Appendix: Energy-Based Cross Attention for Bayesian Context Update in Text-to-Image Diffusion Models

A Proof of Theorem 1

Theorem 1. *For the energy functions*

$$\mathrm{E}(\boldsymbol{Q};\boldsymbol{K}) = \frac{\alpha}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T) - \sum_{i=1}^{N} \mathrm{logsumexp}(\boldsymbol{Q}\boldsymbol{k}_i^T, \beta) \tag{17}$$

and

$$E(\boldsymbol{K}) = \log \sum_{i=1}^{N} \exp(\frac{1}{2}\boldsymbol{k}_i\boldsymbol{k}_i^T), \tag{18}$$

the gradient of the log posterior is given by:

$$\nabla_{\boldsymbol{K}} \log p(\boldsymbol{K} \mid \boldsymbol{Q}) = \mathrm{softmax}_2\left(\beta \boldsymbol{K}\boldsymbol{Q}^T\right)\boldsymbol{Q} - \Bigg(\alpha \boldsymbol{I} + \boldsymbol{D}\Big(\mathrm{softmax}\left(\frac{1}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T)\right)\Big)\Bigg)\boldsymbol{K}, \tag{19}$$

Then, by using the chain rule the update rule of context vectors $\boldsymbol{C}$ is derived as follows:

$$\boldsymbol{C}_{n+1} = \boldsymbol{C}_n + \gamma\Bigg(\mathrm{softmax}_2\left(\beta \boldsymbol{K}\boldsymbol{Q}^T\right)\boldsymbol{Q} - \Bigg(\alpha \boldsymbol{I} + \boldsymbol{D}\Big(\mathrm{softmax}\left(\frac{1}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T)\right)\Big)\Bigg)\boldsymbol{K}\Bigg)\boldsymbol{W}_K^T, \tag{20}$$

where $\gamma > 0$ is a step size, and $\boldsymbol{D}(\cdot)$ is a vector-to-diagonal-matrix operator.

Proof. Based on the Bayes' theorem, the gradient of the log posterior is derived as:

$$\nabla_{\boldsymbol{K}} \log p(\boldsymbol{K} \mid \boldsymbol{Q}) = -\big(\nabla_{\boldsymbol{K}}\,\mathrm{E}(\boldsymbol{Q};\boldsymbol{K}) + \nabla_{\boldsymbol{K}}\,\mathrm{E}(\boldsymbol{K})\big). \tag{21}$$

First, with definition (17),

$$\nabla_{\boldsymbol{K}}\,\mathrm{E}(\boldsymbol{Q};\boldsymbol{K}) = \alpha\boldsymbol{K} - \nabla_{\boldsymbol{K}} \sum_{i=1}^{N} \mathrm{logsumexp}(\boldsymbol{Q}\boldsymbol{k}_i^T, \beta), \tag{22}$$

where $\forall i \in \{1, \ldots, N\}$,

$$\begin{aligned}\nabla_{\boldsymbol{k}_i} \sum_{i=1}^{N} \mathrm{logsumexp}(\boldsymbol{Q}\boldsymbol{k}_i^T, \beta) &= \frac{1}{\beta}\nabla_{\boldsymbol{k}_i} \log \sum_{j=1}^{P_l^2} \exp(\beta \boldsymbol{q}_j \boldsymbol{k}_i^T) \\ &= \sum_{j=1}^{P_l^2} \frac{\exp(\beta \boldsymbol{q}_j \boldsymbol{k}_i^T)}{\sum_{n=1}^{P_l^2} \exp(\beta \boldsymbol{q}_n \boldsymbol{k}_i^T)} \boldsymbol{q}_j \\ &= \mathrm{softmax}(\boldsymbol{Q}\boldsymbol{k}_i^T)^T \boldsymbol{Q}.\end{aligned} \tag{23}$$

Then, by considering that $\boldsymbol{k}_i$ is a i-th row vector of $\boldsymbol{K}$,

$$\begin{aligned}\nabla_{\boldsymbol{K}} \sum_{i=1}^{N} \mathrm{logsumexp}(\boldsymbol{Q}\boldsymbol{k}_i^T, \beta) &= \big(\mathrm{softmax}_1(\beta \boldsymbol{Q}\boldsymbol{K}^T)\big)^T \boldsymbol{Q} \\ &= \mathrm{softmax}_2(\beta \boldsymbol{K}\boldsymbol{Q}^T)\boldsymbol{Q},\end{aligned} \tag{24}$$

where the last equality holds due to the definition of $\mathrm{softmax}_1$ in Section 2.2.

Second, with definition (18), $\nabla_{\boldsymbol{K}} \mathrm{E}(\boldsymbol{K}) = \nabla_{\boldsymbol{K}} \log \sum_{i=1}^{N} \exp(\frac{1}{2}\boldsymbol{k}_i\boldsymbol{k}_i^T)$, where

$$\begin{aligned}\nabla_{\boldsymbol{k}_i} \log \sum_{i=1}^{N} \exp(\frac{1}{2}\boldsymbol{k}_i\boldsymbol{k}_i^T) &= \frac{\exp(\frac{1}{2}\boldsymbol{k}_i\boldsymbol{k}_i^T)}{\sum_{j=1}^{N}\exp(\frac{1}{2}\boldsymbol{k}_j\boldsymbol{k}_j^T)}\boldsymbol{k}_i \\ &= \mathrm{softmax}\left(\frac{1}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T)\right)_i \boldsymbol{k}_i,\end{aligned} \tag{25}$$

where $\mathrm{softmax}(\cdot)_i$ denotes i-th value of a softmax vector. Then,

$$\nabla_{\boldsymbol{K}} \log \sum_{i=1}^{N} \exp(\frac{1}{2}\boldsymbol{k}_i\boldsymbol{k}_i^T) = \boldsymbol{D}\Big(\ \mathrm{softmax}\,(\frac{1}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T))\Big)\boldsymbol{K}, \tag{26}$$

where $\boldsymbol{D}(\cdot)$ is a vector-to-diagonal-matrix operator that takes N-dimensional $\mathrm{softmax}$ vector as an input and returns a $N \times N$ diagonal matrix with $\mathrm{softmax}$ values as main diagonal entries. Then, By combining (22), (24) and (26), one can finally obtain:

$$\nabla_{\boldsymbol{K}} \log p(\boldsymbol{K} \mid \boldsymbol{Q}) = \mathrm{softmax}_2\left(\beta\boldsymbol{K}\boldsymbol{Q}^T\right)\boldsymbol{Q} - \Big(\alpha\boldsymbol{I} + \boldsymbol{D}\Big(\ \mathrm{softmax}\left(\frac{1}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T)\right)\Big)\Big)\boldsymbol{K}. \tag{27}$$

By using the chain rule with $\boldsymbol{K} = \boldsymbol{C}\boldsymbol{W}_K$, the update rule of context vectors $\boldsymbol{C}$ is derived as in (20). □

We introduce vector-to-matrix operator $\boldsymbol{D}(\cdot)$ to avoid confusion and fix the typo in the main paper.

