

Edge-Cloud Routing for Text-to-Image Model with Token-Level Multi-Metric Prediction

Zewei Xin, Qinya Li, Chaoyue Niu, Fan Wu

Department of Computer Science and Engineering, Shanghai Jiao Tong University

Abstract

Large text-to-image models demonstrate impressive generation capabilities; however, their substantial size necessitates expensive cloud servers for deployment. Conversely, light-weight models can be deployed on edge devices at lower cost but often with inferior generation quality for complex user prompts. To strike a balance between performance and cost, we propose a routing framework, called *RouteT2I*, which dynamically selects either the large cloud model or the light-weight edge model for each user prompt. Since generated image quality is challenging to measure directly, *RouteT2I* establishes multi-dimensional quality metrics, particularly, by evaluating the similarity between the generated images and both positive and negative texts that describe each specific quality metric. *RouteT2I* then predicts the expected quality of the generated images by identifying key tokens in the prompt and comparing their impact on the quality. *RouteT2I* further introduces the Pareto relative superiority to compare the multi-metric quality of the generated images. Based on this comparison and predefined cost constraints, *RouteT2I* allocates prompts to either the edge or the cloud. Evaluation reveals that *RouteT2I* significantly reduces the number of requesting large cloud model while maintaining high-quality image generation.

1. Introduction

Nowadays, text-to-image (T2I) models like Imagen [3], Stable Diffusion [25], and DALL-E [24] have achieved significant success in generating diverse, high-quality images given user prompts. However, the impressive generation quality comes with large model and high cost. For example, Stable Diffusion 3.5 [2] has 8 billion parameters. Such a large model necessitates reliance on cloud servers, leading to high serving cost. As shown in Tab. 1, it is particularly costly in commercial scenarios with millions of requests¹.

The high cost of cloud-based image generation drives the

Text-to-Image Model	#Parameter	Pricing (\$/M)
DALL-E [24]	12 B	-
DALL-E 2 [23]	3.5 B	20 K
DALL-E 3 [4]	-	80 K
Imagen [3]	-	50 K
Stable Diffusion 1.6 [25]	0.86 B	9 K
Stable Diffusion XL [21]	2.6 B	9 K
Stable Diffusion 3 [8]	8 B	65 K
Stable Diffusion 3.5 [2]	8 B	65 K
Parti [32]	20 B	-
Playground v3 [16]	24 B	-
FLUX 1.1 [1]	12 B	40 K

Table 1. Overview of popular text-to-image models, including their parameters and the cost of generating millions of images.

trend of deploying T2I models on edge devices. Techniques such as quantization [13, 30, 34], structural pruning [5, 14], and reducing denoising steps [17, 19, 27] are used to minimize model sizes and speed up inference, showing that edge deployment is feasible. Compared to cloud models, edge models leverage users' local computing devices to provide instant services anytime and anywhere without cloud serving cost and communication overhead. However, the lightweight edge T2I models often have lower generation quality compared to the cloud large models.

Although cloud models often offer superior quality, not all user prompts need these large models. For simple user prompts, lightweight edge models can produce comparable or even better results. To ensure high-image quality generation while limiting requests to cloud models to reduce cost, it is necessary to introduce a routing mechanism such that suitable models are selected for different user prompts based on their complexity. Intuitively, as shown in Fig. 1, a routing framework for T2I models can route only hard prompts to the cloud, while handling easy ones on the edge.

Previous routing methods are primarily designed for classification models [11] and large language models (LLMs) [7, 9, 18, 20], presenting significant challenges when applied to T2I models. One major challenge lies in measuring the quality of generated images. Unlike text, im-

¹<https://artificialanalysis.ai/text-to-image>

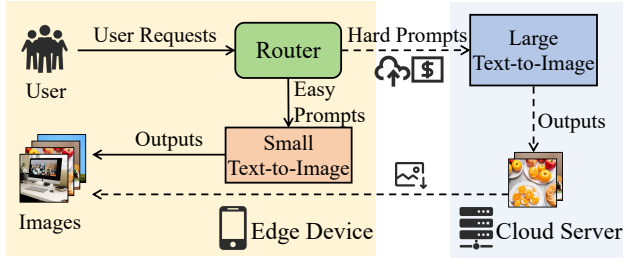


Figure 1. Edge-cloud routing for text-to-image generation models.

age quality does not have unified metrics and is subject to many influencing factors, including color, clarity, and the unique distortions typical of generated images [12]. This complicates the objective and comprehensive evaluation of generated image quality. Furthermore, the output space of T2I models is substantially larger than the input text space, and a single text description can correspond to numerous images, hindering the effective prediction of image quality based on text prompts.

To address these challenges, we propose a new edge-cloud T2I routing framework RouteT2I , deploying a routing model and a small T2I generation model on the edge, and a large T2I generation model on the cloud. The design goal is to optimize the overall quality of generated images at a limited cost, by effectively routing user prompts to the edge or the cloud. To evaluate the quality gap between generated images, we first propose a multi-metric quality measure, including metrics for both traditional photos and those unique to generated images. Each metric assesses image quality by measuring the similarity between the image and a pair of positive and negative prompts that describe the quality metric. We then define Pareto relative superiority (PRS) to quantify quality differences between generated images. RouteT2I predicts PRS based on user prompts, inspired by the cross-attention operation between text and image during generation. In particular, we design a novel Mixture-of-Experts (MoE) [28] network, dual-gate token selection MoE, where a user prompt is treated as a sequence of tokens, with experts aligned to quality metrics. A token selection gate allows experts to proactively choose tokens that markedly impact these metrics, thereby focusing on key tokens. A dual-gate MoE leverages both positive and negative gates to evaluate dominant influences when tokens have opposing effects simultaneously. Furthermore, multiple heads are introduced to predict multiple quality metrics. Based on the multi-metric quality prediction, RouteT2I determines quality differences between images generated on the edge over those in the cloud. Only user prompts that show substantial quality improvement when generated in the cloud are routed there, while others remain at the edge.

