Universal Few-Shot Spatial Control for Diffusion Models

Kiet T. Nguyen **Chanhyuk Lee** **Donggyun Kim** **Dong Hoon Lee** **Seunghoon Hong**
KAIST
{kietngt00, chan3684, kdgyun425, donghoonlee, seunghoon.hong}@kaist.ac.kr

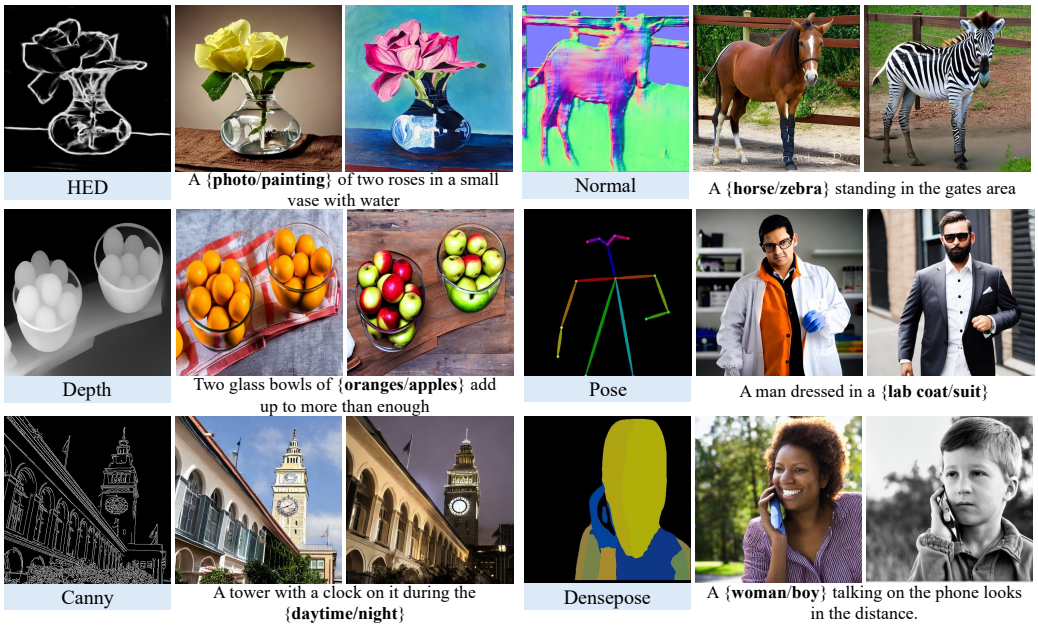

Figure 1: Results of our method learned with **30 examples** on **unseen** spatial conditions. The proposed control adapter guides the pre-trained T2I models in a versatile and data-efficient manner.

Abstract

Spatial conditioning in pretrained text-to-image diffusion models has significantly improved fine-grained control over the structure of generated images. However, existing control adapters exhibit limited adaptability and incur high training costs when encountering novel spatial control conditions that differ substantially from the training tasks. To address this limitation, we propose Universal Few-Shot Control (UFC), a versatile few-shot control adapter capable of generalizing to novel spatial conditions. Given a few image-condition pairs of an unseen task and a query condition, UFC leverages the analogy between query and support conditions to construct task-specific control features, instantiated by a matching mechanism and an update on a small set of task-specific parameters. Experiments on six novel spatial control tasks show that UFC, fine-tuned with only 30 annotated examples of novel tasks, achieves fine-grained control consistent with the spatial conditions. Notably, when fine-tuned with 0.1% of the full training data, UFC achieves competitive performance with the fully supervised baselines in various control tasks. We also show that UFC is applicable agnostically to various diffusion backbones and demonstrate its effectiveness on both UNet and DiT architectures. Code is available at `https://github.com/kietngt00/UFC`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Text-to-Image (T2I) diffusion models [11, 41, 38, 43, 8, 7, 25], trained on large-scale datasets, have achieved remarkable success in generating high-quality, semantically aligned images from natural language prompts. While language-based control offers intuitive and flexible guidance, it often lacks the precision needed for fine-grained visual control, such as specific object positions, shapes, or scene layouts. To overcome this, recent works [19, 35, 28, 58, 27, 39, 59, 53] incorporate explicit spatial signals—like edge maps, depth maps, and segmentation masks to control diffusion models.

To enable spatial control while preserving the generative quality of pre-trained diffusion models, existing methods typically employ control adapters [58, 35, 28] that inject spatial signals into a frozen T2I model. However, these adapters are usually trained independently for each spatial control task, requiring substantial computational resources and extensive labeled data for a new task. Alternatively, reusing pre-trained multi-task adapters - either directly [39, 53] or with minimal updates [59]- struggle to generalize to tasks that differ from their training distribution, and often show poor adaptability.

In this work, we aim to address these limitations by proposing a universal few-shot learning framework, named Universal Few-shot Control (UFC), that efficiently controls diffusion models with novel spatial conditions using only a small number of labeled examples at test time (*e.g.*, dozens of image and condition pairs). Developing such a method would substantially enhance the practicality and flexibility of spatial control methods, but also poses two main challenges. First, significant domain discrepancies across diverse spatial signals make it difficult to learn a consistent, shared representation of control signals that facilitate generalization to novel conditions. Second, in real-world settings where new control tasks may arise dynamically and labeled data is limited, the model must be capable of efficient test-time adaptation from only a few annotated examples.

To overcome these challenges, UFC introduces a universal control adapter that represents novel spatial conditions by adapting the interpolation of visual features of images in a small support set, rather than directly encoding task-specific conditions. The interpolation is guided by patch-wise similarity scores between the query and support conditions, modeled by a matching module [23]. Since image features are inherently task-agnostic, this interpolation-based approach naturally provides a unified representation, enabling effective adaptation across diverse spatial tasks. Furthermore, to facilitate rapid and data-efficient updates, UFC combines episodic meta-learning on a multi-task dataset with a small set of task-specific parameters, enabling effective adaptation under few-shot settings.

Extensive experiments across diverse spatial control modalities, including edge maps, depth maps, normal maps, human pose, and segmentation masks for human body, demonstrate that UFC outperforms prior approaches in few-shot settings by large margins. Notably, UFC achieves fine-grained spatial control using only **30 shots**, and even on par with some fully supervised baselines on Normal, Depth, and Canny with just 150 support examples (0.1% of the training data). Furthermore, UFC is compatible with both UNet [42] and DiT [36] backbones, ensuring applicability across recent T2I diffusion architectures. To our knowledge, we propose the first method for *few-shot spatial control* in text-to-image diffusion models, which enables spatial control image generation with novel conditions at test time using minimal annotated data.

2 Related Works

Controllable Image Diffusion Models Text-to-Image diffusion models [11, 41, 38, 43, 8, 7, 25] have gained widespread adoption for their ability to generate high-quality, semantically aligned images from text prompts. To enable finer control over image structure, various methods [35, 58, 28, 27, 39, 59, 53, 48, 52] proposed incorporating additional spatial control inputs such as edge maps, segmentation masks, or human poses, etc., into diffusion models. ControlNet [58] augments a pretrained T2I diffusion model with a control adapter, initialized from the model itself, to encode spatial conditions and inject them into the frozen diffusion model. Uni-ControlNet [59] extends this by adding lightweight feature extractors to support multiple condition types. Prompt Diffusion [53], built on ControlNet, introduces a vision-language prompting mechanism to enable in-context learning for handling a diverse set of tasks. However, these prior methods are limited to tasks seen during training. To address this limitation, training-free methods [2, 57, 34, 29] have been proposed; but, they often rely on a long process of latent optimization, which substantially increases generation time while providing only limited controllability. Consequently, few-shot adaptation with more fine-grain controllability without incurring significant generation-time overhead remains unexplored.

Few-shot Diffusion Models To extend diffusion models to unseen tasks, several works [5, 60, 55, 46, 12, 26, 20] have explored few-shot learning for image generation. For example, D2C [46] trains a variational autoencoder [24] on a few labeled examples to produce latent representations that condition a diffusion model to generate images in the target domain. Similarly, FSDM [12] uses visual features extracted from a small support set to guide image generation toward unseen classes. While these approaches demonstrate promising few-shot generation capabilities, they primarily focus on domain or class transfer, and are not designed to generalize across diverse spatial control tasks.

Analogy Image Generation. Several recent works [13, 56] leverage the analogy between a query source and a support source-target pair to enable generalization in diffusion models on diverse tasks with visual instruction. Analogist [13] treats visual instruction as an inpainting task, arranging query-support pairs in a 2×2 grid and using a diffusion model to complete the missing cell. This training-free method generalizes well to tasks like colorization, deblurring, and editing. While we also leverage query-support analogy in few-shot settings, our goal differs: prior work preserves the support image's appearance, whereas we aim to generate diverse content guided by spatial conditions.

