PixelPonder: Dynamic Patch Adaptation for Enhanced Multi-Conditional Text-to-Image Generation

Yanjie Pan[1*] **Qingdong He**[2*] **Zhengkai Jiang**[2] **Pengcheng Xu**[3] **Chaoyi Wang**[4] **Jinlong Peng**[2] **Haoxuan Wang**[1] **Yun Cao**[2] **Zhenye Gan**[2] **Mingmin Chi**[1] **Bo Peng**[1†] **Yabiao Wang**[5,2†]

[1]Fudan University [2]Tencent Youtu Lab [3]Western University
[4]University of Chinese Academy of Sciences [5]Zhejiang University

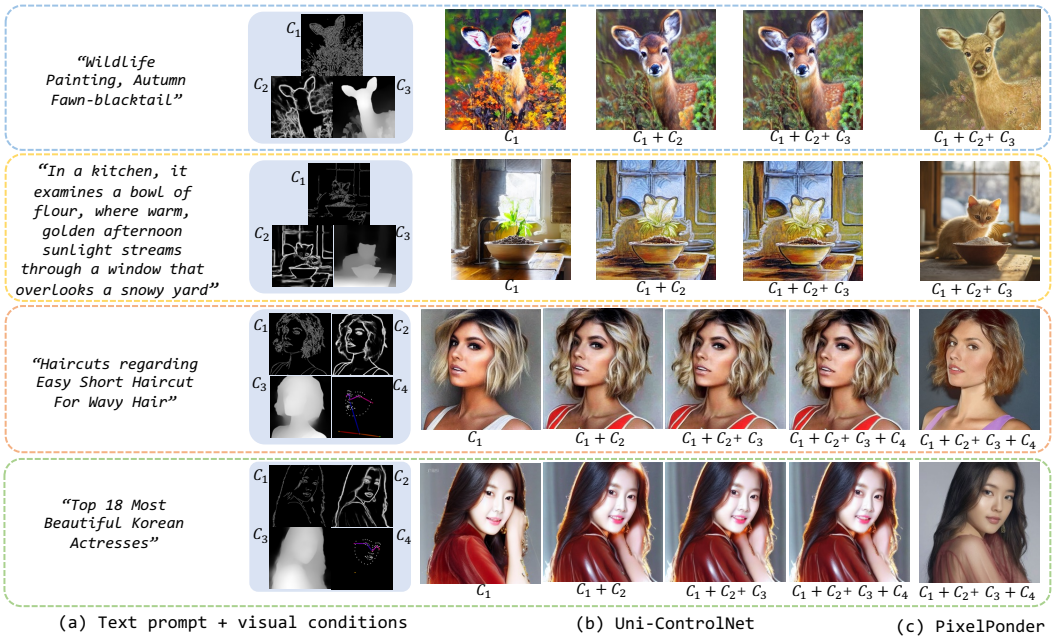

Figure 1: **Visualization comparison between Uni-ControlNet [51] and our proposed method in different conditional controls with the same text prompt.** ***(a, left two columns)*** Text and various visual controls, where C_1, C_2, C_3 and C_4 denotes canny, sketch, depth and pose map respectively. ***(b, middle three or four columns)*** Generation results from UniControlNet. ***(c, last column)*** Generation results from our PixelPonder. Previous methods struggled to generate coherent results under multiple conditions, while our results maintain strong similarity to the respective visual controls.

Abstract

Recent advances in diffusion-based text-to-image generation have demonstrated promising results through visual condition control. However, existing ControlNet-like methods struggle with compositional visual conditioning - simultaneously preserving semantic fidelity across multiple heterogeneous control signals while maintaining high visual quality, where they employ separate control branches

*Equal contributions.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

that often introduce redundant guidance during the denoising process, leading to structural distortions and artifacts in generated images. To address this issue, we present PixelPonder, a novel unified control framework, which allows for effective control of multiple visual conditions under a single control structure. Specifically, we design a patch-level adaptive condition selection mechanism that dynamically prioritizes spatially relevant control signals at the sub-region level, enabling precise local guidance without global interference. Additionally, a time-aware control injection scheme is deployed to modulate condition influence according to denoising timesteps, progressively transitioning from structural preservation to texture refinement and utilizing the control information from different categories to promote finer image generation. Extensive experiments demonstrate that PixelPonder surpasses previous methods across different benchmark datasets, showing superior improvement in spatial alignment accuracy while maintaining high textual semantic consistency. The code and models are available at https://hithqd.github.io/projects/PixelPonder/.

1 Introduction

The emergence of diffusion-based text-to-image synthesis [32, 9, 25, 26, 29, 30] have driven significant progress in single-modality conditioned generation frameworks [48, 21, 23, 4, 22]. While these approaches demonstrate enhanced controllability through visual-textual alignment, fundamental limitations persist due to modality-specific biases in visual conditioning. As evidenced in [35, 11], distinct visual control signals exhibit complementary characteristics, such as depth maps effectively regulate inter-object spatial relationships but lack fine-grained object details, whereas canny maps capture precise texture contours while disregarding global structural context. This modality specialization gap raises a critical challenge: ***How to establish an effective fusion paradigm for multi-visual condition control that synergistically integrates heterogeneous visual features while resolving inter-modality conflicts?***

In practical applications, it is often necessary to describe the visual features of a significant object using multiple visual conditions to achieve precise control over its generation. Users typically desire to simultaneously control both the layout and the intricate details. However, visual conditions that are rich in layout and detail information are challenging to integrate into a single visual condition map. A common scenario involves using canny maps to represent details and depth maps to control the layout. In such case, these visual conditions are highly overlapping and complementary to one another.

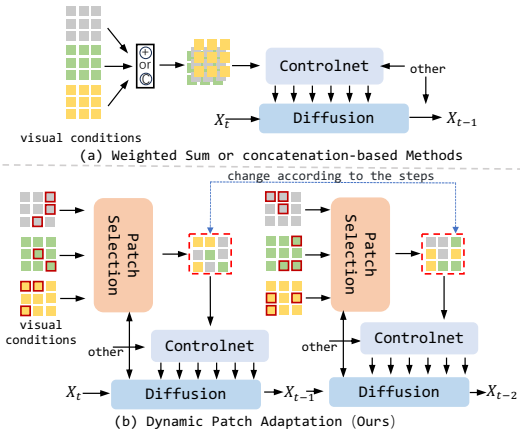

Figure 2: **Comparison of different schemes dealing with multiple condition. (a)** Existing methods integrate various visual controls into a highly consistent visual control signal in the temporal domain through concatenation or weighted summation. **(b)** We employ dynamic patch selection to reconstruct visual conditions at the patch level, enabling multi-visual controllable generation in the generative model.

Bridging the redundant information among these visual conditions and reasonably utilizing the emphasis of various visual conditions to coordinate the generation of the object is exactly the issue that this paper wants to focus on. As shown in Fig.2(a), prior efforts including Cocktail [16] and Uni-ControlNet [51] propose to consolidate various visual conditions into unified latent representations. As illustrated in Fig.1, generating normal images under multi-modal and coupled visual condition controls remains a difficult task for these approaches. While this coarse-grained unification alleviates inter-condition conflicts through dimensionality reduction, it remains suboptimal in preserving modality-specific attributes critical for precise control (e.g., high-frequency textures in cannys vs. low-frequency structures in depth). The inherent information bottleneck induced by

latent compression disproportionately attenuates complementary cross-modal features, fundamentally limiting their capacity to synergize discriminative visual cues. This constitutes a critical limitation: current unification paradigms prioritize conflict mitigation over inter-modal feature reinforcement.

To address the aforementioned challenges, we introduce an innovative visual condition combination method termed *PixelPonder*. As illustrated in Fig.2(b), our method distinguishes itself from previous approaches by integrating diverse visual conditions at the patch level. Recognizing that visual conditions enriched with low-frequency or high-frequency signals significantly influence various denoising stages during the image generation process [15], we have developed an adaptive patch selection mechanism that dynamically adjusts with timesteps. This mechanism decouples each visual condition in the temporal domain, enabling the use of distinct combinations of visual conditions to regulate the denoising process at different stages. Our proposed combination method amalgamates various visual conditions with greater granularity, and facilitating complementary control of high- and low-frequency information within the temporal domain.

