Cost-Aware Routing for Efficient Text-To-Image Generation

Anonymous authors
Paper under double-blind review

Abstract

Diffusion models are well known for their ability to generate a high-fidelity image for an input prompt through an iterative denoising process. Unfortunately, the high fidelity also comes at a high computational cost due to the inherently sequential generative process. In this work, we seek to optimally balance quality and computational cost, and propose a framework to allow the amount of computation to vary for each prompt, depending on its complexity. Each prompt is automatically routed to the most appropriate text-to-image generation function, which may correspond to a distinct number of denoising steps of a diffusion model, or a disparate, independent text-to-image model. Unlike uniform cost reduction techniques (e.g., distillation, model quantization), our approach achieves the optimal trade-off by learning to reserve expensive choices (e.g., 100+ denoising steps) only for a few complex prompts, and employ more economical choices (e.g., small distilled model) for less sophisticated prompts. We empirically demonstrate on COCO and DiffusionDB that by learning to route to nine already-trained text-to-image models, our approach is able to deliver an average quality that is higher than that achievable by any of these models alone.

1 Introduction

While diffusion models have set a new standard for photorealism in generative art (Ho et al., 2020), their operational costs remain a major challenge. The generation of a single image can involve many denoising steps, each utilizes a learned denoiser model with potentially over a billion parameters (Rombach et al., 2022). This makes in-the-wild adoption (i.e., on-device) challenging and raises valid concerns about their environmental sustainability (Sarah Wells, 2023; Kate Crawford, 2024; Kaack et al., 2022). To address this, a significant body of research has explored optimization strategies such as network simplification (Li et al., 2024; 2023a) and model distillation (Sauer et al., 2024; Salimans & Ho, 2022; Meng et al., 2023; Liu et al., 2023).

However, these existing methods typically apply the same degree of optimization irrespective of the task's intrinsic difficulty. This results in a single model with a fixed computational cost, which is inherently suboptimal as the generative effort required to synthesize an image varies with the complexity of the input prompt. For example, a simple prompt like *a white and empty wall* requires fewer denoising steps to generate a high-quality image than a complex one like *a colorful park with a crowd*, as shown in Figure 1.

With the motivation to adaptively allocate computational budget, we present *CATImage*, a framework that allows the amount of computation for text-to-image generation to vary for each prompt. Our framework operates with a pre-defined set of choices that can be chosen adaptively for each input prompt. Each choice represents a text-to-image generation function and has a distinct profile of computational cost and the expected image quality. Concretely, these choices may correspond to different numbers of denoising steps of the same diffusion model (i.e., homogeneous choices), disparate, independent text-to-image generative models (i.e., heterogeneous choices), or a combination of both. The proposed *CATImage* aims to adaptively select the right choice (i.e., "routing") for each input prompt, in such a way that expensive choices (e.g., 100+ denoising steps) are reserved only for complex prompts. Our approach enables a joint deployment of diverse text-to-image models and has a potential to deliver higher average image quality compared to using any individual model in the pool, while allowing the average computational cost to be adapted at deployment time.

(a) *'a white and empty wall'* (b) *'a colorful park with a crowd'* (c) *quality trends across numbers of steps*

Figure 1: *Two input prompts that require different denoising steps to ensure quality.* As shown in Figure 1c, prompt Figure 1a only requires a small number of denoising steps to reach a high CLIPScore. By contrast, the more complex prompt Figure 1b requires over 100 steps to reach a similar quality. Key to our proposed *CATImage* is to allocate an appropriate amount of computation for each prompt, so that the overall computational cost is reduced while the quality remains the same.

In summary, our contributions are as follows.

1. We precisely formulate a constrained optimization problem for the above routing problem (Section 3.1). The formulation aims to maximize average image quality subject to a budget constraint on the generation cost.
2. We study the theoretically optimal routing rule that optimally trades off the average quality and cost (Section 3.2). Based on the optimal rule, we construct a plug-in estimator that can be trained from data.
3. We perform a series of objective analyses on the COCO (Lin et al., 2014) and DiffusionDB datasets (Wang et al., 2022b). Our findings show that, through adaptive routing, our proposal matches the quality of the largest model in the serving pool (namely, Stable Diffusion XL from Radford et al. (2021) with 100 denoising steps) with only a fraction of its computational cost (Table 3).[1]

2 Background: Text-To-Image Generative Models

Let $\mathbf{x} \in \mathcal{X}$ denote an input text prompt, and $\mathbf{i} \in \mathcal{I} \doteq [0,1]^{W \times H \times 3}$ denote an image described by the prompt, where $W, H \in \mathbb{N}$ denote the width and the height of the image (in pixels), and the last dimension denotes the number of color channels. A text-to-image generative model is a stochastic map $h \colon \mathcal{X} \to \mathcal{I}$ that takes a prompt $\mathbf{x}$ as input and generates an image $h(\mathbf{x}) \in \mathcal{I}$ that fits the description in the prompt $\mathbf{x}$. There are many model classes one may use to construct such a model h, including conditional Generative Adversarial Networks (GANs) (Zhang et al., 2017; Goodfellow et al., 2014), Variational Auto-Encoder (VAE) (Kingma & Welling, 2022), and diffusion models (Ho et al., 2020), among others.

Diffusion models A specific class of text-to-image generative models that has recently been shown to produce high-fidelity images is given by diffusion-based models (Saharia et al., 2022; Ho et al., 2020; 2022). A diffusion generative model relies on a function $g \colon \mathcal{X} \times \mathbb{N} \times \mathbb{R}^D \to \mathcal{I}$ that takes as input a prompt $\mathbf{x}$, the number of denoising steps $T \in \mathbb{N}$, a noise vector $\mathbf{z} \in \mathbb{R}^D$ with $D = 3 \cdot WH$, and generates an image $\mathbf{i} = g(\mathbf{x}, T, \mathbf{z})$. Image generation is done by iteratively refining the initial noise vector $\mathbf{z}$ for T iterations to produce the final image. The noise vector $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is typically sampled from the standard multivariate normal distribution and the T refinement steps correspond to the reverse diffusion process, which reconstructs an image from a random initial state (Ho et al., 2020). With $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ understood to be an implicit source of randomness, we define $h_T(\mathbf{x}) \doteq g(\mathbf{x}, T, \mathbf{z})$ to be an image sampled from the diffusion model using T diffusion steps. With T

[1]We will release the code and data upon paper publication.

