{\text{FM}}, t_{\text{FM}}) &= \widehat{x_1^{\text{FM}}} - \widehat{x_0^{\text{FM}}} \\ &= \widehat{x_0^{\text{DM}}} - \widehat{x_T^{\text{DM}}} \\ &= (\alpha_{t_{\text{DM}}} - \sigma t_{\text{DM}})(x_{t_{\text{DM}}}^{\text{DM}} - v_\theta(x_{t_{\text{DM}}}^{\text{DM}}, t_{\text{DM}})). \end{aligned} \tag{16}$$

While we demonstrate the transition specifically from v-parameterization to velocity, this approach is versatile and can be readily applied to other parameterizations, such as ϵ-parameterization, by following the same method outlined above. Algorithm 1 and Algorithm 2 summarize this approach in more detail. In summary, our main objective is to estimate the velocity using the prediction on the diffusion trajectory. We use standard Euler steps for inference.

Algorithm 1 Diff2Flow Training

Require: Data Sample x_1^{FM}, Noise sample x_0^{FM}, Timestep $t_{\text{FM}} \in [0, 1]$

1: Compute the interpolant $x_{t_{\text{FM}}}^{\text{FM}}$ ▷ Eq. (9)
2: Reparameterize t_{FM} to $t_{\overline{\text{DM}}}$ using f_t^{-1} ▷ Eq. (12)
3: Reparameterize $x_{t_{\text{FM}}}^{\text{FM}}$ to $x_{t_{\text{DM}}}^{\text{DM}}$ using f_x^{-1} ▷ Eq. (13)
4: Approximate velocity prediction $\mathbf{v}_\theta$ ▷ Eq. (16)
5: Take gradient descent step on $\mathcal{L}_{\text{FM}}$ ▷ Eq. (5)

Algorithm 2 Diff2Flow Sampling

Require: Noise sample x_0^{FM}, Number of sampling steps N

1: **for** t = $0, \frac{1}{N}, \cdots, \frac{N-1}{N}$ **do**
2: Reparameterize t_{FM} to $t_{\overline{\text{DM}}}$ using f_t^{-1} ▷ Eq. (12)
3: Reparameterize $x_{t_{\text{FM}}}^{\text{FM}}$ to $x_{t_{\text{DM}}}^{\text{DM}}$ using f_x^{-1} ▷ Eq. (13)
4: Approximate velocity prediction $\mathbf{v}_\theta$ ▷ Eq. (16)
5: $x_{t+1/N} \leftarrow x_t + \frac{1}{N}\mathbf{v}_\theta$ ▷ Eq. (6)
6: **end for**
7: **return** x_1

3.3. Parameter-efficient Finetuning

Prior works [36, 47] have demonstrated that parameter-efficient finetuning (PEFT) can be advantageous in tasks like domain adaptation and model distillation for generative models. PEFT reduces the number of parameters requiring updates, lowers memory demands during training, and theoretically mitigates catastrophic forgetting [2, 21]. In our approach, we utilize Low-Rank Adaptation (LoRA) [21], which achieves parameter efficiency by freezing the main model weights and updating only a low-rank decomposition of the weight matrices. Given a weight matrix $W_0 \in \mathbb{R}^{d\times k}$, we freeze W and contrain the matrix's update with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with the rank $r \leq \min(d, k)$. The modified forward pass for an input x is then $h = W_0 x + \Delta W x = W_0 x + BAx$. We observe that LoRA does not work out of the box when training a diffusion model with a flow matching objective, likely due to the shift required to adapt to a new parameterization scheme. In other words, LoRA struggles with the significant adjustment required when reconfiguring the model from a diffusion-based parameterization to a velocity prediction paradigm. This observation also underscores why naïve application of a flow matching objective to a pre-trained diffusion model is not effective; it forces the model to unlearn the old parameterization and switch to an entirely new one. In contrast, LoRA performs well when incorporating our proposed objective change that aligns the trajectories. Here, LoRA provides an efficient balance, achieving strong results with minimal parameter updates and showing further improvements as more parameters are fine-tuned.