It is worth noting that when operating under a single visual condition control, our proposed method effectively reverts to the single visual control scheme. The efficacy of ControlNet's single visual condition control has been extensively validated in numerous studies [49, 51, 27]. Therefore, this paper does not investigate the generative capabilities of our method under single visual condition scenarios. Our main contributions are summarized as follows:

- ***New Insight:*** We reveal that current multi-visual condition control methods perform poorly in terms of generation quality and controllability, making it difficult for generated images to align with various visual conditions and lacking effective solutions for improvement.
- ***Efficient Patch Adaptation:*** We propose PixelPonder, a novel mechanism which can refine the combination of multi-visual conditions by transitioning from the image level to the patch level, thereby enabling finer-grained controllable generation.
- ***Flexible Condition Combination:*** Our PixelPonder supports flexible visual control combination, addressing the redundant information conflict of multiple visual conditions for the same object.
- ***Promising Results:*** We provide a consolidated and public evaluation of controllability and fidelity under various conditional controls, and demonstrate that PixelPonder comprehensively outperforms existing methods.

2 Related Work

2.1 Diffusion-based Generative Models

Diffusion models [36, 15, 24, 26, 29, 5] have made remarkable strides in the field of image generation. Thanks to improvements in sampling strategies [15, 37, 38], as well as the application of latent diffusion methods [32], the computational resources required for these models to generate high-quality images have been substantially reduced, demonstrating promising prospects in various applications [4, 2, 12, 30]. In the domain of text-to-image generation, diffusion models [32] combined with U-Net [33] have successfully integrated text control into the image generation process through cross-attention mechanisms. To further enhance generative capabilities, the Transformer architecture has been integrated into diffusion models [25, 6, 10]. Building on this foundation, FLUX [20], which introduces flow matching objectives, has achieved state-of-the-art performance in image generation.

2.2 Controllable Text-to-Image Diffusion Models

The generation of images controlled by textual descriptions often lacks spatial control. To enhance spatial control capabilities, numerous studies [23, 49, 22, 17, 8, 46, 50] have focused on incorporating visual control signals into the generation process. In this context, ControlNet [49] proposed a spatial control network architecture, encoding additional visual conditions into latent representations and injecting control signals into the corresponding backbone network using zero convolution, thus achieving a spatial alignment mechanism within diffusion models. Due to its straightforward approach and effective control results, it has been widely adopted in various text-to-image diffusion models [32, 20]. Furthermore, recent research has used prompt engineering techniques [22, 46] and inter-attention constraints [7, 44] to facilitate more orderly generation.

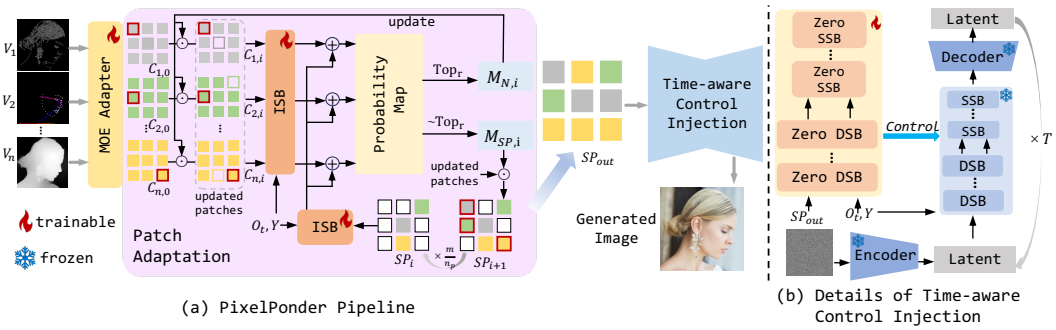

Figure 3: **Overall pipeline of the proposed**. The Patch Adaptaiom Module segments various visual conditions $\{V_1, V_2, \ldots, V_n\}$ into patches and dynamiclly reorganizes the visual conditions at the patch level to obtain a unified visual control signal SP_{out}. Subsequently, SP_{out} is sent to the control network, with the Time-aware Control Injection scheme to achieve high-quality image generation under multiple visual conditions.

Some studies [27, 23, 42, 1] have addressed the differences and diversity of various visual conditions in different modalities using the mix of experts (MOE) to decouple different visual conditions into distinct feature subspaces, thus establishing a unified visual control architecture. Subsequently, other research has explored the possibility of generating multiple condition controls within a single diffusion model through simple feature aggregation methods [16, 51]. However, these methods are based on the low coupling of various conditions, which limits their effectiveness in tasks requiring the control of complementary conditions.

3 Methodology

In this paper, we introduce PixelPonder, a specialized framework for handling complex scenarios of image generation under multiple visual conditions. The pipeline of PixelPonder is demonstrated in Fig. 3. In this section, we first provide a brief overview of the previous Diffusion Transformer in Section 3.1, along with ControlNet. Then, in Section 3.2, we dive into a detailed description of the Patch Adaptation Module, which primarily focuses on the autoregressive reorganization of visual conditions at the patch level. In Section 3.3, we elaborate on the time-step awareness injection strategy and the control implementation of the control network. Finally, we summarize our training optimization objectives in Section 3.4.

3.1 Preliminary

Diffusion Transformer (DiT)[41, 25] is applied in models such as FLUX [20] and Stable Diffusion 3 [10]. Compared to the Latent Diffusion Model [32], these models utilize a transformer [34] instead of U-Net as the denoising network to iteratively process noise tokens.

A model with a DiT architecture handles noisy image tokens $X \in \mathbb{R}^{N \times d}$ and text condition tokens $Y \in \mathbb{R}^{M \times d}$, where d is the dimension of the embedding layer, N and M represent the number of tokens for images and text, respectively. Additionally, they redefine the mapping from images to noise using flow matching. In this rectified mapping, the data at time t, denoted as z_t, is transformed from the real image $x_0 \in \mathbb{R}^{C \times H \times W}$ to pure noise $N \sim \mathcal{N}(0, I)$ at the following position:

$$z_t = (1-t)x_0 + tN, \tag{1}$$

$$\vec{V} : \mathcal{N} \rightarrow \mathcal{Z}\,, \tag{2}$$

where $\mathcal{N}$ represents the spatial distribution of noisy images. $\mathcal{Z}$ represents the spatial distribution of real images. $\vec{V}$ is the vector field from $\mathcal{N}$ to $\mathcal{Z}$.

In FLUX, a series of flow matching modules are included, among which the denoising backbone network consists of the Double Stream Block (DSB) and the Single Stream Block (SSB).

ControlNet [49] is designed to inject visual control information into a denoising network, thereby guiding the iterative process of noise tokens. Specifically, ControlNet freezes the parameters Θ of

the backbone network $F(\cdot;\Theta)$ and creates a trainable copy Θ_c of these parameters to incorporate zero-initialized convolutional layers Z. When applied to generative models based on the U-Net architecture, a common practice is to add the visual condition C_i to the noise x_t as input to the trainable copy, while utilizing other signals s that is consistent with the backbone network. This process can be formalized as follows:

$$y = F(x_t, s, \Theta) + Z(F(x_t + M(C_1, \ldots, C_n), s, \Theta_c)), \tag{3}$$

where M serves as a unified representation of various visual condition fusion methods, n represents the category id.

3.2 Patch Adaptation

The Patch Adaptation Module (PAM) aims to recompose various visual conditions at the patch level into a unified visual condition. This is achieved through an autoregressive iterative combination process that combines patches between different visual conditions. The specific algorithm flow of PAM is shown in algorithm 1.

Specifically, at time step t, the PAM initializes a zero vector unified control condition $SP_0 \in \mathbb{R}^{N\times d}$ to obtain the final unified visual control combination output SP_{out}. The combination selection is defined as an update process U_i from SP_i to SP_{i+1}, with i representing the index of the number of iterations in the update process. This iterative update process can be formalized as follows:

$$SP_{i+1} = U(SP_i, \ldots, O_t), \tag{4}$$

where O_t refers to the time step information, and its specific role will be discussed in next subsection.

