DiffGraph: An Automated Agent-driven Model Merging Framework for In-the-Wild Text-to-Image Generation

Supplementary Material

A. More Details about Experiment Settings

A.1. More Implementation Details

In our main experiments, $N_r = 1000$ reference prompts are used to evaluate each expert during the node calibration process. During expert selection, our Expert Selection Agent (ESA) retrieves $K_1 = 3$ CKPT experts based on the summary s of the user prompt p, and retrieves $K_2 = 3$ PEFT experts for each visual attribute a_m parsed from p. The node feature (i.e., the text embedding) dimensionality d_{node} is 384, the edge feature (i.e., the concatenation of five image quality scores) dimensionality d_{edge} is 5, and the dimensionality d_h of the latent vectors encoded by $Enc(\cdot)$ is 128. Moreover, $Enc(\cdot)$ contains two GNNs, GNN_μ and GNN_σ. Both are instantiated as two-layer GCNs with identical architectures, using an input dimensionality of 384, a hidden dimensionality of 64, and an output (i.e., the latent vector) dimensionality of 128. The decoder $Dec(\cdot)$ is instantiated as a two-layer FFN with an input dimensionality of 128, a hidden dimensionality of 128, and an output dimensionality of 2. The batch size B is set to 64. Notably, the number N_a of visual attributes, the number N_{ckpt} of CKPT experts to be merged, and the number N_{peft} of PEFT experts to be merged are all automatically determined by the LLM. Besides, when $N_{\text{ckpt}} = 0$, our framework considers merging only PEFT experts, and loads these PEFT experts onto the original SD15 or FLUX for image generation.

Our framework uses the default generation configurations adopted in the official implementations of the pre-trained diffusion models for image generation. Specifically, the model merged from SD15-based experts uses 50 denoising steps with an output resolution of 512×512. For the model merged from FLUX-based experts, 28 denoising steps are used with an output resolution of 1024×1024. In addition, some experts are trained with specific trigger words (i.e., special keywords that must be included in the prompt to activate the generation capabilities learned by the experts). These trigger words will be appended to the user prompt when the corresponding experts are selected for merging, ensuring that their specialized skills are properly activated during the generation process.

A.2. More Dataset Details

To train our DiffGraph, we use the DABench [19] dataset, which contains 50,482 in-the-wild prompt samples collected from the online platform Civitai. Following the setting in [19], prompt samples in DABench are split for training and testing with a 9:1 ratio. Notably, from the split training samples, $N_r = 1000$ prompts are further selected and separated to construct the reference prompt set used for node calibration (discussed in Sec. 3.1). The remaining training prompts serve as the final training set for optimizing our framework. We provide details of the reference prompt construction process in Sec. A.3. In addition to evaluating our framework on the DABench test set, we also follow [8] to sample 1000 prompts from the widely used DiffusionDB [14] benchmark, a large-scale dataset containing 1.8 million real-user prompts, to form an additional test set.

A.3. More Details about Reference Prompt Construction

To ensure that experts are evaluated on a sufficiently diverse and representative set of reference prompts during node calibration, we adopt the prompt construction pipeline from [17] to select diverse, high-quality prompts from the training prompts based on key linguistic features, such as nouns, prepositions, and adjectives. Specifically, to assess prompt quality, we use WordNet [2] to extract and count valid nouns, adjectives, and prepositions for each prompt, after which prompts are ranked based on these statistics. Prompts with the highest counts are then selected and added to the reference set. The selection process first emphasizes noun-rich prompts to ensure balanced coverage across key object categories, including *Person, Animal, Plant, Artifacts, and Natural Views*. It then selects prompts with high preposition richness, focusing on those containing spatial and relational terms such as `near` and `on`. Finally, prompts are selected based on adjective richness, with a focus on descriptors related to *color* and *shape*.

The above selection procedure iteratively continues, collecting top-ranked prompts in the order of noun, adjective, and preposition richness, until the desired number ($N_r = 1000$ in our main experiments) of reference prompts is obtained. In addition, to further enhance the diversity of the reference prompt set, before adding newly selected prompts to the reference set, we remove prompts that are overly similar to previously selected ones. To this end, following [7, 13], we compute the maximum ROUGE-L score between each newly selected prompt and all existing prompts in the reference set. Prompts with ROUGE-L scores higher than 0.8 are discarded to maintain diversity in the final reference prompt set. We refer readers to [17] and [13] for more details.

A.4. More Details about Expert Collection

We write a script to drive our Graph Construction Agent (GCA) to automatically collect online expert models derived from SD15 or FLUX from two widely used open-source platforms: Civitai and Hugging Face. Our GCA prioritizes high-quality experts that are well-received by users by considering download counts, upvotes, and user comments (if provided). These metrics allow our framework to gather high-quality expert resources that have been validated through community engagement. Following the preliminary collection, a rigorous filtering process is implemented to further ensure expert quality. This includes excluding outdated or non-downloadable experts, removing duplicate experts, and filtering out experts that produce NSFW content. As a result, our framework organizes 2319 SD15-derived experts (including 613 CKPT experts and 1706 PEFT experts) and 1373 FLUX-derived experts (including 309 CKPT experts and 1064 PEFT experts). For each collected expert, our GCA conducts node registration and node calibration to organize them into our universal graph.