We summarize the key contributions as follows:

- We, for the first time, consider and formulate the problem of text-to-image model routing between cloud and edge.
- We propose a contrastive multi-metric quality measure method for generated images and further design a routing model architecture with dual gate token selection MoE coupled with a routing strategy for user prompts.
- We evaluate the proposed design using the public COCO2014 dataset [15] and 10 pairs of edge-cloud generation models. Key results include a relative quality improvement of 83.97% at a routing rate of 50%, and the cloud request reduction of 70.24% at the quality target of 50%.

2. Related Work

Previous model routing work primarily focused on classification models and large language models (LLMs). Depending on whether routing decision occurs before or after weak model execution, model routing can be categorized into predictive and non-predictive routing.

Non-predictive routing takes the output of a weaker model to decide whether to route to the next, more powerful model. LAECIPS [10] evaluates the confidence of predictions made by the edge semantic segmentation model to determine whether assistance from the cloud’s large visual model is necessary. For classification models, Hybrid [11] employed a routing model to partition the data domain between edge and cloud models, routing cases with incorrect edge outputs to the cloud. In the context of LLM, LLM Cascade [33] routed based on the consistency of responses from weaker LLMs, while FrugalGPT [6] considered current quality and historic total costs in the cascading LLM to decide on routing. Tabi [31] not only routed according to the confidence of responses from weaker LLMs but also enhanced output quality by aggregating historic outputs.

Predictive routing selects the appropriate model before running the initial, weaker model. Previous studies focused on LLM routing, with the difference in routing strategies. RouteLLM [20] predicted the outcomes of quality comparisons between LLM outputs and routed based on prediction confidence. Hybrid LLM [7] relaxed the comparison criteria, allowing the weak model to succeed if the quality gap is within a threshold, thus saving costs with some quality compromise. Recent research [29] advances the confidence routing strategy by identifying out-of-distribution data through confidence thresholds. ZOOTER [18] introduced a new routing strategy by predicting the normalized quality of candidate model outputs, making routing decision based on relative quality, with routing models distilled from existing quality scoring models.

3. Problem Formulation

In text-to-image generation scenario, the cloud normally hosts a large T2I model, denoted as \mathcal{M}_c , while a resource-constraint edge device, such as a smartphone, often deploys a lightweight T2I model, denoted as \mathcal{M}_e . For example, Google has released Imagen [3, 26] for cloud deployment and MobileDiffusion [35] for edge deployment. From the perspective of a user, the large model generally offers superior generation quality, but incurs edge-cloud communication cost and cloud serving fee, which together are denoted as F_c . In contrast, the edge model can leverage the user’s local device, thereby eliminating these additional costs.

In practice, user requests involve not only complex ones but also simple ones. For some requests, particularly simple text prompts, the performance gap between the large model and the light-weight model is negligible, and sometimes, the lightweight model even performs better. In such cases, the purely large model serving not only fails to improve performance but also incurs high cost. Therefore, it is necessary to design a routing mechanism to select the most suitable model based on the complexity of the prompt. Intuitively, the desired routing mechanism should direct simpler prompts to the cost-effective edge model and more complex prompts to the high-quality cloud model, thus maximizing generation quality while minimizing cost.

Formally, a T2I routing framework $R : \mathcal{X} \rightarrow \{0, 1\}$ assigns user prompts \mathcal{X} to the cloud large model \mathcal{M}_c as 1 and to the edge light-weight model \mathcal{M}_e as 0. The generated images after routing can be expressed as $\mathcal{I}_r = R(\mathcal{X})\mathcal{I}_c + (1 - R(\mathcal{X}))\mathcal{I}_e$, where \mathcal{I}_c and \mathcal{I}_e denote the images generated using \mathcal{M}_c and \mathcal{M}_e , respectively. Given a quality scoring function $\mathcal{Q}(\cdot)$ of the generated images, the cost budget τ_{fee} of cloud model serving and edge-cloud communication, and the response latency constraint τ_{time} , the optimization objective is maximizing the quality under the budget and latency constraints, formulated as

$$\begin{aligned} & \max R(\mathcal{X})\mathcal{Q}(\mathcal{I}_c) + (1 - R(\mathcal{X}))\mathcal{Q}(\mathcal{I}_e) \\ & \text{s.t. } \mathbb{P}\{R(\mathcal{X}) = 1\} \cdot F_c \leq \tau_{fee} \\ & \quad \mathbb{P}\{R(\mathcal{X}) = 1\} \cdot D_{\mathcal{M}_c} + \mathbb{P}\{R(\mathcal{X}) = 0\} \cdot D_{\mathcal{M}_e} \\ & \quad + D_R \leq \tau_{time}, \end{aligned} \quad (1)$$

where $D_{\mathcal{M}_c}$, $D_{\mathcal{M}_e}$, and D_R denote the latency of cloud large model serving plus edge-cloud communication, the latency of the edge light-weight model serving, and the latency of the routing model execution. Considering the constraints are linear, the optimization objective can be simplified to imposing an upper bound ρ_r on the routing rate to the cloud large model as

$$\begin{aligned} & \max R(\mathcal{X})\mathcal{Q}(\mathcal{I}_c) + (1 - R(\mathcal{X}))\mathcal{Q}(\mathcal{I}_e) \\ & \text{s.t. } \mathbb{P}\{R(\mathcal{X}) = 1\} \leq \rho_r. \end{aligned} \quad (2)$$