3 Background

A text-to-image diffusion model generates an image x conditioned on a text prompt c_{text} by iteratively denoising a sequence of latent variables. Specifically, the model learns a finite-length Markov chain $\{z_{T-t}\}_{t=0}^{T}$ that starts from standard Gaussian noise $z_T \sim \mathcal{N}(0, I)$, where each transition $z_{t+1} \to z_t$ progressively removes noise. The final clean variable z_0 can be either the image itself or a latent representation $z = E(x)$, which is decoded back to the image by $x = D(z)$, where (E, D) denotes an auto-encoder [41, 38]. The chain is the time-reversal of a diffusion process $\{z_t\}_{t=0}^{T}$ that gradually corrupts the data by adding small Gaussian noise until it becomes pure noise at z_T [47]. To learn each denoising transition, the diffusion network $\mathcal{E}_\phi$ is trained to predict the added noise ϵ at timestep t:

$$\min_{\phi} \mathbb{E}_{(z_0, c_{\text{text}}) \sim \mathcal{D}_{\text{train}}, t, \epsilon} \left[\left\| \epsilon - \mathcal{E}_\phi(z_t, c_{\text{text}}, t) \right\|^2 \right], \tag{1}$$

where (z_0, c_{text}) is drawn from the training dataset $\mathcal{D}_{\text{train}}$ and t is uniformly sampled from $\{1, \cdots, T\}$. z_t can be computed in closed form from z_0 and t using the properties of Gaussian variables [16]. After training, an image is generated by running the learned Markov chain from z_T down to z_0.

Although text-to-image diffusion models control the overall content of an image, a text prompt alone is limited in steering the fine-grained spatial structures. Spatial control approaches [35, 58, 28, 48] address this limitation by introducing an additional spatial condition y_τ, typically a dense map describing the target image (*e.g.,* edges, depth, or semantic masks). For each condition type (or *task*) τ, an auxiliary control adapter $\mathcal{G}_{\theta_\tau}(y_\tau)$[1] is trained and injected into a frozen, pre-trained diffusion backbone $\mathcal{E}_\phi$, while keeping the backbone parameters ϕ fixed to preserve their learned priors:

$$\min_{\theta_\tau} \mathbb{E}_{(z_0, c_{\text{text}}, y_\tau) \sim \mathcal{D}_{\text{train}}, t, \epsilon} \left[\left\| \epsilon - \mathcal{E}_\phi(z_t, c_{\text{text}}, t, \mathcal{G}_{\theta_\tau}(y_\tau)) \right\|^2 \right], \tag{2}$$

However, learning a dedicated adapter for each task τ usually requires thousands of labeled data to achieve reasonable performance [58]. This can be costly if we want to condition the pre-trained model on a novel task τ_{novel}. On the other hand, reusing an adapter trained on a fixed task(s) for novel tasks often suffers from limited adaptation performance, especially when the new tasks are substantially different from training [59]. Therefore, designing a *versatile* control adapter that can accommodate arbitrary unseen spatial conditions with a small amount of data remains a key challenge for data-efficient steering of text-to-image diffusion models.

4 Method

4.1 Problem Setup

Our goal is to build a universal few-shot framework that steers a frozen text-to-image diffusion model with arbitrary *novel* spatial conditions, given only a handful of annotated examples (typically a few dozen). To this end, we introduce a universal control adapter $\mathcal{I}(y_\tau; \mathcal{S}_\tau)$ that can inject any type of spatial conditions into the frozen diffusion model by directly incorporating the given data as follows:

$$\hat{\epsilon} = \mathcal{E}_\phi(z_t, t, c_{\text{text}}, \mathcal{I}(y_\tau; \mathcal{S}_\tau)), \tag{3}$$

[1]The control adapter can take auxiliary inputs such as noisy latent, timestep, and text prompt, but we omit them for clarity.

where $\mathcal{S}_\tau = \{(x^i, y_\tau^i)\}_{i \leq N}$ denotes a support set comprising N image-condition pairs for task τ. We focus exclusively on the design of the adapter $\mathcal{I}$ while the remaining components, such as loss function, diffusion architecture, and injection strategy, are indepedent to the adapter and can be freely chosen from existing spatial control approaches [35, 58, 28].

Unlike a task-specific adapter $\mathcal{G}_{\theta_\tau}$ (Eq. (2)), the universal adapter $\mathcal{I}$ must address two key challenges.

1. **Heterogeneous input formats.** Spatial conditions vary widely—from extremely sparse cues such as human-pose keypoints to dense maps such as depth or segmentation. The adapter must transform these disparate inputs into the *unified* control features that the frozen diffusion model can consume reliably.
2. **Severe data scarcity.** For a new task, we assume only a few dozen labeled examples. To handle distribution shifts without over-fitting the tiny support set, the adapter needs an adaptation mechanism that is both *flexible* and *efficient*.

4.2 Universal Few-Shot Control Adapter

We propose **Universal Few-shot Control (UFC)**, a universal control adapter that addresses the challenges outlined in Section 4.1 by (1) unifying heterogeneous spatial conditions with image features, and (2) adapting to new tasks through patch-wise matching and parameter-efficient fine-tuning. An overview is shown in Figure 2.

To obtain control features that remain consistent across diverse condition types, UFC leverages *task-agnostic* visual patches extracted from support images via **patch-wise matching**. Specifically, given a query condition y_τ^q and a support set $\mathcal{S}_\tau = \{(x^i, y_\tau^i)\}_{i \leq N}$, UFC encodes images and conditions into spatial feature maps with encoders f and g_τ, respectively. The control feature for k-th query patch is then constructed by

$$\mathcal{I}(y_\tau^{q,k}; \mathcal{S}_\tau) = \sum_{i=1}^{N} \sum_{j=1}^{M} \sigma\big(g_\tau(y_\tau^{q,k}),\, g_\tau(y_\tau^{i,j})\big) \cdot f(x^{i,j}), \tag{4}$$

where $\sigma : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ is a patch-wise similarity function, M is the number of spatial patches per image, and $k, j \leq M$ denote the patch indices. Intuitively, the patch embeddings of the support images $f(x^{i,j})$ form *task-agnostic* bases of query conditions, while *task-specific* weights—computed from the conditions—select the relevant entries.

The patch-wise matching formulation in Eq. 4, inspired by few-shot dense prediction methods [23, 22], offers several advantages for the universal control problem. First, since the task-specific conditions are only used to determine the weights for the task-agnostic visual features, the resulting control features $\mathcal{I}(y_\tau^q; \mathcal{S}_\tau)$ reside in a unified visual feature space regardless of the task τ. This ensures consistent guidance of the diffusion model and robust generalization to unseen tasks. Second, by composing the control features patch-wise, it can exploit the locality inductive bias on the image-condition pairs and amplify the effective number of support patches (*i.e.*, bases) for each query patch. This provides sufficient representational power to model complex spatial controls from just a few support images. Finally, the general matching rule can wrap any control adapter G_{θ_τ} by treating it as a condition encoder g_τ and adding an image encoder f. UFC can therefore inherit advances in adapter architectures and injection mechanisms developed for spatial control.

To adapt rapidly to a new task without over-fitting, UFC introduces only a *small* set of task-specific parameters θ_τ while sharing the remainder. Starting from a pre-trained visual backbone $g(\cdot; \theta)$, we obtain a condition encoder

$$g_\tau(y_\tau) = g(y_\tau; \theta, \theta_\tau), \tag{5}$$

where θ is shared across tasks and θ_τ is lightweight (*e.g.,* bias [3] or LoRA [18] parameters). After meta-training on a fixed task set (Section 4.4), θ is frozen and only θ_τ is fine-tuned for each novel task τ. The image encoder f, also initialized from a pre-trained visual backbone, remains frozen throughout to preserve task-agnostic visual features. Leveraging strong priors of large-scale visual pre-training and modern parameter-efficient tuning techniques, UFC generalizes to unseen tasks with minimal task-specific parameters.

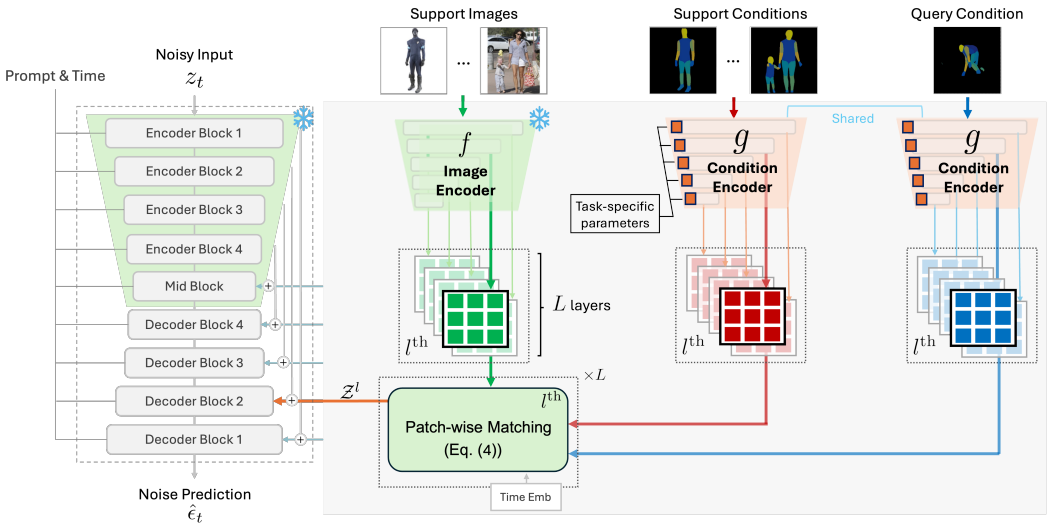

Figure 2: Overview of the proposed framework. The control adapter $\mathcal{I}$ consists of an image encoder f, a condition encoder g_τ, and a matching module implementing Eq. 4. The support image-condition pairs and the query conditions are encoded to extract multi-layer features. The matching module at each layer is applied to produce control features. The features are then injected into the generation process following the mechanism in Section 4.3 to control the structure of images.