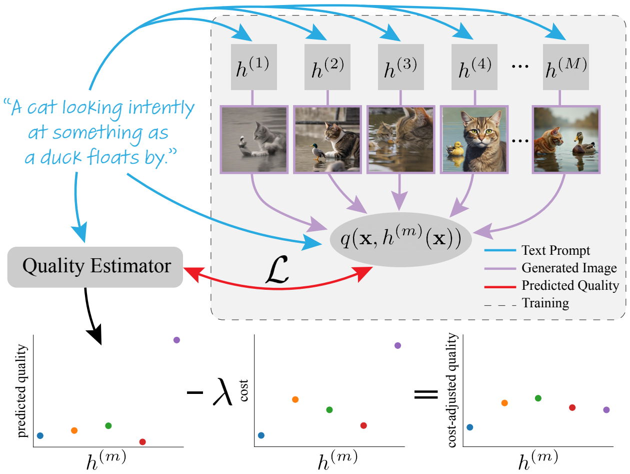


Figure 2: *Illustration of our pipeline.* During training (dashed box), a quality estimator is trained to predict per-prompt quality scores for all routing candidates $h^{(1)}, \ldots, h^{(M)}$. At inference time (bottom), given a prompt, predicted quality scores of all routing candidates are adjusted by their respective costs. The routing candidate that has the highest cost-adjusted score is chosen (see Eq. (3)).

chosen, $h_T : \mathcal{X} \to \mathcal{I}$ is thus an instance of text-to-image generative models as described earlier. The importance of this view will be apparent when we describe our proposed method in Section 3, which enables an automatic selection of the number of denoising steps, separately for each prompt. Typically, the number of denoising steps is pre-chosen according to the computational budget available at inference time, with a low value of T, giving a lower computational cost at the expense of image quality.

3 Cost-Aware Text-To-Image Generation

We now describe our main proposal termed *CATImage* (Cost-Aware Text-based Image Generation), which seeks to minimize inference cost by adaptively adjusting the cost per prompt, depending on its complexity. As illustrated in Figure 1, in the case of a diffusion model, our key observation is that not all prompts require a large number of denoising steps to ensure quality. Thus, inference efficiency can be achieved by spending a small amount of computation for easy prompts. Our proposed framework is general and allows cost adjustment in a per-prompt manner via selecting an appropriate amount of resources from homogeneous choices (i.e., adaptively varying the number of denoising steps of a single diffusion model), or heterogeneous choices (i.e., adaptively route prompts to disparate, independent generative models).

We start by formalizing the cost-aware text-to-image generation task as a learning-to-route problem in Section 3.1. The formulation can be theoretically shown (Section 3.2) to have a simple Bayes optimal routing rule, involving subtracting off the expected quality metrics with the costs of candidate numbers of denoising steps. We show that the optimal rule can be estimated from data, and propose two estimators: a Transformer-based estimator (Vaswani et al., 2017), and a K-nearest neighbors (KNN) model.

3.1 Problem Formulation

Let $[n] \doteq \{1, 2, \ldots, n\}$ denote the set of counting numbers up to n. Suppose that we are given a fixed set of M choices $\mathcal{H} \doteq \{h^{(1)}, \ldots, h^{(M)}\}$ where each choice $h^{(i)} : \mathcal{X} \to \mathcal{I}$ represents a trained generative model (see Section 2 for a precise definition). Our goal is to derive a routing rule that optimally (in the sense of quality-cost trade-offs) chooses the best model to invoke for each input prompt. These M *base models* may be homogeneous, being derived from a single diffusion model with varying numbers of diffusion steps, a mix of heterogeneous generative model classes, or a combination of both. For example, if we want to decide whether to use 20 or 50 number of denoising steps in the Stable Diffusion XL (SDXL) model (Podell et al., 2023), then $M = 2$, and $\mathcal{H} = \{h^{(1)}, h^{(2)}\}$ where the two models are both SDXL with the number of denoising steps fixed to 20 and 50, respectively. We will abstract away the details of the underlying M base models and propose a general framework that supports both the homogeneous and heterogeneous cases (as shown in our experiments in Section 5).

Suppose we are given a quality metric of interest $q\colon \mathcal{X} \times \mathcal{I} \to \mathbb{R}$ (see *Quality Metrics* under Section 5.1), which takes as input a prompt-image tuple, and estimates a quality score. We seek a router $r\colon \mathcal{X} \to [M]$ that predicts the index of the M choices from a given prompt. We posit two desirable properties that the router ought to possess:

1. The router must respect a specified budget constraint on the inference cost.

2. Routing prompts to candidates in $\mathcal{H}$ must maximize average quality metric.

Following similar formulations considered in Jitkrittum et al. (2023; 2025); Mao et al. (2023); Tailor et al. (2024), the above desiderata may be realized as a constrained optimization problem:

$$\max_r Q(r) \;\text{ subject to }\; C(r) \le B, \quad \text{where} \tag{1}$$

$$Q(r) \doteq \mathbb{E}\left[\sum_{m\in[M]} \mathbf{1}\left[r(\mathbf{x}) = m\right] \cdot q(\mathbf{x}, h^{(m)}(\mathbf{x}))\right], \text{ and } \; C(r) \doteq \mathbb{E}\left[\sum_{m\in[M]} \mathbf{1}\left[r(\mathbf{x}) = m\right] \cdot c^{(m)}\right], \tag{2}$$

where for $m \in [M]$, $c^{(m)} \ge 0$ denotes the cost for the model $h^{(m)}$ to produce one image for a given prompt, $\mathbb{E}$ denotes the expectation with respect to the population joint distribution on all random variables (i.e., prompt $\mathbf{x}$, and the sampled output of $h^{(m)}$), $B \ge 0$ is a hyperparameter specifying an upper bound on the average cost. The optimization problem (1) thus seeks a router r that maximizes the average quality $Q(r)$ subject to the constraint that the average cost (over all prompts) is bounded above by B.