Due to the lack of a universal standard for evaluating the quality of generated images, a multi-dimensional quality scoring function $\mathcal{Q} : \mathcal{I} \rightarrow [0, 1]^N$ is required for a more comprehensive comparison. As a result, different from existing routing frameworks for typical classification and text generation tasks, the T2I routing framework’s optimization objective is a new multi-objective problem. On the other hand, from the perspective of input-to-output mapping complexity, the considered T2I task presents a unique challenge, because the text input space is significantly smaller than the image output space. In contrast, the classification task has a more constrained output space, while the text generation task involves comparable input and output spaces, rendering previous routing designs inapplicable for T2I tasks. By examining the interaction between text and images during the T2I generation process, we conduct a detailed analysis of how different prompt tokens influence the generated image and its multi-dimensional quality metrics. In what follow, we define the contrastive multi-metric quality measure in Sec. 4 and introduce the T2I routing framework in Sec. 5.

4. Contrastive Multi-Metric Image Generation Quality Measure

The connection between text and image enables the depiction of an image’s properties through corresponding textual descriptions. We thus propose a quality metric by evaluating the alignment between the text prompt that describe this metric and the image, typically, the cosine similarity between text features and image features using CLIP [22]. To enhance the accuracy and stability of image quality measure, we introduce positive and negative text prompts that describe opposing levels of the quality metric for contrast. The contrastive quality metric of image I is defined as

$$q(I, m) = \sigma(\text{CLIP}(I, m^+) - \text{CLIP}(I, m^-)), \quad (3)$$

where $m = (m^+, m^-)$ denotes positive and negative pair, and σ denotes the sigmoid function that transforms output values in the range from 0 to 1. According to this definition, if an image is more related to the positive prompt, and is less related to the negative prompt, the contrastive quality metric is higher. Compared to quality measure with only the positive prompt, the contrastive method evaluates whether positive or negative quality is more dominant in the image, leading to a more robust and reliable evaluation.

Considering that the quality of real photos relies on many factors, such as definition and color, we introduce multi-dimensional metrics to comprehensively measure the quality of generated images, offering stability against noise and uncertainty. In addition to the above metrics for real photos, we also incorporate unique metrics specifically for generated images, such as realism and object integrity, to establish a comprehensive set of multiple metrics. For each

original quality metric i out of all the N metrics, we design a pair of negative and positive prompts $m_i = (m_i^+, m_i^-)$ that describe i in text. The N -dimensional contrastive quality metrics for an image I can be expressed as

$$\mathcal{Q}(I) = [q(I, m_i) | i = 1, 2, \dots, N], \quad (4)$$

where m_i is an instantiation of m in Eq. (3). For example, for the quality metric of definition, we use the positive prompt ‘‘High definition photo’’ and the negative prompt ‘‘Low definition photo’’; and for the the quality metric of object integrity, we take the positive prompt ‘‘Object completion’’ and the negative prompt ‘‘Object twisting’’.

In the multi-objective optimization problem setting for routing with multi-metric quality, the initial goal is to find a Pareto optimal generated image that excels in all the metrics. However, this is challenging in practice due to the potential nonexistence of such an image generated from the cloud or edge model. We thus relax the constraint of Pareto optimality by allowing suboptimal performance in some metrics if the image significantly outperforms in others when selecting higher-quality images. We define Pareto relative superiority (PRS) to quantify the quality advantage of the images \mathcal{I}_e generated by the edge model over the images \mathcal{I}_c generated by the cloud model. In particular, we first normalize the quality distance for the metric i between an edge-generated image $I_e \in \mathcal{I}_e$ and a cloud-generated image $I_c \in \mathcal{I}_c$ as

$$D_i(I_e, I_c) = \sigma \left(\frac{q(I_e, m_i) - q(I_c, m_i)}{\Gamma |\mu_i(\mathcal{I}_e) - \mu_i(\mathcal{I}_c)|} \right), \quad (5)$$

where $\mu_i(\cdot)$ is the average quality of the set of cloud or edge generated images, while the sigmoid function σ and the temperature parameter Γ are used to modulate the data distribution, effectively distinguishing similar qualities and preventing centralization. We then define the overall quality gap as a weighted sum of distances across all N metrics:

$$PRS(I_e, I_c) = \sum_{i=1}^N w_i D_i(I_e, I_c), \quad (6)$$

where w_i denotes the importance weight of the metric i , and $\sum_i w_i = 1$. We can evaluate the relative quality of I_e compared to I_c by examining how much Pareto relative superiority deviates from 0.5. This evaluation helps to route a user prompt to the model that generates an image with higher overall quality.