Algorithm 1 Patch Adaptation

Input: visual condition features $C_n \in \mathbb{R}^{N\times d}$, time step information O_t and textual feature Y.
Output: the unified visual control SP_{out}

$SP_0 = \mathbf{0}_{N\times d}$
$C_n = [P_n^0, \ldots, P_n^m]$
$\mathcal{C}_0 = \{C_1, C_2, \ldots, C_n\}$
$i \leftarrow 0$
$n_p \leftarrow$ number of selected patches
while $i \cdot n_p < m$ **do**
  $M_{\text{prob}} = [\text{ISB}_{\text{C}}(\mathcal{C}_i, O_t, Y)] + \text{ISB}_{\text{SP}}(SP_i, O_t, Y)$
  $M_{N,i} = \text{Zero}(\text{Top}_r(M_{prob}))$, $M_{sp,i} = \sim M_{N,i}$
  $\mathcal{C}_{i+1} = \mathcal{C}_i M_{N,i}$
  $SP_{i+1} = SP_i + M_{sp,i}\mathcal{C}_i$
  $i \leftarrow i + 1$
end while
return SP_i

Assuming that the visual input to the PAM consists of N types of condition images, denoted as $V_1, V_2, \ldots, V_n$, we use the corresponding visual condition feature encoders E_i to obtain various visual condition features $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, $C_k \in \mathbb{R}^{e\times d}$ and d denotes the feature dimension. To analyze and process the visual condition features in a more fine-grained manner, we can consider the visual condition feature C_k as represented by patches $P_i^m \in \mathbb{R}^d$, expressed as:

$$C_k = [P_k^0, \ldots, P_k^m], \tag{5}$$

where P refers to a patch, k denotes the random category id, and m denotes the number of patches.

In an update process U_i, the PAM selects a set $\mathcal{P}_i = \{P_{n,i}^s \mid s \in [0, m], k \in [0, n]\}$ composed of multiple patches $P_{n,i}^s$ from various visual characteristics $\mathcal{C}_i$. For convenience, we define this selection process as F_{sp}. The composition of this set $\mathcal{P}_i$ considers two aspects of information: ***a) The dominant role of textual information*** Y ***in the current time step in image generation*** , and ***b) The influence of previously selected patches*** $\mathcal{P}_0, \ldots, \mathcal{P}_{i-1}$ ***on the overall composition of image control.*** The update process from SP_i to SP_{i+1} can be more specifically expressed as:

$$U_i(SP_i, \ldots, O_t) = SP_i + \mathcal{P}_i, \tag{6}$$

$$\mathcal{P}_i = F_{\text{sp}}(\mathcal{C}_i, \mathcal{P}_0, \ldots, \mathcal{P}_{i-1}, O_t, Y). \tag{7}$$

Specifically, to achieve F_{sp}, we design the Image Stream Block (**ISB**). To ensure that the textual control signal remains consistent globally, we change the DSB text input selection strategy, abandoning the textual information Y_{DSB} generated in dual-stream matching as input and instead using a globally unified textual feature representation Y as the input textual information. We modified the DSB text input selection strategy to adopt a globally unified textual feature representation Y as the input textual information. The ISB is used to obtain the weights of each patch in various visual condition features

under the current textual control Y at time step t, denoted as $\mathcal{W}_c = \{W_{0,i}, \ldots, W_{n,i}\}$. Furthermore, to meet requirement ***b)*** of F_{sp}, we apply an ISB to SP_i outside of $\mathcal{C}_i$ to obtain a global weight W_{sp} that represents the global composition information of SP_i under the current U_i. During training, the Gumbel-Softmax [18] is adopted to ensure the differentiability of the Top-r selection operation. The final obtained patch selection probability map M_{prob} is given by:

$$M_{\text{prob}} = [\mathcal{W}_c] + W_{\text{sp}}, \tag{8}$$

$$\mathcal{W}_c = \text{ISB}_{\text{C}}(\mathcal{C}_i, O_t, Y), \quad W_{\text{sp}} = \text{ISB}_{\text{SP}}(SP_i, O_t, Y). \tag{9}$$

In each iteration, to mask the influence of previously selected patch blocks $\{P_0, \ldots, P_{i-1}\}$ on the combination selection process U_i, we use a binary mask $M_{k,i}$ to update C_k^{i-1} to obtain C_k^i. The binary mask $M_{k,i}$ is derived from M_{prob}:

$$M_{N,i} = \text{Zero}(\text{Top}_r(M_{prob})), \tag{10}$$

$$[\bigcup_{k=1}^{N} M_{k,i}] = M_{N,i}, \tag{11}$$

where Top_r retrieves the positions of the top r patches with the highest probabilities in M_{prob}, and Zero sets the corresponding positions in the binary mask to zero.

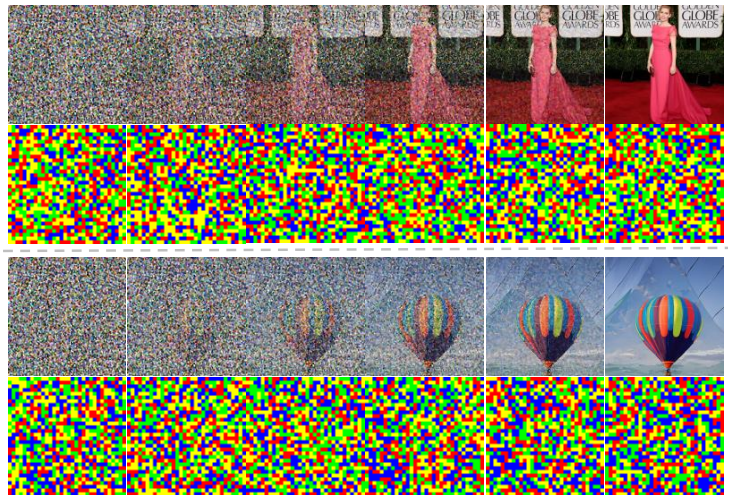

Figure 4: **Denoising results at steps 25, 20, 15, 10, 5, and 1**, where each pair includes the generated result and the selection results SP_{out} at the current time step. In each SP_{out}, canny, pose, depth and sketch are indicated in red, blue, green, and yellow respectively.

Fig.4 illustrates the visualization results of SP_{out} obtained by PAM in different time steps, as well as the generated images. In the denoising process, the generated images exhibit a noticeable tendency towards patchiness. The segmentation contours between the generated image patches are prominent. Both examples demonstrate the characteristics of PAM: a) PAM exhibits a balanced tendency in its selections over various time steps. b) For patches at the same location, the global selections include patches from all categories.

It is noted that C_n^0 is equivalent to C_n, which means that we do not process C_n in U_0. The representation of the visual condition C_n during the iteration process is as follows:

$$C_k^i = \begin{cases} \mathbf{E}_k(V_k) & \text{if } i = 0 \\ M_{k,i} C_k^{i-1} & \text{if } i > 0. \end{cases} \tag{12}$$

While obtaining $M_{N,i}$, we can also acquire the mask $M_{sp,i}$ corresponding to the selected patch P_i during the update process U_i:

$$M_{sp,i} = \sim M_{N,i}, \mathcal{P}_i = M_{sp,i} \mathcal{C}_i, \tag{13}$$

where $\sim$ represents the negation operation.

3.3 Time-aware Control Injection

The time-aware control injection is designed to fully utilize the visual control effects of various visual conditions at different time steps during the denoising process. This is achieved by introducing additional temporal information O_t in the PSM at each time step, which controls the patch selection tendency in each update process U. To ensure that the patch selection tendencies in all selection processes U within a time step are consistently influenced by the temporal information, we use a unified temporal signal O_t at the current time step for control, as shown in Eq. 4.



Figure 5: **Visualization comparison** between official methods and our proposed model in different conditional controls.

Specifically, we introduce the time-step information O_t into the visual conditions and the global ISB module. This allows the ISB to simultaneously acquire time-step control information to obtain patch selection weights that are more aligned with the denoising temporal flow, which can be formalized as follows:

$$Control = D'(SP_{out}, Y, O_t, \Theta_c), \tag{14}$$

$$I_r = D(N, Control, Y, \Theta), \tag{15}$$

where D denotes the original model, Θ indicates the model parameters, D' refers to the control network, and Θ_c signifies the model parameters of the control network, I_r represents the generated image. Considering the large number of parameters in the model, we reduce the number of modules in the control network and align the image streams sequentially to the corresponding backbone modules.

To achieve stable control of visual composition under time awareness, we initialize the last linear layer to zero from the ControlNet and FLUX, denoted as zero-initialized Double Stream Block (**Zero DSB**) and zero-initialized Single Stream Block (**Zero SSB**), which can be formulated as:

$$Attention(Q, K, V) = Z((\frac{QK^T}{\sqrt{d_k}}), V). \tag{16}$$

3.4 Training

Objective. Given a set of multi-visual prompts $\mathcal{C}$ and a text prompt Y, we fine-tune our method to reconstruct the original image x. During the training process, we freeze the backbone network and only optimize the patch selection network and the zero controllable network. Our training objective combines V and V' to obtain the visual representation $V_{n\to c}$ under the visual prompts C. This encourages the visual control network to learn the internal visual condition control flow V' based on the V pre-learned by the backbone network. The loss function $\mathcal{L}$ is formalized as:

$$\mathcal{L} = \|(N - x_0) - \int_0^1 \vec{V} \oplus \vec{V}'(z_t, \mathcal{C}, Y, t)dt\|^2, \tag{17}$$

where the $\oplus$ represents the vector sum of two fields.