Remark. The optimization problem is general and allows the per-model costs to be in any unit suitable for the application (e.g., latency in seconds, FLOP counts). Further, no practical constraint is imposed on the quality metric function q. For instance, q could be the CLIP score (Radford et al., 2021). Intuitively, if the budget B is large, the cost constraint $C(r) \le B$ would have little effect, and the optimal router is expected to route each prompt to the base model that can produce the highest quality metric score, disregarding the cost of the model. In practice, such a model is often the largest one in the pool $\mathcal{H}$, or the diffusion model with the largest number of denoising steps. On the contrary, if B is small, the router would prioritize cost over quality, preferring to choose a small base model (or a small number of denoising steps) over a larger candidate. This proposal offers a framework to allow trading off average quality with cost in a unified way by varying B.

3.2 Theoretically Optimal Routing Rule

Having formulated the constrained problem in (1), we now investigate its theoretically optimal solution. We will use the optimal solution to guide us on how to design a practical router. Based on the results in Jitkrittum et al. (2023; 2025), the optimal solution to (1) is shown in Proposition 1.

Proposition 1. *For a cost budget $B > 0$, the optimal router $r^*\colon \mathcal{X} \to \{1, \ldots, M\}$ to the constrained optimization problem (1) is*

$$r^*(\mathbf{x}) = \arg\max_{m\in[M]} \mathbb{E}\left[q(\mathbf{x}, h^{(m)}(\mathbf{x})) \mid \mathbf{x}\right] - \lambda \cdot c^{(m)},$$

where the conditional expectation is over the sampled output from the model $h^{(m)}$, and $\lambda \ge 0$ is a Lagrange multiplier inversely proportional to B.

The result follows from Proposition 1 in Jitkrittum et al. (2025). The result states that the choice/model we choose to route a prompt $\mathbf{x}$ to is the one that maximizes the average quality, adjusted additively by the cost of the model. The hyperparameter λ controls the trade-off between quality and cost, and is inversely proportional to the budget B. For instance, if $\lambda = 0$ (corresponding to $B = \infty$), then the model with the highest expected quality for $\mathbf{x}$ will be chosen, regardless of its cost. Increasing λ enforces the routing rule to account more for model costs, in addition to the expected quality.

Estimating the Optimal Rule The optimal rule r^* in Proposition 1 depends on the population conditional expectation $\gamma^{(m)}(\mathbf{x}) \doteq \mathbb{E}\left[q(\mathbf{x}, h^{(m)}(\mathbf{x})) \mid \mathbf{x}\right]$, which is unknown. Following a similar reasoning as in Jitkrittum et al. (2025), we propose plugging in an empirical estimator $\hat{\gamma}^{(m)} \colon \mathcal{X} \to \mathbb{R}$ in place of $\gamma^{(m)}$, resulting in the empirical rule $\hat{r}_\lambda$:

$$\hat{r}_\lambda(\mathbf{x}) = \arg\max_{m \in [M]} \hat{\gamma}^{(m)}(\mathbf{x}) - \lambda \cdot c^{(m)}. \tag{3}$$

For each $m \in [M]$, the idea is to train an estimator $\hat{\gamma}^{(m)}$ to estimate the true expected quality. That is, suppose we are given a collection of N training prompts $\{\mathbf{x}_i\}_{i=1}^N$. For each prompt $\mathbf{x}_i$, we may sample S times from $h^{(m)}$ to produce output images $\mathbf{i}_{i,1}^{(m)} \ldots, \mathbf{i}_{i,S}^{(m)}$. These output images allow one to estimate the empirical expectation of the quality $\hat{y}_i \doteq \frac{1}{S}\sum_{s=1}^S q(\mathbf{x}, \mathbf{i}_{i,s}^{(m)})$. With the labeled training set $\{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^N$, we may then proceed to train a predictive model $\hat{\gamma}(\mathbf{x}) \doteq \left(\hat{\gamma}^{(1)}(\mathbf{x}), \ldots, \hat{\gamma}^{(M)}(\mathbf{x})\right)$, which has M output heads for predicting the expected qualities of the M models. There are several standard machine learning models one can use as the model class for $\hat{\gamma}$.

We emphasize that we do not advocate a specific model class as part of our proposal since different model classes offer distinct properties on training and inference costs, which may be best tailored to the application. What we propose is an application of the generic routing rule in (3) to text-to-image model routing. The rule is guaranteed to give a good quality-cost trade-off provided that the estimator $\hat{\gamma}^{(m)}$ well estimates $\gamma^{(m)}$. In experiments (Section 5), we demonstrate estimating $\gamma^{(m)}$ with two model classes: 1) K-nearest neighbors, and 2) Multi-Layer Perceptron (MLP) with a Transformer backbone (Vaswani et al., 2017). Likewise, we do not propose or advocate a specific value of λ. The parameter is left to the user as a knob to control the desired degree of quality-cost trade-off. In experiments, we evaluate our proposed routing rule by considering a wide range of λ and show the trade-off as a deferral curve (see Section 3.3). An illustration summarizing our pipeline is displayed in Figure 2.

3.3 Deferral Curve

In general, any methods that offer the ability to trade off quality and cost may be evaluated via a *deferral curve* (Bolukbasi et al., 2017; Cortes et al., 2016; Gupta et al., 2024; Narasimhan et al., 2022). A deferral curve is a curve showing the average quality against the average cost, in a quality-cost two-dimensional plane. Specifically, for our proposed routing rule $\hat{r}_\lambda$ in (3), the curve is precisely given by $\mathcal{C} = \{(C(\hat{r}_\lambda), Q(\hat{r}_\lambda)) \mid \lambda \in [0, \infty)\}$ where Q and C denote the average quality and cost, and are defined in Eq. (2). In practice, the population expectation in Q and C is replaced with an empirical expectation over examples in a test set. More generally, one evaluates the deferral curve of a method by computing its average quality and cost as we vary parameters that control the trade-off. For instance, for the SDXL diffusion model, we may produce a deferral curve by varying the number of denoising steps.