5. Design of RouteT2I

To achieve the optimization objective in Eq. (2), we propose a text-to-image model routing framework RouteT2I. As shown in Fig. 2, RouteT2I comprises a routing model that predicts the multi-metric generation quality given user

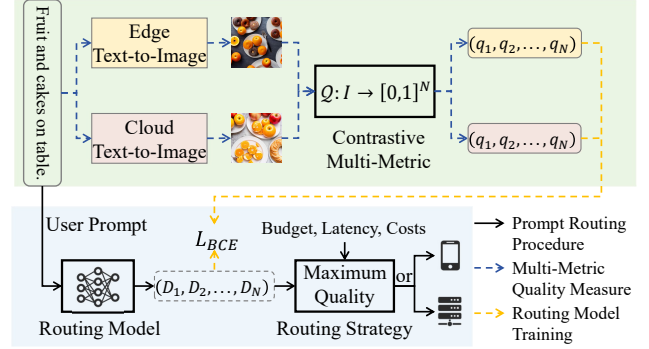


Figure 2. Overview of RouteT2I. RouteT2I assembles a pair of off-the-shelf edge and cloud text-to-image models and evaluates them using multi-metric quality across diverse prompts. RouteT2I utilizes Pareto relative superiority between qualities as supervision to train the routing model. Then, to balance quality and cost, the routing strategy determines the most suitable model for each user prompt, choosing between the edge or cloud model.

prompts and a routing strategy that selects cloud or edge model. In particular, (1) the routing model predicts the quality relationship between images generated on the edge and cloud, namely, Pareto relative superiority. Inspired by the role of prompts in generation processes, where prompt as sequences of tokens interact with images through cross-attention to determine the image content, we treat prompt as a set of tokens in the routing model and select key tokens with influential contextual information for each quality metric. Then, the dominant influence of these tokens is accessed by comparing their positive and negative effects; and (2) the routing strategy aims to maximize quality under cost constraints. The key idea is to use Pareto relative superiority to describe the quality disparity of images generated by the cloud and the edge models. Only prompts that show a notable quality gap when generated in the cloud are routed there, while the rest remain on the cost-effective edge.

5.1. Routing Model Architecture Design with Dual-Gate Token Selection MoE

To predict Pareto relative superiority between the images generated by the cloud model and the edge model from user prompts, we design a dual-gate token selection mixture-of-experts (MoE), as shown in Fig. 3, and introduce it into Transformer network to replace linear layers.

Our model treats the user prompt as a sequence of tokens, whose influences on the image quality differ due to various weights in cross-attention during generation. To focus on key tokens for each metric, we design a token selection gate, where experts align with quality metrics and actively choose the most relevant tokens to simulate metric attention to different image attributes. Only tokens that significantly impact the quality metrics corresponding to the

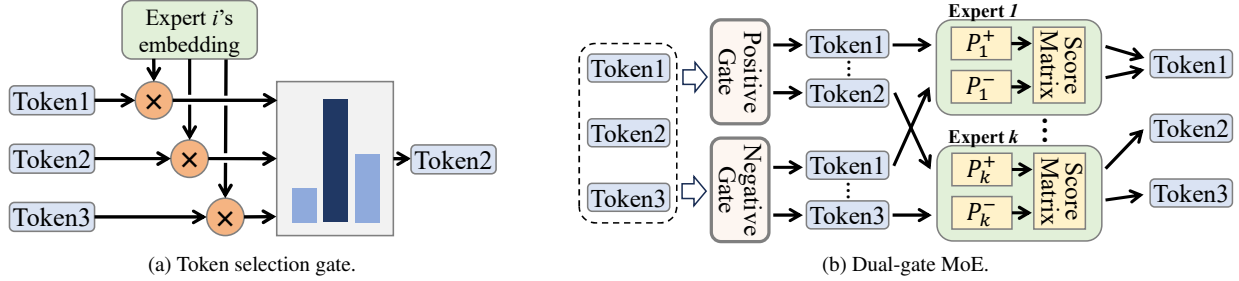


Figure 3. Architecture of a dual-gate MoE that uses the token selection gate as the gate network. First, the gate selects the most relevant tokens for each expert. Then, experts project the tokens selected by positive and negative gates into their respective spaces and evaluate them using score matrices. Finally, the evaluations of the tokens are gathered and compared to form new tokens.

experts are selected, while those with minimal influences are disregarded, reducing potential interference.

Regarding the design of a dual-gate MoE, both positive and negative gates are introduced to distinguish the opposing aspects of token influences on image quality. Similar to images that have both superior and inferior regions when evaluated by metrics, these tokens are assessed by the corresponding experts for their positive and negative influences to each metric. Additionally, we introduce a comparison between two gates to identify which influence of each token predominates overall. We further improve the structure of expert to handle both positive and negative influences on a single metric simultaneously.

To support multi-metric quality prediction, our design incorporates multiple prediction heads within the model architecture. Each head outputs the prediction of a specific quality metric. Such multi-head design not only enhances noise resistance, thereby increasing robustness, but also facilitates a comprehensive evaluation of quality.