A.5. More Details about the Main Experiments

Experts collected to form the fixed expert set. To form the fixed expert set, we collect 13 popular experts derived from SD15 and FLUX, respectively, based on the category tags of the online platform Civitai. We list these experts in Tab. 10.

More Baseline and Variant Details. In our main experiments, in addition to the simple baseline Direct, which directly uses the original SD15 or FLUX models for image generation, we include seven recent model-merging methods as baselines: DARE, Model Swarms, Diffusion Soup, ESA*+K-LoRA, ESA*+LoRA.rar, AutoLoRA, and DiffAgent, as well as a constrained variant (Ours-fixed) of our framework. Below, we provide more details about these methods.

1. **DARE** [18] sparsifies the model parameters of each expert to mitigate expertise interference during model merging. We run this algorithm with different sparsification ratios and use the best merged model obtained.
2. **Diffusion Soup** [1] performs a greedy merging procedure. Starting from the expert with the best performance, it iteratively adds the next-best expert into a uniformly averaged parameter "soup". The added expert is retained only if it improves performance; otherwise, it is discarded. This process continues until all experts are considered.
3. **Model Swarms** [3] treats each model as a point in the parameter space and employs a PSO-based algorithm to explore linear parameter interpolations between experts, iteratively searching for an optimal merged model.
4. **ESA*+K-LoRA**: K-LoRA [9] is a training-free merging method that flexibly merges a fixed number of experts (two in their case) by comparing parameter magnitudes across expert models. To adapt it to leverage online expert resources, we equip it with our modified ESA module. Specifically, we modify the prompt used in the *expert filtering* process to let the LLM automatically select the two suitable experts from the retrieved experts, which are then fed into K-LoRA for merging.
5. **ESA*+LoRA.rar**: LoRA.rar [12] trains a supernetwork that can flexibly merge two different experts (typically one character LoRA and one style LoRA). Similar to ESA*+K-LoRA, we augment LoRA.rar with our modified ESA module to support the use of online expert resources.
6. **AutoLoRA** [8] employs contrastive learning to train a network that maps experts' model parameters into the text space for prompt-expert retrieval. To this end, we map all the experts organized in our universal graph into the text space for expert retrieval and subsequent merging. In addition, AutoLoRA requires prespecifying the number of experts to merge during testing. We set AutoLoRA to merge 3 experts, which we found to yield the best performance.
7. **DiffAgent** [19] collects user preference data (i.e., user prompts and the corresponding experts preferred by users) from Civitai and uses it to fine-tune an LLM for selecting the most suitable expert based on a user (testing) prompt. Notably, since the original DiffAgent paper did not include preference data for experts derived from FLUX, we follow their data-collection pipeline to gather FLUX-based preference data and subsequently fine-tune the LLM, enabling a feasible comparison between DiffAgent and our method on leveraging FLUX-derived experts.
8. **Ours fixed**: In this variant, we remove our ESA module and consider only the nodes corresponding to experts in the fixed expert set, together with the reference-prompt nodes connected to them, to form the subgraph. The VGAE model then takes this subgraph as input to predict the merging coefficients used to combine experts in the fixed set.

A.6. More Details about the Training Algorithm

In DiffGraph, considering the excessively deep computational graph induced by the full denoising process, we adopt a policy-gradient strategy to optimize the VGAE model instead of employing direct backpropagation, as shown in Eq.4 of our main paper. We provide here the detailed formulation of the image-quality evaluation metric $u(\cdot,\cdot)$ in Eq.4 of our main paper, which serves as the reward signal for optimizing the VGAE.

Inspired by [19], we instantiate $u(\cdot,\cdot)$ as a combination of several widely used image-quality metrics, including:

First, **CLIP Score (CS)** [4] measures the correlation between the user prompt p and the correspondingly generated image I using cosine similarity of their respective embeddings obtained through CLIP [10] :

$$u_{\theta_{\text{CS}}}(p, I) = w \cdot \max(\cos(\text{Enc}_{\text{text}}(p), \text{Enc}_{\text{img}}(I)), 0), \quad (1)$$

where θ_{CS} represents the parameters in the CLIP model, and w is set to 2.5 as stated in [4]. *Second*, **ImageReward (IR)** [16], a text-to-image human-preference reward model fine-tuned from BLIP [6], extracts features from p and I, and based on which it predicts a scalar quality score through an MLP:

$$u_{\theta_{\text{IR}}}(p, I) = \text{MLP}\Big(\text{Enc}_{\text{text}}\big(p,\ \text{Enc}_{\text{img}}(I)\big)\Big), \quad (2)$$

where θ_{IR} denotes the parameters of the IR model. *Third*, **Aesthetic Score (AS)** [11] evaluates the aesthetic appeal of I using a regression head built on top of CLIP image embeddings:

$$u_{\theta_{\text{AS}}}(I) = \text{MLP}\big(\text{Enc}_{\text{img}}(I)\big), \quad (3)$$

where θ_{AS} is the parameters of the AS model. *Fourth*, **PickScore (PS)** [5], a CLIP-based human-preference model trained on large-scale datasets, measures the alignment between p and I by computing the similarity between the text and image embeddings:

$$u_{\theta_{\text{PS}}}(p, I) = \text{Enc}_{\text{text}}(p) \cdot \text{Enc}_{\text{img}}(I) \cdot T, \quad (4)$$

where θ_{PS} represents the parameters of the PS model, and T is the learned scalar temperature parameter. *Finally*, Human Preference Score v2.1 (**HPS**) [15], a scoring model fine-tuned from the CLIP-H model, estimates human preference for generated images via the similarity between text and image embeddings:

$$u_{\theta_{\text{HPS}}}(p, I) = \frac{\text{Enc}_{\text{text}}(p) \cdot \text{Enc}_{\text{img}}(I)}{\tau}, \quad (5)$$

where θ_{HPS} denotes the HPS model's parameters, and τ represents the learned temperature scalar.

Following [19], each of the above image quality scores is normalized to the range $[0, 1]$, and the final image quality evaluation metric $u(\cdot, \cdot)$ is obtained as:

$$\begin{aligned} u(p, I) = \frac{1}{5}\big(&u_{\theta_{\text{CS,norm}}}(p, I) + u_{\theta_{\text{IR,norm}}}(p, I) \\ &+ u_{\theta_{\text{AS,norm}}}(I) + u_{\theta_{\text{PS,norm}}}(p, I) \\ &+ u_{\theta_{\text{HPS,norm}}}(p, I)\big), \end{aligned} \quad (6)$$

where $u_{\theta_{\cdot,\text{norm}}}$ denotes the normalized quality score.

A.7. More Details about Expert Evaluation in the Node Calibration Mechanism

During node calibration, each expert generates images for all reference prompts, and we compute image-quality scores for these generated images to form the edge features. Notably, through detailed analysis (discussed in Sec. B.3), we observe that using reduced resolutions (e.g., half of the default resolution of the original pretrained diffusion models) for image generation during node calibration can still achieve good performance. Thus, we use half of the default resolution of the original SD15 and FLUX models to evaluate experts during the preparation stage of our framework.

B. More Framework Analysis

With SD15 as the default pretrained diffusion model, we here conduct additional experiments on the DABench dataset to further analyze our framework.

B.1. Further Analysis of Parameter-based Merging

In our DiffGraph, instead of directly using raw model parameters as input features to derive the merging schemes, the VGAE model in our MP module takes the text-encoded expert node features and the performance–score-based edge features obtained through node registration and calibration mechanisms as input features to predict the merging schemes. Here, we further analyze the impact of this design by evaluating the following variants: **Parameter-based merging** (also discussed in Sec. 4.2), where we follow existing parameter-dependent methods [8, 9, 12] and replace the node features of the selected experts with their model parameters. These model parameter-based node features, together with the edge features, are then fed into the VGAE to produce the merging coefficients. In addition, similar to the aforementioned settings (discussed in Sec. 4.2 in our main paper), we compare four variants (**Parameter-based merging 2023**, **Parameter-based merging 2023→2025**, **Ours 2023**, and **Ours 2023→2025**) to assess the scalability to new experts. As shown in Tab. 1, our method achieves better results than all other variants, showing that our node registration and calibration mechanisms effectively facilitate generating high-quality merging coefficients for image generation. Moreover, the substantial performance degradation observed in the parameter-based variants indicates that using model parameters as input features is not suitable for effectively harnessing large-scale online experts.

Table 1. Evaluation on parameter-based merging.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
Parameter-based merging 2023	6.42	26.03	6.01	19.37	81.12
Parameter-based merging 2023→2025	10.20	25.96	5.98	19.48	81.25
Parameter-based merging 2025	26.62	27.24	6.07	20.03	81.95
Ours 2023	43.96	29.27	6.35	20.42	84.31
Ours 2023→2025	69.64	29.66	6.43	20.57	**84.81**
Ours 2025	**73.11**	**30.06**	**6.54**	**20.62**	84.79

B.2. Impact of the Number of Reference Prompts

In our main experiments, we construct a set of N_r = 1000 reference prompts for node calibration. Here, we explore the impact of varying the number N_r of reference prompts. As shown in Tab. 2, the performance improves noticeably when N_r is smaller than 1000, and the improvement trend plateaus beyond this point. Based on this observation, we choose to set $N_r = 1000$ in our experiments to achieve good results, meanwhile taking computational efficiency into consideration.