4. Experiments

4.1. Text-to-Image Synthesis

We start with the text-to-image task, the original task on which the diffusion prior model was trained. Specifically,

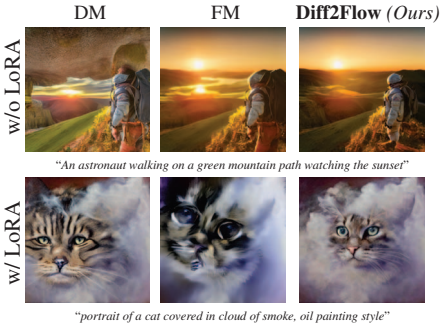


Figure 3. Qualitative results for our finetuned Text-to-Image models using different objectives. **DM**, **FM** and **Diff2Flow** stand for diffusion finetuning, flow matching finetuning, and our method. Images are generated with the same seed, NFEs, and CFG $= 4.0$.

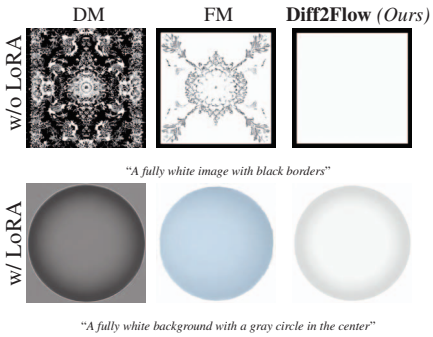


Figure 4. Our finetuning objective, both with and without PEFT effectively addresses the issue of non-zero-terminal SNR of DMs. The generations align faithfully to the input prompts.

we use Stable Diffusion 2.1 [46] as our diffusion prior, pretrained at a resolution of 768×768, and fine-tune it to generate images at a lower resolution of 512×512. This resolution shift allows us to investigate the effectiveness of different fine-tuning techniques in bridging resolution gaps.

Starting from the SD2.1 prior, we continue training with different objectives on the LAION-Aesthetics dataset [53] and evaluate model performance on COCO 2017 [29]. Fig. 5 shows that simply continuing diffusion training with an FM objective yields weak performance, especially when constraining model capacity with LoRA. In contrast, our method consistently surpasses continued diffusion training, both with and without classifier-free guidance, converging in as few as 2.5k iterations. Note that all models show poor FID results without training due to the resolution mismatch.

We present qualitative results in Fig. 3 and Fig. 4, where we use the same sampling hyperparameters and fix the initial Gaussian noise. A key issue that we also address in the transition from the diffusion model to flow matching is the problem of non-zero terminal SNR [28], which typically results in images generated by stable diffusion models having, on average, gray tones instead of true black or white. Our training paradigm solves this problem, illustrated in Fig. 4.

Method	FID ↓	CLIP ↑	Aesthetics Score ↑
SD1.5	56.77	26.34	5.32
SD1.5 (cont. training)	56.36	26.33	5.90
Diff2Flow	**52.80**	**26.54**	**5.99**

Table 1. Eval on COCO-2017 5k with 25 Euler/DDIM steps.

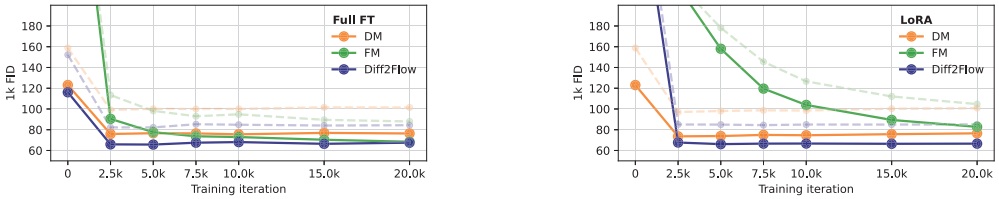
Furthermore, we investigate the scenario where diffusion training is resumed with our method without any changes in resolution or task objectives. Specifically, we initialize from an SD1.5 checkpoint and continue training on the LAION-Aesthetics dataset at a resolution of 512×512. The corresponding evaluation metrics are reported in Tab. 1. Consistent with observations in SD3 [8], the rectified trajectories of Diff2Flow yield superior performance compared to the baseline SD model. We show that converting a diffusion model to its flow-matching counterpart also leads to performance improvements within the same generative task.