B Pseudo-code for BCU and CACAO

This section provides the description of the pseudocode for the proposed Bayesian Context Update (BCU) and Compositional Averaging of Cross-Attention Output (CACAO). Algorithm 1 outlines the cascaded context propagation across cross-attention layers within the UNet model during the sampling step t. Note that the context is reinitialized at the beginning of each sampling step. On the other hand, Algorithm 2 details the BCU implemented in each cross-attention layer. Remark that $\boldsymbol{D}$ in line 5 denotes vector-to-diagonal-matrix operator. Specifically, the proposed BCU provides a significant computational efficiency by reusing the similarity $\boldsymbol{Q}\boldsymbol{K}^T$, which requires computational cost $\mathcal{O}(N^2)$, to compute $\nabla_{\boldsymbol{K}} E(\boldsymbol{Q}; \boldsymbol{K})$. Consequently, there is only a small amount of additional computational overhead associated with the proposed BCU.

Algorithm 1 Context cascade at sampling step t

Require: $\boldsymbol{Q}_t, \boldsymbol{C}_{clip}$, UNet

1: $\boldsymbol{C}_t \leftarrow \boldsymbol{C}_{clip}$ // Re-initialize
2: **for** layer in UNet **do**
3:     **if** layer is CrossAttention **then**
4:         $\boldsymbol{Q}_t, \boldsymbol{C}_t \leftarrow$ layer($\boldsymbol{Q}_t, \boldsymbol{C}_t$) // Algorithm 2
5:     **else**
6:         $\boldsymbol{Q}_t \leftarrow$ layer($\boldsymbol{Q}_t$)
7:     **end if**
8: **end for**
9: $\boldsymbol{Q}_{t+1} \leftarrow \boldsymbol{Q}_t$
10: **return** $\boldsymbol{Q}_{t+1}$

Algorithm 2 Bayesian Context Update (BCU)

Require: $\boldsymbol{Q}, \boldsymbol{C}, \boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v, \alpha, \beta, \gamma_{\text{attn}}, \gamma_{\text{reg}}$

$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \leftarrow \boldsymbol{Q}\boldsymbol{W}_q, \boldsymbol{C}\boldsymbol{W}_k, \boldsymbol{C}\boldsymbol{W}_v$

$\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{K}^T$

$\boldsymbol{Q} \leftarrow \text{softmax}_2(\beta \boldsymbol{S})\boldsymbol{V}$

$\nabla_{\boldsymbol{K}} E(\boldsymbol{Q}; \boldsymbol{K}) = \text{softmax}_2(\beta \boldsymbol{S}^T)\boldsymbol{Q}$

$\nabla_{\boldsymbol{K}} E(\boldsymbol{K}) = -(\alpha \boldsymbol{I} + \boldsymbol{D}(\text{softmax}(\frac{1}{2}\text{diag}(\boldsymbol{K}\boldsymbol{K}^T))))\boldsymbol{K}$

$\Delta \boldsymbol{C} = (\gamma_{\text{attn}} \nabla_{\boldsymbol{K}} E(\boldsymbol{Q}; \boldsymbol{K}) + \gamma_{\text{reg}} \nabla_{\boldsymbol{K}} E(\boldsymbol{K}))\boldsymbol{W}_k^T$

$\boldsymbol{C} \leftarrow \boldsymbol{C} + \Delta \boldsymbol{C}$

return $\boldsymbol{Q}, \boldsymbol{C}$

Algorithm 3 outlines the pseudocode for the CACAO implemented for M given contexts. For the simplicity, we exclude the BCU from the algorithm. Nontheless, the BCU and the CACAO could be leveraged together.

Algorithm 3 Compositional Averaging of Cross-Attention Output (CACAO)

Require: $\boldsymbol{Q}, \boldsymbol{C} = \{\boldsymbol{C}_1, ..., \boldsymbol{C}_M\}, \boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v, \alpha_s, \beta$

$\boldsymbol{Q} \leftarrow \boldsymbol{Q}\boldsymbol{W}_q$

for s in $[1, ..., M]$ **do**

  $\boldsymbol{K}_s, \boldsymbol{V}_s \leftarrow \boldsymbol{C}_s\boldsymbol{W}_k, \boldsymbol{C}_s\boldsymbol{W}_v$

  $\boldsymbol{S}_s = \boldsymbol{Q}\boldsymbol{K}_s^T$

end for

$Q \leftarrow \frac{1}{M}\sum_{s=1}^{M} \alpha_s \text{softmax}_2(\beta \boldsymbol{S}_s)\boldsymbol{V}_s$

return $\boldsymbol{Q}$

C Experimental setups

In this section, we describe detailed experimental setups for three applications including baseline method, hyper-parameter of the proposed method, and dataset if it is the case. Code: `https://github.com/EnergyAttention/Energy-Based-CrossAttention`.

C.1 Common experimental setup

We mainly leverage pre-trained Stable Diffusion v1-5 (except Table 1: v1-4) which is provided by *diffusers*, a Python library that offers various Stable Diffusion pipelines with pre-trained models. All images are sampled for 50 steps via PNDM sampler [20] using NVIDIA RTX 2080Ti. In every experiment, we set the parameter α in Equation (18) to zero, focusing solely on controlling the values of γ_{attn} and γ_{reg}. BCU is applied to every task, and CACAO is additionally employed in C.4.

Different learning rate for each token It is worth noting that the γ_{attn} and γ_{reg} could be expressed as vectors. In other words, if the context $C \in \mathrm{R}^{N \times d_c}$ is given, γ_{attn} and γ_{reg} are N-dimensional vectors. Hence, we have the flexibility to adjust the learning rate $\gamma_{\{\cdot\}}$, allowing us to increase or decrease the impact of certain tokens based on the user's intent. Unless otherwise noted, γ_{attn} and γ_{reg} is set to a constant for each text token.

