ReNeg: Learning Negative Embedding with Reward Guidance

Xiaomin Li[1,2,*], Yixuan Liu[1,*,‡], Takashi Isobe[1], Xu Jia[2,†], Qinpeng Cui[3], Dong Zhou[1],
Dong Li[1], You He[3], Huchuan Lu[2], Zhongdao Wang[3,†], Emad Barsoum[1]
[1]Advanced Micro Devices Inc., [2]Dalian University of Technology, [3]Tsinghua University
xmli22@mail.dlut.edu.cn, {yixuan.liu,takashi.isobe,dong.zhou,d.li,ebarsoum}@amd.com,
{xjia,lhchuan}@dlut.edu.cn, heyou_f@126.com, {cqp22,wcd17}@tsinghua.edu.cn

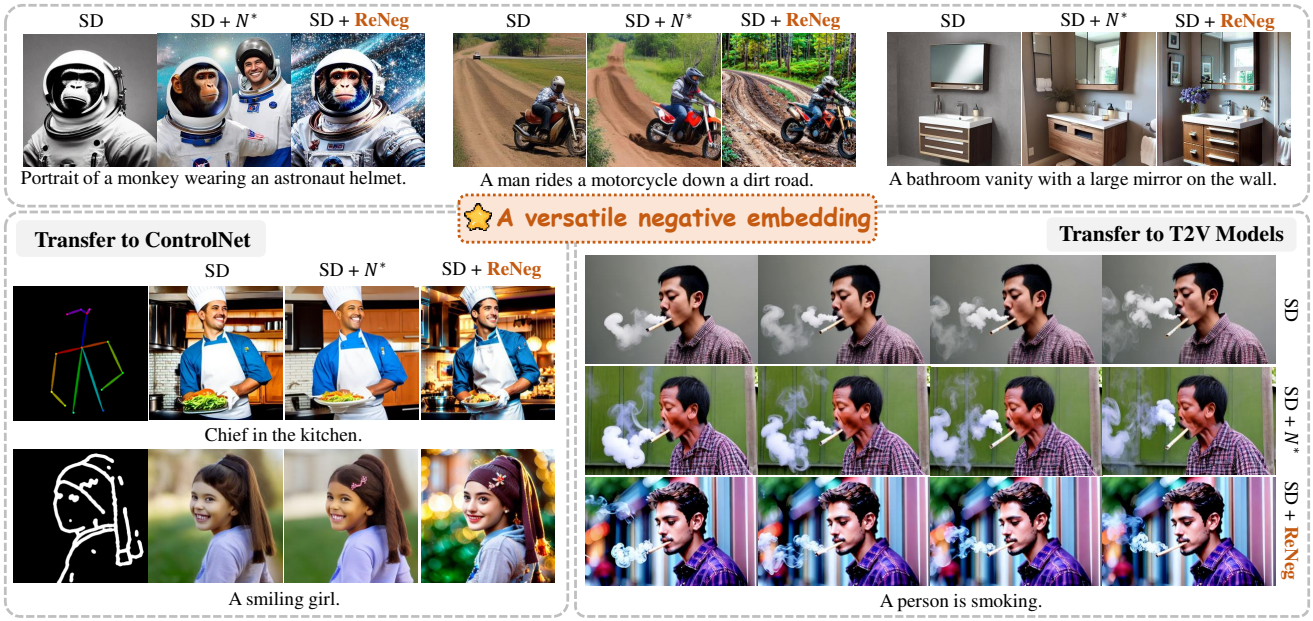

Figure 1. We develop ReNeg, a versatile negative embedding seamlessly adaptable to text-to-image and even text-to-video models. Strikingly simple yet highly effective, ReNeg amplifies the visual appeal of outputs from base Stable Diffusion (SD) models. '$+N^*$' and '+ReNeg' indicate improved results with handcrafted negative prompts and our negative embedding, respectively.

Abstract

In text-to-image (T2I) generation applications, negative embeddings have proven to be a simple yet effective approach for enhancing generation quality. Typically, these negative embeddings are derived from user-defined negative prompts, which, while being functional, are not necessarily optimal. In this paper, we introduce ***ReNeg****, an end-to-end method designed to* learn *improved* ***Neg****ative embeddings guided by a* ***Re****ward model. We employ a reward feedback learning framework and integrate classifier-free guidance (CFG) into the training process, which was previously utilized only during inference, thus enabling the effective learning of negative embeddings. We also propose two strategies for learning both global and per-sample negative embeddings. Extensive experiments show that the learned negative embedding significantly outperforms null-text and handcrafted counterparts, achieving substantial improvements in human preference alignment. Additionally, the negative embedding learned within the same text embedding space exhibits strong generalization capabilities. For example, using the same CLIP text encoder, the negative embedding learned on SD1.5 can be seamlessly transferred to text-to-image or even text-to-video models such as ControlNet, ZeroScope, and VideoCrafter2, resulting in consistent performance improvements across the board. Code is available at https://github.com/AMD-AIG-AIMA/ReNeg.*

* Equal contribution. Work done during an internship at AMD.
† Corresponding authors.
‡ Project lead.

1. Introduction

Recent advancements in diffusion models [15, 29, 30] have led to significant breakthroughs in image generation [10, 27, 32, 38] and video generation [3, 6, 20, 36, 41]. A pivotal technique in these models is Classifier-free Guidance (CFG) [14], which enhances text control capabilities while also improving the realism and aesthetic quality of the generated outputs. CFG operates by concurrently training a conditional probability model alongside an unconditional probability model, merging their score predictions during inference. The unconditional model takes a null-text prompt as input, serving as guidance for the score predictions. In practice, this null-text prompt can be substituted with a negative prompt [1, 9, 33, 35], prompting the model to generate outputs that deviate from the characteristics specified by the negative prompt. Empirical evidence suggests that negative prompts generally yield superior results compared to null-text prompts, leading to their prevalent use in text-to-image [9, 33] and text-to-video generation [12].