4.2. Rectifying the Trajectories

In addition to fine-tuning diffusion models for resolution changes, we can also optimize them for straighter sampling trajectories to enable faster generation speeds. A popular approach to this is *Reflow* [32], which modifies the flow-matching training objective by replacing the random Gaussian noise terms in Eq. (5) with calculated noise derived from a pre-trained prior ODE to ensure that each noise-image pair is aligned according to the ODE. Quantitative results are presented in Tab. 3, where our method demonstrates competitive performance over various number of function evaluations (NFEs), including low NFEs such as 2 and 4, while only fine-tuning less than 7% of the parameters. Qualitative results using reflow with 4 inference steps are shown in Fig. 6 and in the Appendix.

4.3. Monocular Depth Estimation

In addition to common tasks, we also show the versatility of our approach for domain adaptation finetuning. Previous methods have shown strong performance in using large-scale image generative models for affine-invariant monocular depth estimation [9, 14, 24]. *Marigold* [24] directly finetune Stable Diffusion on synthetic image-depth data using the standard v-parameterized Diffusion loss (Equation (3)). Following a similar setup, *DepthFM* [14] propose to convert Stable Diffusion into a flow matching model by training with the flow matching (see Equation (5)) loss. This approach requires aligning the model's outputs to a different objective, interpolant, and timestep scaling (with $t_{\text{DM}} \in [0, 1000]$ in diffusion versus $t_{\text{FM}} \in [0, 1]$ in flow matching).



(a) (b)

Figure 5. Text-to-Image FID on the COCO 2017 [29] validation dataset. Light curves indicate results without Classifier-free Guidance [18]. **a)** We show that both FM and Diff2Flow converge to the same performance, given sufficient training and model capacity (full fine-tuning). However, our method removes the computational overhead from learning how to adjust the network output, resulting to faster convergence. **b)** Given limited modeling capacity (LoRA) the difference is more pronounced, where the FM model fails to close the gap to Diff2Flow, indicating that finetuning a portion of the parameters is enough to transfer a diffusion model to a flow matching model.

Method	Training samples	NYUv2 [40]		KITTI [13]		ETH3D [52]		ScanNet [5]		DIODE [62]	
		AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑	AbsRel↓	$\delta1$↑
Depth Anything [65] (CVPR '24)	62M	**4.3**	**98.1**	7.6	94.7	12.7	88.2	—	—	6.6	95.2
Depth Anything v2 [66] (arXiv '24)	62M	4.4	97.9	7.5	94.8	13.2	86.2	—	—	**6.5**	**95.4**
Metric3D [68] (ICCV '23)	8M	5.0	96.6	5.8	97.0	6.4	96.5	**7.4**	**94.1**	22.4	80.5
Metric3D v2 [22] (TPAMI '24)	16M	**4.3**	**98.1**	**4.4**	**98.2**	**4.2**	**98.3**	—†	—†	13.6	89.5
Marigold [24] (CVPR '24)	74K	5.5	96.4	9.9	91.6	6.5	96.0	6.4	95.1	30.8	77.3
GeoWizard [9] (ECCV '24)	278K	5.2	96.6	9.7	92.1	6.4	96.1	6.1	95.3	29.7	79.2
reproduced by [38]	278K	5.7	96.2	14.4	82.0	7.5	94.3	6.1	95.8	31.4	77.1
DepthFM [14] (AAAI '25)	74K	6.0	95.5	9.1	90.2	6.5	95.4	6.6	94.9	22.4	78.5
E2E FT [38] (WACV '25)	74K	**5.2**	96.6	9.6	91.9	6.2	95.9	**5.8**	**96.2**	30.2	77.9
Lotus-G [16] (arXiv '24)	59K	5.4	96.6	11.3	87.7	6.2	96.1	6.0	96.0	—	—
Diff2Flow	74K	5.7	**96.7**	**8.7**	**92.0**	**5.5**	**97.4**	6.2	95.7	**21.6**	**79.5**
Diff2Flow (LoRA)	74K	5.9	96.4	9.5	90.8	6.0	96.8	6.4	95.5	22.6	78.5

Table 2. **Comparison to state-of-the-art depth estimation methods.** We compare discriminative (top) and generative models (bottom). Numbers obtained from [38]. †Metric3D v2 [22] was trained on ScanNet, so zero-shot evaluation is not possible. We also show the results reproduced by [38] for GeoWizard. All of our results are generated with NFE$=10$ and an ensemble size of 4.