4 Related Work

Uniform Optimization Strategies for Diffusion Models Diffusion models have recently exploded in popularity due to their high performance on tasks such as image and video generation, audio generation, and 3D shape generation (Ho et al., 2020; Ramesh et al., 2021). Latent diffusion models (Rombach et al., 2022) have significantly improved training and inference efficiency, but still require a large number of forward denoising neural network evaluations to produce high-quality results. To address this, an extensive body of literature has been proposed to optimize and accelerate diffusion models, which are typically applied *uniformly across all prompts*. For example, optimizing the sampling strategy may enable more efficient denoising computation (Li et al., 2024; Chen et al., 2023b; Li et al., 2023a), such as timestep integration (Nichol & Dhariwal, 2021) or conditioning on the denoising (Preechakul et al., 2022). Optimizing solvers for the denoising step can also efficiently reduce the computation to avoid re-training or fine-tuning (Song et al., 2020; Lu et al., 2022; Liu et al., 2022; Karras et al., 2022). Alternatively, reducing the redundant computations by caching the internal results within the denoising network is also explored in (Ma et al., 2024a;b). Another common approach includes model-based optimizations, such as distilling a fully trained model into a smaller student model that achieves comparable results with fewer denoising steps (Sauer et al., 2024; Salimans & Ho, 2022; Meng et al., 2023; Liu et al., 2023) or combining multiple denoising models with different sizes to accelerate the denoising process (Yang et al., 2023; Li et al., 2023b; Pan et al., 2024). An alternative strategy is to approximate the direct mapping from initial noise to generated images, further reducing the number of denoising steps (Luo et al., 2023; Song et al., 2023).

Adaptive Optimization Strategies for Diffusion Models Instead of a fixed reduction in computational resources, AdaDiff (Tang et al., 2023) explores a more dynamic approach where the number of denoising steps is decided based on the uncertainty estimation of the intermediate results during denoising. Our work shares a similar motivation for flexible resource allocation. However, we adaptively allocate resources according to prompt complexity and thus can select the most suitable number of steps or model before any denoising process. Concurrently, AdaDiff (Zhang et al., 2023) tackles optimal number of steps selection using a prompt-specific policy, with a lightweight network trained on a reward function that balances image quality and computational resources. In contrast, we decouple the quality estimation from the routing decision, which allows our framework to adapt to different resource constraints without any retraining.

Learning-To-Defer, and Modeling Routing The idea of adaptively invoking a different expert on each input is a widely studied area in machine learning under the topic of *learning to defer*. Here, each expert may be a human expert (Mozannar & Sontag, 2020; Mozannar et al., 2023; Sangalli et al., 2023), or a larger model (Narasimhan et al., 2022; Jitkrittum et al., 2023; Mao et al., 2023; Gupta et al., 2024). In the latter, depending on the topology or order the models are invoked, a learning-to-defer method may yield a *cascade* if models are arranged in a chain (Wang et al., 2022a; Jitkrittum et al., 2023; Kolawole et al., 2024); or yield a *routed model* if there is a central routing logic (i.e., the router) which selectively sends input traffic to appropriate models (Jiang et al., 2023; Mao et al., 2023; Gupta et al., 2024; Jitkrittum et al., 2025). The latter setup is also known as *model routing* and receives much attention of late, especially in the natural language processing literature. Model routing has been successfully applied to route between many Large Language Model (LLMs) of various sizes and specialties (see Chen et al. (2023a); Hu et al. (2024); Zhuang et al. (2025); Ong et al. (2025); Jitkrittum et al. (2025) and references therein). To our knowledge, our work is one of the first that connects the model routing problem to efficient text-to-image generation.

5 Experiments

In this section, we show how our proposed routing method (Section 3) can be realized in practice by evaluating its effectiveness on real data. We experiment with both homogeneous (i.e., all routing candidates are derived from the same diffusion model with different candidate numbers of denoising steps), and heterogeneous settings (i.e., the routing candidates also include different generative models). Our goal is to optimally select the best model (or number of denoising steps) for each input prompt given a specified cost constraint.

5.1 Experimental Setup

Text-To-Image Generative Models As defined in Section 3.1, our method selects from a set of generative models $\mathcal{H}$ for each input prompt. We consider a diverse range of models with varying configurations, each offering a different trade-off between image quality and computational cost:

1. SDXL: a widely-used SD architecture (Rombach et al., 2022). To see the full extent of the achievable trade-off, we consider representative numbers of denoising steps in a wide range between 1 and 100.

2. Turbo (Sauer et al., 2024) and Lightning (Lin et al., 2024): distilled versions of SDXL for faster generation. We use the SDXL variant with 1 step for Turbo, and 4 steps for Lighting.

3. DDIM (Song et al., 2020): a non-Markovian diffusion process allowing faster sampling. We use this sampling strategy on the SDXL variant at 50 steps.

4. DeepCache (Ma et al., 2024b): a caching method that reduces redundant computation in SDXL. We use the official implementation released from Ma et al. (2024b), and set the cache interval parameter to 3.

5. Infinity (Han et al., 2024): a non-diffusion, auto-regressive text-to-image model based on the Transformer encoder-decoder. We use the pre-trained Infinity-2B variant with a visual vocabulary size of 2^{32}.