To create effective negative prompts, the prevailing method involves manually selecting negative terms—such as "low resolution" and "distorted"—and employing trial and error to identify optimal combinations. However, this approach suffers from significant limitations. The manually defined search space is inherently incomplete, as it cannot encompass all permutations of negative vocabulary. Additionally, the quality assessment of generated outputs often relies on subjective human judgment, resulting in inefficiencies during the search process. Consequently, manually crafted negative prompts, while yielding decent performance, are suboptimal.

In this work, we introduce ReNeg, a **Re**ward-guided approach that directly *learns* **Neg**ative embeddings through gradient descent. Our method enhances the process along two dimensions compared to manual searches. First, learning occurs within a comprehensive search space. We utilize the continuous text embedding space—specifically, the output embedding space of text encoders—since the original language space is discrete [33], which makes gradient-based optimization more challenging. Second, we fully automate the evaluation criterion using an image reward model (RM) [39], enabling continuous gradient descent rather than relying on trial and error. We formulate the learning objective within a reward feedback framework, leveraging a pretrained image reward model to guide the gradient descent. Negative embeddings are treated as a set of model parameters, and to ensure they receive gradients from the reward guidance, we modify the training process to incorporate CFG, which was previously utilized only during inference. Furthermore, we propose a tuning approach that learns per-sample negative embeddings, adapting to various prompts and yielding additional improvements.

Extensive experiments demonstrate that both the learned global and per-sample negative embeddings consistently outperform null-text and carefully crafted negative embeddings in terms of generation quality and human preference. In human preference benchmarks such as HPSv2 [37] and Parti-Prompts [40], ReNeg, while significantly simpler, achieves remarkable results, even rivaling methods that require full model fine-tuning. Moreover, we highlight that the global negative embeddings learned using ReNeg exhibit *strong generalization capabilities* and are *easily transferable*. Any text-conditioned generative model utilizing the same text encoder can share these negative embeddings. We demonstrate that negative embeddings learned with Stable Diffusion 1.5 transfer seamlessly to text-to-image and text-to-video models, including ControlNet [42], ZeroScope [5], and VideoCrafter2 [6].

Our primary contributions are summarized as follows:

- We propose ReNeg, an innovative approach for learning negative embeddings guided by reward models.
- We introduce two strategies for learning both global and per-sample negative embeddings, each yielding significant improvements, with per-sample embeddings demonstrating superior performance.
- We establish that the learned global negative embeddings can be readily transferred to other models and tasks, resulting in consistent performance improvements.

2. Related Works

Text-to-Image Diffusion Models. Existing T2I models [26, 27] have demonstrated impressive generative capabilities, but generating satisfactory content from user-provided positive text prompts remains challenging, particularly for common issues such as hand and face generation defects. A straightforward solution is to improve the user prompt through prompt engineering [11, 34]. Recent works [4, 13] have optimized pretrained large language models (LLMs) [19, 24], enabling the modified models to generate refined prompts based on the original user input. While the image quality generated using these refined prompts shows some improvement over the original, exploration of negative prompts remains limited. DNP [9], for instance, samples a negative image based on the positive prompt and uses a captioning model to generate a corresponding negative prompt. In practice, this process often requires multiple attempts to find an appropriate negative prompt. On the other hand, DPO-Diff [33] searches for negative prompts within the discrete language space. In contrast, we attempt to directly learn negative embeddings through gradient descent in the continuous text embedding space, which allows for a more efficient search of optimal negative embedding.

Reward Optimization for Text-to-Image Models. Numerous efforts [8, 21, 39] have been made to optimize existing T2I models by leveraging reward models [18, 28,

37, 39]. Inspired by Reinforcement Learning from Human Feedback (RLHF) [7, 23] in LLMs, DDPO [2] employs reinforcement learning (RL) to finetune diffusion models, aiming to maximize the reward score within a relatively constrained vocabulary. Alternatively, Diffusion-DPO [31] adopts the Direct Preference Optimization (DPO) [25] strategy that aligns diffusion models with human preferences without the need for RL, significantly improving the visual appeal of generated content. Moreover, methods like ReFL [39] and TextCraftor [21] calculate the reward score directly from the predicted initial image to guide T2I model finetuning, typically targeting components such as the text encoder [21] or UNet [2, 31, 39] in the diffusion model. Distinct from these approaches, our method seeks to optimize the negative embedding under reward guidance, enabling comparable or even superior generation quality at minimal storage cost.

3. Preliminary

Diffusion Models. Diffusion models consist of two processes: a forward noising process and a backward denoising process. In the forward process, Gaussian noise is gradually added to the data $x_0 \sim p(x_0)$ through a fixed-length Markov chain. As the time steps increase, a series of noisy latent variables $\{x_1, x_2, ..., x_T\}$ with increasing noise levels are progressively generated:

$$q(x_t|x0) = \mathcal{N}(x_t; \sqrt{1-\bar{\beta}_t}x_0, \bar{\beta}_t \boldsymbol{I}), \tag{1}$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. ϵ is a standard Gaussian noise, and α_t decreases with the timestep t. In the backward process, diffusion models restore the original data distribution by progressively denoising variables sampled from a Gaussian distribution $x_T \sim \mathcal{N}(0, \boldsymbol{I})$.