Learning rate scheduling Since the proposed BCU is leveraged for the diffusion model, one can readily introduce scheduling strategies for γ_{attn} and γ_{reg} along the sampling step t. We implement multiple variants such as 'constant', 'step', and 'exponential decay' as follows.

$$\begin{aligned} &[\text{constant}] \quad \gamma(t) = \gamma_0 \\ &[\text{step}] \quad \gamma(t) = \gamma_0 \cdot \text{ReLu}(t - \tau) \\ &[\text{exp-decay}] \quad \gamma(t) = \gamma_0 \cdot \lambda^t \end{aligned} \tag{28}$$

where γ_0 is the initial value, $\text{ReLu}(x) = 0$ if $x \leq 0$, otherwise 1, τ denotes the temporal threshold, and λ denotes the decay ratio. Unless stated otherwise, the scheduling strategy is set to the 'constant'.

C.2 Multi-concept image generation

We compared the performance of the proposed method with Structured Diffusion [11] which does not require additional training as our method. We leveraged the open-sourced official implementation [1].

For the proposed method, we set the γ_{attn} and γ_{reg} differently for each sample within [1e-2, 1.5e-2, 2e-2]. As shown in the following ablation studies E, large γ_{attn} tends to generate saturated images while large γ_{reg} results in mixed/vanished contents.

We found that using different learning rates for each context token is useful for multi-concept generation, especially when a single concept tends to dominate with a constant learning rate. For example, given the main prompt `"A cat wearing a shirt"`, we set the γ_{attn} for the `"shirt"` to 3e-2, while γ_{attn} is set to 1.5e-2 for other tokens. We have observed that doubling the γ_{attn} for a text token to be emphasized is sufficient to achieve balanced multi-concept image generation for most cases.

C.3 Text-guided image inpainting

Additionally, we conducted a performance comparison between our proposed method and two alternative approaches: (**a**) Stable Inpaint[2], which fine-tunes the weights of Stable Diffusion through inpainting training, and (**b**) Stable Repaint[3], which leverages the work of Lugmayr et al. [22] on the latent space of Stable Diffusion for the inpainting task. In the case of Stable Repaint, the mask is downsized and transferred into the latent space. We applied the Bayesian Context Update (BCU) technique to both methods, resulting in improved results compared to their respective baselines.

Masked BCU. To further enhance the performance for the inpainting task, we introduce the concept of masked Bayesian Context Update (masked BCU). Specifically, let $M \in \mathbb{R}^{P_l^2 \times P_l^2}$ represent a diagonal matrix where the main diagonal values are derived from the downsampled inpainting mask for the l-th cross-attention layer, with an output spatial size of P_l^2. In Equation (29), we modify the attention term (12) by incorporating the downsampled mask, effectively covering the query matrix as follows:

$$\boldsymbol{C}_{n+1} = \boldsymbol{C}_n + \gamma\Big(\operatorname{softmax}_2\left(\beta \boldsymbol{K}\boldsymbol{Q}^T\right)\boldsymbol{M}\boldsymbol{Q} - \Big(\alpha \boldsymbol{I} + \boldsymbol{D}\Big(\operatorname{softmax}\left(\frac{1}{2}\operatorname{diag}(\boldsymbol{K}\boldsymbol{K}^T)\right)\Big)\Big)\boldsymbol{K}\Big)\boldsymbol{W}_K^T. \tag{29}$$

As evident in Equation (12), the attention term updates the context vectors, aligning $\boldsymbol{k}_i$ towards $\boldsymbol{q}_j, j = 1, \ldots, P_l^2$, while considering the alignment strength between each $\boldsymbol{q}_j$ and $\boldsymbol{k}_i$. However, in the inpainting task, we have prior knowledge that the context vectors should be most aligned with the semantically relevant masked regions. Therefore, we mask out unrelated background spatial representations, allowing for the context vectors to be updated with a specific focus on the masked regions. This approach facilitates the incorporation of semantic information encoded by $\boldsymbol{k}_i$ specifically into the spatial mask regions.

In our proposed method, we set different values for γ_{attn} and γ_{reg} for each sample, selected from the set [1e-2, 1.5e-2, 2e-2, 2.5e-2], to account for variations in the input samples.

C.4 Image editing via compositional generation

We present empirical evidence demonstrating the effectiveness of our energy-based framework for compositional synthetic and real-image editing. The Bayesian Context Update (BCU) technique can be readily applied to both the main context vector ($\boldsymbol{C}_1$ in Section 3.2, $s = 1$) and editorial context vectors ($\boldsymbol{C}_{s>1}$). Each BCU operation influences the attention maps used in Compositional Averaging of Cross-Attention Output (CACAO), enhancing the conveyance of semantic information associated with each context. Note that α_s in (16) represents the degree of influence of the s-th concept in the composition. In practice, we fix $\alpha_1 = 1$ for the main context, while $\alpha_{s>1}$ is tuned within the range of (0.5, 1.0).

---

[1] `https://github.com/weixi-feng/Structured-Diffusion-Guidance`

[2] `https://huggingface.co/runwayml/stable-diffusion-inpainting`

[3] `https://github.com/huggingface/diffusers/tree/main/examples/community#stable-diffusion-repaint`

Let $\gamma_{attn,s}$ and $\gamma_{reg,s}$ denote the step sizes for BCU of the s-th context vector. If the editing process involves changing the identity of the original image (e.g., transforming a "cat" into a "dog"), we set both $\gamma_{attn,1}$ and $\gamma_{reg,1}$ to zero. Otherwise, if the editing maintains the original identity, we choose values for $\gamma_{attn,1}$ and $\gamma_{reg,1}$ from the range of (5e-4, 1e-3), similar to $\gamma_{attn,(s>1)}$ and $\gamma_{reg,(s>1)}$. All hyperparameters, including α_s and γ_s, are fixed during the quantitative evaluation process (more details in Section D and Table 2).

To ensure consistent results, we maintained a fixed random seed for both real and synthetic image editing. For real image editing, we employed null-text pivotal inversion [24] to obtain the initial noise vector.