Quality Metrics The effectiveness of generative models largely depends on the criteria used to evaluate their output. Our proposed method can adaptively identify the optimal allocation of generative model for *any* instance-level image quality metric. As there is no consensus on the optimal metric for evaluating image quality, we explore several widely-used metrics: *CLIPScore* (Radford et al., 2021) for text-image semantic alignment, *ImageReward* (Xu et al., 2023) with a reward model tuned to human preferences, and *Aesthetic Score* (Beaumont & Schuhmann, 2022) trained on human aesthetic ratings from LAION (Schuhmann et al., 2022). Additionally, we also introduce *Sharpness* metric adapted from Paris et al. (2011), defined as, $q_{\text{Sharp}}(\mathbf{x}, \mathbf{i}) = \frac{\sum_{ij}(\mathbf{i}_{ij} - [\mathbf{i} \circledast G]_{ij})^2}{\sum_{ij} \mathbf{i}_{ij}^2}$, where $\circledast$ denotes the convolution operator, $\mathbf{i}_{i,j}$ is the pixel intensity at location (i, j), and G is a Gaussian kernel with standard deviation of 1. Intuitively, this metric measures the relative distance between the given image $\mathbf{i}$ and itself after a Gaussian blur filter is applied.

Quality Estimator $\hat{\gamma}$ One of the key components of our routing method is the quality estimator which estimates the expected quality of the m-th model given an input prompt (see $\hat{\gamma}^m$ in Eq. (3)). We explore two model classes: a K-Nearest Neighbors (K-NN) model and a Transformer-based model. Both of these models incur a negligible inference cost: less than 0.001 TFLOPs compared to 1.5 TFLOPs of the smallest base model in the pool (Infinity).

The K-NN approach provides a non-parametric way to estimate quality by averaging the quality scores of K nearest training prompts in the space of CLIP embeddings (Radford et al., 2021). This method is simple, and can generalize well with sufficient data. The Transformer model takes as input the per-token embeddings produced by the frozen CLIP text encoder. A two-layer MLP with M output heads is added to each output token embedding. Pooling across all tokens gives M output scores $\hat{\gamma}^{(1)}(\mathbf{x}), \ldots, \hat{\gamma}^{(M)}(\mathbf{x})$ (see Eq. (3)), each estimating the expected quality of the m-th model on prompt $\mathbf{x}$ (see Appendix Section E for details).

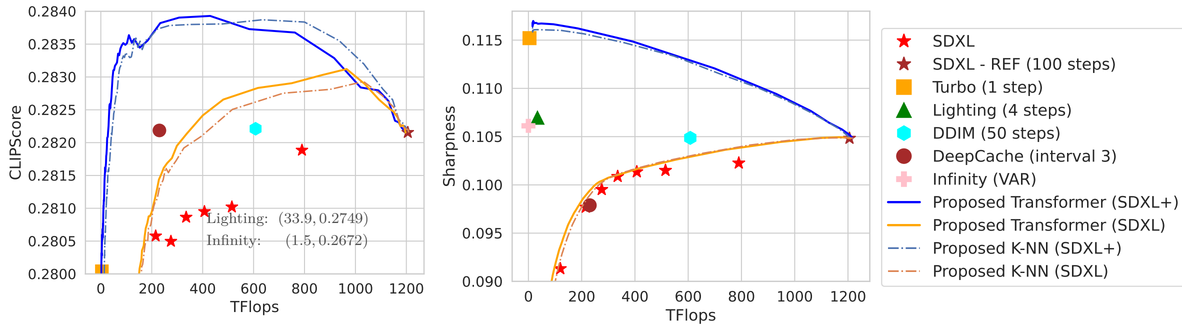
All base models except Infinity already use CLIP embeddings, making router overhead negligible. Infinity uses Flan-T5 embeddings ($\approx$ 13 GFLOPs overhead), but this cost is minimal compared to one SDXL call ($\approx$ 200 TFLOPs for 17 steps).

We train a separate model for each of the quality metrics considered. In each case, the quality scores are linearly scaled across all training examples to be in [0, 1]. These scaled metrics are treated as ground-truth probabilities, and the model is trained by minimizing the sum of the sigmoid cross-entropy losses across all heads.

5.2 Dataset Details

We utilize two datasets: 1) the COCO captioning dataset (Lin et al., 2014), which contains high-quality and detailed image captioning, and 2) the DiffusionDB dataset (Wang et al., 2022b), which contains a larger collection of realistic, user-generated text prompts for text-to-image generation. From both datasets, we sub-sample prompts by retaining only those with pairwise CLIP similarity below 0.75, resulting in a diverse set of 18,384 prompts in COCO dataset, and 97,841 prompts on the DiffusionDB dataset. We split each dataset independently into 80% for training, 10% for validation, and 10% for testing. We then generate images from those prompts using all the base text-to-image models as described earlier. For SDXL, we generate images with various numbers of denoising steps ranging from 1 to 100. The costs in terms of FLOPs from these candidates cover the full range of costs of all other baselines.

For each model, we generate four images per prompt (i.e., $S = 4$ in Section 3.2) using different random seeds, with a fixed seed across different numbers of steps for SDXL. The generated images for each prompt $\mathbf{x}_i$ allow us to compute the average quality metric, which is then used as the training label $\hat{y}_i$ (as described in Section 3.2). Unless otherwise specified, we use the widely used Euler Scheduler (Karras et al., 2022) for diffusion-based image generation.



(a) CLIPScore (b) Sharpness

Figure 3: Deferral curves of our proposed methods and baselines on the COCO dataset as described in Section 5.1, where the quality metric is measured by CLIPScore (Sub-figure **(a)**.) and pixel sharpness (Sub-figure **(b)**.), which are presented in *Quality Metrics* under Section 5.1. Our *Proposed Transformer (SDXL+)*, which considers all the numbers of diffusion steps of SDXL and other baselines as candidate choices to route to, offers the best quality-cost trade-off, where cost is measured in TFLOPs. In Figure 3a, baselines that are not visible are shown at the bottom-right corner in the format of (cost, CLIPScore).