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{3}$$

μ_θ and σ_θ are predicted statistics. Observed by [15], only predicting the noise through a neural network $\epsilon_\theta(x_t, t)$ works well. For diffusion-based text-to-image tasks, the textual prompt c is introduced as the condition. The training objective can be represented by a reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{\mathbf{x_0}, \mathbf{c}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}), t} \left[\|\boldsymbol{\epsilon}_\theta(x_t, E(c), t) - \boldsymbol{\epsilon}\|_2^2 \right]. \tag{4}$$

Here, E is a pre-trained text encoder.

Classifier-free Guidance. During training diffusion models, [14] proposes to jointly train a conditional diffusion model $\boldsymbol{\epsilon}_\theta(x_t, c, t)$ and an unconditional one $\boldsymbol{\epsilon}_\theta(x_t, \phi, t)$. During inference, it allows control over the balance between realism and diversity of generated samples by adjusting the guidance scale γ for the conditional generation task:

$$\tilde{\epsilon}_\theta(x_t, c, t) = \boldsymbol{\epsilon}_\theta(x_t, \phi, t) + \gamma(\boldsymbol{\epsilon}_\theta(x_t, c, t) - \boldsymbol{\epsilon}_\theta(x_t, \phi, t)) \tag{5}$$

The unconditional model is realized by inputting a null-text embedding ϕ as condition. Numerous works [9, 33] indicate that replacing ϕ with a handcrafted negative embedding n further improves generation quality. In this work, we aim to learn the negative embedding. The main challenge here is that CFG is usually performed during inference only. To learn the negative embedding, we have to incorporate CFG into training and make the gradient w.r.t. n tractable.

Table 1. Parameter efficiency comparison between the negative embedding n, the full model parameters θ_0, and the added LoRA parameters with rank $r = 8$ (θ_l^8) and $r = 16$ (θ_l^{16}).

$E(n)$	$E(\theta_0)$	$E(\theta_l^8)$	$E(\theta_l^{16})$
5.1×10^{-4}	1.5×10^{-6}	8.1×10^{-8}	1.9×10^{-9}

4. Method

In this section, we begin by discussing our motivation and the feasibility of learning a negative embedding in Section 4.1, illustrating why it could possibly work. Next, we introduce how to learn a universal negative embedding using ReNeg in Section 4.2, and present a per-sample variant in Section 4.3 to further enhance image generation quality.

4.1. On feasibility of learning negative embeddings

Our key insight is that the negative embedding can be seen as part of the model parameters. Based on a pretrained diffusion model, we make an important observation that *tuning the negative embedding is much more efficient than tuning the model parameters*. To illustrate this, we conduct a pilot study as follows: we randomly select a set of N prompts, feed them into a pretrained SD1.5 model, and obtain a set of denoised latents $X \in \mathbb{R}^{N \times D}$, where $D = h \times w \times c$ represents the dimension of the latents. When N is large enough, X can be seen as a nice sampling of the learned distribution. We care about how the distribution changes when small perturbations are applied to the model parameters θ (or similarly to the negative embedding n). For a desired "efficient" set of parameters, we expect the distribution to vary fast with small perturbations, therefore it would bring a greater chance to reach better solutions when tuning this set of parameters. To quantify parameter efficiency, we define a criterion based on the Jacobian matrix. Let $\mathcal{J} = \frac{\partial X}{\partial \theta} \in \mathbb{R}^{d_\theta \times ND}$ be the Jacobian matrix, computed via iterative backpropagation (See Appendix for details), where d_θ is the dimension of parameter θ. We define the parameter efficiency $E(\theta) = \frac{1}{ND d_\theta} ||\mathcal{J}||_F$, where $|| \cdot ||_F$ denotes the

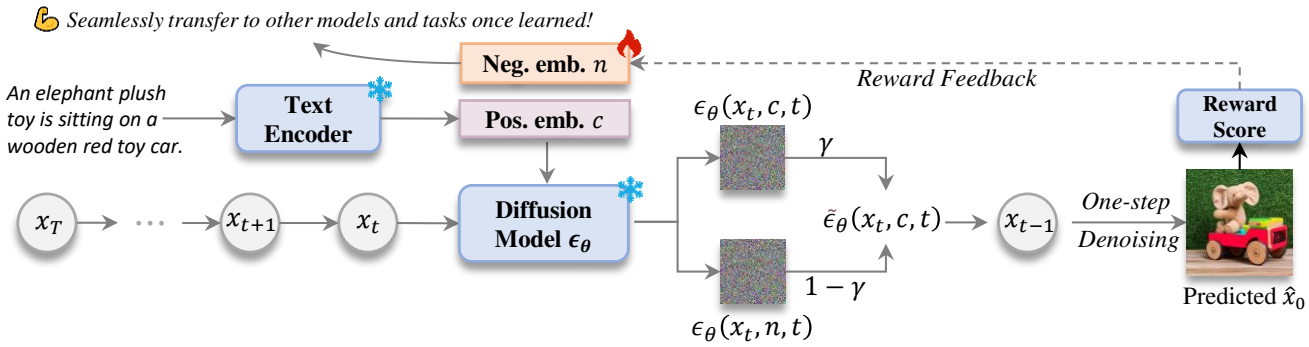

Figure 2. Overview of the training pipeline of our ReNeg. We learn the negative embedding by integrating Classifier-Free Guidance into the training process. The negative embedding is directly optimized using reward feedback, with gradients backpropagated during the final one-step generation process. Once optimized, it can be seamlessly transferred to any T2I or T2V models.

Frobenius Norm of a given matrix. A larger $E(\theta)$ implies a higher rate of distributional change, indicating that tuning the parameters θ is more efficient.