Table 2. Evaluation on the number (N_r) of reference prompts.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
$N_r = 300$	57.28	28.86	6.41	20.40	84.62
$N_r = 500$	65.57	29.41	6.46	20.59	84.78
$N_r = 1000$	73.11	30.06	6.54	20.62	84.79
$N_r = 2000$	74.30	29.98	6.57	20.61	84.86

B.3. Impact of the Generation Configuration for Node Calibration

In our main experiments, during node calibration, the performance of each SD15-derived expert is evaluated on images generated at half the resolution (denoted **res 1/2**) of the original diffusion model's default resolution. We also explore the impact of using different resolutions for node calibration. As shown in Tab. 3, compared to the default configuration (i.e., res default) used in the official implementation, performing node calibration with a smaller resolution can still achieve similar performance. Therefore, we choose the "res 1/2" configuration to obtain good results while maintaining efficiency.

Table 3. Evaluation on the generation configuration for node calibration.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
res 1/8	67.32	28.88	6.43	20.52	84.57
res 1/4	72.55	29.93	6.50	20.57	84.45
res 1/2	73.11	30.06	6.54	20.62	84.79
res default	73.49	30.17	6.54	20.58	84.95

B.4. Impact of the Key Steps in Expert Selection

Here, we also explore the impact of the key steps in the expert selection process, namely prompt parsing, expert retrieval, and expert filtering, by comparing the following variants: 1) **w/o parsing**, where we remove the prompt parsing step and directly encode the original user prompt p into a text embedding to retrieve the top-K_3 most relevant experts from a joint pool containing both CKPT and PEFT experts (we find setting $K_3 = 8$ achieves optimal results). 2) **w/o retrieval**, where we remove the expert retrieval step and instead randomly select K_1 CKPT experts and $N_a \times K_2$ PEFT experts to serve as the retrieved experts. 3) **w/o filtering**, where we remove the expert filtering step and directly merge all the retrieved experts, namely $(K_1 + N_a \times K_2)$ experts in total. As shown in Tab. 4, our full framework achieves better performance than all other variants, demonstrating the effectiveness of all three key steps in the expert selection process.

Table 4. Evaluation on the key steps in expert selection.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
w/o parsing	47.01	29.14	6.42	20.31	84.02
w/o retrieval	27.80	28.73	6.22	20.26	83.37
w/o filtering	37.95	28.90	6.34	20.39	84.19
Ours (full)	73.11	30.06	6.54	20.62	84.79

B.5. Impact of the number (K_1 and K_2) of Retrieved Experts

During the expert selection process, we retrieve K_1 CKPT experts and $N_a \times K_2$ PEFT experts (notably N_a is automatically determined by the LLM). Here, we evaluate the impact of setting different values of K_1 and K_2. Notably, when setting K_1 or K_2 to 0, our framework only considers PEFT or CKPT experts for model merging. Moreover, when merging only PEFT experts, they are loaded onto the original pretrained diffusion model for image generation. As shown in Tab. 5, the performance improves noticeably when K_1 and K_2 are below 3, and the improvement trend plateaus beyond this point. Based on this observation, we thus set $K_1 = 3$ and $K_2 = 3$ in our experiments to obtain good results.

Table 5. Evaluation on the number (K_1 and K_2) of retrieved experts.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
(*with* $K_2 = 3$)					
$K_1 = 0$	39.15	28.97	6.33	20.35	84.35
$K_1 = 1$	59.30	29.33	6.38	20.36	84.61
$K_1 = 2$	68.95	29.57	6.40	20.41	84.64
$K_1 = 3$	73.11	30.06	6.54	20.62	84.79
$K_1 = 4$	71.75	30.04	6.46	20.63	84.71
(*with* $K_1 = 3$)					
$K_2 = 0$	54.37	29.31	6.36	20.37	84.02
$K_2 = 1$	61.84	29.55	6.42	20.43	84.21
$K_2 = 2$	68.98	29.94	6.46	20.46	84.56
$K_2 = 3$	73.11	30.06	6.54	20.62	84.79
$K_2 = 4$	70.38	29.64	6.51	20.58	84.53

B.6. Impact of Incorporating Different Hops of Neighboring Nodes to Form the Subgraph

In our main experiments, we activate the selected expert nodes together with their one-hop neighboring nodes to

form the subgraph for deriving the merging coefficients. We also investigate the impact of incorporating different hops of neighboring nodes when forming the subgraph by comparing the following variants: 1) **zero-hop (w/o calibration)**, where we ignore all neighbor nodes and construct the subgraph using only the nodes corresponding to the selected experts. 2) **two-hop (using the entire graph)**, where we incorporate both one-hop and two-hop neighboring nodes of the selected expert nodes to construct the subgraph, which is equal to taking the entire universal graph as the input of VGAE. As shown in Tab. 6, we observe that both variants yield inferior results. We therefore choose to incorporate the one-hop neighboring nodes of the selected experts during subgraph activation to obtain optimal performance.