During the reverse diffusion process in Sections C.2 and C.3, we kept γ fixed as a constant value. However, for compositional generation, we utilized step scheduling (Equation 28) for γ_s and α_s. After converting the initial noise vector for real images or using a fixed random seed for synthetic images, BCU and CACAO are applied after a threshold time $\tau_s > 0$ for the s-th editorial context. This scheduling strategy helps to preserve the overall structure of generated images during the editing process. In our observations, a value of $\tau_s \in [10, 25]$ generally produces satisfactory results, considering a total number of reverse steps set to 50. However, one can increase or decrease τ_s for more aggressive or conservative editing, respectively.

The exemplary real images presented in Figures 5 and 6 of the main paper were sampled from datasets such as FFHQ [16], AFHQ [5], and ImageNet [7]. For a detailed quantitative analysis, please refer to Section D.

D Quantitative Comparison

In this section, we conducted a comparative analysis of the proposed framework against several state-of-the-art diffusion-based image editing methods [23, 12, 24, 27], following the experimental setup of [27]. To ensure a fair comparison, all methods utilize the pre-trained Stable Diffusion v1-4, employ the PNDM sampler with an equal number of sampling steps, and adopt the same classifier-free guidance scale.

D.1 Baseline Methods

In addition to the Plug-and-Play method discussed in the main paper, we include the following baselines for comprehensive quantitative comparison:

SDEdit [23] + word swap. This method introduces the Gaussian noise of an intermediate timestep and progressively denoises images using a new textual prompt, where the source word (e.g., `Cat`) is replaced with the target word (e.g., `Dog`).

Prompt-to-prompt (P2P) [12]. P2P edits generated images by leveraging explicit attention maps from a source image. The source attention maps M_t are used to inject, re-weight, or override the target maps based on the desired editing operation. These original maps act as hard constraints for the edited images.

DDIM + word swap [24]. This method applies null-text inversion to real input images, achieving high-fidelity reconstruction. DDIM sampling is then performed using inverted noise vectors and an edited prompt generated by swapping the source word with the target.

pix2pix-zero [27]. pix2pix-zero first derives a text embedding direction vector $\triangle c_{\text{edit}}$ from the source to the target by using a large bank of diverse sentences generated from a state-of-the-art sentence generator, such as GPT-3 [2]. Inverted noise vectors are denoised with the edited text embedding, $c + \triangle c_{\text{edit}}$, and cross-attention guidance to preserve consensus.

D.2 Dataset

For our quantitative evaluations, we focus on three image-to-image translation tasks: (1) translating cats to dogs (cat $\rightarrow$ dog), (2) translating horses to zebras (horse $\rightarrow$ zebra), and (3) adding glasses to cat input images (cat $\rightarrow$ cat with glasses). Following the data collection protocol of [27], we retrieve 250 relevant cat images and 213 horse images from the LAION 5B dataset [33] using CLIP embeddings of the source text description. We select images with a high CLIP similarity to the source word for each task.

D.3 Metrics

Motivated by [38, 27], we measure CLIP Accuracy and DINO-ViT structure distance. Specifically, (**a**) CLIP Acc represents whether the targeted semantic contents are well reflected in the generated images. It calculates the percentage of instances where the edited image has a higher similarity to the target text, as measured by CLIP, than to the original source text [27]. On the other hand, (**b**) structure distance [38, 37] measures whether the overall structure of the input image is well preserved. It is defined as the difference in self-similarity of the keys extracted from the attention module at the deepest DINO-ViT [3] layer.

D.4 Details

The main context vector $\boldsymbol{C}_{main}$ is encoded given a main prompt automatically generated by BLIP [19]. In addition, the editorial context vectors $\boldsymbol{C}_{src}$ and $\boldsymbol{C}_{tgt}$ are encoded given the text descriptions of the source and target concept, i.e. source and target prompt. For example, for a cat $\rightarrow$ dog task (cat $\rightarrow$ cat w/ glasses), the source prompt is `"cat"` (`"cat wearing glasses"`), and the target prompt is `"dog"` (`"without glasses"`). Then we apply BCU and CACAO based on the obtained context vectors. Please refer to Table 2 for the hyperparameter configurations.

D.5 Results

Table 1 shows that the proposed energy-based framework gets a high CLIP-Acc while having low Structure Dist. It implies that the proposed framework can perform the best edit while still retaining the structure of the original input image. This is a remarkable result considering that the proposed framework is not specially designed for the real-image editing task. Moreover, the proposed framework does not rely on the large bank of prompts and editing vector $\triangle c_{\text{edit}}$ [27] which can be easily incorporated into our method.

While DDIM + word swap records remarkably high CLIP-Acc in horse $\rightarrow$ zebra task, Figure 7 and 12 show that such improvements are based on unintended changes in the overall structure. Table 2 summarizes the hyperparameter settings for each task. Examples of results are presented in Figure 13 and 12.

Table 1: Comparison to state-of-the-art diffusion-based editing methods. Dist for DINO-ViT Structure distance. Baseline results are from [27].

Method	(**a**) **Cat** $\rightarrow$ **Dog**		(**b**) **Horse** $\rightarrow$ **Zebra**		(**c**) **Cat** $\rightarrow$ **Cat w/ glasses**	
	CLIP Acc (↑)	Dist (↓)	CLIP Acc (↑)	Dist (↓)	CLIP Acc (↑)	Dist (↓)
SDEdit [23] + word swap	71.2%	0.081	92.2%	0.105	34.0%	0.082
DDIM + word swap	72.0%	0.087	**94.0**%	0.123	37.6%	0.085
prompt-to-prompt [12]	66.0%	0.080	18.4%	0.095	69.6%	0.081
pix2pix-zero [27]	92.4%	0.044	75.2%	0.066	71.2%	**0.028**
Stable Diffusion + ours	**93.7**%	**0.040**	90.4%	**0.061**	**81.1**%	0.052

Table 2: Hyperparameter configurations for each editing task. Each task index comes from Table 1. $\gamma_{attn,main} = 0$ and $\gamma_{reg,main} = 0$ as mentioned in section C.4. Note that $\alpha_{src} < 0$ for the concept negation (related ablation study in Figure 9). τ_s denotes the warm-up period for step scheduling in (28) and Section C.4.