Table 1: Comparison on the MultiGen-20M [21] and Subject-200K [40] datasets. The best and second-best results are highlighted in **bold** and underline, respectively. In the **Conditions** column, "all" denotes the use of canny maps, sketches, pose maps, and depth maps simultaneously during generation.

Methods	Conditions	MultiGen-20M				Subject-200K				Inference Time (↓)
		FID (↓)	CLIP Score (↑)	SSIM (↑)	MUSIQ (↑)	FID (↓)	CLIP Score (↑)	SSIM (↑)	MUSIQ (↑)	
T2I-Adapter	all	66.95	71.47	24.03	57.95	64.72	74.35	31.76	55.07	9.33
Uni-ControlNet	all	32.58	78.08	29.37	65.85	44.35	77.40	37.98	66.75	7.22
UniControl	all	25.15	74.09	35.58	**72.05**	30.95	72.96	47.31	**67.50**	7.39
Cocktail	pose+hed	24.67	**79.87**	24.14	67.13	51.51	**79.38**	29.59	64.16	4.94
Ours	all	**11.85**	78.60	**43.99**	69.54	**10.61**	77.12	**66.60**	67.32	9.59

4 Experiments

We validate the effectiveness of PixelPonder on the MultiGen-20M [21] and Subject-200K[40] datasets. Our evaluation primarily focuses on several leading methods, including T2I-Adapter [23], Uni-ControlNet [51], UniControl [27], and Cocktail [16] in the realm of multiple controllable text-to-image diffusion models. It is worth noting that we conducted experiments on Cocktail using only pose and sketch maps, as it only supports pose, sketch, and segmentation maps. Although the models of other approaches such as AnyControl [39] are public, their code cannot be successfully run after many attempts. More implementation details can be found in the supplementary material.

4.1 Main Results

Comparison of Controllability. As shown in Tab.1, we report the controllability comparison results across different datasets, where the structural similarity index (SSIM) is employed to evaluate image consistency. Our PixelPonder surpasses other methods with improvements of 8.41% and 19.29% on two benchmark datasets, establishing a new state-of-the-art in image generation. This notable accomplishment demonstrates that PixelPonder is capable of processing intricate combinations of multiple spatial conditions, generating images that are highly consistent with the given spatial constraints.

Comparison of Image Quality. Previous methods often encounter two primary issues when handling multiple visual control conditions: a) Coarsening of lines within the image, and b) Disappearance or distortion of the main subject. In contrast, PixelPonder, leveraging its unique architectural design, effectively avoids these issues by harmonizing multiple visual control conditions without compromising image quality. This is demonstrated by the quantitative results in Tab.1, including significant improvements in FID (12.82 on MutiGen-20M and 20.34 on Subject-200K) and leading MUSIQ scores, and qualitative comparison in Fig.5 further supports the effectiveness of PixelPonder in handling complex visual conditions.

Comparison of CLIP Score. Tab.1 compares CLIP Score metric across methods under maximal conditions. The results reveal a consistent trade-off: methods with fewer visual constraints achieve higher text adherence. Specifically, PixelPonder demonstrates superior performance under full conditional control, outperforming the compared methods. Although Cocktail attains higher scores under significantly fewer visual constraints, the strong text adherence of Cocktail significantly compromises the consistency of its visual performance. As illustrated in Fig.5, the images generated by Cocktail exhibit pronounced issues of object loss and mutations when

Table 2: **Results of combining different condition types.** A quantitative comparison with models controlled by single visual conditions and various combinations of visual conditions. "Source" refers to the original reference image.

Methods	Conditions	MultiGen-20M		
		FID (↓)	SSIM (↑)	MUSIQ (↑)
Source	-	-	-	69.30
ControlNet++	canny	17.69	36.69	65.67
ControlNet++	hed	13.93	42.12	71.22
ControlNet++	depth	17.56	27.79	71.23
Ours	pose	30.00	27.91	58.16
Ours	pose+depth	19.76	31.63	62.63
Ours	pose+depth+hed	12.41	40.48	67.82
Ours	all	11.85	43.99	69.54



Figure 6: **Comparison of multi-condition image generation** by PixelPonder with different combinations against single-condition image generation by ControlNet++.

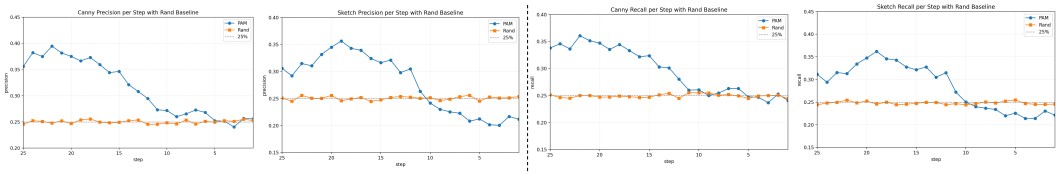

Figure 7: **Comparison of precision and recall** of PAM and random scheme on canny and sketch conditions. Taking Canny and Sketch as examples.

Table 3: **Ablation on zero controllable flow.** × represents the non-use of zero controllable flow, while the ✓ represents the use of zero controllable flow.

Zero	MultiGen-20M		
	FID (↓)	SSIM (↑)	MUSIQ (↑)
×	11.26	43.21	69.15
✓	11.85	43.99	69.54

Table 4: **Ablation on timestep information.** × represents the non-use of timestep information, while the ✓ represents the use of timestep information.

Timestep	MultiGen-20M		
	FID (↓)	SSIM (↑)	MUSIQ (↑)
×	29.53	28.55	67.21
✓	11.85	43.99	69.54

compared to those produced by other methods. This highlights PixelPonder's unique capability to balance precise visual control with competitive text adherence.

Comparison of Inference Speed. Tab.1 presents the inference speed. Due to the fact that Cocktail involves only half the number of conditions compared to other methods, any direct comparison of time efficiency may not be entirely representative or meaningful. Patch Adaptation introduces a marginal increase in generation time; however, given the substantial improvement in image quality, the associated computational overhead is acceptable.

Condition Types Comparison. To examine the effect of the number of condition types, Fig.6 shows the visual results under various condition combinations, with qualitative evaluation metrics provided in Tab.2. Examples in Fig.6 demonstrate that as the number of conditions increases, the layout and texture within images become increasingly refined and accurate, demonstrating that the control effects of different visual conditions are not identical. Furthermore, as can be seen from Tab.2, an increase in visual conditions is beneficial for the overall quality and controllability of the images, which aligns with the observed visual effects. As evidenced by the comparison with ControlNet++ [21], integrating multi-conditions improves control precision and enables finer-grained manipulation of the generated output.

4.2 Ablation Study

Patch Selection. In Fig. 8, we illustrate the statistical distribution of each visual condition patch during the denoising steps. During the initial denoising stages from step 0 to 5, the denoising process comes from low-frequency regions to high-frequency regions. However, from step 5 to 25, the trend is completely opposite. This suggests that after establishing a rough layout in the initial stages, generating low-frequency or high-frequency regions does not necessarily require consistent information with the layout. On the contrary, it requires visual information from other frequencies to leave sufficient space to meet other visual requirements. More discussion is shown in the supplementary material.

Spatial Correlation Analysis for Patch Selection Interpretability. To quantitatively evaluate how well PAM's selection aligns with local spatial energy, we constructed modality-specific importance reference maps. Taking Canny and Sketch as examples, (1) canny conditions: We computed gradient magnitude maps via Sobel operators (5,500 samples from MultiGen-20M). Images were partitioned into a patch grid matching PAM's dimensions. After applying a global threshold, the top 25% of patches with the highest strong-edge pixel density were labeled as High-Density Edge Regions (Ground Truth), (2) sketch conditions: Similarly, importance was defined by line density (ratio of non-zero pixels). The top 25% densest patches were designated as the importance ground truth. As illustrated in Fig. 7, the random baseline maintains a stable precision of approximately 25% across all timesteps, which is expected given the uniform random selection across four condition types. In contrast, both Canny and Sketch exhibit a clear temporal evolution trend. This non-random, time-aware pattern demonstrates that PAM's routing mechanism is not arbitrary. Its early-stage selection explicitly prioritizes spatially important regions, contributing to its empirical advantages over the random baseline.