5.3 Experiments on COCO dataset

We present experimental results on a subset of COCO's test set (Lin et al., 2014) consisting of 1.8k image-caption pairs in Figure 3. We evaluate the deferral curves (see Section 3.3) of our proposed method and all the baselines. The results are shown in Figures 3a and 3b for the two different quality metrics: CLIPScore and image sharpness (Section 5.1), respectively. The deferral curves plot average quality against average cost measured in TFLOPs (Tera Floating Point Operations). Baselines that do not support dynamic quality-cost trade-off are shown as isolated dots in the same quality-cost plane; these baselines use the same compute cost for image generation for each input prompt. For instance, each point ★ of SDXL represents the performance of the SDXL model with the number of denoising steps fixed. For our proposed methods, *Proposed (SDXL)* refers to the homogeneous configuration in which the model candidate set $\mathcal{H}$ consists solely of the SDXL model at multiple numbers of denoising steps settings. *Proposed (SDXL+)* extends this configuration by incorporating other text-to-image models considered, namely, Turbo, DDIM, DeepCache, and Infinity. Each of these has two variants based on Transformer or K-NN as the model class for estimating the expected quality metric.

Homogeneous vs. Heterogeneous setting In both settings, our methods outperform baselines with static inference costs per prompt. The heterogeneous setting further benefits from models with strong quality-to-cost trade-offs (e.g., Infinity, Turbo), improving our dynamic routing's effectiveness and cost-efficiency. Moreover, our strategy remains adaptive, seamlessly allocating prompts to higher-performance models when additional computational resources are available, improving performance beyond what is attainable using each model alone (see Appendix Section H for details on model selection rates).

Transformer vs. KNN Between the two proposed variants, the Transformer-based variant generally outperforms the K-NN variant, suggesting that directly learning to predict the quality metric can be more effective than estimating it from neighboring prompts.

Qualitative Analysis In Figure 4, we analyze scenarios showing both successes and failures of our adaptive routing method (*Proposed Transformer (SDXL+)* on CLIPScore metric). Specifically, we focus on cases where our method uses the same overall computational cost as the baseline (SDXL with a fixed 22 denoising steps). Within these scenarios, we consider cases where our method allocates more than 22 denoising steps, indicating that the prompts are particularly complex and require additional refinement.

For the prompt *A young kid stands before a birthday cake decorated with Captain America*, our method correctly recommends more denoising steps, as fewer would not generate accurate images. In contrast, the prompt *There are two traffic signals on a metal pole, each with three light signals on them* includes an exact

Success case. A young kid stands before a birthday cake decorated with Captain America

Failure case. There are two traffic signals on a metal pole, each with three light signals on them.

(a) 22 steps (fixed) (b) 27 steps (routed) (c) 22 steps (fixed) (d) 27 steps (routed)

Figure 4: Success and failure cases of the baseline SDXL with static 22 denoising steps, and our approach *Proposed Transformer (SDXL+) in Figure 3a* operating at the same average cost as the baseline. **(a)**, **(b)**: Our approach is able to recognize the need for a larger number of denoising steps to generate an image that matches the prompt. **(c)**, **(d)**: Prompts that specify an exact number of objects are difficult for diffusion models in general. The number of objects may fluctuate during the denoising process, making it difficult to predict the right number of steps.

Table 1: We train the k-NN quality estimator over $T = 100$ trials on COCO. In each trial, we generate S images for each training prompt with 1) $S = 1$ and 2) $S = 3$. Each of these two cases results in T k-NN models, and hence T (random) deferral curves evaluated on the same test set as used in Fig 3. We report the mean performance (CLIPScore) across the T trials for our approach.

CLIPScore ↑								
	Infinity	Turbo	Lighting	SDXL-9	DeepCache	SDXL-22	DDIM	SDXL-65
Ours (S=1)	0.2816	0.2814	0.2828	0.2828	0.2828	0.2828	0.2824	0.2819
Ours (S=3)	0.2816	0.2816	0.2831	**0.2832**	**0.2832**	**0.2832**	0.2828	0.2819
Fixed	0.2816	0.2794	0.2798	0.2773	0.2749	0.2805	0.2820	0.2819
Win Rate (S=1)	-	1	1	1	1	1	0.98	-
Win Rate (S=3)	-	1	1	1	1	1	1	-
TFLOPs	1.5	1.54	23.92	107.73	210	263.34	598.5	778.05

number of objects, a concept which both diffusion models and CLIP often struggle with (Binyamin et al., 2024; Paiss et al., 2023). Our approach accounts for this difficulty by recommending more steps than average. However, in this case, more denoising steps actually degrade image quality which is uncommon and ends up hurting the router performance.

We also perform a user study to compare the subset of these routing decisions with the fixed cost baseline (see Appendix Section I). All participants rate Figure 4b (ours) as the better image, while 14 of 19 participants select Figure 4c (baseline) as the better image.

5.4 Effect of Sample Size S

As the estimation error depends on the randomness in image generation, we aggregate signals from multiple generated images by averaging their quality metric values across multiple random seeds. To further quantify variability across trials, we vary the number of generated images for each prompt and train a separate model for each case. In Table 1, we report the mean performance of CLIPScores on the COCO dataset, as well as the Win Rate, defined as the fraction of trials that our router has higher test quality than the baseline. The maximum standard deviation across trials of both above variants is less than 2×10^{-4} throughout the cost range.

Table 2: Quality-cost trade-off of our proposed approach on DiffusionDB (Section 5.5). We report the average quality (as measured by four different quality metrics) achieved by our routing approach when operating at the cost (TFLOPs) of each model in the pool. For each metric, the highest score achieved is highlighted in bold, which in all cases corresponds to our routing method. Additionally, our approach is able to consistently maintain or exceed the quality using the same cost as each model baseline.