Task	α_{src}	α_{tgt}	$\gamma_{\cdot,main}$	$\gamma_{attn,src}$	$\gamma_{reg,src}$	$\gamma_{attn,tgt}$	$\gamma_{reg,tgt}$	τ_s
(**a**)	-0.65	0.75	0	5e-4	5e-4	6e-4	6e-4	25
(**b**)	-0.5	0.6	0	4e-4	4e-4	5e-4	5e-4	15
(**c**)	-0.6	0.7	0	1e-3	1e-3	1e-3	1e-3	17

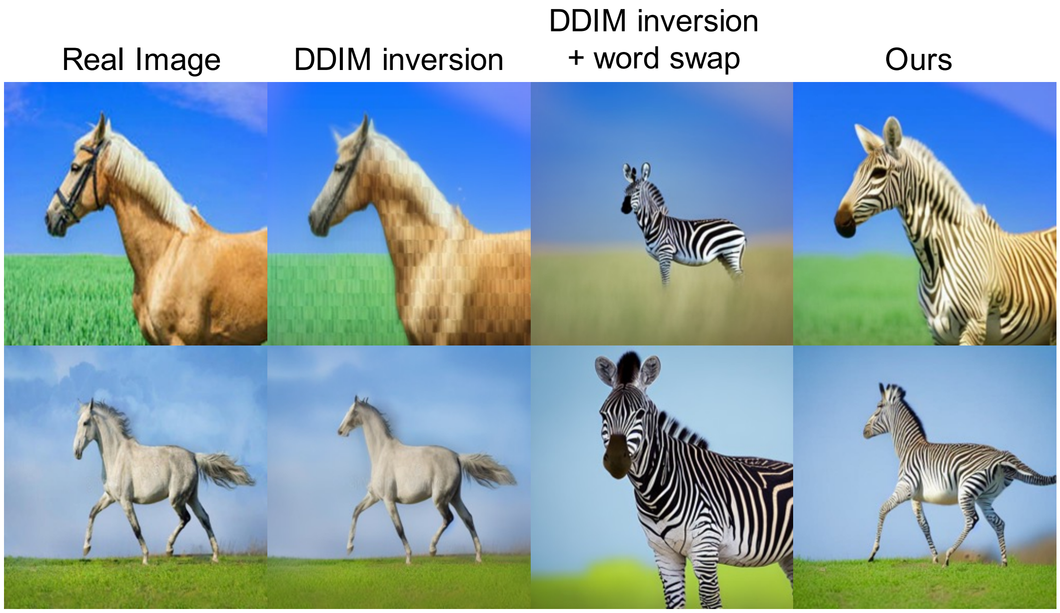

Figure 7: Image editing comparison with DDIM-inversion. Generated samples by DDIM-inversion with word swap readily deviate the original data contents, while the proposed method avoids undesired changes.

E Ablation study and more results

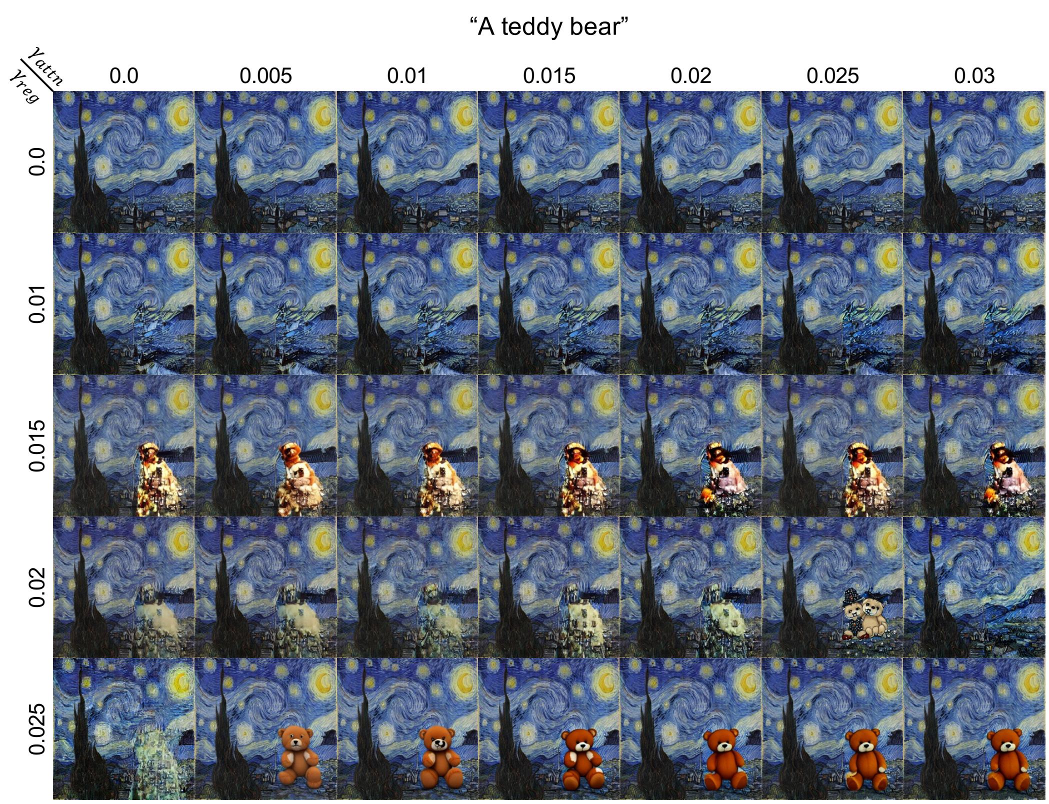

Figure 8: Ablation results for γ_{attn} and γ_{reg}. All samples are generated from the same random noise.