We compare the parameter efficiency between the full model parameters θ_0, the added LoRA [16] parameters with rank $r = 8$ (θ_l^8) and $r = 16$ (θ_l^{16}), and the negative embedding n. We observe in Table 1 that $E(n) \gg E(\theta_0) > E(\theta_l^8) > E(\theta_l^{16})$, indicating that tuning the negative embedding is the most efficient. The lower efficiency of θ_0, θ_l^8 and θ_l^{16} can be attributed to the fact that they start from a well-pretrained checkpoint where local minimum has already been reached, leading to a rather low rate of distributional change. In contrast, the negative embedding is usually set empirically and is thus far from optimal. The high parameter efficiency shows tuning the negative embedding is a promising alternative to tuning the entire model, highlighting the feasibility of *learning* negative embeddings.

4.2. Learning Negative Embeddings with ReNeg

Overview. The key idea behind ReNeg is to learn the negative embedding using a reward feedback learning (ReFL) framework [39]. Starting with a pretrained diffusion model, *e.g.*, Stable Diffusion 1.5 [27], we sample random prompts and feed them into the model to generate predicted images. An image reward model evaluates the reward score according to the generation quality, and then backpropagate gradients to the diffusion model. A graphical overview of this process is provided in Fig. 2.

Reward model. In language models, reinforcement learning from human feedback (RLHF) [7, 23] is a widespread technique that is proven to be effective in human preference alignment. Typically, the reward for a given prediction is evaluated by a pretrained reward model (RM) $\mathcal{R}$. In the domain of visual data generation, we adopt a similar approach, leveraging an RM to guide diffusion models. Specifically, using the human preference-based reward model HPSv2.1 [37], we optimize the negative embedding to maximize the reward score. Formally, an image RM outputs a scalar reward score $r = \mathcal{R}(c, x)$ for a given pair of prompt c and image x.

Learning objective. Unlike in language models, the iterative denoising nature of diffusion models makes it challenging to estimate the likelihood of the generated samples using reward models. Specifically, the reward model is typically trained on natural images and struggles to estimate the rewards of intermediate images that are not fully denoised. A feasible solution is to perform a one-step prediction from intermediate denoised latents and then supervise the resulting prediction. Concretely, suppose a canonical denoising iteration is $x_T \to \ldots \to x_t \to \ldots \to x_0$. At an intermediate timestep t, we directly predict $\hat{x}_0$ from x_t, *i.e.*, the iteration can be depicted as $x_T \to \ldots \to x_t \to \hat{x}_0$. The one-step prediction is given by

$$\hat{x}_0 = \frac{x_t - \left(\sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_\theta(x_t, c, t)\right)}{\sqrt{\bar{\alpha}_t}}. \tag{6}$$

The final learning objective is to maximize the expectation of reward scores over the prompt distribution $\mathcal{D}$:

$$\mathcal{J}_\theta(\mathcal{D}) = \mathbb{E}_{c\sim\mathcal{D}}(\mathcal{R}(c, \hat{x}_0)). \tag{7}$$

For implementation simplicity, the gradient is backpropagated through $\hat{x}_0$ to x_t, but stops to go through further to $x_{t+1}, \ldots, x_T$. [39] reveal that the selection of the intermediate timestep t is non-trivial. When $T - t$ is small, rewards for all generations remain indistinguishably low. When $T - t$ is sufficiently large, rewards for generations of different quality become distinguishable. To ensure efficient training, we set $T = 30$ and randomly sample $t \in [0, 10]$.

Training with CFG. In order to learn negative embedding, we incorporate CFG into the aforementioned training framework. We register the negative embedding as a set of the model parameters and initialize it with the pre-extracted null-text embedding. During training, all other parameters

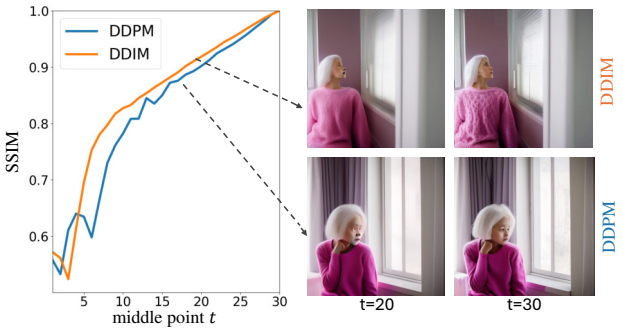

Figure 3. Deterministic ODE sampler (DDIM) improves $\hat{x}_0$ prediction. We plot the similarity score between $\hat{x}_0$ and x_0 against varying selection of middle point t. DDIM consistently outperforms DDPM. **Prompts:** *A white-haired girl in a pink sweater looks out a window in her bedroom.*

Figure 4. Comparison of results using global negative embedding and per-sample negative embedding. Red boxes highlight the improvement in image details achieved by the per-sample negative embedding. **Prompts: (Left)** *A girl in a school uniform making a scissor hand gesture.* **(Right)** *An oil painting of a muscular cat wielding a weapon with dramatic clouds in the background.*

are frozen, and only the negative embedding is updated using gradient $\frac{\partial \mathcal{J}_\theta(\mathcal{D})}{\partial n}$. We also experimented with making the guidance scale factor γ learnable, but found similar results to setting it as a constant. Therefore, we opt to keep it as a constant. Furthermore, effective alignment between training and inference processes can be achieved through this CFG training strategy, where the predicted noise at each step is reparameterized using Eq. 5 during both processes. The sample x_{t-1} is then predicted from the latents of the previous timestep and the reparameterized noise.