4.3 Architecture

The formulation of UFC's control adapter in Eq. (4) is general and can be incorporated into various diffusion backbones and adapter architectures. In this section, we describe the instantiation of UFC within the ControlNet framework [58] due to its popularity and strong performance. For clarity, we consider the pre-trained Stable Diffusion [41, 11] with UNet backbone [41] in this section, while the one with DiT backbone [11] is described in Appendix D.

We initialize the condition encoder g_τ and image encoder f as separate copies of the UNet encoder of the pre-trained diffusion model, where the image encoder is kept frozen. For the task-specific parameters θ_τ of the condition encoder, we adopt bias-tuning [3], which has been proven to be flexible for few-shot learning in dense visual prediction [23, 22]. The matching module is implemented as a multi-head cross-attention block [50], where the timestep embedding is fused through *adaptive layer normalization* [37, 36] inside the attention mechanism. More implementation details about the matching module can be found in Appendix A.1.

To flexibly control the diffusion model, we perform patch-wise matching (Eq. (4)) at multiple levels of the UNet encoders g_τ and f using L separate matching modules, where L denotes the number of UNet encoder layers. The extracted features from g_τ and f at each encoder layer l are passed through the l-th matching module to obtain the corresponding control features $\mathcal{I}^l(y_\tau; \mathcal{S}_\tau)$. These control features pass through a zero-initialized linear projection layer $\mathcal{Z}^l$ and are added to the original features of the target diffusion model:

$$e_c^l = e^l + \mathcal{Z}^l(\mathcal{I}^l(y_\tau; \mathcal{S}_\tau)), \tag{6}$$

where e^l is the output from the l-th layer of the diffusion encoder. Finally, the resulting feature e_c^l replaces the activation e^l and is integrated via skip connections into the corresponding UNet decoder layer of the diffusion model.

4.4 Training and Inference

To equip UFC with few-shot learning capabilities, we adopt the standard episodic meta-training protocol, wherein each training iteration is an episode that mirrors the test-time procedure. In each episode, we first sample a specific control task τ from a meta-training dataset $\mathcal{D}_{\text{train}}$ that consists of multiple tasks. The sampled task τ supplies a support set $\mathcal{S}_\tau$ and a query set Q_τ, where the model exploits the support set to control the diffusion model with the query condition. The model is optimized using the standard denoising loss [16] (or its flow-matching variant [31]):

$$\min_{\theta,\theta_\tau,\sigma,\mathcal{Z}} \mathbb{E}_{\mathcal{S}_\tau, Q_\tau \sim \mathcal{D}_{\text{train}}} \mathbb{E}_{(z_0, y_\tau, c_{\text{text}}) \sim Q_\tau, t, \epsilon} \left[\| \epsilon - \mathcal{E}_\phi(z_t, c_{\text{text}}, t, \mathcal{I}(y_\tau^q; \mathcal{S}_\tau)) \|^2 \right]. \tag{7}$$

During meta-training, the parameters θ and θ_τ in the condition encoder, the matching modules σ, and the projection layers $\mathcal{Z}$ are updated while the pre-trained diffusion model $\mathcal{E}_\phi$ and the image encoder f remain frozen. By repeatedly training on the few-shot learning episodes, the model learns a generalizable mapping from the query condition and the support set to the unified control features.

For an unseen task τ_{novel}, we perform fine-tuning using its support set $\mathcal{S}_{\tau_{\text{novel}}}$. To this end, we randomly partition the support set into two disjoint subsets $\mathcal{S}_{\tau_{\text{novel}}} = \tilde{\mathcal{S}} \cup \tilde{\mathcal{Q}}$, where $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{Q}}$ act as the pseudo-support and the query sets, respectively. Then the universal control adapter is fine-tuned with an objective similar to the meta-training:

$$\min_{\theta_\tau, \sigma, \mathcal{Z}} \mathbb{E}_{\tilde{\mathcal{S}}, \tilde{\mathcal{Q}} \sim \mathcal{S}_{\tau_{\text{novel}}}} \mathbb{E}_{(z_0, y_\tau, c_{\text{text}}) \sim \tilde{\mathcal{Q}}, t, \epsilon} \left[\| \epsilon - \mathcal{E}_\phi(z_t, c_{\text{text}}, t, \mathcal{I}(y_\tau; \tilde{\mathcal{S}})) \|^2 \right]. \tag{8}$$

Fine-tuning a small number of parameters $\theta_\tau, \sigma, \mathcal{Z}$, while freezing the shared parameters θ, makes the model robust to over-fitting to the few-shot support set. After fine-tuning, UFC can produce a control feature for unseen condition types τ_{novel} using the support set $\mathcal{S}_{\tau_{\text{novel}}}$ as described in Section 4.2.

5 Experiments

5.1 Experimental setup

Model settings For experiments using the UNet architecture, we adopt Stable Diffusion v1.5 [41] as the diffusion backbone. Spatial conditions and support images are first projected into the latent space using the pre-trained VAE [24] from the diffusion model, then processed by respective encoders. When encoding these inputs, we fix the denoising timestep to zero, provide empty text prompts, and omit the noisy latent input in order to focus solely on spatial information. Detailed model setting using DiT backbone is provided in the Appendix D.

Datasets To enable episodic meta-training, we sample 300K text-image pairs from LAION-400M [45], with 150K containing humans (for Pose and Densepose) and 150K randomly sampled. Images are resized to the resolution 512×512. Spatial conditions are extracted using pretrained off-the-shelf models. We consider six condition modalities that cover diverse semantics: Canny edge [4], HED edge [54], MiDaS Depth and surface Normal [40], human Pose [6], and human Densepose [14]. We consider Densepose as a control signal for semantic segmentation to constrain all segmentation classes to be included in the support set. During the training process, we maintain the setting that the models will be trained on 150K image-condition pairs for each task.

To evaluate the performance over novel conditions, the six tasks are grouped into three disjoint splits: (Canny, HED), (Normal, Depth), and (Pose, Densepose). Models are trained on two splits and evaluated on the remaining one. These splits ensure test tasks differ significantly from training tasks, providing a robust measure of few-shot generalization. To simulate few-shot learning, we randomly sample support sets and evaluate the performance on the remaining ones, with the exception of human pose tasks (Pose and DensePose), for which we manually select support examples to ensure coverage of diverse human poses.

Implementation details We train UFC (UNet diffusion backbone) for 12.5K iterations with a batch size of 96 on 8 NVIDIA RTX 3090 GPUs, using AdamW [33] with a learning rate of 1×10^{-5}. During inference with the UNet backbone, we adopt the PNDM sampler [32] with 50 denoising steps, classifier-free guidance (CFG) [17] scale of 7.5, and seed 42. For testing, we fine-tune UFC using all image–condition pairs in the support set, but use only a subset of them as conditions during inference to accommodate memory constraints and reduce generation-time overhead.

Baselines We compare our method in the main experiment against three types of baselines:

- **Fully-supervised baselines** ControlNet [58] and Uni-ControlNet [59] are both trained in a fully-supervised manner on the entire dataset (150K images for each task). ControlNet is trained separately for each condition type, while Uni-ControlNet is jointly trained with all conditions.
- **Few-shot baselines** As the first to tackle spatial control in diffusion models under a few-shot setting, we adapt existing methods, Uni-ControlNet and Prompt Diffusion [53], to align with our few-shot adaptation. For Uni-ControlNet, as mentioned in the original paper, we extend the input layer with additional channels for novel conditions and few-shot fine-tune this layer. For Prompt Diffusion, we evaluate both the zero-shot setup as proposed in the original paper and the full fine-tuning setup. For fair comparison, these few-shot methods are pre-trained using the same meta-training dataset as our method and fine-tuned on the same few-shot data.

Table 1: Controllability measurement on COCO 2017. Few-shot baselines are fine-tuned with **30 shots** for evaluation on novel tasks.

Baseline Type	Method	Canny SSIM (↑)	HED SSIM (↑)	Depth MSE (↓)	Normal MAE (↓)	Pose AP^{50} (↑)	Densepose mIoU (↑)
Fully-supervised	ControlNet [58]	**0.3598**	**0.5972**	**89.09**	**14.14**	**0.525**	**0.4824**
	Uni-ControlNet [59]	0.3378	0.5808	92.60	14.59	0.340	0.4695
Training-free	Ctrl-X [29]	0.2901	0.3002	98.10	19.38	0.005	0.1352
Few-shot	Prompt Diffusion [53]	0.2120	0.2887	98.81	20.34	0.012	0.2266
	Uni-ControlNet + FT [59]	0.2222	0.2917	99.21	20.36	0.010	0.2687
	Prompt Diffusion + FT [53]	0.2773	0.4810	95.64	18.04	0.034	0.3548
	UFC (Ours - UNet)	**0.3239**	**0.5121**	**94.38**	**15.09**	**0.229**	**0.4340**

Table 2: FID evaluation on COCO 2017. Few-shot baselines are fine-tuned with **30 shots** for evaluation on novel tasks.