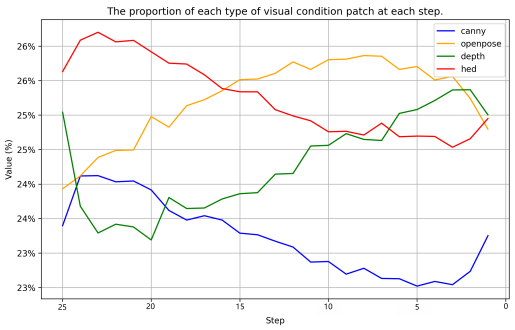

Figure 8: **Statistical distribution of each visual condition patch** during the denoising steps.

Zero Controllable Flow. As shown in Tab.3, the use of zero controllable flow significantly enhances the aesthetic quality and color richness of the images. However, this comes with a slight loss of feature consistency in the images, while maintaining better visual coherence.

Timestep Information Injection. As shown in Tab.4, controllable generation without timestep guidance exhibits a pronounced degradation in FID and SSIM, compared to generation with timestep guidance. This confirms that timestep information is essential for achieving effective structural controllability in generated images.

5 Conclusion

In this paper, we propose PixelPonder, a framework for compositional visual conditioning in diffusion models that addresses guidance redundancy in multi-control generation. Our key contributions include: a Patch-level Adaptive Condition Adaptation mechanism to resolve spatial conflicts via learnable attention gates, and a Time-aware Control Injection scheme to orchestrate condition influence across denoising phases. Experiments show significant improvements over state-of-the-art methods, enabling users to achieve precise, multi-aspect object creation through diverse visual conditions. With PixelPonder, users can depict different aspects of objects by utilizing various visual conditions, thereby realizing their creations.

References

[1] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023.

[2] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[4] P. Cao, F. Zhou, Q. Song, and L. Yang. Controllable generation with text-to-image diffusion models: A survey. *arXiv preprint arXiv:2403.04279*, 2024.

[5] J. Chen, J. Yu, C. Ge, L. Yao, E. Xie, Y. Wu, Z. Wang, J. Kwok, P. Luo, H. Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.

[6] J. Chen, Y. Wu, S. Luo, E. Xie, S. Paul, P. Luo, H. Zhao, and Z. Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024.

[7] M. Chen, I. Laina, and A. Vedaldi. Training-free layout control with cross-attention guidance. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5343–5353, 2024.

[8] J. Cheng, X. Liang, X. Shi, T. He, T. Xiao, and M. Li. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023.

[9] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[10] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

[11] Q. He, J. Peng, P. Xu, B. Jiang, X. Hu, D. Luo, Y. Liu, Y. Wang, C. Wang, X. Li, et al. Dynamiccontrol: Adaptive condition selection for improved text-to-image generation. *arXiv preprint arXiv:2412.03255*, 2024.

[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. Hessel, A. Holtzman, M. Forbes, R. L. Bras, and Y. Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[14] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[15] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[16] M. Hu, J. Zheng, D. Liu, C. Zheng, C. Wang, D. Tao, and T.-J. Cham. Cocktail: Mixing multi-modality control for text-conditional image generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[17] L. Huang, D. Chen, Y. Liu, Y. Shen, D. Zhao, and J. Zhou. Composer: Creative and controllable image synthesis with composable conditions. *arXiv preprint arXiv:2302.09778*, 2023.

[18] E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[19] J. Ke, Q. Wang, Y. Wang, P. Milanfar, and F. Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157, 2021.

[20] B. F. Labs. Flux: Official inference repository for flux.1models accessed:2024-11-12. 2024.

[21] M. Li, T. Yang, H. Kuang, J. Wu, Z. Wang, X. Xiao, and C. Chen. Controlnet ++ : Improving conditional controls with efficient consistency feedback. In *European Conference on Computer Vision*, pages 129–147. Springer, 2025.

[22] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22511–22521, 2023.

[23] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, and Y. Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4296–4304, 2024.

[24] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[25] W. Peebles and S. Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[26] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[27] C. Qin, S. Zhang, N. Yu, Y. Feng, X. Yang, Y. Zhou, H. Wang, J. C. Niebles, C. Xiong, S. Savarese, et al. Unicontrol: A unified diffusion model for controllable visual generation in the wild. *arXiv preprint arXiv:2305.11147*, 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] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[30] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

[31] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

[32] 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.

[33] 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.

[34] L. Ruan, Y. Ma, H. Yang, H. He, B. Liu, J. Fu, N. J. Yuan, Q. Jin, and B. Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10219–10228, 2023.

[35] C. Si, Z. Huang, Y. Jiang, and Z. Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2024.

[36] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

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

[38] 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.

[39] Y. Sun, Y. Liu, Y. Tang, W. Pei, and K. Chen. Anycontrol: Create your artwork with versatile control on text-to-image generation. *arXiv preprint arXiv:2406.18958*, 2024.

[40] Z. Tan, S. Liu, X. Yang, Q. Xue, and X. Wang. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*, 2024.

[41] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[42] X. Wang, T. Darrell, S. S. Rambhatla, R. Girdhar, and I. Misra. Instancediffusion: Instance-level control for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6232–6242, 2024.

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[44] J. Xie, Y. Li, Y. Huang, H. Liu, W. Zhang, Y. Zheng, and M. Z. Shou. Boxdiff: Text-to-image synthesis with training-free box-constrained diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7452–7461, 2023.

[45] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[46] Z. Yang, J. Wang, Z. Gan, L. Li, K. Lin, C. Wu, N. Duan, Z. Liu, C. Liu, M. Zeng, et al. Reco: Region-controlled text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14246–14255, 2023.

[47] Z. Yang, A. Zeng, C. Yuan, and Y. Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023.

[48] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[49] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[50] T. Zhang, Y. Zhang, V. Vineet, N. Joshi, and X. Wang. Controllable text-to-image generation with gpt-4. *arXiv preprint arXiv:2305.18583*, 2023.

[51] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Appendix

A Reproducibility & Impact Statement

We have already elaborated on all the models or algorithms proposed, experimental configurations, and benchmarks used in the experiments in the main body or appendix of this paper. Furthermore, we declare that the entire code used in this work will be released after acceptance. Moreover, we plan to make the dataset and associated code publicly available for research. Nonetheless, we acknowledge the potential for misuse, particularly by those aiming to generate misinformation using our methodology. We will release our code under an open-source license with explicit stipulations to mitigate this risk.

B The Use of Large Language Models

We use large language models solely for polishing our writing, and we have conducted a careful check, taking full responsibility for all content in this work.

C Implementation Details

Dataset. Our training dataset is randomly sampled from the MultiGen-20M [21] dataset, resulting in a training set of 2.5 million samples. Each training sample contains an input image x, a text prompt, and a set of visual conditions $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$. This set $\mathcal{C}$ of visual conditions includes various types of visual cues, such as canny maps, sketches, depth maps, and pose maps, obtained from several other methods [47, 31, 3, 45]. In the testing dataset, due to the small size of the MultiGen-20M test set, and to ensure fairness in comparison, we combine the test and validation sets of MultiGen-20M to form a test set of up to 5,500 text-image pairs. Additionally, to increase the confidence in our conclusions, we randomly sample 5,000 text-image pairs from the Subject-200K[40] dataset as a second test set. This dataset provides more complex text compared to MultiGen-20M, and the Subject-200Ks in each image are clearly defined.

Training Setting. We conduct the training on 8 GPUs with 96GB of VRAM, using a batch size of 4 for every GPU, for a total of 200,000 training steps. The AdamW optimizer is employed with a learning rate of 2×10^{-5}. The default setting for patch size is 2, and the default number of patches selected per U_i is set to $\frac{W}{\text{patch size}}$ (where W is the width of the image features). The default depth of the ISB module is 1, and the default number of Zero DSBs and Zero SSBs is set to 2 and 4, respectively. During training, we randomly select 3 out of 4 types of conditions (canny maps, depth maps, sketch maps, and pose maps) as visual control inputs. Regarding the pose condition, since not all images contain humans, we do not discard the pose condition to ensure its effective training.