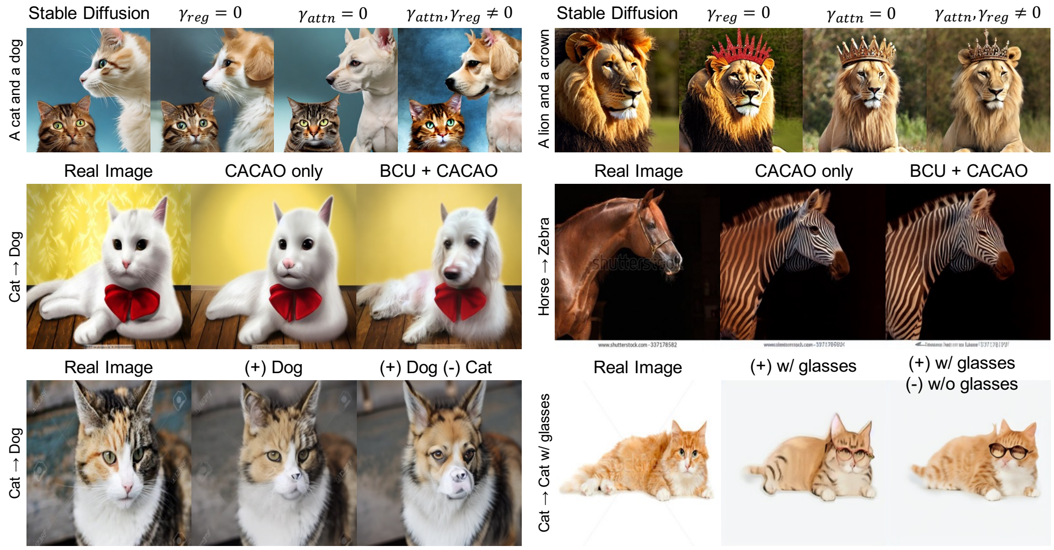

Figure 9: Ablation study results. The first row shows multi-concept generation examples with varying γ_{attn} and γ_{reg}, while the second row shows real image editing examples with varying the usage of BCU and CACAO. The last row shows the effect of negative prompt for the image editing application.

Attention and regularization terms. To access the degree of performance improvement attained by the proposed BCU, we conducted an ablation study for the attention and the regularization terms by regulating γ_{attn} and γ_{reg} for the text-guided image inpainting (Figure 8) and the multi-concept image generation (Figure 9). From the Figure 8, we can observe that the desired content is generated when proper range of γ_{attn} and γ_{reg} are given. Specifically, once γ_{reg} is set to a valid value, the BCU consistently generate a `"teddy bear"` with various γ_{attn}, otherwise it generates background or imperfect objects. This result emphasizes the role of the introduced prior energy $\mathrm{E}(\boldsymbol{K})$. Furthermore, the γ_{attn} also affects to the context alignment of the generated sample (for instance $\gamma_{attn} = 0.025$ and $\gamma_{reg} = 0.02$), which highlights the importance of the introduced conditional energy function $\mathrm{E}(\boldsymbol{Q}; \boldsymbol{K})$. The same evidences could be found in the first row in Figure 9 which are the multi-concept image generation examples.

Synergy between BCU and CACAO. While both BCU and CACAO are designed from the common energy-based perspective, each operation is originated from different energy functions $\mathrm{E}(\boldsymbol{K}; \boldsymbol{Q})$ and $\hat{\mathrm{E}}(\boldsymbol{Q}; \{\boldsymbol{K}_s\}_{s=1}^{M})$, respectively. This fact suggests the synergistic energy minimization by combining the BCU and CACAO, which could further improve the text-conditional image generation. To investigate this further, we conducted an ablation study using a real image editing application. Specifically, we compared the editing performance when solely utilizing CACAO and when combining BCU with CACAO. The second row in Figure 9 is the result of the ablation study that shows fully-compatibility of the BCU and CACAO. Importantly, the incorporation of the BCU improves the quality of the generated images. While the CACAO alone effectively captures the context of the given editing concept, the addition of BCU enhances the fine-grained details in the generated outputs.

Importance of concept negation. Remark that a negative α_s in (16) denotes the negation of given editing prompt. We empirically observed that the concept negation may significantly contribute to the performance of compositional generation. Specifically, for the image-to-image translation task in Table 1, we apply both positive and negative guidance with the target (e.g. `Dog`) *and* source (e.g. `Cat`) concepts, respectively, following the degree of guidance denoted in Table 2. The third row in Figure 9 shows the impacts of source concept negation in the image-to-image translation task. While the positive guidance alone may fail to remove the source-concept-related features, e.g. eyes of the `Cat`, the negative guidance removes such conflicting existing attributes. This implies that the proposed framework enables useful arithmetic of multiple concepts for both real and synthetic image editing.

Prior energy and α. While $\frac{\alpha}{2}\,\mathrm{diag}(\boldsymbol{K}\boldsymbol{K}^T)$ in (7) penalizes norm of each context vectors uniformly, the proposed prior energy function $\mathrm{E}(\boldsymbol{K})$ adaptively regularizes the smooth maximum of $\|\boldsymbol{k}_i\|$. Intuitively, adaptive penalization prevents the excessive suppression of context vectors, potentially resulting in images that are more semantically aligned with a given context. To demonstrate the effectiveness of adaptive penalization in the prior energy function, we conducted a multi-concept image generation task with varying α in (20) from 0 to 1, while fixing other hyperparameters. Figure 10 illustrates the gradual disappearance of salient contextual elements in the generated images depending on the change of α. Specifically, the crown is the first to diminish, followed by subsequent context elements, with the lion being the last to vanish with $\alpha = 1$. This result highlights the validity of the adaptive penalization for the context vectors which stems from the prior energy function.

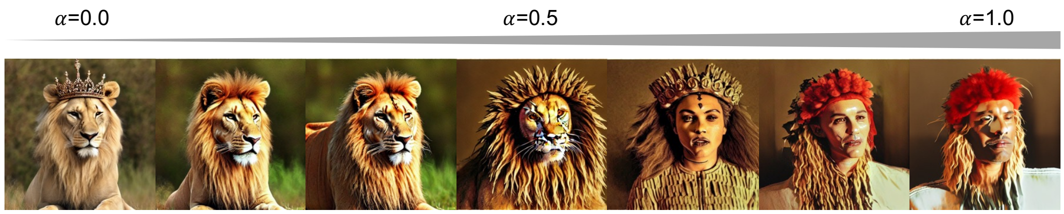

Figure 10: Generated samples with varying α values. As α increases, the generated images progressively deviate from the intended context, `"A lion and a crown"`.