CLIPScore (Radford et al., 2021) ↑									
Ours	0.259 ± 6e-4	0.304 ± 4e-4	0.308 ± 4e-4	0.314 ± 4e-4	0.316 ± 4e-4	0.317 ± 4e-4	**0.318** ± 4e-4	0.318 ± 4e-4	0.318 ± 4e-4
Fixed	0.259 ± 6e-4	0.304 ± 4e-4	0.300 ± 4e-4	0.308 ± 4e-4	0.316 ± 4e-4	0.315 ± 4e-4	0.317 ± 4e-4	0.315 ± 4e-4	0.318 ± 4e-4
Sharpness (Section 5.1) ↑									
Ours	0.131 ± 4e-4	0.135 ± 3e-4	0.135 ± 3e-4	0.136 ± 3e-4	**0.137** ± 3e-4	0.137 ± 3e-4	0.137 ± 3e-4	0.137 ± 3e-4	0.126 ± 3e-4
Fixed	0.131 ± 4e-4	0.122 ± 3e-4	0.110 ± 3e-4	0.103 ± 2e-4	0.101 ± 2e-4	0.114 ± 2e-4	0.123 ± 3e-4	0.107 ± 3e-4	0.126 ± 3e-4
Aesthetic Score (Beaumont & Schuhmann, 2022) ↑									
Ours	6.824 ± 8.3e-3	6.913 ± 8.5e-3	**7.042** ± 9.1e-3	7.032 ± 9.1e-3	7.012 ± 9.0e-3	7.012 ± 9.0e-3	6.935 ± 8.9e-3	6.935 ± 8.9e-3	6.707 ± 8.1e-3
Fixed	6.824 ± 8.3e-3	6.780 ± 9.2e-3	7.010 ± 9.3e-3	6.285 ± 8.2e-3	6.625 ± 9.1e-3	6.588 ± 8.1e-3	6.690 ± 8.1e-3	6.600 ± 9.0e-3	6.707 ± 8.1e-3
ImageReward (Xu et al., 2023) ↑									
Ours	1.029 ± 8.5e-3	1.083 ± 8.0e-3	**1.086** ± 8.0e-3	1.086 ± 7.9e-3	1.079 ± 7.8e-3	1.076 ± 7.8e-3	1.037 ± 7.5e-3	1.037 ± 7.5e-3	0.891 ± 7.4e-3
Fixed	1.029 ± 8.5e-3	0.960 ± 8.1e-3	0.932 ± 8.4e-3	0.497 ± 8.2e-3	0.809 ± 8.8e-3	0.769 ± 7.7e-3	0.866 ± 7.5e-3	0.861 ± 8.6e-3	0.891 ± 7.4e-3
	INFI	**TURB**	**LIGH**	**SDXL**	**DEEP**	**SDXL**	**SDXL**	**DDIM**	**SDXL**
TFLOPs	**1.50**	**1.54**	**23.92**	**119.70**	**210.00**	**239.40**	**598.50**	**598.50**	**1197.00**

The results show that using S=3 improves the test performance compared to S=1. Additionally, the win rate of "Ours (S=1)" compared to the baseline is 100% in almost all cost ranges. This win rate implies that our approaches are statistically significantly better than the baseline according to the sign test at significance level $a < 10^{-6}$. This means using one image per prompt is already sufficient to improve the baseline of using fixed compute costs. Deviation across trials is minimal relative to the mean metric, suggesting that S=3 is sufficient. In all other experiments in the paper, we used S=4.

5.5 Experiments on DiffusionDB dataset

In this section, we present results on a subset of prompts from the DiffusionDB dataset (Wang et al., 2022b), which aligns more closely with real-world prompts used in text-to-image generation. We evaluate the performance across four metrics: *CLIPScore*, *ImageReward*, *Aesthetic Score*, and *Sharpness*.

Quantitative results comparing our dynamic routing method to the fixed-model baselines are summarized in Table 2. This table effectively captures the trade-offs shown in the deferral curves at a specific cost equal to each baseline. We use KNN as a quality estimator to efficiently evaluate multiple metrics at scale. The results show that our method consistently matches or exceeds fixed-model baseline performance across all four quality metrics. Additionally, the highest value of each score (highlighted in Table 2 in bold) is attainable *only* with our routing strategy. In other words, even under an unconstrainedand computational budget, none of the individual baselines can attain the quality that our adaptive routing achieves through prompt-based allocation across the model pool.

Table 3: Cost ratio (%) of our method compared to baselines to match the quality score (Sharpness)

Model	Our cost
Infinity	100%
Turbo	97.40%
Lighting	6.27%
DeepCache	0.71%
DDIM	0.25%
SDXL100	0.13%

Table 3 quantifies the computational cost reduction achieved by our routing method compared to the baseline at equivalent quality levels (on Sharpness metric). For inherently efficient models (e.g. Infinity(Han et al., 2024), Turbo (Sauer et al., 2024)), the savings appear marginal. However, compared to Lighting (Lin et al., 2024), a *distilled* SDXL variant, our method achieves the same performance at only 6% of its computational cost. For higher-performance models, such as SDXL at 100 denoising steps, the savings are even more significant.

We refer to the appendix for additional qualitative and quantitative analysis, as well as an extended evaluation that includes more recent generative models (i.e. FLUX).

5.6 Conclusion, Limitation, and Future Work

In this paper, we present *CATImage*, a cost-aware routing approach that dynamically selects optimal models and numbers of denoising steps based on prompt complexity. We show that incorporating multiple base models, such as distilled versions of diffusion models and alternative architectures, improves the quality–cost trade-off. Extensive experiments on COCO and DiffusionDB datasets across multiple quality metrics validate our method's effectiveness and generalization capability. Nevertheless, several limitations are worth highlighting. To determine optimal routing decisions, *CATImage* relies on estimating the expected quality *per* prompt, which excludes metrics such as Fréchet Inception Distance (FID) (Heusel et al., 2017) that measure statistical similarity across the entire image distribution. Additionally, our method does not explicitly account for uncertainty arising from different noise variations during the generation process. Addressing these limitations remains a direction for future research.

References

Romain Beaumont and Christoph Schuhmann. LAION-Aesthetics Predictor V1. https://github.com/LAION-AI/aesthetic-predictor, 2022. GitHub repository, commit 6d122ad (15 Aug 2022); accessed 15 May 2025.