Baseline Type	Method	Canny	HED	Depth	Normal	Pose	Densepose
Fully-supervised	ControlNet [58]	21.06	17.15	**20.16**	21.73	**44.33**	**34.72**
	Uni-ControlNet [59]	**16.79**	**16.76**	20.35	**19.59**	47.13	36.77
Training-free	Ctrl-X [53]	29.83	30.18	30.32	30.35	56.60	45.88
Few-shot	Prompt Diffusion [53]	24.85	23.00	24.07	23.28	**43.60**	**34.15**
	Uni-ControlNet [59] + FT	24.58	25.48	24.77	25.12	47.82	36.06
	Prompt Diffusion [53] + FT	20.57	**19.86**	24.56	24.30	45.94	36.83
	UFC (Ours - UNet)	**19.24**	20.58	**21.04**	**21.60**	47.91	37.79

- **Training-free baselines** We also compare our method with training-free methods, Ctrl-X [29] and FreeControl [34], which directly employ the frozen T2I diffusion models for spatial control. We consider Ctrl-X as our main evaluation but leave comparisons with FreeControl in Appendix B using a smaller evaluation set due to its slow inference speed.

Evaluation Protocol We quantitatively evaluate all methods on COCO 2017 [30] validation split, which contains 5,000 images with associated captions. All images and their corresponding conditions are resized to 512×512. For tasks Pose and Densepose, we restrict evaluation to images containing humans. To assess image generation quality, we report the Fréchet Inception Distance (FID) [15]. To evaluate controllability, we extract spatial conditions from the generated images using the same pre-trained networks used for dataset construction and compare them with the query conditions using task-specific metrics. Specifically, we use the Structural Similarity Index (SSIM) for Canny and HED, Average Precision (AP^{50}) computed using Object Keypoint Similarity for human Pose, Mean Squared Error (MSE) for Depth prediction, Mean Angular Error (MAE) for surface Normal, and mean Intersection over Union (mIoU) for human segmentation in Densepose.

5.2 Main Results

Quantitative Results We report the **30-shot** performance of our model with UNet backbone [41] in terms of two aspects: controllability (Table 1) and image quality (Table 2). Despite using only a few annotated examples, UFC achieves comparable image quality and over 90% of the controllability performance of fully supervised baselines across most conditions. We observe that dense condition maps benefit more from the matching mechanism, as fine-grained similarity modeling enables more effective control features. This is reflected in the smaller relative performance gap with fully supervised baselines on Densepose compared to Pose.

UFC consistently outperforms all few-shot and training-free methods in controllability while maintaining equal or better image quality. Among the few-shot baselines, Uni-ControlNet shows clear underfitting, and zero-shot Prompt Diffusion fails to generalize, both having weak controllability and FID. Compared to fine-tuned (FT) Prompt Diffusion, UFC has significant gains in both metrics on Canny, Depth, and Normal, with up to 16.8% improvement in controllability. For Densepose and HED, our method achieves similar FID scores but significantly exceeds FT Prompt Diffusion in controllability. On Pose, although FT Prompt Diffusion yields lower FID, its structural control is ineffective ($AP^{50} \approx 0$), while UFC achieves 0.229.

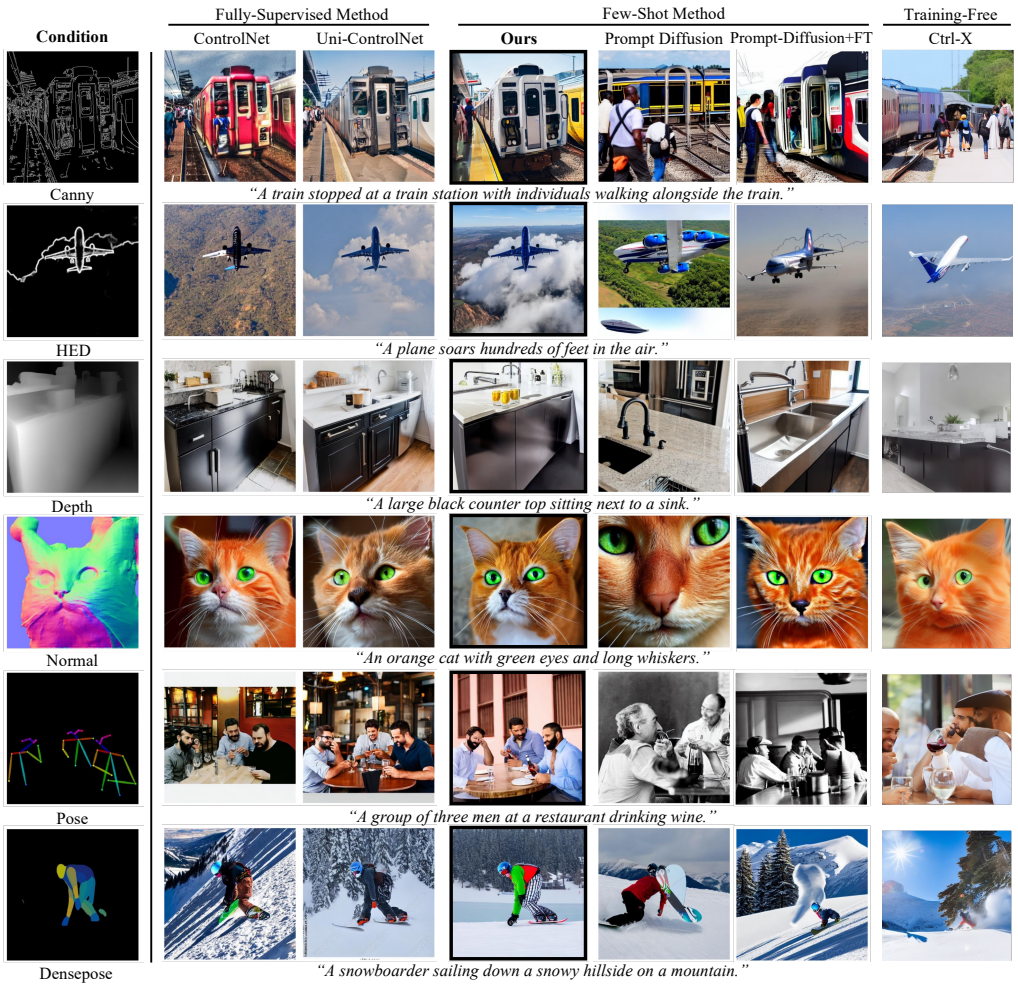

Figure 3: Qualitative comparison across six spatial control tasks. Our method (highlighted in black boxes), fine-tuned with **30-shot** on unseen tasks, demonstrates competitive controllability with fully supervised baselines. In contrast, other baselines struggle to follow the spatial guidance accurately.

UFC's strong few-shot performance stems from two factors. First, it effectively handles the large discrepancy between condition modalities by adapting the interpolation of support image features into task-specific control signals, unlike Prompt Diffusion or Uni-ControlNet, which lack robust strategies for handling distribution shifts. Second, UFC introduces an efficient adaptation mechanism by fine-tuning the matching module σ, projection layers $\mathcal{Z}$, and task-specific parameters θ_τ in the condition encoder from the meta-training weights. In contrast, Uni-ControlNet updates only a tiny subset of parameters, leading to underfitting, while Prompt Diffusion lacks any explicit adaptation mechanism, and naively full fine-tuning is ineffective in the few-shot regime.

Qualitative Results We present qualitative comparisons in Figure 3. As shown, our method with UNet backbone, fine-tuned with only **30 shots**, consistently adheres to the given query conditions across various novel tasks. In contrast, the images generated by the training-free method, Ctrl-X, exhibit low generation quality and poor alignment to the spatial condition, indicating its limited adaptability to diverse spatial conditions. Similarly, zero-shot Prompt Diffusion fails to follow spatial guidance, suggesting that its in-context learning mechanism struggles when there are significant differences between training and unseen tasks. Lastly, while fine-tuned Prompt Diffusion loosely follows the spatial inputs, it often introduces undesired visual artifacts, particularly under the HED and DensePose conditions.

Table 3: Ablation study on matching module and parameters adaptation.

Methods	Canny		HED		Depth		Normal		Pose		Densepose	
	SSIM↑	FID	SSIM↑	FID	MSE↓	FID	MAE↓	FID	AP^{50}↑	FID	mIoU↑	FID
UFC (UNet)	**0.3239**	**19.24**	**0.5121**	20.58	**94.38**	**21.04**	**15.09**	**21.60**	**0.229**	**47.91**	**0.4340**	37.79
w\o Matching	0.2984	20.43	0.4972	**20.43**	96.17	21.10	16.06	23.70	0.150	48.47	0.3995	40.84
w\o Fine-tuning	0.2443	21.64	0.3688	26.42	97.92	22.12	18.90	22.82	0.002	47.97	0.1855	**35.59**

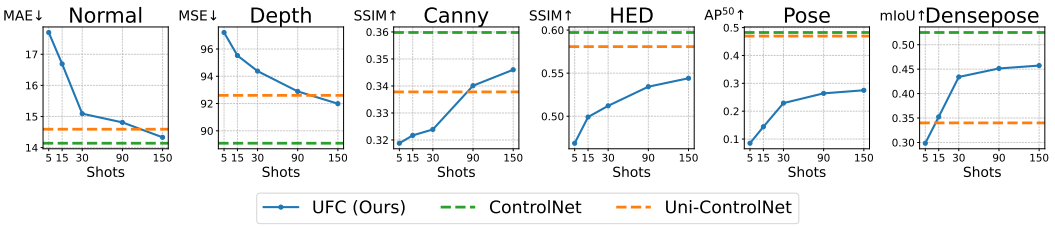

Figure 4: Performance of UFC when fine-tuned with different numbers of support data. Overall, UFC consistently improves the controllability with the increasing size of support sets. The results on FID are presented in the Appendix C, Figure 8.