Inference Setting. During the inference process, the sampling step for PixelPonder is set to 25. For the comparative experiments with other methods, such as UnicontrolNet and Cocktail, the hyperparameters during inference is set to the official default configurations. In all visual displays except for the multiple generated images of the same object, the global random seed is set to 100. It is worth mentioning that assuming the visual conditions consist of $H \times W$ tokens, the number of patches selected in each update process U_i is W, and the total number of update processes is H.

Evaluation Metrics. For multi-visual condition control tasks, we evaluate generation quality, controllability, and text-image consistency using a comprehensive set of mathematical and model-based metrics. **Generation quality** is assessed through **FID**[14] and **MUSIQ** [19]. FID, a model-based metric, quantifies the similarity between feature distributions of real and generated images, providing a robust measure of image fidelity. MUSIQ, another perceptual model-based metric, evaluates image quality from a human perceptual perspective by capturing both local and global quality attributes. **Controllability** is measured using **SSIM** [43], a mathematically grounded metric that computes the structural similarity between reference and generated images, offering a precise assessment of visual fidelity. The evaluation metric for **inference speed** is the time cost per image, measured in seconds per image (s/per image). Finally, **text-image consistency** is evaluated via **CLIP Score** [13, 28], a model-based metric that measures the semantic alignment between input text and generated images.

Table A1: **Ablation on patch size.** We tested the impact of patch size in Patch Selection on controlling the generation.

Patch Size	MultiGen-20M			
	FID (↓)	CLIP Score (↑)	SSIM (↑)	MUSIQ (↑)
2	11.26	78.62	43.21	69.15
4	12.80	78.24	40.73	68.43
8	13.46	78.07	39.75	68.15

D Patch Size

Tab.A1 presents the performance of PixelPonder under different patch sizes. The experimental results indicate that finer-grained image condition combination can better resolve conflicts among various visual elements in the visual conditions.

E Scale Study

To investigate whether PixelPonder demonstrates better image quality with larger parameter scales and training datasets, we evaluated the fidelity and consistency of the generated images using FID, SSIM and MUSIQ on different data scales and module depths.

Data Scale. We evaluate the performance of PixelPonder on different scales of training datasets, with the results shown in Tab.A2. Due to limitations in computational resources and time, we only explored up to 2.5 million data. As shown in the table, PixelPonder demonstrates significant improvements in FID and SSIM as the scale of the training dataset increases. However, MUSIQ, constrained by the original reference images, does not show further improvement after a substantial increase in the size of the data set. Compared to other methods, such as ControlNet++ and UniControl, our method achieves results comparable to those obtained by tens of millions of training samples, while being trained on only millions of samples. This indicates that there is still room for improvement in the precise control capabilities of PixelPonder.

Table A2: **Ablation on data scale.** FID, SSIM and MUSIQ are reported on MultiGen-20M datasets. k represents thousand, and m represents million.

Data scale	MultiGen-20M		
	FID (↓)	SSIM (↑)	MUSIQ (↑)
105k	12.20	40.79	68.93
686k	11.43	41.13	69.16
2.5m	11.26	43.21	69.15

F Patch Selection Analysis

To further explore the specific internal processes of patch selection, we conduct some investigations, including the selection tendencies of patch selection for various visual conditions, as well as the characteristics of the denoising process based on patch selection.

Previous methods exhibit a progression of features from coarse to fine during the denoising process. Specifically, generative models first create the rough outlines and layouts of various objects, followed by the addition of details, which means generating from low-frequency regions to high-frequency regions. Generally, we believe that generating low-frequency visual elements requires low-frequency information, while high-frequency visual elements require high-frequency information.

To further validate our idea, we design a Fourier-based low-frequency and high-frequency signal correction, aiming at adjusting the intensity of low-frequency and high-frequency signals to be consistent with previous conclusions. Its mathematical form is as follows:

$$f_{low} = \alpha M_{low} ft \tag{A1}$$

Table A3: **Ablation on fourier.** × represents the non-use of fourier, while the ✓ represents the use of fourier.

Fourier	MultiGen-20M		
	FID (↓)	SSIM (↑)	MUSIQ (↑)
×	12.79	44.97	68.38
✓	13.21	44.49	68.09

$$f_{high} = \frac{\tilde{M}_{low} f}{min(t, \frac{2}{3})} \tag{A2}$$

where t is the current time step, M_{low} is a low-frequency mask, $\tilde{M}_{low}$ is a high-frequency mask, α is the correction hyperparameter, f is the spectrogram, and f_{low} and f_{high} represent the corrected low-frequency and high-frequency signals, respectively. f_{low} decreases over time steps, while f_{high} increases. We incorporate this Fourier correction into the control network, and the results are shown in Tab.A3. It demonstrates that the image quality generated by the Fourier correction constructed under the previous conclusions has declined in various aspects, which indirectly supports the validity of our conclusions.

In the examples shown in Fig.4 of the main paper, due to the fine-grained division of patches, the pose maps are mostly blank noise at many patch locations, resulting in a lack of control effect. However, contrary to our expectations, in the invalid areas corresponding to the pose maps, patch selection still chooses the blank noise from the pose maps to perturb the generation, which leads to better generation results. This is supported by the comparisons in Tab.2 of the main paper, where the generation results under all conditional control scenarios significantly outperform other combinations of conditions, particularly in consistency evaluation metrics such as FID and SSIM. This indicates that appropriately adding noise during the denoising process does not negatively impact image generation; rather, it frees up more space for other visual condition controls to exert their influence, which is consistent with our conclusions regarding the selection tendencies of various visual conditions in patch selection.

G Generation Stability Comparison

As shown in Fig.A1, the previous single-condition control framework encounters significant challenges in preserving both stability and consistency regarding the detailed features and morphological characteristics of target objects across multiple generated images. Particularly in the cake and furniture scenarios, Uni-ControlNet manifests several limitations, including texture degradation, contour deformation, and background inconsistency. In stark contrast, PixelPonder demonstrates superior performance in maintaining both consistency and stability across these dimensions. A notable example can be observed in the cake scenario, where PixelPonder effectively maintains precise control over the cream's morphological features while simultaneously ensuring consistent dimensional attributes of the cake across all generated images.

As shown in Fig.A2, randomly selected visual results exhibit significant deviations in terms of image layout, fine details, and intra-image contextual understanding of the garment-person interaction. Although the statistical difference is only a few percentage points, the visual discrepancy is highly noticeable. This demonstrates the effectiveness of the patch adaptation strategy in the generation process.

H Image Consistency Exploration

To investigate whether the images generated by PixelPonder are consistent with the visual control conditions in terms of visual elements, we provide additional relevant visual demonstrations in Fig.A3 and Fig.A4.

In each visual example, we present the reference along with its corresponding visual conditions. Additionally, the generated images are displayed under the visual conditions of the reference, as well as the extracted visual conditions from these generated images to investigate the consistency of each



Figure A1: **Comparison of multiple random generations under single-condition and multi-condition.** ***Ours*** employs all available conditions, whereas **canny** and **depth** denote the use of only the canny or depth conditions, respectively.

method with respect to various visual conditioning details in terms of visual fidelity and adherence to the specified conditions.

Tab.A4 presents quantitative metrics evaluating the consistency between the condition maps of the generated images and those of the original reference images. Following the evaluation metrics of ControlNet++, we employ F1 Score for Canny edges, SSIM for HED edges, and RMSE for depth maps to assess consistency. The main comparison focuses on schemes with identical multi-visual conditions. Thus, ControlNet++ is excluded from direct comparison as it is a single-condition method.

Notably, **ControlNet++** represents the state-of-the-art single-condition control baseline within our knowledge. To investigate the performance of single-condition methods in terms of comprehensive

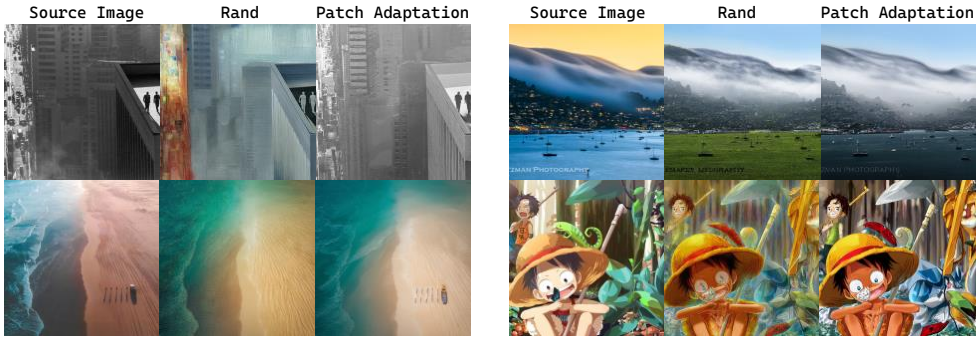

Figure A2: **Generation quality comparison** between Random Selection and Patch Adaptation. Random selection can lead to a loss of information, resulting in reduced capability to generate details and clarity.