5.1.1. Token Selection Gate

In the token selection gate, the dot product between the token representations $T \in \mathbb{R}^{n \times d}$ and the expert embeddings $E \in \mathbb{R}^{k \times d}$ is computed as an affinity matrix:

$$A = \text{Softmax}(T \cdot E^T), \quad (7)$$

where n denotes the number of tokens in user prompt, k denotes the number of experts, d denotes the hidden dimension, and the Softmax function is applied along the expert axis. $A[t, i]$ in the affinity matrix captures the correlation between token t and expert i . Then, top- K tokens with the highest relevance for each expert are selected

$$M = \text{Top-K}(A), \quad (8)$$

where $M \in \mathbb{R}^{n \times k}$ is the binary mask matrix such that $M[t, i] = 1$ denotes token t is selected by expert i , and $M[t, i] = 0$ denotes not selected. Experts tend to have similar weights because the metrics they are responsible for are

weighted similarly. This ensures that the tokens selected by these experts possess significant global importance. This practice minimizes interference from tokens that are insignificant across all metrics. The influences of key tokens are further evidenced by their frequent selection through the top- K function and their corresponding scores within the affinity matrix. Moreover, by allocating an equal number of K tokens to each expert, workloads remain balanced, and bias in quality dimensions is avoided, ultimately resulting in more comprehensive and reliable quality prediction. Overall, the token selection gate is summarized as

$$A, M = G(T, E), \quad (9)$$

where G denotes the gate network.

5.1.2. Dual-Gate MoE

A pair of a positive gate G^+ and a negative gate G^- are introduced for different experts to select tokens that have positive and negative impacts on their corresponding quality metrics, while capturing the weights of their influences. The token selection results are denoted as

$$A^o, M^o = G^o(T, E^o), \text{ where } o \in \{+, -\}. \quad (10)$$

The affinity matrix A^o and the binary mask matrix M^o are captured by the gate G^o for the positive or negative aspect using the corresponding expert embedding E^o . The tokens selected by two gates are then evaluated by the corresponding positive or negative experts.

To eliminate the redundancy of having separate experts for evaluating both positive and negative aspects of the same quality metric, the structure of the expert is revised to allow an expert to assess both aspects simultaneously. This revision acknowledges that both positive and negative evaluations are inherently part of the same quality metric. Specifically, the original linear layer $W_i \in \mathbb{R}^{d \times h}$ of expert on metric i is split into two projection matrices $P_i^+, P_i^- \in \mathbb{R}^{d \times l}$ and one score matrix $S_i \in \mathbb{R}^{l \times h}$. Two projection matrices are specific to positive and negative aspects, projecting tokens into the corresponding representations in an l -dimensional low-dimensional space. Since $l \ll h, d$, the

parameter size and the computation cost are reduced from $O(hd)$ to $O(l(h+d))$, allowing an expert to efficiently focus on both positive and negative aspects of a metric with a low cost. The shared score matrix then evaluates the impact of these representations on metric i . Thus, the impact of t -th token $T[t] \in \mathbb{R}^d$ on quality metric i for positive or negative aspect o is described as

$$T_i^o[t] = T[t] \cdot P_i^o \cdot S_i, \text{ where } o \in \{+, -\}. \quad (11)$$

By separating the projection matrices and score matrix, we can evaluate both positive and negative impacts on a metric within the same expert. Then, the overall positive or negative impact of the t -th token on image quality can be obtained by a weighted sum of its impacts for corresponding aspects across all metrics

$$T^o[t] = \sum_{i=0}^k \lambda_{t,i}^o \cdot T_i^o[t], \text{ where } o \in \{+, -\}. \quad (12)$$

Here, $\lambda_{t,i}^o = \frac{\mathbb{1}\{M^o[t,i]=1\}A^o[t,i]}{\sum_{j=0}^k \mathbb{1}\{M^o[t,j]=1\}A^o[t,j]}$ denotes the normalized affinity of the i -th expert for the t -th token. By contrasting the impact of a token from both positive and negative aspects, we can assess the token’s predominant influence on the image quality as

$$\hat{T}[t] = \sigma(T^+[t] - T^-[t]). \quad (13)$$

The introduction of contrasting helps diminish ambiguity in predictions, especially when tokens exhibit influences on multiple metrics from both positive and negative aspects.

5.2. Routing Strategy Design

Given the Pareto relative superiority predicted by routing model, we propose a routing strategy to effectively route user prompts to suitable models to achieve the trade-off between quality and costs. With a preset routing rate, we can determine the proportion of user prompts sent to the cloud, focusing on those where the cloud model significantly outperforms the edge model in quality. Since Pareto relative superiority represents this quality gap, we can efficiently filter prompts by setting a threshold α on the Pareto relative superiority. In particular, the prompts with a Pareto relative superiority above the threshold are better handled by the edge model for cost efficiency, while those below the threshold achieve superior quality when processed in the cloud. Thus, the optimization objective in Eq. (2) can be expressed as:

$$\max_{\alpha \leq 1/2} \mathbb{P}\{PRS(I_e, I_c) < \alpha \mid I_e, I_c \in \mathcal{I}_e, \mathcal{I}_c\} \leq \rho_r. \quad (14)$$

Since Pareto relative superiority being greater than or less than $1/2$ indicates a relative advantage or disadvantage, respectively, we set an upper bound $1/2$ on α to prevent user prompts with images generated at the edge of better quality from being routed to the cloud.

6. Experiment

6.1. Experimental Setup

Dataset. We take COCO2014 [15], a comprehensive resource for object detection, segmentation, and captioning tasks. We select captions from this dataset to serve as the user prompts \mathcal{X} . Given these prompts, we generate images $\mathcal{I}_e, \mathcal{I}_c$ using different open-source text-to-image models.

Models. We take stable diffusion models with varying performances and sizes for cloud and edge usage, include Stable Diffusion 1.5 (SD1.5) [25], Stable Diffusion 2.1 (SD2.1) [25], Stable Diffusion XL (SDXL) [21], Stable Diffusion XL-Refiner (XL-Refiner) [21], and Stable Diffusion 3 (SD3) [8]. We adopt the default hyperparameters in their official documentation for image generation.