Deterministic ODE solver improves $\hat{x}_0$ prediction. Due to the nature of reward guidance being applied to the one-step prediction $\hat{x}_0$, we aim for the prediction of $\hat{x}_0$ to be as accurate as possible. This accuracy allows for a broader sampling range at time t, leading to a more precise learned marginal distribution. We found that, in this context, selecting the deterministic ODE solver of DDIM for sampling from T to t yields better results than using the original SDE solver of DDPM, resulting in more stable and accurate predictions of $\hat{x}_0$. In Figure 3, we visualize the differences between $\hat{x}_0$ predicted from x_t and x_0 obtained through complete sampling. As t varies within the interval $[0, T]$, it is evident that the discrepancies between $\hat{x}_0$ and x_0 with DDIM are generally smaller than those with DDPM.

Transferability. Although our training relies on the generative model and the reward model, the learned negative embedding is independent of these models, as they are merely vectors in the output space of the text encoder. This means that as long as the same text encoder is used, the learned negative embedding can be seamlessly transferred across different generative models. We validate the effectiveness of this transfer in the experimental section.

4.3. Per-sample Negative Embedding

While it is feasible to learn a globally applicable and effective negative embedding, the optimal negative embedding may vary depending on the prompt. For instance, when the prompt requests realistic images resembling photography, "cartoonish" may serve as a negative prompt. Conversely, when the prompt asks for a cartoon image, "realism" may instead function as the negative prompt. Therefore, we propose an *optional* procedure that allows further adaptation of the learned global negative embedding to generate adaptive negative embedding tailored to a specific prompt.

Given a specific prompt, our method for learning the per-sample negative embedding is similar to that described in Eq. (7), with several key differences. First, we no longer need to take the expectation over the prompt distribution, instead training on individual samples. Second, the optimization of per-sample negative embeddings is initialized with the learned global negative embedding. Finally, we propose a search strategy that guarantees convergence to a solution that outperforms the global negative embedding. Specifically, we define a maximum training step N and a patience value P, training stops when all N steps finish, or if early-stop is triggered when the reward does not increase for at least P consecutive steps. Implementation details are provided in Algorithm 1. We observe consistent improvements, particularly in details refinement and text-image alignment, when training per-sample negative embedding compared to the learned global negative embedding, as shown in Fig. 4.

5. Experiments

5.1. Implementation Details

Dataset. In this work, we use the prompts provided by ImageReward [39] for training. The training set consists of 10,000 prompts spanning various categories, including people, art, and outdoor scenes. We evaluate the proposed ReNeg on two popular benchmarks: Parti-Prompts [40] and HPSv2 [37]. Parti-Prompts contains 1,632 prompts encompassing various categories. Meanwhile, HPSv2 comprises 3,200 prompts, covering four styles of image descriptions: animation, concept art, paintings, and photo.

Training Setting. The proposed method is built upon the open-sourced *Stable Diffusion 1.5*. To optimize the negative

Table 2. Quantitative results on HPSv2 and Parti-Prompts benchmarks. '$+N^*$' represents the handcrafted negative prompt. Methods marked with † indicate our reproduced version. The best result is highlighted in **bold**, and the second best is underlined.

	Model	HPSv2				Parti-Prompts		
		Animation	Concept Art	Painting	Photo	Pickscore	Aesthetic	HPSv2.1
Direct Inference	SD1.5	25.92	24.66	24.65	25.62	18.40	5.23	25.67
	SD1.5 + N^*	27.29	26.33	26.39	27.01	19.14	5.26	26.79
Prompt Refinement	BeautifulPrompt [4]	24.07	23.95	23.99	21.40	19.38	5.78	22.72
	Promptist [13]	26.22	25.11	25.14	24.25	19.44	5.42	25.24
	DNP† [9]	26.02	25.08	24.89	25.49	19.81	5.21	25.83
Finetuning SD	DDPO-Aesthetic [2]	18.20	19.03	19.15	18.93	19.23	4.99	20.69
	DDPO-Alignment [2]	20.45	20.53	20.12	20.33	19.29	4.93	19.00
	Diffusion-DPO† [31]	27.60	26.42	26.36	26.32	19.48	5.26	26.62
	ReFL† [39]	29.04	28.34	28.21	27.48	18.17	5.48	27.97
	TextCraftor (Text) [21]	30.16	30.48	30.46	28.36	19.17	**5.90**	28.36
ReNeg	Global Neg. Emb.	31.37	31.67	32.00	29.27	19.90	5.45	29.16
	Per-sample Neg. Emb.	**32.21**	**32.52**	**32.83**	**30.00**	**19.97**	5.50	**29.84**

Algorithm 1 Learning per-sample negative embedding

Require: Prompt c, learned global negative embedding n, maximum steps N, patience value P
Variable $\mathcal{J}_{best} \leftarrow 0$, $p_{ctr} \leftarrow 0$
for $i = 1$ to N **do**
  $x_T \sim \mathcal{N}(0, I)$ // Sample noise as latent
  $\hat{x}_0 = \text{Sample}(x_T, c, n)$ // Sampling using Eq. (6)
  $\mathcal{J}_n(c) = \mathcal{R}(c, \hat{x}_0)$
  if $\mathcal{J}_n(c) > \mathcal{J}_{best}$ **then**
    $\mathcal{J}_{best} \leftarrow \mathcal{J}_n(c)$
    Reset $p_{ctr} \leftarrow 0$
  else // Early stopping
    $p_{ctr} \leftarrow p_{ctr} + 1$
    if $p_{ctr} \geq P$ **then break**
    end if
  end if
  Update n using gradient descend.
end for
Return per-sample negative embedding n.

embedding, we employ the AdamW [22] optimizer and a Cosine Scheduler for 4,000 steps, with a learning rate of $5e-3$ and batch size of 64. The weights of the pretrained T2I diffusion model are frozen throughout the optimization. We further refine the negative embedding for an additional 10 steps with a patience value of 3 to obtain the per-sample negative embedding. At inference, DDIM scheduler with 30 steps is used for sampling and the classifier-free guidance weight is set to 7.5 with the resolution 512×512.