Method	#Params ↓	Steps	FID ↓	CLIP ↑
SDv1.5+DPM Solver [34] (NeurIPS '22)	0.9B	25	20.10	0.318
Rectified Flow [32] (ICLR '23)	0.9B	25	21.65	**0.315**
Diff2Flow (LoRA)	62M	25	**21.45**	0.314
PeRFlow [64] (arXiv '24)	0.9B	4	**22.97**	0.294
Diff2Flow (LoRA)	62M	4	25.29	**0.313**
Rectified Flow [32] (ICLR '23)	0.9B	2	**31.35**	0.296
Diff2Flow (LoRA)	62M	2	32.31	**0.305**

Table 3. We fine-tune all our models with LoRA, and apply only 1-rectified flow according to Liu et al. [32]. We get competitive results with state-of-the-art reflow models.

This process incurs significant computational cost, limiting the training efficiency and final performance. However, the flow matching nature of DepthFM allows for faster inference, at slightly reduced performance when given enough training. Our method mitigates this problem, accelerating training, and improving performance. The training and evaluation metrics are provided in the Appendix.

Figure 6. SD1.5 [46] + Diff2Flow-Reflow, 4-step inference results.

Results Table 2 compares our method with both discriminative and generative depth predictors. Trained exclusively on synthetic data, our method matches or outperforms the prior state-of-the-art in monocular depth estimation, as shown in Figure 8. The diffusion-based Marigold model [24] struggles to achieve high accuracy with a limited number of sampling steps. In contrast to both DepthFM and Marigold, our method achieves competitive performance after very few training iterations and with only two sampling steps. The performance difference becomes even more pronounced when training with LoRA, where model capacity is limited. Figure 8b shows that this limitation prevents a

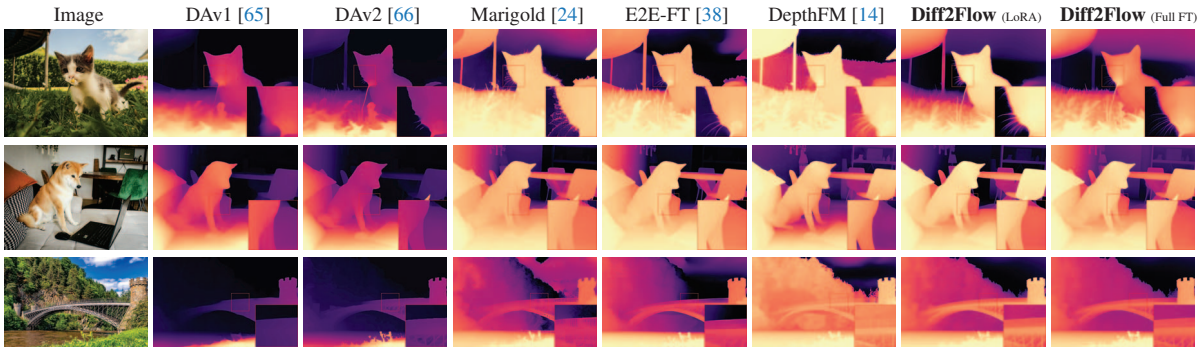
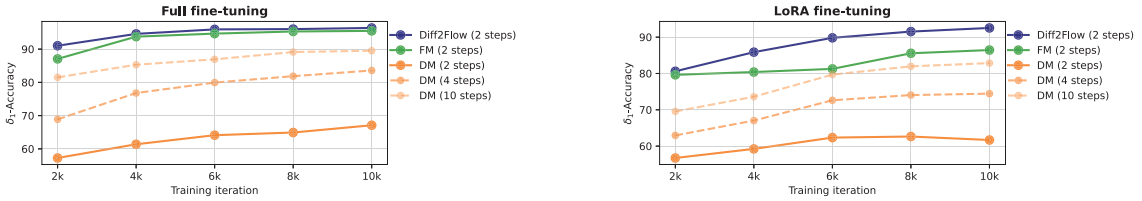


Figure 7. Zero-shot qualitative results on real-world imagery. Our methods produce depth predictions with perceptually higher fidelity and finer detail. All predictions are made using the optimal settings of the models. Best viewed when zoomed in.