Baselines and Settings. We first introduce *random routing* as a baseline, where prompts are randomly assigned to candidate models, given a specified routing rate to the cloud. Additionally, although there does not exist any previous work on text-to-image generation routing, for a comprehensive comparison, we also reproduce several representative routing methods for LLM by adapting them to our scenarios, including RouteLLM [20] with BERT classifier or matrix factorization, Hybrid LLM [7], and ZOOTER [18]. Unless specified in their paper, the routing models are based on Transformers and take the same hyperparameter settings as in our method. All the routing models are implemented in PyTorch 2.3.1 and trained using Adam optimizer with a learning rate of $2e-5$, a weight decay of 0, a batch size of 16, on a NVIDIA 4090D for about 10 epochs.

Metrics. For cost efficiency, we introduce *routing rate*, namely, the proportion of user prompts to the cloud model:

$$p = \mathbb{P}\{R(\mathcal{X}) = 1\}. \quad (15)$$

Smaller routing rate indicates better cost efficiency.

For image generation quality, we first fix a routing rate p and introduce the *winning rate* of the selected models over the cloud model after routing:

$$\begin{aligned} w(\mathcal{I}_{r,p}) &= \mathbb{P}\{Q(I_r) \geq Q(I_c) \mid I_r, I_c \in \mathcal{I}_{r,p}, \mathcal{I}_c\} \quad (16) \\ &= \mathbb{P}\{PRS(I_r, I_c) \geq \frac{1}{2} \mid I_r, I_c \in \mathcal{I}_{r,p}, \mathcal{I}_c\}, \quad (17) \end{aligned}$$

where $\mathcal{I}_{r,p}$ denotes the generated images after routing with the routing rate p . To give a clear comparison of the improvement, we introduce the *winning rate improvement ratio*, which measures the winning rate improvement compared to the random baseline over the improvement achieved by the oracle of the optimal route under ideal conditions:

$$\Delta w(p) = \frac{w(\mathcal{I}_{r,p}) - w(\mathcal{I}_{b,p})}{w(\mathcal{I}_{o,p}) - w(\mathcal{I}_{b,p})}, \quad (18)$$

Router	Multi-Dimensional Metric										$\Delta P(\%)$
	Definition	Detail	Clarity	Sharpness	Harmony	Realism	Color	Consistency	Layout	Integrity	
Edge Model	0.6251	0.6685	0.6076	0.6537	0.5949	0.5575	0.4680	0.5088	0.4860	0.4690	0.00
Cloud Model	0.6337	0.6847	0.6346	0.6703	0.5930	0.5868	0.5134	0.5199	0.5345	0.4972	80.00
Random	0.6294	0.6766	0.6211	0.6620	0.5939	0.5721	0.4907	0.5144	0.5102	0.4831	40.03
RouteLLM-BERT [20]	0.6347	0.6792	0.6305	0.6651	0.5960	0.5788	0.4982	0.5160	0.5167	0.4866	71.51
RouteLLM-MF [20]	0.6364	0.6814	0.6299	0.6660	0.5952	0.5776	0.4970	0.5164	0.5149	0.4850	69.90
Hybird LLM [7]	0.6327	0.6784	0.6306	0.6677	0.5964	0.5787	0.5008	0.5161	0.5191	0.4864	73.49
ZOOTER [18]	0.6350	0.6796	0.6315	0.6672	0.5966	0.5788	0.5004	0.5166	0.5179	0.4854	77.95
RouteT2I (Ours)	0.6350	0.6786	0.6318	0.6679	0.5975	0.5804	0.5010	0.5167	0.5189	0.4865	83.97

Table 2. The multi-dimensional quality of images generated by the routed edge and cloud text-to-image model with RouteT2I at the routing rate 50%. The higher the metrics, the better.

Router	Routing Rate (p)				
	40%	50%	60%	70%	80%
RouteLLM-BERT [20]	24.29	22.45	19.07	17.62	20.59
RouteLLM-MF [20]	25.65	23.29	19.92	16.67	20.59
Hybird LLM [7]	23.77	19.75	14.92	13.62	16.09
ZOOTER [18]	26.77	21.97	17.97	16.97	21.04
RouteT2I (Ours)	30.60	25.81	20.32	18.02	21.94

Table 3. Winning rate improvement ratios Δw (%) at different routing rates.

where $\mathcal{I}_{b,p}$ and $\mathcal{I}_{o,p}$ denote the generated images using the random baseline and oracle, respectively. To further quantify the quality improvement, we introduce *relative performance improvement*, which measures the improvement of the selected model with routing design over the edge model in relation to the improvement achieved by the cloud model:

$$\Delta P = \frac{1}{N} \sum_{i=1}^N \frac{\mu_i(\mathcal{I}_r) - \mu_i(\mathcal{I}_e)}{|\mu_i(\mathcal{I}_c) - \mu_i(\mathcal{I}_e)|}. \quad (19)$$

This metric effectively quantifies routing effectiveness while accounting for the original quality gap.

For cost-quality balance, we introduce the *cost saving ratio* to measure the reduction of the routing rate to the cloud model, compared to the random baseline at a given relative performance improvement ΔP :

$$\gamma(\Delta P) = \frac{p_b(\Delta P) - p_r(\Delta P)}{p_b(\Delta P)}, \quad (20)$$

where $p_b(\Delta P)$ and $p_r(\Delta P)$ represent the routing rates of baseline and router at a given ΔP , respectively.