Evaluation metrics. We adopt the Human Preference Score v2.1 (HPSv2.1) [37], PickScore [18], and an aesthetic predictor [28] to comprehensively evaluate our method. Both HPSv2.1 and PickScore are derived from CLIP-based models trained on large-scale human preference datasets, allowing them to approximate human perception in assessing image quality. These metrics have demonstrated strong alignment with actual human preferences. The aesthetic predictor assesses visual appeal by analyzing high-dimensional image features, focusing on aspects of style and semantics. To further evaluate the performance of existing T2V models enhanced by ReNeg, we employ four metrics from VBench [17]: Aesthetic Quality, Motion Smoothness, Temporal Flickering, and Background Consistency.

5.2. Comparison with State-of-the-Arts

We compare ReNeg with three categories of methods: (1) Null-text embedding and handcrafted negative embedding; (2) Prompt refinement methods that utilize automatic or manual recaptioning for either positive or negative prompts; and (3) Human preference alignment methods that typically involve tuning all model parameters. The quantitative results are presented in Tab. 2.

Quantitative results. First, we compare the proposed method with a handcrafted negative prompts, which consolidates commonly suggested negative prompts from the community. By incorporating the handcrafted negative prompts, SD1.5 achieves performance gains on both benchmarks. However, it still lags behind our optimized negative embedding. We further compare our method with prompt optimization (recaptioning) methods and outperform them by a large margin. Remarkably, our method achieves comparable performance against the TextCraftor, which involves full finetuning of the UNet weights. The quantitative results highlight that our method not only enhances the visual appeal of generated results but also aligns more closely with human preferences. Notably, our method can achieve additional performance gains by combining with positive prompt refinement. Moreover, adaptively finetuning the negative embedding for a given positive prompts leads to further improvements in generation quality.

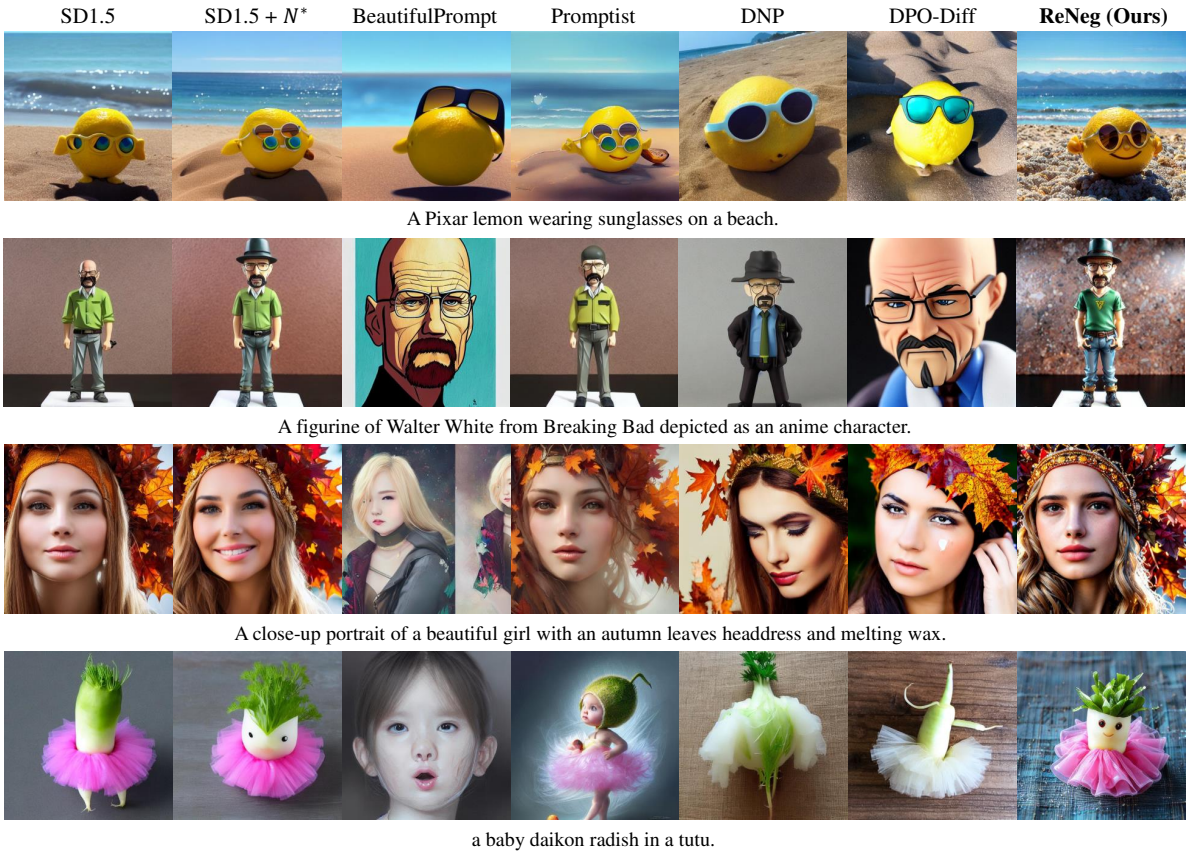

Figure 5. Qualitative comparisons. The prompts source from HPSv2 and Parti-Prompts benchmarks. All images are generated at a resolution of 512×512. For a fair comparison, the images above are generated using the same initial noise and seed.