Lital Binyamin, Yoad Tewel, Hilit Segev, Eran Hirsch, Royi Rassin, and Gal Chechik. Make it count: Text-to-image generation with an accurate number of objects. *arXiv preprint arXiv:2406.10210*, 2024.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 527–536. JMLR.org, 2017.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance, 2023a. URL https://arxiv.org/abs/2305.05176.

Yu-Hui Chen, Raman Sarokin, Juhyun Lee, Jiuqiang Tang, Chuo-Ling Chang, Andrei Kulik, and Matthias Grundmann. Speed is all you need: On-device acceleration of large diffusion models via gpu-aware optimizations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4650–4654, 2023b.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *Algorithmic Learning Theory: 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings 27*, pp. 67–82. Springer, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KgaBScZ4VI.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis, 2024. URL https://arxiv.org/abs/2412.04431.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 6626–6637, 2017.

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

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23 (47):1–33, 2022.

Qitian Jason Hu, Jacob Bieker, Xiuyu Li, Nan Jiang, Benjamin Keigwin, Gaurav Ranganath, Kurt Keutzer, and Shriyash Kaustubh Upadhyay. Routerbench: A benchmark for multi-LLM routing system. In *Agentic Markets Workshop at ICML 2024*, 2024. URL https://openreview.net/forum?id=IVXmV8Uxwh.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise comparison and generative fusion. In *Proceedings of the 61th Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 2023.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. When does confidence-based cascade deferral suffice? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=4KZhZJSPYU.

Wittawat Jitkrittum, Harikrishna Narasimhan, Ankit Singh Rawat, Jeevesh Juneja, Zifeng Wang, Chen-Yu Lee, Pradeep Shenoy, Rina Panigrahy, Aditya Krishna Menon, and Sanjiv Kumar. Universal model routing for efficient llm inference, 2025. URL https://arxiv.org/abs/2502.08773.

Lynn H Kaack, Priya L Donti, Emma Strubell, George Kamiya, Felix Creutzig, and David Rolnick. Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, 12(6):518–527, 2022.

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.

Kate Crawford. Generative ai's environmental costs are soaring — and mostly secret. *Nature World View*, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Steven Kolawole, Don Dennis, Ameet Talwalkar, and Virginia Smith. Agreement-based cascading for efficient inference, 2024. URL https://arxiv.org/abs/2407.02348.

Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. Autodiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7105–7114, 2023a.

Wenhao Li, Xiu Su, Shan You, Tao Huang, Fei Wang, Chen Qian, and Chang Xu. Not all steps are equal: efficient generation with progressive diffusion models. *arXiv preprint arXiv:2312.13307*, 2023b.

Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36, 2024.

Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *ArXiv*, abs/2402.13929, 2024. URL https://api.semanticscholar.org/CorpusID:267770548.

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*, pp. 740–755. Springer, 2014.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

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.

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.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *Advances in Neural Information Processing Systems*, 37:133282–133304, 2024a.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. Two-stage learning to defer with multiple experts. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GIlsH0T4b2.

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*, pp. 14297–14306, 2023.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International conference on machine learning*, pp. 7076–7087. PMLR, 2020.

Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. Who should predict? exact algorithms for learning to defer to humans. In *International conference on artificial intelligence and statistics*, pp. 10520–10545. PMLR, 2023.

Harikrishna Narasimhan, Wittawat Jitkrittum, Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Post-hoc estimators for learning to defer to an expert. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_jg6Sf6tuF7.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. *arXiv preprint arXiv:2302.12066*, 2023.

Zizheng Pan, Bohan Zhuang, De-An Huang, Weili Nie, Zhiding Yu, Chaowei Xiao, Jianfei Cai, and Anima Anandkumar. T-stitch: Accelerating sampling in pre-trained diffusion models with trajectory stitching. *arXiv preprint arXiv:2402.14167*, 2024.

Sylvain Paris, Samuel W Hasinoff, and Jan Kautz. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph.*, 30(4):68, 2011.

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, 2023. URL https://arxiv.org/abs/2307.01952.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10619–10629, 2022.

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.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pp. 8821–8831. Pmlr, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. URL https://arxiv.org/abs/2205.11487.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Sara Sangalli, Ertunc Erdil, and Ender Konukoglu. Expert load matters: operating networks at high accuracy and low manual effort. *Advances in Neural Information Processing Systems*, 36:16283–16301, 2023.

Sarah Wells. Generative ai's energy problem today is foundational. *IEEE Spectrum*, 2023.

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: an open large-scale dataset for training next generation image-text models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

Dharmesh Tailor, Aditya Patra, Rajeev Verma, Putra Manggala, and Eric Nalisnick. Learning to defer to a population: A meta-learning approach, 2024. URL https://arxiv.org/abs/2403.02683.

Shengkun Tang, Yaqing Wang, Caiwen Ding, Yi Liang, Yao Li, and Dongkuan Xu. Adadiff: Accelerating diffusion models through step-wise adaptive computation. *arXiv preprint arXiv:2309.17074*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Xiaofang Wang, Dan Kondratyuk, Eric Christiansen, Kris M. Kitani, Yair Movshovitz-Attias, and Elad Eban. Wisdom of committees: An overlooked approach to faster and more accurate models. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=MvO2t0vbs4-.

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. *arXiv:2210.14896 [cs]*, 2022b. URL https://arxiv.org/abs/2210.14896.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2096–2105, 2023.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: learning and evaluating human preferences for text-to-image generation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 15903–15935, 2023.

Shuai Yang, Yukang Chen, Luozhou Wang, Shu Liu, and Yingcong Chen. Denoising diffusion step-aware models. *arXiv preprint arXiv:2310.03337*, 2023.

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

Hui Zhang, Zuxuan Wu, Zhen Xing, Jie Shao, and Yu-Gang Jiang. Adadiff: Adaptive step selection for fast diffusion. *ArXiv*, abs/2311.14768, 2023. URL https://api.semanticscholar.org/CorpusID:265456553.

Richard Zhuang, Tianhao Wu, Zhaojin Wen, Andrew Li, Jiantao Jiao, and Kannan Ramchandran. EmbedLLM: Learning compact representations of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Fs9EabmQrJ.