5.3 Analysis

Ablation study To assess the effectiveness of our design in handling the large discrepancy across condition modalities, we conduct an ablation study with two variants of UFC on UNet backbone [41]: (1) **UFC w/o Matching** and (2) **UFC w/o Fine-tuning**. Both variants are trained on the same dataset as our main method and evaluated on the COCO 2017 validation split. In (1), we remove the matching mechanism and encode the query condition directly using the condition encoder *i.e.*, $\mathcal{G}_\tau(y_\tau)$ while retaining task-specific parameters and meta-learning paradigm. At test time, we fine-tune θ_τ and $\mathcal{Z}$ using the small support set. In (2), we disable adaptation entirely by sharing all parameters across tasks during training and evaluating directly on unseen tasks without any fine-tuning.

Quantitative results are presented in Table 3, with qualitative examples in Appendix C. Both ablation variants show a clear drop in performance compared to our full model. Notably, **UFC w/o Matching** achieves similar image quality but significantly lower controllability, underscoring the importance of leveraging task-agnostic features from support images via the matching mechanism. We provide the attention map visualization in Figure 7, Appendix C for a better understanding of our matching mechanism. **UFC w/o Fine-tuning** performs the worst overall, highlighting the essential role of parameter adaptation in generalizing to novel spatial conditions.

Impact of Support Set Sizes We evaluate UFC on varying sizes of support sets for unseen condition types and plot the performance curves in Figure 4. We can observe a clear trend that the controllability of our method improves considerably with the increasing size of the support set. With 150 support image-condition pairs (0.1% of full training data), UFC begins to surpass the controllability of the fully-supervised baseline (Uni-ControlNet) on several tasks. We also present the FID score over varying support set sizes in Figure 8 in Appendix C, which shows that our method maintains the image quality across different shots. This improvement in controllability demonstrates the effectiveness of our method in adapting to unseen spatial condition modalities through lightweight fine-tuning, eliminating the costly fully-supervised training for novel tasks.

Impact of Support Samples To evaluate the sensitivity of UFC to the choice of support set, we evaluate 30-shot performance with various support sets and report the results in Table 6, Appendix C. It shows that performance of our method tends to remain consistent under diverse support sets in most tasks. However, in some tasks such as Pose and Densepose, we observe that the model performed worse than the main results that used curated supports maximizing diversity in scale, deformation, crowding, and occlusion. It shows that the importance of support-set diversity is itself task-dependent, and ensuring sample diversity can potentially benefit the few-shot performance.

Table 4: Quantitative evaluation of UFC with different backbones in *30-shot* setting.

Backbone	Canny		HED		Depth		Normal		Pose		Densepose	
	SSIM↑	FID	SSIM↑	FID	MSE↓	FID	MAE↓	FID	AP^{50}↑	FID	mIoU↑	FID
Ours (UNet)	0.3239	**19.24**	0.5121	20.58	94.38	**21.04**	15.09	**21.60**	0.229	**47.91**	0.4340	**37.79**
Ours (DiT)	**0.3984**	20.43	**0.6001**	**19.84**	**91.51**	22.70	**13.79**	23.06	**0.336**	51.88	**0.4708**	39.27

Impact of Diffusion Backbone Since UFC is applicable to various diffusion backbones, we employ the stronger pre-trained model based on DiT [11] and compare it with the one based on UNet used in our main table. The quantitative results are reported in Table 4, showing that the DiT-based model with a stronger diffusion backbone consistently outperforms the UNet-based model in controllability across all tasks. The lower image quality compared to the UNet variant, which is also observed in [48], can be due to the mismatch between the optimized image resolution of the pre-trained diffusion backbone with the testing resolution (1024 and 512). Figure 9 in Appendix D shows that UFC (DiT) qualitatively achieves more fine-grained controllability than the UNet counterpart. More qualitative examples using the DiT backbone are shown in Figure 10 in Appendix.

Evaluation on More Novel Conditions To further validate the few-shot capability of our method in more challenging settings, we evaluate UFC on novel spatial conditions on 3D structures, such as meshes, wireframes, and point clouds extracted from iso3d dataset [10]. Importantly, these control tasks involve not only novel spatial conditions but also different output image distributions from meta-training datasets, simulating a more realistic generalization scenario to unseen control tasks.

We meta-train UFC on all six tasks in our original dataset (*i.e.*, subset of LAION-400M), then fine-tune the model on 30-shot support sets manually collected for each condition type. The fine-tuned model is then tested on unseen query conditions. The qualitative results in Figure 11, Appendix E demonstrate that generated images align closely with the spatial conditions provided by 3D meshes, wireframes, and point clouds. These results confirm UFC's effectiveness when encountering novel spatial conditions.

6 Conclusion

In this paper, we present Universal Few-shot Control (UFC), a unified control adapter that is capable of few-shot controlling text-to-image diffusion models with unseen spatial conditions. UFC adapts interpolated visual features from support images into task-specific control signals, guided by the analogy between query and support conditions. When evaluated on unseen control conditions, episodic meta-trained UFC demonstrated strong few-shot ability: it consistently outperformed all few-shot baselines and achieved precise spatial control over generated images. Notably, UFC can be on par with a fully supervised baseline, despite using only 0.1% of the full data for fine-tuning. Finally, we demonstrated that our method is applicable to both recent architectures of diffusion models, including UNet and DiT backbones.

Limitations and Future Research While UFC demonstrates strong few-shot performance in spatially-conditioned image generation with T2I diffusion models, several limitations remain for future work. First, our framework is designed primarily for spatial control generation rather than tasks that require preserving the appearance of the condition image, such as style transfer, inverse problems (*e.g.,* colorization, deblurring, inpainting). Extending the framework to handle such tasks can be a future research direction. Next, our approach requires fine-tuning on a small annotated set for each new task, unlike Large Language Models, which adapt to new tasks via in-context learning from just a few examples without fine-tuning. Developing similar capabilities for spatial control image generation remains an open and promising challenge.

Acknowledgments This work was in part supported by the National Research Foundation of Korea (RS-2024-00351212 and RS-2024-00436165) and the Institute of Information & communications Technology Planning & Evaluation (IITP) (RS-2022-II220926, RS-2024-00509279, RS-2021-II212068, RS-2022-II220959, and RS-2019-II190075) funded by the Korea government (MSIT). The work was also supported by Hyundai Motor Chung Mong-Koo Foundation to Kiet T. Nguyen.

References

[1] Stability AI. 2024. Stable Diffusion 3.5 Medium. `https://huggingface.co/stabilityai/stable-diffusion-3.5-medium`. Accessed: 2025-05-22.

[2] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Universal Guidance for Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 843–852.

[3] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple Parameter-efficient Fine-tuning for Transformer-based Masked Language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

[4] John Canny. 1986. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

[5] Yu Cao and Shaogang Gong. 2024. Few-shot image generation by conditional relaxing diffusion inversion. In *European Conference on Computer Vision*, pages 20–37. Springer.

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Junsong Chen, Simian Luo, and Enze Xie. 2024. PIXART-δ: Fast and Controllable Image Generation with Latent Consistency Models. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*.

[8] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. 2024. PixArt-α: Fast Training of Diffusion Transformer for Photorealistic Text-to-Image Synthesis. In *ICLR*.

[9] CompVis. 2022. Stable Diffusion v1-5. `https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5`. Accessed: 2025-05-22.

[10] Dylan Ebert. 2025. 3D Arena: An Open Platform for Generative 3D Evaluation. *arXiv preprint arXiv:2506.18787*.

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

[12] Giorgio Giannone, Didrik Nielsen, and Ole Winther. 2022. Few-shot diffusion models. *arXiv preprint arXiv:2205.15463*.

[13] Zheng Gu, Shiyuan Yang, Jing Liao, Jing Huo, and Yang Gao. 2024. Analogist: Out-of-the-box visual in-context learning with image diffusion model. *ACM Transactions on Graphics (TOG)*, 43(4):1–15.

[14] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. 2018. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7297–7306.

[15] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

[17] Jonathan Ho and Tim Salimans. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

[19] Lianghua Huang, Di Chen, Yu Liu, Shen Yujun, Deli Zhao, and Zhou Jingren. 2023. Composer: Creative and Controllable Image Synthesis with Composable Conditions. *arXiv preprint arxiv:2302.09778*.