6.2. Main Results

We present the routing performance of our RouteT2I and baselines using SD3 as the cloud model and SD2.1 as the edge model.

Wining Rates. Tab. 3 shows the wining rate improvement ratios at various routing rates, and our RouteT2I demonstrates a significant improvement across all routing

Router	Relative Performance Improvement (ΔP)				
	40%	50%	60%	70%	80%
RouteLLM-BERT [20]	56.15	51.39	46.92	42.70	40.21
RouteLLM-MF [20]	48.86	49.90	48.20	44.89	41.50
Hybird LLM [7]	62.06	58.85	53.63	49.92	33.38
ZOOTER [18]	69.28	65.76	60.81	57.35	49.64
RouteT2I (Ours)	71.81	70.24	66.61	60.01	53.53

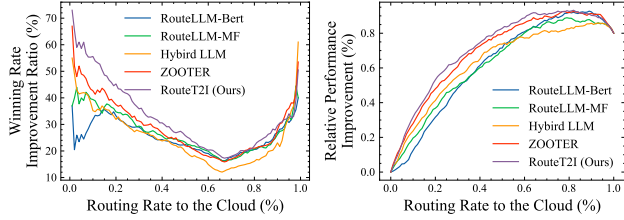
Table 4. Cost saving ratio γ (%) compared to random routing at different relative performance improvements.

rates. Specifically, at a 40% routing rate, our improvement reaches 30.60%, exceeding the baselines by at least 3.83% and demonstrating strong consistency with the oracle.

Multi-Metric Quality. Tab. 2 presents multiple image quality metrics at a routing rate of 50%. We can observe that RouteT2I outperforms all the baselines on 6 out of 10 metrics, because our design with a multi-metric optimization objective incorporates each metric simultaneously. The overall relative performance improvement of RouteT2I is highest, improving at least 6% than existing baselines.

Cost Efficiency. Tab. 4 presents the cost saving ratios at given relative performance improvement targets. We can see that, at a 40% relative performance improvement, RouteT2I reduces the number of cloud serving calls by up to 71.81% compared to random routing, and reduces at least 5.80% compared to other baselines. In other cases, RouteT2I still reduces over 50% compared to random routing, demonstrating high cost efficiency.

Visualization. In Fig. 4, we visualize the wining rate improvement ratio and relative performance improvement. The results indicate that when the routing rate is below approximately 70%, RouteT2I significantly improves quality by effectively assigning the appropriate model for each prompt. However, this advantage diminishes at higher routing rates, because most prompts are processed by the cloud model, limiting the functionality of routing. When the routing rate exceeds 80%, the increase in Fig. 4b suggests that RouteT2I approaches the performance of the oracle, confirming this observation.



(a) Winning rate improvement ratio. (b) Relative performance improvement.

Figure 4. Visualization of winning rate improvement ratio Δw and relative performance improvement ΔP when varying routing rates p to the cloud.

	SD3	SD2.1	XL-Refiner	SDXL	SD1.5
SD3		68.03%	71.58%	75.13%	71.95%
SD2.1	31.96%		52.75%	57.05%	54.12%
XL-Refiner	28.42%	47.24%		54.91%	50.75%
SDXL	24.87%	42.95%	45.09%		46.46%
SD1.5	28.05%	45.88%	49.24%	53.54%	

Figure 5. Winning rate w between commonly used open-source text-to-image models

Cloud Model	Edge Model	Routing Rate (p)				
		40%	50%	60%	70%	80%
SD3	SD2.1	30.36	25.81	20.32	18.02	21.94
SD3	XL-Refiner	26.40	21.70	18.83	16.26	21.40
SD3	SDXL	27.10	23.35	20.50	16.24	19.75
SD3	SD1.5	25.16	20.28	17.64	14.13	17.94
SD2.1	XL-Refiner	21.83	18.68	20.99	24.78	28.56
SD2.1	SDXL	19.85	16.61	17.04	19.81	25.54
SD2.1	SD1.5	11.62	9.88	9.75	11.72	13.69
XL-Refiner	SDXL	10.88	8.98	9.07	9.82	10.51
XL-Refiner	SD1.5	18.52	15.74	18.43	21.30	26.54
SDXL	SD1.5	16.52	15.88	19.57	23.98	30.49

Table 5. Winning rate improvement ratio Δw (%) of our RouteT2I with different edge and cloud text-to-image models.

6.3. Routing Performance on Different T2I Models

To verify the generality of our routing design across different T2I model combinations, Tab. 5 shows the performance of RouteT2I when deploying various T2I models on the cloud and the edge. The quality rankings among these models are depicted in Fig. 5. The results demonstrate that RouteT2I achieves significant improvements, particularly when there is a pronounced quality gap between

Router	Routing Rate (p)				
	40%	50%	60%	70%	80%
w/o Multi-Metric	27.37	22.81	18.87	16.47	19.92
w/o Token Selection	27.82	23.05	18.27	16.77	19.24
w/o Dual-Gate	27.22	22.09	19.27	17.37	21.62
RouteT2I	30.60	25.81	20.32	18.02	21.94

Table 6. Ablation experiment of RouteT2I, showing winning rate improvement ratio Δw (%) on given routing rates.

models, such as with SD3 as the cloud model and others as the edge model. In these cases, RouteT2I consistently achieves over 25% of the oracle’s improvement compared to a random baseline at a 40% routing rate. Notably, even with closely related models like XL-Refiner and SDXL, where XL-Refiner is essentially an SDXL model enhanced with an added refinement stage, RouteT2I still reaches about 10% of the oracle’s improvement. These results highlight RouteT2I’s capability to discern subtle differences in ability between cloud and edge models.