Table 6. Evaluation on different approaches to form the subgraph.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
zero-hop (w/o calibration)	11.92	25.87	5.94	19.56	81.53
two-hop (using the entire graph)	56.77	29.20	6.42	20.47	84.61
Ours (one-hop)	73.11	30.06	6.54	20.62	84.79

B.7. Impact of the Reference Prompt Construction Mechanism

In our framework, we adopt a comprehensive prompt construction pipeline [17] to select N_r prompts from the raw training prompt set to form the reference prompt set. Here, we further explore the impact of the reference prompt construction mechanism by comparing the following variant: **Random selection**, where we randomly select N_r prompts from the raw training samples to form the reference prompt set. As shown in Tab. 7, our framework achieves better performance than this variant, demonstrating the efficacy of the reference prompt construction mechanism.

Table 7. Evaluation on the reference prompt construction mechanism.

Methods	IR ↑	HPS ↑	AS ↑	PS ↑	CS ↑
Random selection	63.74	29.70	6.43	20.49	84.69
Ours	73.11	30.06	6.54	20.62	84.79

B.8. User Study

Following [9], we conducted a user study to evaluate our results based on human subjective judgment. Five participants were shown images generated by all methods (500 images in total), presented in a random order to avoid potential bias. They were asked to rate each image on a scale of 1 to 5 according to two evaluation criteria: overall image quality and prompt–image alignment. The average scores for each method are reported in Tab. 8. As shown, our method obtained the highest ratings on both metrics.

Table 8. Comparison based on user study.

Method	Image Quality ↑	Prompt-Image Alignment ↑
Direct	1.9	2.2
DARE [18]	3.1	2.8
Model Swarms [3]	3.6	3.3
Diffusion Soup [1]	3.4	3.1
Ours fixed	3.8	3.5
ESA*+K-LoRA [9]	3.8	3.7
ESA*+LoRA.rar [12]	3.9	3.6
AutoLoRA [8]	3.8	3.4
DiffAgent [19]	4.0	3.7
Ours	**4.5**	**4.6**

C. More Qualitative Results

In this section, we provide more qualitative results of our methods (see Fig. 5 and Fig. 6).

D. Prompt Templates

D.1. Prompt for Node Registration

In Fig. 1, we provide the prompt used to generate textual descriptions of the experts' capabilities during the node registration process. We also provide several examples of the generated textual descriptions in Tab. 9.

D.2. Prompts for Prompt Parsing

Next, we present the prompts used to parse the user prompt p: (1) the prompt used to generate the summary s (as shown in Fig. 2), and (2) the prompt used to identify the fine-grained visual attributes $\{a_m\}_{m=1}^{N_a}$ (as shown in Fig. 3).

D.3. Prompts for Expert Filtering

In Fig. 4, we offer the prompt used to filter out relatively irrelevant and redundant experts selected during the expert retrieval process.

References

[1] Benjamin Biggs, Arjun Seshadri, Yang Zou, Achin Jain, Aditya Golatkar, Yusheng Xie, Alessandro Achille, Ashwin Swaminathan, and Stefano Soatto. Diffusion soup: Model merging for text-to-image diffusion models. In *European Conference on Computer Vision*, pages 257–274. Springer, 2024. 2, 5

[2] Christiane Fellbaum. *WordNet: An electronic lexical database*. MIT press, 1998. 1

[3] Shangbin Feng, Zifeng Wang, Yike Wang, Sayna Ebrahimi, Hamid Palangi, Lesly Miculicich, Achin Kulshrestha, Nathalie Rauschmayr, Yejin Choi, Yulia Tsvetkov, et al. Model swarms: Collaborative search to adapt llm experts via swarm intelligence. In *Forty-second International Conference on Machine Learning*. 2, 5

[4] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 confer-*

Prompt for Textual Description Generation

System Prompt
You are a professional capability evaluator tasked with producing a compact textual description of a text-to-image expert model's generative capabilities. Using the expert introduction and two usage examples (if provided), each consisting of a user prompt and its corresponding image generated by the expert, summarize the expert's strengths, visual effects, and the types of prompts it handles well. The output description must be precise, concise, and grounded in observable image characteristics.

Instruction
Please generate a textual description based on the following expert introduction and two prompt-image usage examples.
- Expert introduction: {EXPERT INTRODUCTION}.
- Prompt example 1: {PROMPT EXAMPLE 1}. Image example 1: {IMAGE EXAMPLE 1}
- Prompt example 2: {PROMPT EXAMPLE 2}. Image example 2: {IMAGE EXAMPLE 2}

Example Output: {This expert is suitable for prompts that aim to produce oil painting–style imagery, especially when keywords such as "oil painting," "canvas texture," or "classical style" are present. It enhances the visual output with rich textures and painterly effects, making it ideal for artistic renderings, stylized portraits, or narrative scenes that seek the aesthetic of traditional oil paintings.}

Figure 1. Prompt used to generate textual descriptions of experts' capabilities.