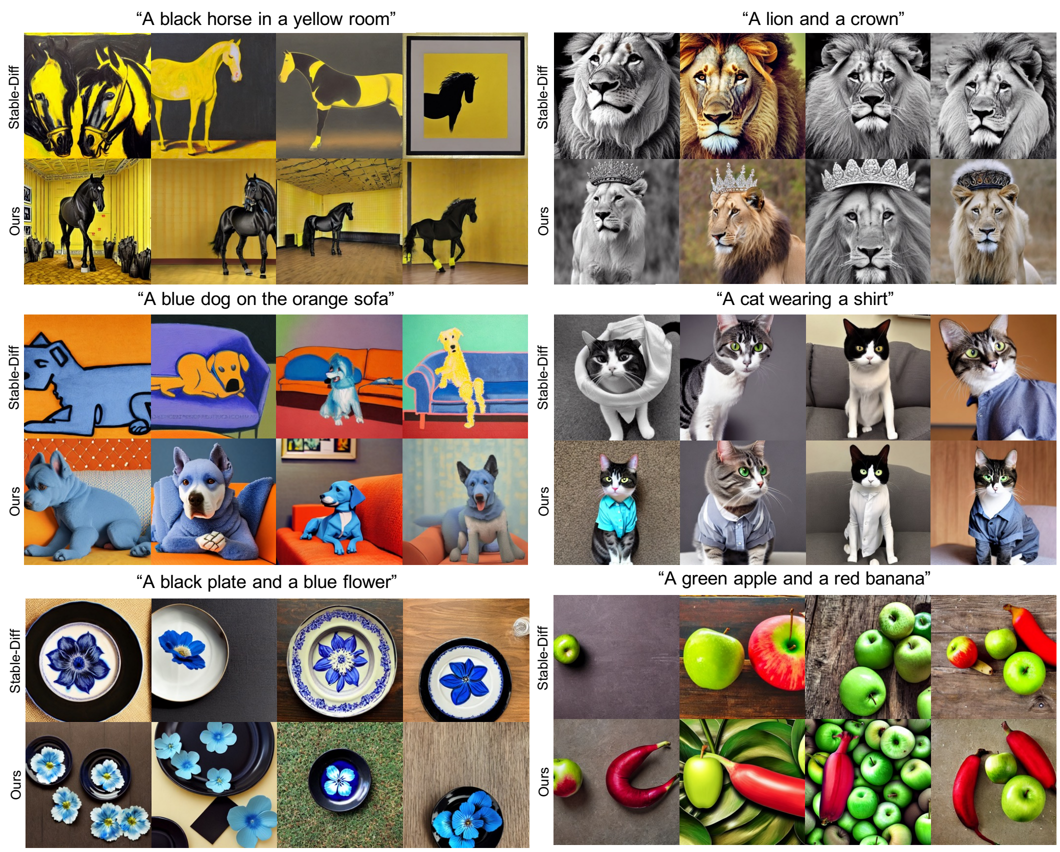

Figure 11: Further results for multi-concept image generation. Best views are displayed.

Figure 12: Further results for real image editing: horse to zebra.

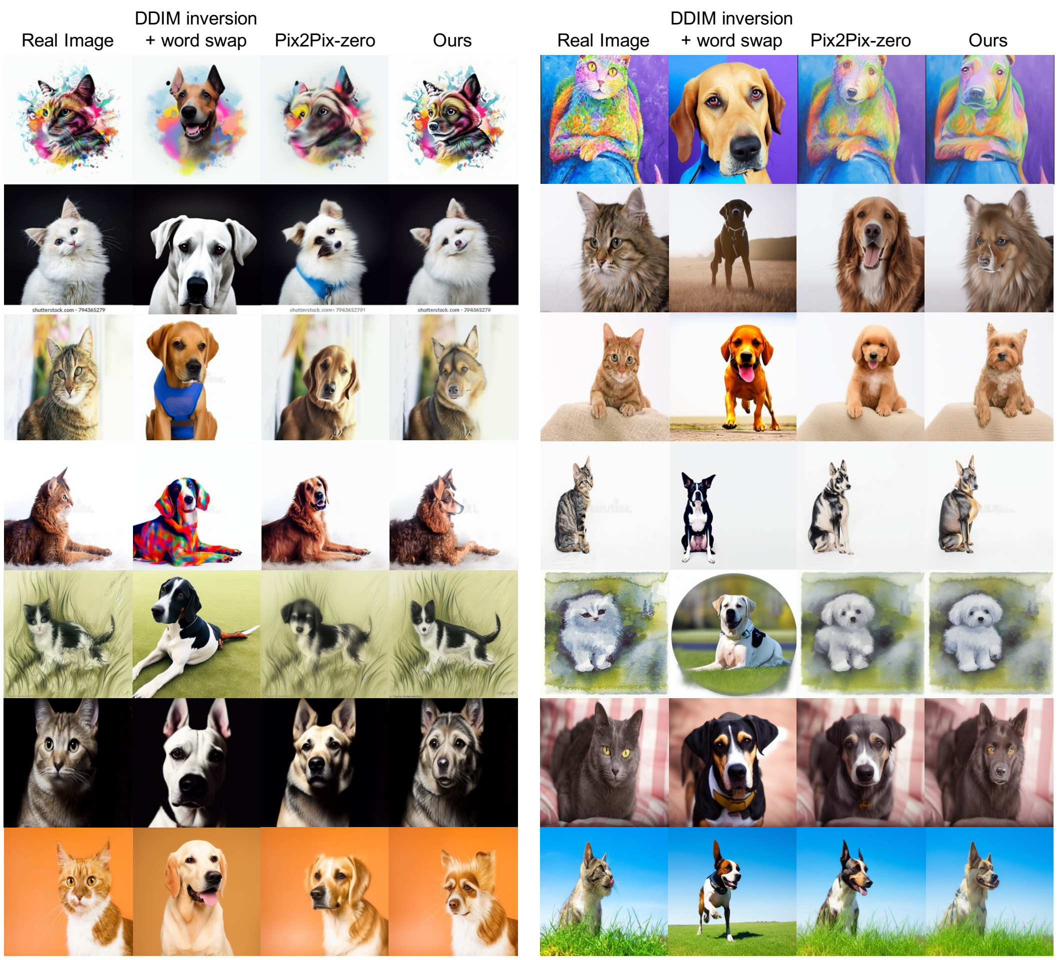

Figure 13: Further results for real image editing: cat to dog.