Table A4: **Comparison of image consistency and controllability on MultiGen-20M.** Note that **ControlNet++ is a single-condition model**, while other compared methods are **multi-condition models**. The diagonal cells (highlighted in gray) represent the optimal performance of ControlNet++ when evaluated under its corresponding visual condition.

Methods	Conditions	MultiGen-20M		
		Canny Edge (F1 Score↑)	Hed Edge (SSIM↑)	Depth Map (RMSE↓)
ControlNet++	canny	31.98	49.61	14.58
ControlNet++	hed	30.33	66.49	12.17
ControlNet++	depth	17.49	39.22	11.60
T2I-Adapter	all	11.35	28.93	24.97
Uni-ControlNet	all	21.97	41.96	17.14
UniControl	all	26.8	46.26	13.26
Ours	all	**28.35**	**60.51**	**11.93**

control precision, we extracted multi-visual condition maps from images generated under single-condition constraints and calculated the corresponding metrics. As synthesized from Tab. A4 and Figs. A3-A4, the *specialization* of the single-condition scheme induces a significant **visual imbalance** phenomenon: while achieving exceptionally high consistency under the specific designated condition, its control precision for other unassigned conditions remains notably weak. This further highlights the critical need for coordinated multi-visual control tasks to achieve balanced and precise generation.

Moreover, multi-condition combination approaches (e.g., T2I-Adapter, Uni-ControlNet, UniControl) exhibit significantly inferior visual consistency compared to single-condition control methods overall. This indicates that existing multi-condition visual control methods struggle to harmonize the representations of diverse visual conditions, leading to degraded controllability as condition complexity increases.

PixelPonder, benefiting from its unique visual condition integration scheme, effectively leverages the complementary representational strengths of diverse visual conditions, achieving superior adherence across multiple scales (Canny, HED, and depth) compared to existing single- and multi-condition control approaches.

I More Qualitative Comparison

In Fig.A5 and Fig.A6, we present more visual results across various scenes. Previous methods do not separate multiple classes of visual conditions in visual control; instead, they inject multiple visual conditions simultaneously at a single time step, redundantly introducing visual elements of cannys. Our proposed PixelPonder, through patch-level separation of visual conditions and temporal domain separation, avoids the generation of overlapping conflicting visual signals and achieves the injection of complementary information among various visual elements in the temporal domain, resulting in superior generation performance under multi-visual condition control.

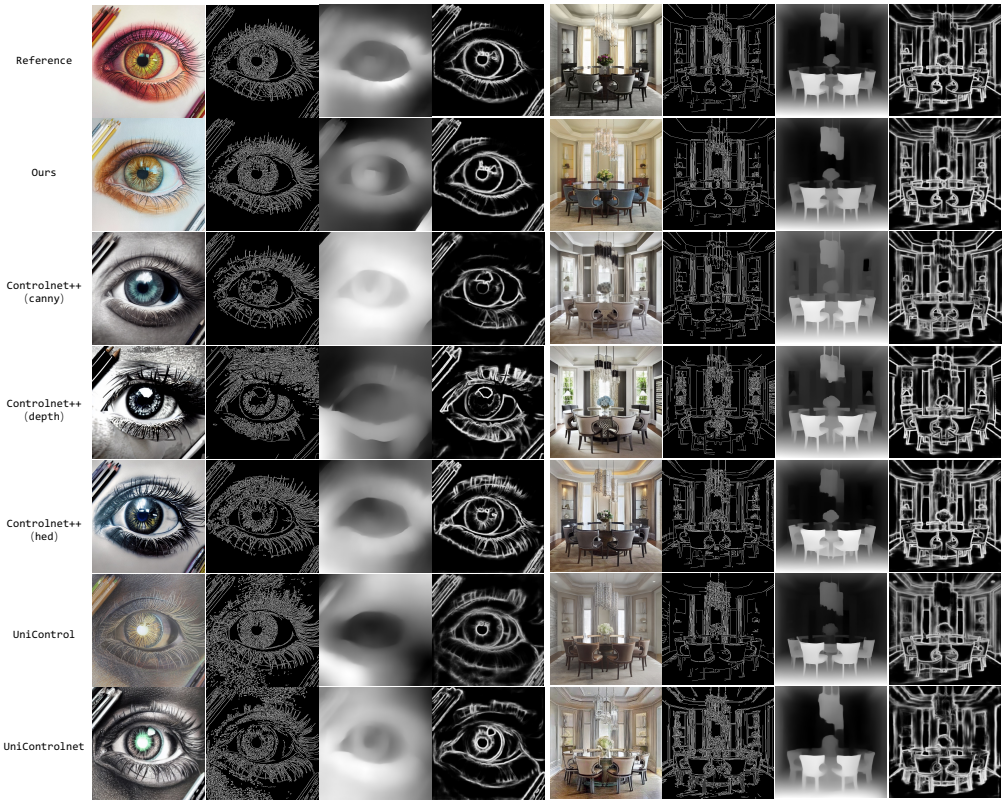

Figure A3: **Image consistency comparison** between official or re-implemented methods and our proposed model.

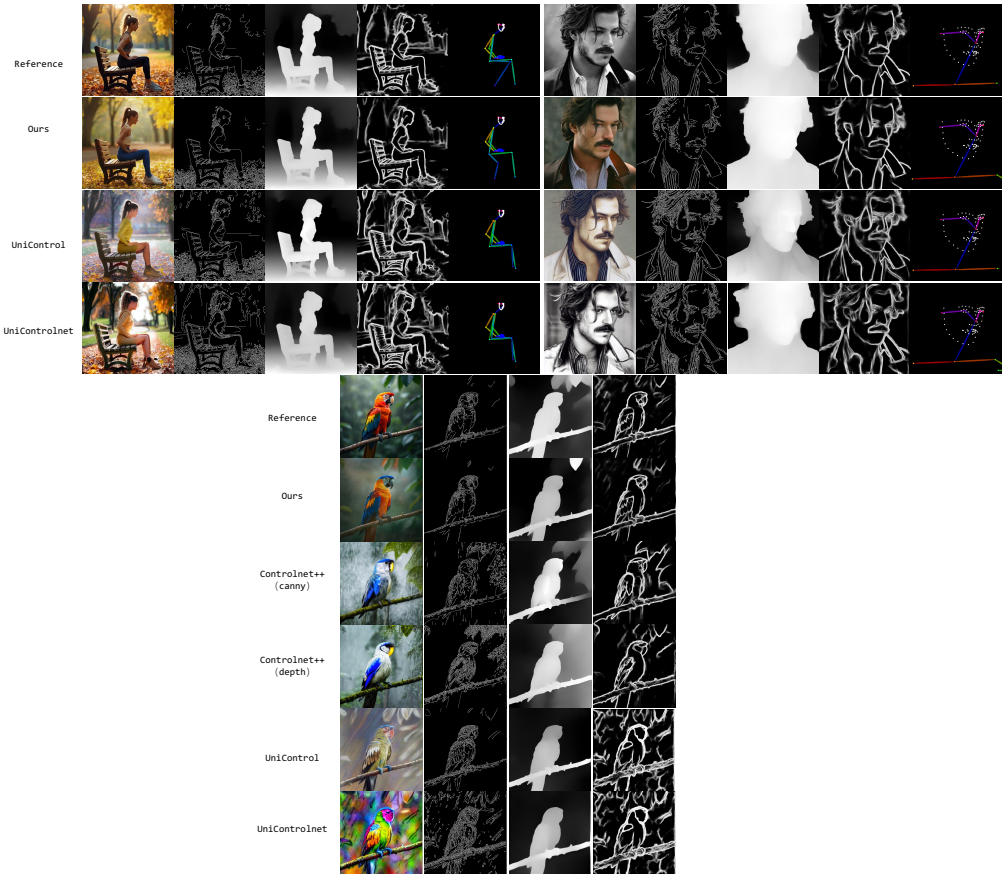

Figure A4: **Image consistency comparison** between official or re-implemented methods and our proposed model.