Prompt for Summary Generation

System Prompt
You are a professional text summarizer. Given a user prompt describing image-generation requirements, your task is to produce a concise textual summary that clearly captures the core intent of the user's generation needs.

Instruction
First, filter out irrelevant or redundant information from the user prompt, such as repeated or semantically redundant nouns, adjectives, or modifiers, unintelligible or meaningless terms or strings, and descriptions that do not contribute to the actual image-generation intent.
Then, identify the key visual-generation needs, such as the primary subject or character, the main artistic style, and the major scene or environment when applicable. Please ensure that these extracted elements are generalized into broader conceptual domains when appropriate, so that the summary reflects the essential user intent rather than overly specific details. Based on this, generate a concise summary (no more than `15` words), using a clear "*object or character* + *in* + *style* + *in scene*" structure whenever feasible.

Below is an Example:
Given the user prompt: {short hair, brown hair, girl, waist up, portrait framing, cinematic lighting, soft ambient lighting, cinematic shadows, glowing skin, glossy skin, perfect jawline, beautiful lips, detailed eyes, kawaii, elegant, goddess aura, alluring gaze}.
Your summary should be: {An Asian girl portrait in cinematic style}.

Figure 2. Prompt used to generate the summary of the user prompt

ence on empirical methods in natural language processing, pages 7514–7528, 2021. 3

[5] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 36652–36663, 2023. 3

[6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 3

[7] Jialu Li, Jaemin Cho, Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. Selma: Learning and merging skill-specific text-to-

Table 9. Examples of generated textual descriptions of experts' capabilities.

Expert Name	Description
High-Polygon 3D Model LoRA	This expert is trained on high-polygon 3D model imagery and excels at generating clean, high-resolution textures for skin, hair, and clothing. It enhances visual outputs with crisp surface details and realistic material rendering, making it ideal for character design, digital modeling, and scenes that require a polished 3D aesthetic.
Beautiful Realistic Asians	This expert is mainly suitable for generating realistic, cinematic images of Asian women, particularly excelling in fashion and aesthetic settings. It is effective at producing high-quality, lifelike portraits with a cinematic flair, often highlighted by bokeh effects and vivid cityscapes.
Cetus-Mix	This expert excels at generating flat anime styles with a distinctive oil painting texture in the background, making it ideal for creating visually appealing 2D and 2.5D anime-style artworks. It is particularly strong in producing detailed scenes with complex lighting and shading effects.

Table 10. The list of the fixed expert set.

SD15	FLUX
XXMix_9realistic	Edge of Reality
hellomecha	UltraReal Fine-Tune
Urban Samurai — v0.14 — Clothing LoRA	Add Micro Details - Concept
Realistic Vision V6.0 B1	The Space Marines Warhammer 40K
multiple view	Hand Detail FLUX
GHIBLI_Background	Sci-fi Environments
Ink scenery	FLUX Image Upgrader / Detail Maximizer
Add More Details - Detail Enhancer / Tweaker	SpaceCraft
JR East E235 series / train interior	Feet XL + SD 1.5 + FLUX.1-dev + Pony + Illustrious
Hands SD 1.5	Wakfu Environments
dvArch - Multi-Prompt Architecture Tuned Model	Guns (Lots Of Guns) XL + F1D (Choose from list)
Product Design (minimalism-eddiemauro)	Velvet's Epic Dragons — Flux
WildLifeX - Animals	POV face-touching

image experts with auto-generated data. In *Advances in Neural Information Processing Systems*, pages 38530–38558. Curran Associates, Inc., 2024. 1

[8] Zhiwen Li, Zhongjie Duan, Die Chen, Cen Chen, Daoyuan Chen, Yaliang Li, and Yingda Chen. Autolora: Automatic lora retrieval and fine-grained gated fusion for text-to-image generation. *arXiv preprint arXiv:2508.02107*, 2025. 1, 2, 3, 5

[9] Ziheng Ouyang, Zhen Li, and Qibin Hou. K-lora: Unlocking training-free fusion of any subject and style loras. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13041–13050, 2025. 2, 3, 5

[10] 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

[11] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022. 3

[12] Donald Shenaj, Ondrej Bohdal, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. Lora. rar: Learning to merge loras via hypernetworks for subject-style conditioned image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16132–16142, 2025. 2, 3, 5

[13] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023. 1

[14] Zijie J Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. Diffu-

siondb: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 893–911, 2023. 1

[15] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 3

[16] 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. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023. 3