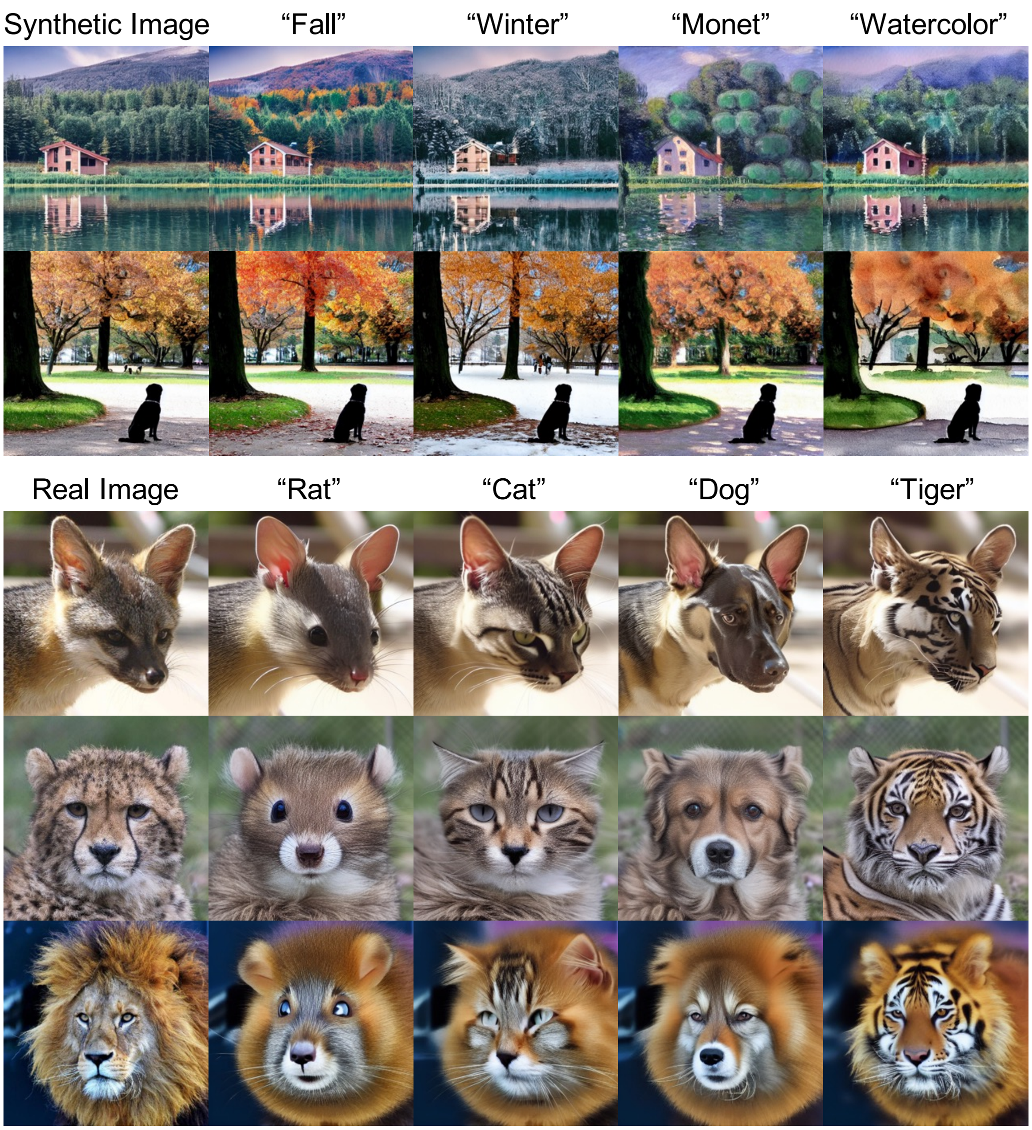

Figure 14: Further results for image editing with varying text prompts. Best views are displayed.

References

[1] M. Brack, F. Friedrich, D. Hintersdorf, L. Struppek, P. Schramowski, and K. Kersting. Sega: Instructing diffusion using semantic dimensions. *arXiv preprint arXiv:2301.12247*, 2023.

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[4] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *arXiv preprint arXiv:2301.13826*, 2023.

[5] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[6] M. Demircigil, J. Heusel, M. Löwe, S. Upgang, and F. Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168:288–299, 2017.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Y. Du, S. Li, and I. Mordatch. Compositional visual generation with energy based models. *Advances in Neural Information Processing Systems*, 33:6637–6647, 2020.

[9] Y. Du, S. Li, J. Tenenbaum, and I. Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.

[10] Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

[11] W. Feng, X. He, T.-J. Fu, V. Jampani, A. Akula, P. Narayana, S. Basu, X. E. Wang, and W. Y. Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *arXiv preprint arXiv:2212.05032*, 2022.

[12] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

[13] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[14] B. Hoover, Y. Liang, B. Pham, R. Panda, H. Strobelt, D. H. Chau, M. J. Zaki, and D. Krotov. Energy transformer. *arXiv preprint arXiv:2302.07253*, 2023.

[15] T. Karras, M. Aittala, T. Aila, and S. Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

[16] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[17] D. P. Kingma, T. Salimans, B. Poole, and J. Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

[18] D. Krotov and J. Hopfield. Dense associative memory is robust to adversarial inputs. *Neural computation*, 30(12):3151–3167, 2018.

[19] J. Li, D. Li, C. Xiong, and S. 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.

[20] L. Liu, Y. Ren, Z. Lin, and Z. Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

[21] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 423–439. Springer, 2022.

[22] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.

[23] C. Meng, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

[24] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or. Null-text inversion for editing real images using guided diffusion models. *arXiv preprint arXiv:2211.09794*, 2022.

[25] W. Nie, A. Vahdat, and A. Anandkumar. Controllable and compositional generation with latent-space energy-based models. *Advances in Neural Information Processing Systems*, 34:13497–13510, 2021.

[26] E. Nijkamp, M. Hill, S.-C. Zhu, and Y. N. Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. *Advances in Neural Information Processing Systems*, 32, 2019.

[27] G. Parmar, K. K. Singh, R. Zhang, Y. Li, J. Lu, and J.-Y. Zhu. Zero-shot image-to-image translation. *arXiv preprint arXiv:2302.03027*, 2023.

[28] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[29] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. K. Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

[30] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. 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.

[31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[32] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[33] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[34] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[35] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[36] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[37] N. Tumanyan, O. Bar-Tal, S. Bagon, and T. Dekel. Splicing vit features for semantic appearance transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10748–10757, 2022.

[38] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. *arXiv preprint arXiv:2211.12572*, 2022.

[39] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. *arXiv preprint arXiv:2212.05034*, 2022.

[40] A. L. Yuille and A. Rangarajan. The concave-convex procedure (cccp). *Advances in neural information processing systems*, 14, 2001.