Qualitative results. The qualitative comparisons are presented in Fig. 5. By incorporating a handcrafted negative prompts, SD1.5 achieves significant improvements in both visual quality and semantical reasonability. Compared to that, the methods for positive prompt refinement [4, 13] yield minimal gains in visual quality. This limited improvement can be attributed primarily to the fact that the optimized prompts generated by these methods sometimes deviate from the intended meaning of the original description, resulting in outputs that do not fully capture the desired content. Images generated by DNP [9] and DPO-Diff [33] lack finer details and aesthetic appeal. This is partially because of their constrained search strategies and search space, which pose a challenge to finding an optimal negative prompts. Our method, however, produces higher-quality images with finer details and coherent semantic alignment to the text prompts.

5.3. Visualization of Negative Embedding

To elucidate the differences among our learned negative embedding, the null-text prompt, and the handcrafted negative prompts, we visualize their corresponding embeddings. Utilizing SD1.5 as the foundational model, we treat negative embeddings as pseudo-positive prompts to generate corresponding images. The observed variations in appearance between the two columns of negative embeddings arise from differing random noise. As illustrated in Fig. 6, our learned embedding exhibits more muted colors and lacks significant semantic information compared to both null-text and handcrafted prompts. Specifically, the null-text prompt represents an average distribution of naturally generated images, closely aligning with authentic visual data. In contrast, the handcrafted prompts displays slight deviations from natural images; while it abstracts certain features, it retains recognizable semantic elements, such as textures and discernible semantic information. Conversely, the images generated from our learned embedding appear atypical and lack clear semantic content. Notably, the outputs from our method align more closely with human aesthetic preferences and maintain semantic coherence.

5.4. Generalization Capability

Generalization across different SD models. The proposed ReNeg is flexible and can be easily applied to various SD models, including SD1.4 and SD2.1. We calculate the win rate between the SD models using handcrafted nega-

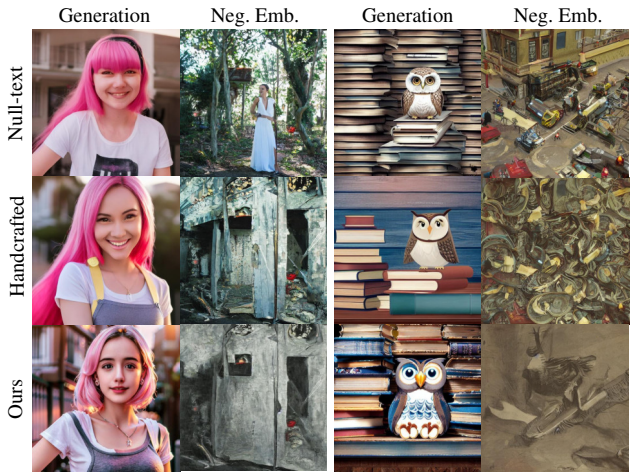

Figure 6. Example of generation results using different negative embeddings for the same prompt and the corresponding negative embeddings visualization. **Prompts: (Left)** *Frontal portrait of anime girl with pink hair wearing white t-shirt and smiling.* **(Right)** *A plushy tired owl sits on a pile of antique books in a humorous illustration.*

Table 3. Automated win rates on the HPSv2 benchmark comparing SD1.4, SD1.5, and SD2.1 using our negative embedding versus a handcrafted negative prompt. Results are calculated based on the HPSv2.1 metric.

Model	Animation	Concept Art	Painting	Photo	Average
SD1.4	0.91	0.95	0.97	0.87	0.93
SD1.5	0.99	0.99	0.98	0.99	0.99
SD2.1	0.96	0.98	0.98	0.93	0.96

Table 4. Performance comparison on video generation models. 'Handcrafted Prompt' denotes the handcrafted negative prompts.

Model	Aesthetic Quality	Motion Smoothness	Temporal Flickering	Background Consistency
VideoCrafter2	58.0	97.7	96.2	97.6
w/ Handcrafted Prompt	57.8	97.8	**96.5**	97.9
w/ Our ReNeg	**58.6**	**97.8**	96.4	**98.5**
ZeroScope	49.9	**98.9**	**98.3**	97.7
w/ Handcrafted Prompt	53.1	98.6	97.9	98.2
w/ Our ReNeg	**58.7**	98.1	97.3	**98.7**

tive prompts and the learned negative embedding on HPSv2 and Parti-Prompts. To provide a more meaningful comparison, we calculate the win rate of using our negative embedding relative to the handcrafted negative prompts, rather than comparing it to a setup that exclusively uses a positive prompts (*i.e.,* without any negative prompts). As shown in Tab. 3 and Fig. 7, the learned negative embedding significantly outperforms the handcrafted counterpart, achieving substantial improvements in human preference alignment.

Generalization across T2I and T2V Models. Here, we conduct experiments to examine the generalization capability of the proposed method across different generative

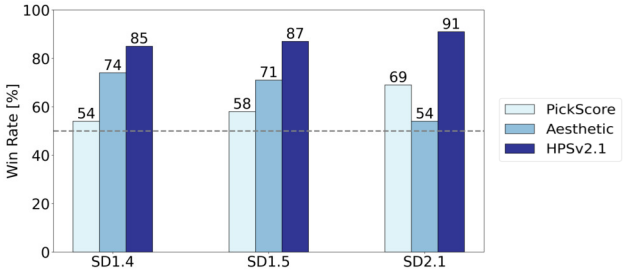

Figure 7. Comparison of the win rates on Parti-Prompts with and without our ReNeg. Across three metrics, our negative embedding achieves strong performance on various T2I models.