[17] Zilyu Ye, Zhiyang Chen, Tiancheng Li, Zemin Huang, Weijian Luo, and Guo-Jun Qi. Schedule on the fly: Diffusion time prediction for faster and better image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23412–23422, 2025. 1, 5

[18] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024. 2, 5

[19] Lirui Zhao, Yue Yang, Kaipeng Zhang, Wenqi Shao, Yuxin Zhang, Yu Qiao, Ping Luo, and Rongrong Ji. Diffagent: Fast and accurate text-to-image api selection with large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6390–6399, 2024. 1, 2, 3, 5

Prompt for Visual Attribute Identification

System Prompt

You are an expert in analyzing user requirements for image generation. Given a user-provided text prompt, your task is to identify the image-generation needs and produce a structured semantic analysis. Think step by step, following the instructions below, and output the final result in strictly valid JSON format.

Instruction

Step 1: User Prompt Filtering and Refinement.

Filter out irrelevant or redundant information from the user prompt, including:

- Repeated or semantically redundant nouns, adjectives, or modifiers,
- Unintelligible or meaningless terms or strings,
- Descriptions that do not affect the image generation intent.

Reorganize the remaining effective content into a concise version composed of short, meaningful phrases.

Step 2: Semantic Component Identification

Decompose the refined prompt into a set of semantic components (each semantic component being a syntactic or semantic unit such as a noun phrase or an adjective phrase) that may represent the user's image-generation needs. These components may include, but are not limited to:

- Character or subject,
- Artistic style,
- Character names,
- Man-made objects such as clothing,
- Scene or environment,
- Viewpoint or composition,
- Resolution or quality

Construct a *semantic component set* from these elements.

Step 3: Visual Attribute Reasoning.

For each semantic component, infer a set of visual attributes that the image generation model may need to express when depicting this semantic component. For example:

- For "character" components, the model may need to focus on visual attributes such as eyes, hair, skin texture, and other facial features.
- For components that include terms like "photography", the model may need to consider attributes such as cinematic lighting, realistic rendering, and overall visual fidelity.
- For "style" components, the model may need to capture visual attributes such as color palette, brushstroke patterns, and medium effects (e.g., ink splashing, watercolor diffusion).

In addition, please note that:

- Each semantic component itself must also be included as one of its own visual attributes.
- If a semantic component has no additional visual attributes, its attribute set should contain only the component itself.

Step 4: Output the final result strictly in the following JSON structure.

```
{"<component_1>": [
      "<visual_attribute_1>",
      "<visual_attribute_2>"],
 "<component_2>": [
      "<visual_attribute_3>",
      "<visual_attribute_4>",
      "<visual_attribute_5>"],
 ...}
```

Think step by step and carefully follow each guideline.

Figure 3. Prompt used to identify visual attributes.

Prompt for Expert Filtering

System Prompt

You are a specialist in selecting expert models to achieve user-desired image generation. Given (i) a user prompt, (ii) a concise summary of this user prompt, (iii) a set of visual attributes (each representing fine-grained visual requirements to be considered during image generation), and (iv) textual descriptions of candidate experts' capabilities (including both powerful CKPT experts and lightweight PEFT experts), your task is to filter out irrelevant or redundant candidates and determine the final subset of selected experts that can collaboratively accomplish the user's intended image generation. The total number of finally selected experts is typically no greater than eight; please retain the most appropriate expert candidates. Think step by step, following the instructions below.

Instruction

Step 1: CKPT Expert Filtering

You will be given an input in JSON format, for example:

```
{
  "<user_prompt_summary>": {
    "<CKPT_name_1>" : "<CKPT_description_1>",
    "<CKPT_name_2>" : "<CKPT_description_2>",
    ...}
}
```

Carefully analyze the <user_prompt_summary> and each CKPT expert's description. Your goal is to evaluate the relevance of each CKPT expert to the summary and filter out experts that are irrelevant or whose capabilities are redundant with others. The remaining CKPT experts will determine the overall style and primary visual characteristics of the generation.

Step 2: PEFT Expert Filtering

You will be given an input in JSON format, for example:

```
{"<user_prompt>": {
    "<visual_attribute_1>": {
      "<PEFT_name_1>" : "<PEFT_description_1>",
      "<PEFT_name_2>" : "<PEFT_description_2>"},
    "<visual_attribute_2>": {
      "<PEFT_name_3>" : "<PEFT_description_3>",
      "<PEFT_name_4>" : "<PEFT_description_4>"
    },...}
}
```

For each visual attribute, analyze the corresponding PEFT expert candidates and evaluate how well each expert supports that attribute based on its description. Your goal is to filter out PEFT experts that are irrelevant or whose capabilities are redundant with others. When filtering PEFT experts, please consider:

- Whether the PEFT experts align with the visual attributes and the overall creative intent specified by the user prompt.
- Whether the capabilities of the PEFT experts are compatible with the selected CKPT experts.