6.4. Ablation Experiment

Tab. 6 presents the results of ablation study on the multi-metric quality optimization objective, the token selection gate, and the dual-gate MoE in RouteT2I. By utilizing multi-dimensional quality metrics and multiple classification heads, RouteT2I can robustly and comprehensively predict image quality. Omitting them leads to a significant performance drop of approximately 2%. Token selection gates have a significant impact at high routing rates, where quality distribution tends to be sparse. By concentrating on key tokens, RouteT2I can effectively pinpoint essential factors, minimizing interference from unrelated tokens. Meanwhile, the dual-gate MoE performs significantly at moderate routing rates, where tokens with critical quality show no significant difference when generated in the cloud or on the edge. The dual-gate can distinguish subtle variations through the contrast between gates.

7. Conclusion

In this work, we have proposed a new text-to-image model routing design RouteT2I between edge and cloud. RouteT2I adopts the routing model with a dual-gate token selection MoE to predict how user prompts affect multiple image quality metrics by identifying and evaluating key tokens with contrastive method. RouteT2I further introduces a routing strategy that exploits the predicted advantage of the edge model over the cloud model to determine which side for image generation. Extensive evaluation has demonstrated that RouteT2I can significantly enhance generation quality at a specified routing rate and meanwhile reduce cost at a given quality target.

References

- [1] GitHub - black-forest-labs/flux: Official inference repo for FLUX.1 models — github.com. <https://github.com/black-forest-labs/flux>. 1
- [2] stabilityai/stable-diffusion-3.5-large · Hugging Face — huggingface.co. <https://huggingface.co/stabilityai/stable-diffusion-3.5-large>. 1
- [3] Jason Baldridge, Jakob Bauer, Mukul Bhutani, Nicole Brich-tova, Andrew Bunner, Kelvin Chan, Yichang Chen, Sander Dieleman, Yuqing Du, Zach Eaton-Rosen, et al. Imagen 3. *arXiv preprint arXiv:2408.07009*, 2024. 1, 3
- [4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023. 1
- [5] Thibault Castells, Hyoung-Kyu Song, Bo-Kyeong Kim, and Shinkook Choi. Ld-pruner: Efficient pruning of latent diffusion models using task-agnostic insights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 821–830, June 2024. 1
- [6] Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023. 2
- [7] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*, 2024. 1, 2, 6, 7
- [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, 6
- [9] 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. *arXiv preprint arXiv:2403.12031*, 2024. 1
- [10] Shijing Hu, Ruijun Deng, Xin Du, Zhihui Lu, Qiang Duan, Yi He, Shih-Chia Huang, and Jie Wu. Laecips: Large vision model assisted adaptive edge-cloud collaboration for iot-based perception system. *arXiv preprint arXiv:2404.10498*, 2024. 2
- [11] Anil Kag and Igor Fedorov. Efficient edge inference by selective query. In *International Conference on Learning Representations*, 2023. 1, 2
- [12] Tony Lee, Michihiro Yasunaga, Chenlin Meng, Yifan Mai, Joon Sung Park, Agrim Gupta, Yunzhi Zhang, Deepak Narayanan, Hannah Teufel, Marco Bellagente, et al. Holistic evaluation of text-to-image models. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [13] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023. 1
- [14] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snap-fusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36, 2024. 1
- [15] 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. 2, 6
- [16] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024. 1
- [17] 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. 1
- [18] Keming Lu, Hongyi Yuan, Runji Lin, Junyang Lin, Zheng Yuan, Chang Zhou, and Jingren Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models. *arXiv preprint arXiv:2311.08692*, 2023. 1, 2, 6, 7
- [19] 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. 1
- [20] 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 with preference data. *arXiv preprint arXiv:2406.18665*, 2024. 1, 2, 6, 7
- [21] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 6
- [22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3
- [23] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 1
- [24] 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*, pages 8821–8831. Pmlr, 2021. 1
- [25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of*

- the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. [1](#), [6](#)
- [26] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. [3](#)
- [27] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2025. [1](#)
- [28] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017. [2](#)
- [29] Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023. [2](#)
- [30] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. [1](#)
- [31] Yiding Wang, Kai Chen, Haisheng Tan, and Kun Guo. Tabi: An efficient multi-level inference system for large language models. In *Proceedings of the Eighteenth European Conference on Computer Systems*, pages 233–248, 2023. [2](#)
- [32] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gungjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. [1](#)
- [33] Murong Yue, Jie Zhao, Min Zhang, Liang Du, and Ziyu Yao. Large language model cascades with mixture of thoughts representations for cost-efficient reasoning. *arXiv preprint arXiv:2310.03094*, 2023. [2](#)
- [34] Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. *arXiv preprint arXiv:2405.17873*, 2024. [1](#)
- [35] Yang Zhao, Yanwu Xu, Zhisheng Xiao, and Tingbo Hou. Mobilediffusion: Subsecond text-to-image generation on mobile devices. *arXiv preprint arXiv:2311.16567*, 2023. [3](#)