(a) (b)

Figure 8. Results on NYUv2 depth benchmark [40] with 4 ensemble members. **a)** With full fine-tuning we find that FM adapts to the I2D task very quickly, but adding the objective change leads to even quicker convergence and better results. In contrast, the diffusion-based counterpart [24] suffers from slower convergence and requires more inference steps. **b)** The difference between FM and Diff2Flow is even more pronounced if we limit the capacity of the model by using Low-Rank adaptations. With Diff2Flow, the model does not need to learn to change the objective and can focus only on the task itself.

straightforward application of the FM loss to the diffusion model. DepthFM struggles here as the model has to learn the adaptation from the v-parameterization to the FM objective. The successful adaptation of the diffusion-based image prior to flow matching supports our initial hypothesis that the integration of our objective change facilitates this transformation. In Table 4, we present an ablation study on performance with different numbers of trainable parameters. Fine-tuning just $1/4$ of the original parameters is sufficient to achieve competitive performance with previous methods, particularly with the fully finetuned DepthFM model. Figure 7 qualitatively compares our method with state-of-the-art depth estimators on in-the-wild images. While discriminative methods often lack detail and fidelity, generative models capture fine details and avoid the averaging effect common to discriminative approaches.

Method	#Params ↓	AbsRel ↓	δ_1 ↑
Marigold + E2E FT [38]	866M	5.2	96.6
Diff2Flow	866M	5.7	96.7
Diff2Flow (LoRA base)	222M	5.9	96.4
Diff2Flow (LoRA small)	62M	6.9	95.0

Table 4. Zero-shot evaluation on NYUv2 [40] for different number of trainable parameters. We can achieve competitive performance with only a fraction of previous SOTA methods.

5. Conclusion

We introduce a novel framework, *Diff2Flow*, that effectively converts pre-trained diffusion into flow matching models, providing a twofold optimization. First, this approach allows us to efficiently use the diffusion model's prior by exploiting its powerful generative capabilities without retraining from scratch. Second, by adapting it to the flow matching paradigm, we gain key benefits unique to flow matching: fast inference suitable for downstream tasks, and the ability to straighten sampling trajectories through techniques such as reflow, which further improves efficiency and performance. As a result, *Diff2Flow* not only improves training and inference efficiency but also delivers competitive performance in diverse tasks, including text-to-image synthesis and monocular depth estimation. By using parameter-efficient fine-tuning methods such as LoRA, our framework further minimizes computational requirements and demonstrates a practical, scalable way to merge diffusion and flow matching paradigms in generative modeling.

Acknowledgement

This project has been supported by the bidt project KLIMA-MEMES, the German Federal Ministry for Economic Affairs and Climate Action within the project "NXT GEN AI METHODS – Generative Methoden für Perzeption, Prädiktion und Planung", the project "GeniusRobot" (01IS24083), funded by the Federal Ministry of Education and Research (BMBF), and the German Research Foundation (DFG) project 421703927. The authors acknowledge the Gauss Center for Supercomputing for providing compute through the NIC on JUWELS at JSC and the HPC resources supplied by the Erlangen National High Performance Computing Center (NHR@FAU funded by DFG project 440719683) under the NHR project JA-22883.

References

[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 1, 2, 3, 4

[2] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, et al. Lora learns less and forgets less. *Transactions on Machine Learning Research*, 2024. 2, 5

[3] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 2

[4] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 12

[5] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 7, 12

[6] Xiaoliang Dai, Ji Hou, Chih-Yao Ma, Sam Tsai, Jialiang Wang, Rui Wang, Peizhao Zhang, Simon Vandenhende, Xiaofang Wang, Abhimanyu Dubey, et al. Emu: Enhancing image generation models using photogenic needles in a haystack. *arXiv preprint arXiv:2309.15807*, 2023. 2

[7] Xiaodan Du, Nicholas Kolkin, Greg Shakhnarovich, and Anand Bhattad. Generative models: What do they know? do they know things? let's find out! *arXiv preprint arXiv:2311.17137*, 2023. 1, 2

[8] 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. 1, 2, 3, 6

[9] Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *ECCV*, 2024. 2, 6, 7

[10] Michael Fuest, Pingchuan Ma, Ming Gui, Johannes Schusterbauer, Vincent Tao Hu, and Bjorn Ommer. Diffusion models and representation learning: A survey. *arXiv preprint arXiv:2407.00783*, 2024. 2

[11] Frank Fundel, Johannes Schusterbauer, Vincent Tao Hu, and Björn Ommer. Distillation of diffusion features for semantic correspondence. *WACV*, 2025. 1, 2