Step 3: Output the final result strictly in the following format.

```
{"final_selected_CKPT_experts": [...],
 "final_selected_PEFT_experts": [...]}
```

Think step by step and carefully follow each guideline.

Figure 4. Prompt used to filter the retrieved experts.

Figure 5. Qualitative comparisons of different methods on T2I generation with experts derived from SD15. Different attributes in the prompt text are labeled with different colors. We illustrate visual attributes in a paint box manner, where a full colored cell denotes an attribute is successfully reflected in the generated image, a half colored cell denotes the attribute is reflected but at low quality, and an empty (white) cell means the corresponding attribute is totally missing. Zoom in for a better view.

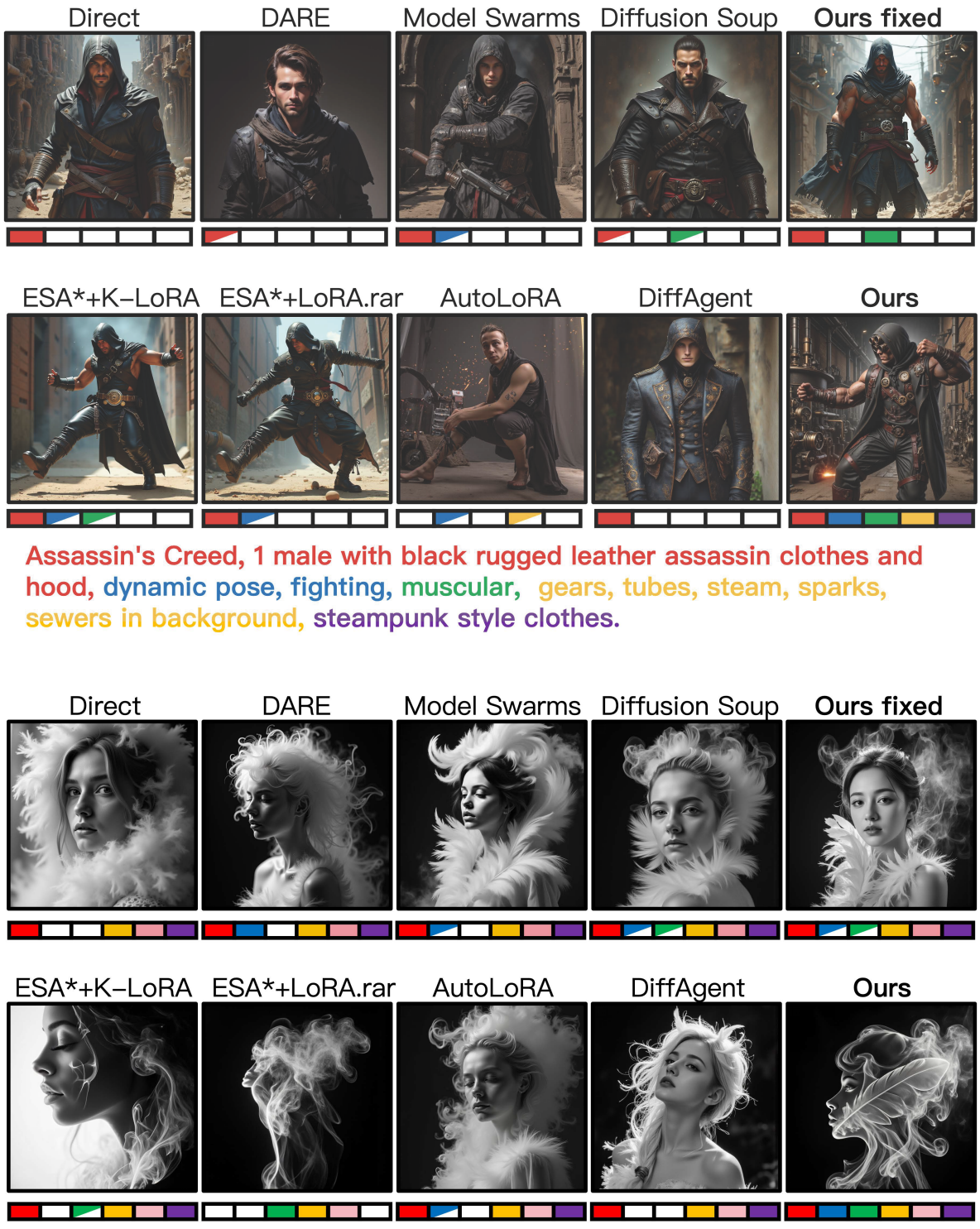

Figure 6. Qualitative comparisons of different methods on T2I generation with experts derived from FLUX. Different attributes in the prompt text are labeled with different colors. We illustrate visual attributes in a paint box manner, where a full colored cell denotes an attribute is successfully reflected in the generated image, a half colored cell denotes the attribute is reflected but at low quality, and an empty (white) cell means the corresponding attribute is totally missing. Zoom in for a better view.