[20] Ying Jin, Jinlong Peng, Qingdong He, Teng Hu, Jiafu Wu, Hao Chen, Haoxuan Wang, Wenbing Zhu, Mingmin Chi, Jun Liu, and Yabiao Wang. 2025. Dual-Interrelated Diffusion Model for Few-Shot Anomaly Image Generation.

[21] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023. Ultralytics YOLO.

[22] Donggyun Kim, Seongwoong Cho, Semin Kim, Chong Luo, and Seunghoon Hong. 2024. Chameleon: A data-efficient generalist for dense visual prediction in the wild. In *European Conference on Computer Vision*, pages 422–441. Springer.

[23] Donggyun Kim, Jinwoo Kim, Seongwoong Cho, Chong Luo, and Seunghoon Hong. 2023. Universal Few-shot Learning of Dense Prediction Tasks with Visual Token Matching. In *The Eleventh International Conference on Learning Representations*.

[24] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[25] Black Forest Labs. 2024. FLUX. `https://github.com/black-forest-labs/flux`.

[26] Lingxiao Li, Yi Zhang, and Shuhui Wang. 2023. The Euclidean Space is Evil: Hyperbolic Attribute Editing for Few-shot Image Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22714–22724.

[27] Ming Li, Taojiannan Yang, Huafeng Kuang, Jie Wu, Zhaoning Wang, Xuefeng Xiao, and Chen Chen. 2024. ControlNet++: Improving Conditional Controls with Efficient Consistency Feedback. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VII*, page 129–147, Berlin, Heidelberg. Springer-Verlag.

[28] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. 2023. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22511–22521.

[29] Kuan Heng Lin, Sicheng Mo, Ben Klingher, Fangzhou Mu, and Bolei Zhou. 2024. Ctrl-x: Controlling structure and appearance for text-to-image generation without guidance. *Advances in Neural Information Processing Systems*, 37:128911–128939.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'ar, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

[31] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. 2023. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*.

[32] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2022. Pseudo Numerical Methods for Diffusion Models on Manifolds. In *International Conference on Learning Representations*.

[33] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[34] Sicheng Mo, Fangzhou Mu, Kuan Heng Lin, Yanli Liu, Bochen Guan, Yin Li, and Bolei Zhou. 2024. Freecontrol: Training-free spatial control of any text-to-image diffusion model with any condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7465–7475.

[35] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 2024. 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*.

[36] William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205.

[37] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*.

[38] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2024. SDXL: Improving Latent Diffusion Models for High-Resolution Image Synthesis. In *The Twelfth International Conference on Learning Representations*.

[39] Can Qin, Shu Zhang, Ning Yu, Yihao Feng, Xinyi Yang, Yingbo Zhou, Huan Wang, Juan Carlos Niebles, Caiming Xiong, Silvio Savarese, Stefano Ermon, Yun Fu, and Ran Xu. 2023. UniControl: a unified diffusion model for controllable visual generation in the wild. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

[40] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2020. 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.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. 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.

[43] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Raphael Gontijo-Lopes, Burcu Karagol Ayan, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. 2022. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In *Advances in Neural Information Processing Systems*.

[44] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved techniques for training gans. *Advances in neural information processing systems*, 29.

[45] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. 2021. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*.

[46] Abhishek Sinha, Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. D2c: Diffusion-decoding models for few-shot conditional generation. *Advances in Neural Information Processing Systems*, 34:12533–12548.

[47] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr.

[48] Zhenxiong Tan, Songhua Liu, Xingyi Yang, Qiaochu Xue, and Xinchao Wang. 2024. Ominicontrol: Minimal and universal control for diffusion transformer. *arXiv preprint arXiv:2411.15098*.

[49] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

[52] Haoxuan Wang, Jinlong Peng, Qingdong He, Hao Yang, Ying Jin, Jiafu Wu, Xiaobin Hu, Yanjie Pan, Zhenye Gan, Mingmin Chi, et al. 2025. UniCombine: Unified Multi-Conditional Combination with Diffusion Transformer. *CoRR*.

[53] Zhendong Wang, Yifan Jiang, Yadong Lu, yelong shen, Pengcheng He, Weizhu Chen, Zhangyang Wang, and Mingyuan Zhou. 2023. In-Context Learning Unlocked for Diffusion Models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[54] Saining Xie and Zhuowen Tu. 2015. Holistically-Nested Edge Detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403.

[55] Ruofeng Yang, Bo Jiang, Cheng Chen, Baoxiang Wang, Shuai Li, et al. 2024. Few-shot diffusion models escape the curse of dimensionality. *Advances in Neural Information Processing Systems*, 37:68528–68558.

[56] Yifan Yang, Houwen Peng, Yifei Shen, Yuqing Yang, Han Hu, Lili Qiu, Hideki Koike, et al. 2023. Imagebrush: Learning visual in-context instructions for exemplar-based image manipulation. *Advances in Neural Information Processing Systems*, 36:48723–48743.

[57] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. 2023. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184.

[58] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847.

[59] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K Wong. 2023. Uni-controlnet: All-in-one control to text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:11127–11150.

[60] Jingyuan Zhu, Huimin Ma, Jiansheng Chen, and Jian Yuan. 2022. Few-shot image generation with diffusion models. *arXiv preprint arXiv:2211.03264*.

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 accurately reflect our contributions and scopes.

   Guidelines:

   - The answer NA 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 NA 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: Limitations are discussed in the Appendix.

   Guidelines:

   - The answer NA 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: [NA]

Justification: We do not include theoretical claims.

Guidelines:

- The answer NA 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: We include sufficient information and provide a URL to our released code and checkpoints for reproducing the results.

   Guidelines:

   - The answer NA 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: We released our code, checkpoints with detailed scripts on GitHub.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe our experimental settings including the hyper-parameters in the main paper and appendix.

   Guidelines:

   - The answer NA 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: [No]

   Justification: We follow the experimental settings from the baselines and belive we provide appropriate experimental results.

   Guidelines:

   - The answer NA 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: We specify the computing resource we use for experiments in our paper.

   Guidelines:

   - The answer NA 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: Our research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [NA]

    Justification: We expect no additional societal impacts or negative societal impacts from our work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We expect no additional risks to be posed by our work.

    Guidelines:

    - The answer NA 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: We clearly state the creators and owners of the assets used in our work.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: We do not introduce new assets.

    Guidelines:

    - The answer NA 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: [No]

    Justification: We do not include crowdsourcing.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: Our paper does not include research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA] .

Justification: Our method does not use LLM.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Implementation Details

A.1 Implementation Details on Matching Module

We implement our matching module described in Section 4.3 using the multi-head attention mechanism [51]. To incorporate denoising timestep t, we adopt *adaptive normalization* [37], modulating the output of the matching module using the denoising timestep. Specifically, we use the embedding of the query condition $g_\tau(y_\tau)$, the support conditions $\{g_\tau(y_\tau^i)\}_{i=1}^N$, and the support images $\{f(x^i)\}_{i=1}^N$ as the query, key, and value inputs, respectively. Let $Q \in \mathbb{R}^{M\times d}$, $K, V \in \mathbb{R}^{(N\cdot M)\times d}$, and the timestep embedding be denoted as $t_{\text{emb}} \in \mathbb{R}^{d_t}$, our matching mechanism operates as below:

$$Q = \text{LayerNorm}(Q), \quad K = \text{LayerNorm}(K), \tag{9}$$

$$(\alpha, \beta, \gamma) = \text{Linear}(t_{\text{emb}}), \quad V = \text{LayerNorm}(V)\cdot(1+\alpha)+\beta \tag{10}$$

$$O = \text{Concat}_{i=1}^{H}\left(\text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\right) \tag{11}$$

where H is the number of attention heads, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d\times d_{\text{head}}}$

The final output is calculated with residual connection as below:

$$O = O + \gamma \cdot act(OW^O) \tag{12}$$

where $W^O \in \mathbb{R}^{H\cdot d_{\text{head}}\times d}$, and act denotes a non-linear activation function.

We apply the matching module at each layer of the 12 encoding blocks and the mid-block in the UNet backbone of the diffusion model, and each attention module uses 8 heads.

A.2 Meta-training Settings

The below setting is for the UNet backbone. The setting for the DiT backbone can be found in Section D.

- **Checkpoint**: We use the Stable Diffusion v1.5 checkpoint available from the HuggingFace [9].
- **Hyperparameters**: We train using the AdamW optimizer [33] with a learning rate of 1×10^{-5} and a weight decay of 0.01. For each training batch, we randomly select two tasks per batch, each accompanied by a support set of three example pairs sampled for its query condition.
- **Spatial condition representation**: Following ControlNet [58], we represent all conditioning inputs as RGB images with a resolution of 512×512.

A.3 Few-shot Fine-tuning Settings

When fine-tuning our meta-trained model on 30 support examples of the novel condition type, at most 600 fine-tuning steps with a batch size of 10 (approximately 1 hour on an RTX 3090 GPU) suffice for all six tasks. Early-stopping with a held-out validation example to monitor the denoising can be applied to avoid overfitting. We found that low-level conditions (e.g., edges) tend to converge faster than high-level conditions (e.g., pose). We also adopt AdamW optimizer [33] with a learning rate of 1×10^{-5} and a weight decay of 0.01.