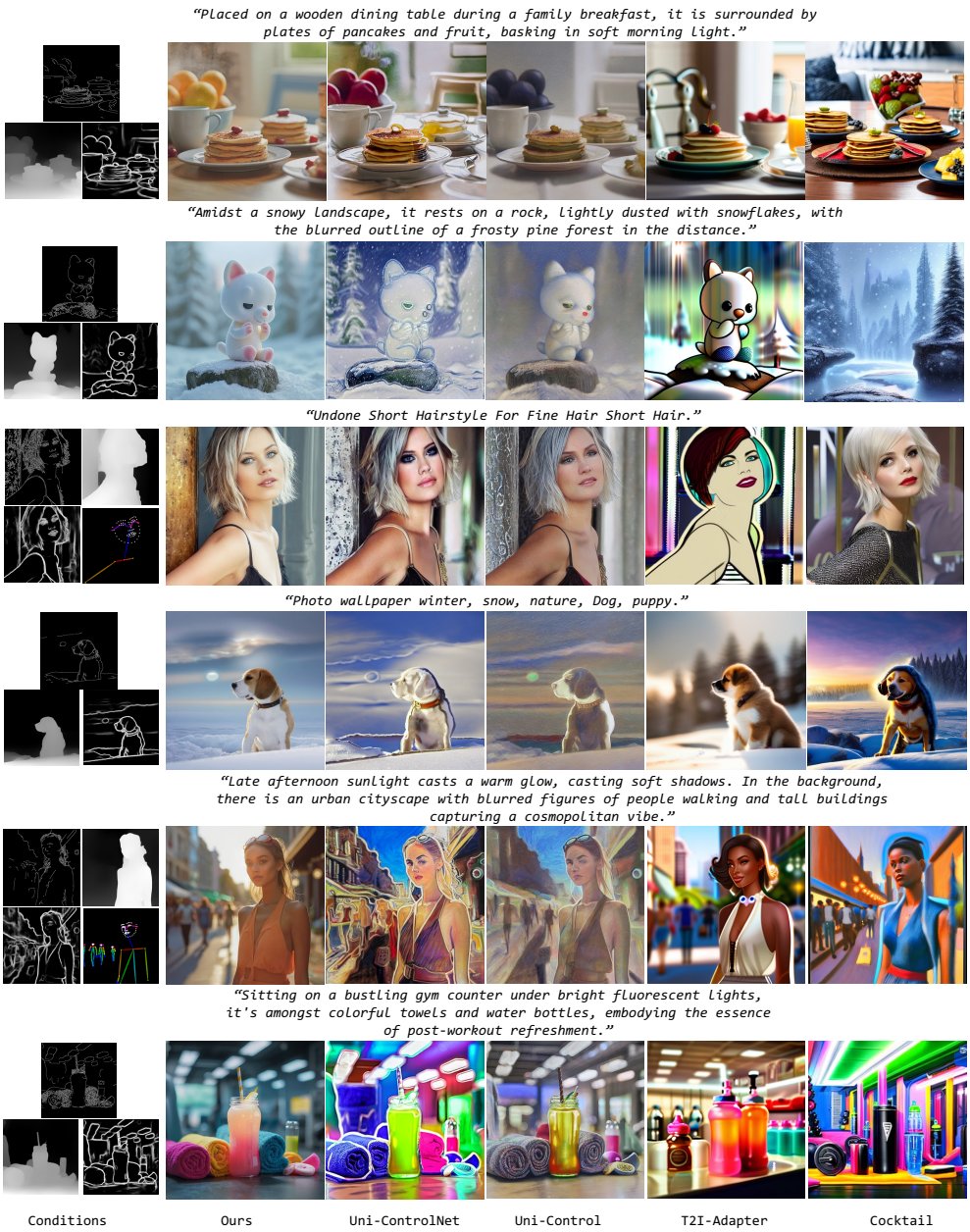

Figure A5: **Image consistency comparison** between official methods and our proposed model.

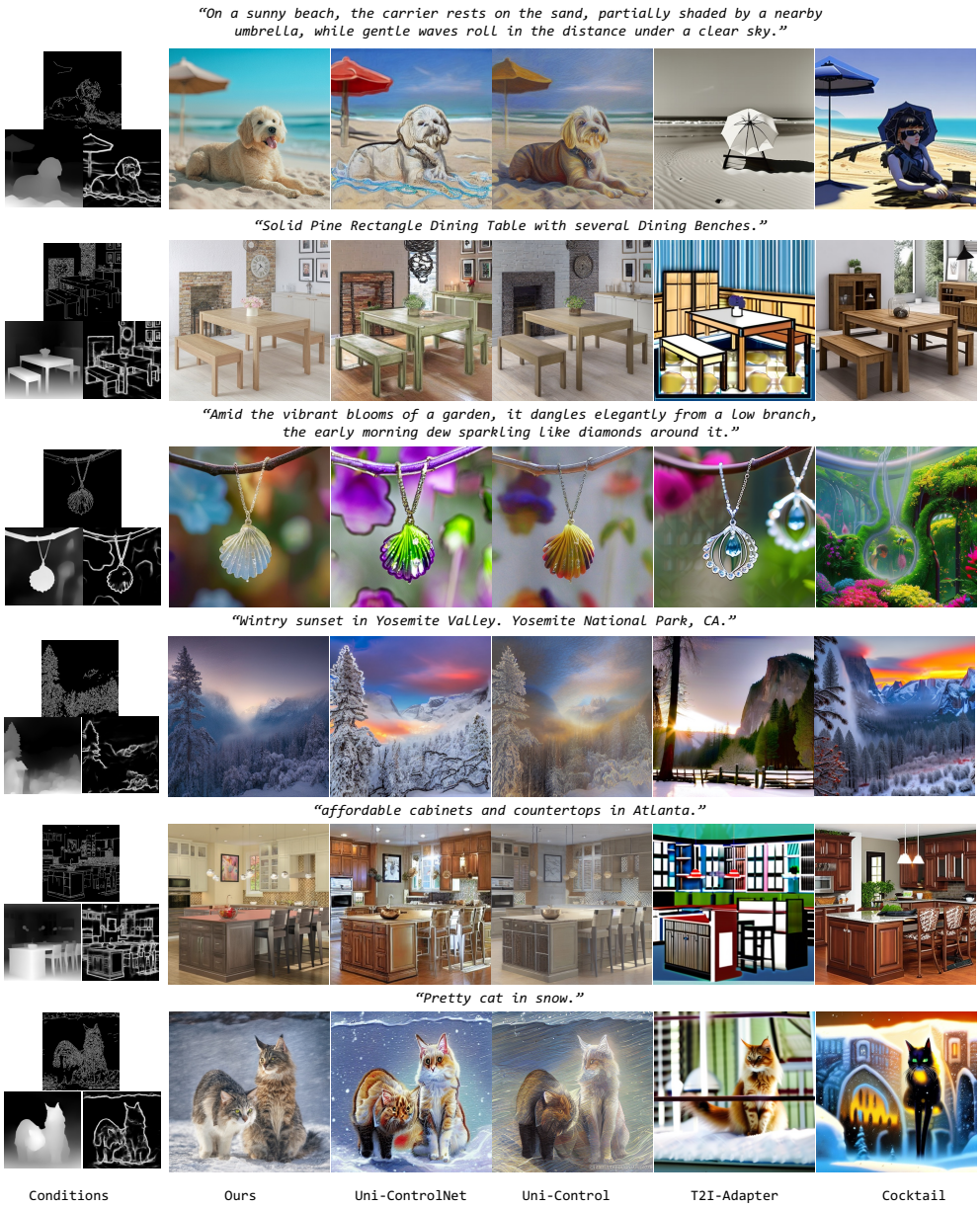

Figure A6: **Image consistency comparison** between official methods and our proposed model.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction have clearly included the motivations, important assumptions, and contributions made in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The authors have discussed the limitations of the work in the Limitations section (App. I).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The authors have clearly provided the full set of assumptions and complete proofs in the Method section.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The authors have presented all the experimental details in the Implementation Details section (App. E).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The authors have presented all the experimental details in the App. E, which provided sufficient instructions to faithfully reproduce the main experimental results. The code and dataset will be released publicly upon acceptance of the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The authors have presented all the experimental details in the Implementation Details section (App. E).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The authors have included experimental details in the Experiments section and Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The authors have included sufficient information on the computer resources in the Implementation Details (App. E).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The authors have explained the broader impacts of the work in the Appendix.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The authors have cited the original paper that produced the code package or dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The authors have submitted the details of the code/model which includes details about training, license, limitations, etc.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.