[12] Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept sliders: Lora adaptors for precise control in diffusion models. *arXiv preprint arXiv:2311.12092*, 2023. 2

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 7, 12

[14] Ming Gui, Johannes Schusterbauer, Ulrich Prestel, Pingchuan Ma, Dmytro Kotovenko, Olga Grebenkova, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Depthfm: Fast monocular depth estimation with flow matching. In *AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence (AAAI), 2025. 1, 2, 3, 5, 6, 7, 8, 12, 15, 16

[15] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *International Conference on Learning Representations*, 2024. 2

[16] Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Liu, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024. 2, 7

[17] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023. 2

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 7

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 2, 3, 4

[20] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 2

[21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 2, 5

[22] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface

normal estimation. *arXiv preprint arXiv:2404.15506*, 2024. 7

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 2

[24] Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024. 1, 2, 6, 7, 8, 12, 13, 15, 16

[25] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36: 65484–65516, 2023. 1

[26] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024. 1

[27] Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in Neural Information Processing Systems*, 37:63082–63109, 2024. 2

[28] Shanchuan Lin, Bingchen Liu, Jiashi Li, and Xiao Yang. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 5404–5411, 2024. 2, 6

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6, 7, 12

[30] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1, 2, 3, 4

[31] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning*, pages 21450–21474. PMLR, 2023. 2

[32] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 1, 2, 3, 4, 6, 7, 12

[33] Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023. 3

[34] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 2, 7

[35] Grace Luo, Lisa Dunlap, Dong Huk Park, Aleksander Holynski, and Trevor Darrell. Diffusion hyperfeatures: searching through time and space for semantic correspondence. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pages 47500–47510, 2023. 2

[36] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023. 2, 5

[37] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. *arXiv preprint arXiv:2401.08740*, 2024. 1, 2, 3

[38] Gonzalo Martin Garcia, Karim Abou Zeid, Christian Schmidt, Daan de Geus, Alexander Hermans, and Bastian Leibe. Fine-tuning image-conditional diffusion models is easier than you think. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025. 2, 7, 8, 15, 16

[39] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14297–14306, 2023. 2

[40] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7, 8, 12

[41] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 2

[42] 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. 1

[43] 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 *The Twelfth International Conference on Learning Representations*, 2024. 1, 2

[44] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[45] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *International Conference on Computer Vision (ICCV) 2021*, 2021. 12

[46] 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)*, pages 10674–10685. IEEE, 2022. 1, 2, 3, 6, 7

[47] Simo Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning. 2022. *URL https://github.com/cloneofsimo/lora*, 2022. 2, 5

[48] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Lit, Jay Whang, Emily Denton, Seyed Kamyar Seyed

Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Raphael Gontijo-Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pages 36479–36494, 2022. 1, 2

[49] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 2, 3

[50] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2025. 2

[51] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 2

[52] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, pages 3260–3269, 2017. 7, 12

[53] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 6, 12

[54] Johannes Schusterbauer, Ming Gui, Pingchuan Ma, Nick Stracke, Stefan Andreas Baumann, Vincent Tao Hu, and Björn Ommer. Boosting latent diffusion with flow matching. In *ECCV*, pages 338–355, 2024. 2

[55] Neta Shaul, Juan Perez, Ricky TQ Chen, Ali Thabet, Albert Pumarola, and Yaron Lipman. Bespoke solvers for generative flow models. *arXiv preprint arXiv:2310.19075*, 2023. 4

[56] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 2, 3

[57] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 1, 2, 3, 4

[58] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 2

[59] Nick Stracke, Stefan Andreas Baumann, Joshua M Susskind, Miguel Angel Bautista, and Björn Ommer. Ctrloralter: Conditional loradapter for efficient 0-shot control & altering of t2i models. *arXiv preprint arXiv:2405.07913*, 2024. 2

[60] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in neural information processing systems*, 2023. 1, 2

[61] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pages 1–34, 2024. 3

[62] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z. Dai, Andrea F. Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R. Walter, and Gregory Shakhnarovich. DIODE: A Dense Indoor and Outdoor DEpth Dataset. *CoRR*, 2019. 7, 12

[63] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2955–2966, 2023. 1

[64] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator, 2024. 3, 5, 7

[65] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024. 7, 8, 15, 16

[66] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024. 7, 8, 15, 16

[67] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2

[68] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023. 7