A.4 Computation Resources

	GPU(s)	Batch size/GPU	Mem/GPU	Time
Training	8 RTX 3090	6	16GB	12 hours
Fine-tuning	1 RTX 3090	10	21GB	<= 1 hour
Inference	1 RTX 3090	8	11.5GB	3.06 s / image

A.5 Dataset Construction

- **Training data**: We randomly sample a subset of data from the LAION dataset [45]. To identify images containing humans, we use the YOLO11x model [21]. Based on this filtering, we construct a dataset consisting of 150K images with humans and 150K images without humans.
- **Evaluation data**: For the Canny, HED, Depth, and Normal tasks, we evaluate our model on 5,000 images from the COCO2017 validation set [30]. For the Pose task, evaluation is limited to images where humans are successfully detected by the Openpose model [6]. Similarly, for the Densepose task, we only evaluate our model with images where the Densepose model [14] detects a human.

B Comparison with Training-Free Methods

We propose a few-shot framework for adapting to new spatial conditions in T2I diffusion models. A natural question arises: *why use a few-shot approach when training-free methods exist [2, 57, 34, 29]*. To illustrate the advantages of our method (UNet backbone), as mentioned earlier in Section 5.1, we compare it with Ctrl-X [29] and FreeControl [34] state-of-the-art training-free approaches capable of handling diverse spatial conditions without relying on additional pretrained networks.

To perform spatial control using a condition image, Ctrl-X [29] replaces the features and attention maps when processing the noisy latent with the ones encoded from the noisy condition image. We re-implement Ctrl-X using the Stable Diffusion v1.5 backbone for a comparable setting. FreeControl [34] controls image structure by constructing a PCA basis from object features, projecting both the condition and noisy image feature maps onto this basis, and minimizing an energy function that encourages alignment between the two projections during generation.

Evaluation protocol We follow FreeControl's original evaluation protocol and assess controllability of UFC and two training-free baselines on 30 images from the ImageNet-R-TI2I dataset [49], which includes 10 object categories with three captions each. FreeControl has limited flexibility in generating diverse object categories, as it requires constructing a separate PCA basis for each. This makes large-scale evaluation on 5,000 images from the COCO 2017 validation set [30] impractical, since each prompt must be individually inspected to identify objects and build corresponding PCA bases.

As the evaluation dataset of FreeControl excludes human subjects, we compare UFC (30-shot) against FreeControl on four tasks: Canny, HED, Depth, and Normal. Due to the limited number of images, we do not report FID [15] or Inception Score [44], as they would not provide reliable estimates of image quality.

Result comparison Table 5 shows that UFC (*30-shot*) significantly outperforms both training-free baselines in controllability across all four tasks. Moreover, on a single NVIDIA RTX 3090 GPU, FreeControl takes approximately 100 times longer per image to generate compared to UFC when generating 30 images for evaluation. This overhead stems from the need to construct a PCA basis for each new object category and perform 200 denoising steps for latent optimization. Ctrl-X requires self-recurrence iteration to avoid out-of-distribution sampling, which increases the generation time for several seconds. In contrast, UFC only introduces a negligible time increase.

Table 5: Controllability scores and average generation time of training-free baselines and UFC (Ours, *30-shot*) on generating 30 images from ImageNet-R-TI2I [49].

Method	Canny SSIM (↑)	HED SSIM (↑)	Depth MSE (↓)	Normal MAE (↓)	Time (Second)
FreeControl [34]	0.3139	0.3821	97.36	19.68	251.3
Ctrl-X [29]	0.3896	0.3693	96.12	20.84	8.01
UFC (Ours)	**0.4074**	**0.5718**	**93.09**	**17.75**	3.06

Figure 5 presents qualitative comparisons. UFC accurately follows the spatial condition, while FreeControl and Ctrl-X fail on fine details control.

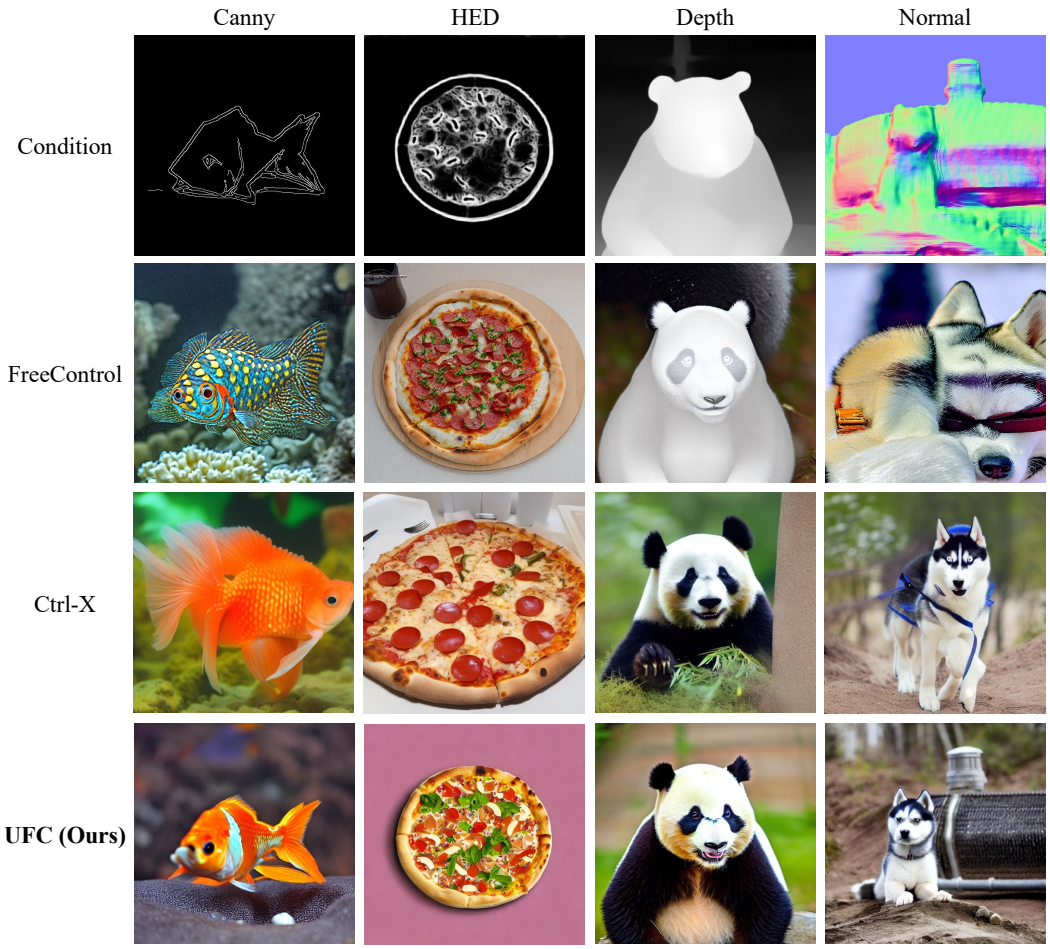

Figure 5: Qualitative comparison between UFC (*30-shot*) and FreeControl [34], Ctrl-X [29] on four control tasks.

C More Results on Analysis

Qualitative results with two variants As mentioned in the ablation study in Section 5.3, Figure 6 presents a qualitative comparison of our method with its two variants: (1) UFC w\o Matching and (2) UFC w\o fine-tuning. Both variants exhibit degraded image generation in terms of controllability, as they often only partially adhere to the spatial conditions. For instance, in the 2nd row with HED condition, the generated images from UFC w\o Matching and UFC w\o Fine-tuning capture the cat's head structure but fail to adhere to the condition for the cat's legs and body. In contrast, our full method, which include both matching and fine-tuning, consistently follows all given conditions.

Figure 6: Qualitative comparison results of our method and its two variants: (1) UFC w\o matching and (2) UFC w\o fine-tuning.

Attention map visualization We present an example of attention maps in Figure 7. For each task, the selected query patch (highlighted by a white box in the Query column) can attend to relevant support patches, rather than unrelated regions such as background areas. For instance, in the Canny and HED tasks (first two rows), the query patches focus on support regions that preserve similar edge structures. On the other hand, for the Pose and Densepose tasks (last two rows), the query patches attend to regions related to human body parts.