Table 5. Performance comparison on ControlNet. Our negative embedding enables a more visually appealing pose-to-image effect and better aligns with human preferences.

Model	PickScore	Aesthetic	HPSv2.1
ControlNet	19.49	5.54	26.83
w/ Handcrafted Prompt	**19.54**	5.54	27.60
w/ Our ReNeg	19.52	**5.95**	**30.79**

models and tasks. As shown in Fig. 1, the learned negative embedding can be transferred seamlessly to the text-to-image and text-to-video models, including ControlNet, ZeroScope, and VideoCrafter2. The quantitative results are reported in Tab. 4 and Tab. 5. We observe a consistent performance improvement on both ControlNet and T2V models by incorporating the learned negative embeddings, which reveal strong generalization capabilities of the proposed ReNeg. To conclude, the learned negative embedding can be easily shared across any text-conditioned generative models using the same text encoder. More qualitative results can be found in the appendix.

6. Conclusion

We propose ReNeg, a framework that searches for a global negative embedding under reward feedback. Building on this, we adaptively learn a distinct negative embedding tailored to each positive prompt, which exhibits consistent improvements in detail refinement and textual alignment. To enable effective gradient propagation through reward guidance, we incorporate the CFG-training strategy. Despite its simplicity, our negative embedding proves to be highly useful, surpassing results generated with only positive prompts or handcrafted negative prompts and rivaling those achieved through full finetuning of diffusion models. Moreover, our negative embedding can be easily transferred to other T2I or T2V models, provided they share the same text encoder. Although ReNeg can generate visually appealing images, limited by the generative ability of the base model, it sometimes shows semantic deviations from the prompt. The improvement is left to our subsequent work.

Acknowledgements. This work was partially supported by the National Natural Science Foundation of China under Grant Nos. 62472065, 62441231, U23B2010.

References

[1] Yuanhao Ban, Ruochen Wang, Tianyi Zhou, Minhao Cheng, Boqing Gong, and Cho-Jui Hsieh. Understanding the impact of negative prompts: When and how do they take effect? *arXiv preprint arXiv:2406.02965*, 2024. 2

[2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 3, 6

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 2

[4] Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. Beautifulprompt: Towards automatic prompt engineering for text-to-image synthesis. *arXiv preprint arXiv:2311.06752*, 2023. 2, 6, 7

[5] cerspense. `https://huggingface.co/cerspense/zeroscope_v2_576w`, 2023. 2

[6] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024. 2

[7] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *NeurIPS*, 2017. 3, 4

[8] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023. 2

[9] Alakh Desai and Nuno Vasconcelos. Improving image synthesis with diffusion-negative sampling. In *ECCV*, 2025. 2, 3, 6, 7

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021. 2

[11] Jindong Gu, Zhen Han, Shuo Chen, Ahmad Beirami, Bailan He, Gengyuan Zhang, Ruotong Liao, Yao Qin, Volker Tresp, and Philip Torr. A systematic survey of prompt engineering on vision-language foundation models. *arXiv preprint arXiv:2307.12980*, 2023. 2

[12] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2

[13] Yaru Hao, Zewen Chi, Li Dong, and Furu Wei. Optimizing prompts for text-to-image generation. *NeurIPS*, 2024. 2, 6, 7

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2, 3

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33, 2020. 2, 3

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[17] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 6

[18] 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. *NeurIPS*, 2023. 2, 6

[19] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. 2023. 2

[20] Xiaomin Li, Xu Jia, Qinghe Wang, Haiwen Diao, Mengmeng Ge, Pengxiang Li, You He, and Huchuan Lu. Motrans: Customized motion transfer with text-driven video diffusion models. In *ACM MM*, 2024. 2

[21] Yanyu Li, Xian Liu, Anil Kag, Ju Hu, Yerlan Idelbayev, Dhritiman Sagar, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Textcraftor: Your text encoder can be image quality controller. In *CVPR*, 2024. 2, 3, 6

[22] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[23] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. 3, 4

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 2

[25] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2024. 3

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

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 4

[28] 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. *NeurIPS*, 2022. 2, 6

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[30] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based

generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[31] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, 2024. 3, 6

[32] Qinghe Wang, Xu Jia, Xiaomin Li, Taiqing Li, Liqian Ma, Yunzhi Zhuge, and Huchuan Lu. Stableidentity: Inserting anybody into anywhere at first sight. *arXiv preprint arXiv:2401.15975*, 2024. 2

[33] Ruochen Wang, Ting Liu, Cho-Jui Hsieh, and Boqing Gong. On discrete prompt optimization for diffusion models. *arXiv preprint arXiv:2407.01606*, 2024. 2, 3, 7

[34] Sam Witteveen and Martin Andrews. Investigating prompt engineering in diffusion models. *arXiv preprint arXiv:2211.15462*, 2022. 2

[35] Max Woolf. Stable diffusion 2.0 and the importance of negative prompts for good results, 2022. https://minimaxir.com/2022/11/stable-diffusion-negative-prompt/. 2

[36] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 2

[37] 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. 2, 3, 4, 5, 6

[38] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *International Journal of Computer Vision*, 2024. 2

[39] 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. 2024. 2, 3, 4, 5, 6

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

[41] David Junhao Zhang, Jay Zhangjie Wu, Jia-Wei Liu, Rui Zhao, Lingmin Ran, Yuchao Gu, Difei Gao, and Mike Zheng Shou. Show-1: Marrying pixel and latent diffusion models for text-to-video generation. *arXiv preprint arXiv:2309.15818*, 2023. 2

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2