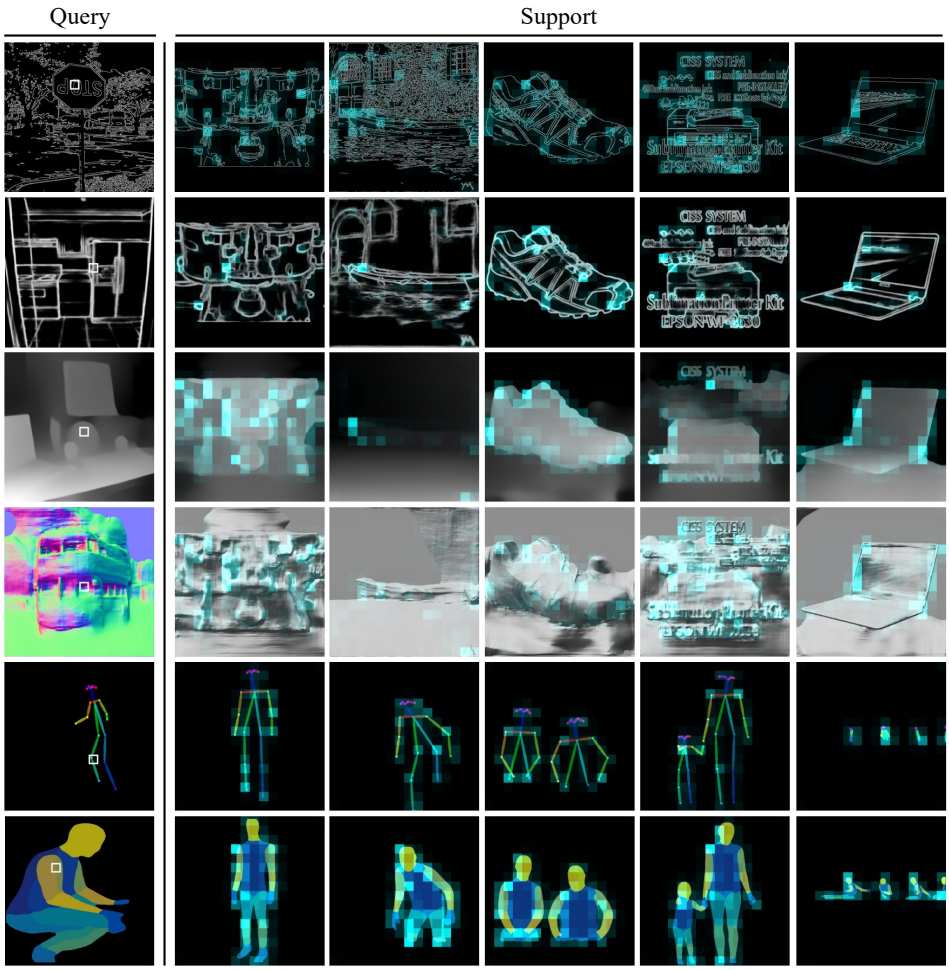

Figure 7: Attention Maps of UFC (*30-shot*) from the 7th layer and a selected head for each task. Support normal maps are converted to grayscale to enhance the visibility of attention regions. Query patches attend primarily to the most relevant support patches.

FID Results with different number of shots In addition to the controllability measurement in Figure 4, Section 5.3, we provide FID results obtained by fine-tuning our method (UFC) using different numbers of support data (i.e., shots) in Figure 8. The results show that our method maintains FID scores across various shots. Specifically, for Normal, Depth, Canny, and HED tasks, our method preserves image quality with FID changes remaining within 1. For Pose, the FID difference is within 4.5, and for DensePose, it is within 1.7. Both changes are relatively small given the FID scores of each task, and using more support data often slightly improves FID. The results confirm that our method improves controllability as the number of support data increases, while maintaining image quality.

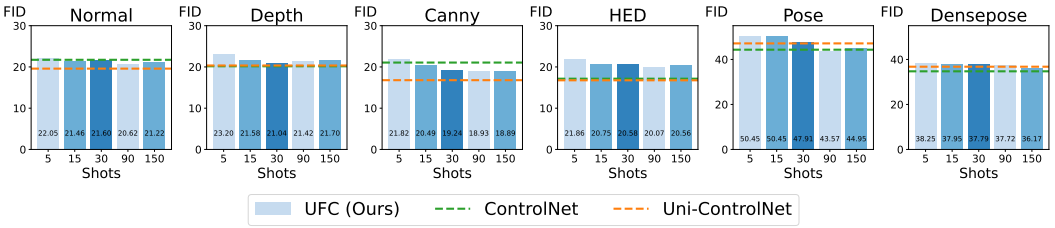

Figure 8: FID score over varying support set sizes (shots).

Effect of support samples As discussed in Section 5.3, UFC is evaluated in 30-shot settings with different investigate the effect of support sets' diversity. In this section, we would like to discuss the implications of *diversity* for each task. Because our matching strategy hinges on condition-specific similarity, the notion of *diversity* depends on tasks. For edge inputs such as Canny and HED, the support set must span a wide range of edge-map densities and orientations. For Pose and DensePose, it needs to cover varied human scales, occlusions, and pose deformations. To measure how support choice influences performance, we repeated fine-tuning three times, each time drawing a new random set of 30 supports and carrying out few-shot generation. The results are summarized in the Table 6.

Table 6: Analysis on the effect of support set's diversity on model performance.

seed	Canny		HED		Depth		Normal		Pose		Densepose	
	SSIM↑	FID	SSIM↑	FID	MSE↓	FID	MAE↓	FID	AP^{50}↑	FID	mIoU↑	FID
1	0.3239	19.24	0.5121	20.58	94.38	21.04	15.09	21.60	0.229	47.91	0.4340	37.79
2	0.3295	21.06	0.4957	20.70	93.47	20.51	15.82	20.85	0.176	48.22	0.3827	39.52
3	0.3273	20.90	0.5128	23.84	94.84	22.21	15.93	22.56	0.191	48.66	0.3831	37.46
4	0.3290	20.41	0.4942	20.78	94.50	21.80	15.56	21.89	0.202	47.83	0.3914	38.30
Mean	0.3274	20.40	0.5037	21.73	94.30	21.39	15.60	21.73	0.1995	48.16	0.3978	38.27
Std	0.0025	0.82	0.0101	1.49	0.5851	0.76	0.3737	0.71	0.0224	0.38	0.0245	0.90

D Experiments with DiT Backbone

Model As discussed in Section 4.3, our framework can work with both UNet and DiT backbones. We extend UFC to the DiT architecture [36], using Stable Diffusion v3 [11] as an example, initialized from the publicly available v3.5-medium checkpoint [1] on Hugging Face. This model consists of 24 Transformer layers in the diffusion backbone. We initialize the image encoder with pretrained weights of the diffusion backbone and keep it frozen during training. For the condition encoder, we initialize it as a trainable copy of the first 12 layers of the diffusion backbone. We train our model on 8 NVIDIA RTX A6000 GPUs.

Control feature injection Unlike the UNet backbone, the DiT architecture does not have skip connections. Therefore, we inject the output of the matching module directly into the hidden representations at each layer of the DiT backbone. Specifically, image features are extracted from 12 even-numbered layers of the image encoder, while query and support condition embeddings are obtained from the condition encoder. The matching mechanism is applied across these 12 layers to generate multi-layer control features. The 24-layer DiT diffusion backbone is divided into 12 sequential chunks, each containing 2 layers. The control features are then added to every hidden representation of these chunks in the corresponding order.

Hyper-parameters We follow the same optimizer settings, training dataset, and evaluation protocol described in Section 5.1. For inference, we use the flow-matching Euler scheduler introduced in Stable Diffusion 3[11], with a classifier-free guidance (CFG) scale of 5.0, 28 generation steps, and a fixed seed of 42.

Qualitative Results Qualitative examples using the DiT backbone are shown in Figures 9, 10. Our DiT-based model, adapted with only **30 shots**, is able to control the structure of generated images with various novel condition tasks.

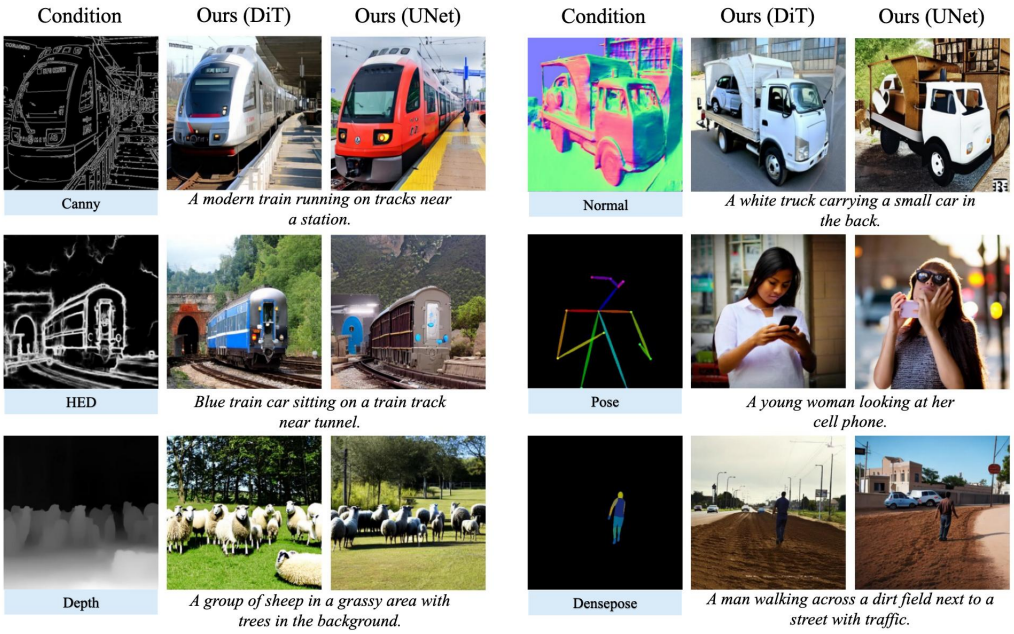

Figure 9: Qualitative comparison of UFC using DiT and UNet backbones in *30-shot* setting. Our method with the DiT backbone yields more fine-grained spatial control than the UNet counterpart.

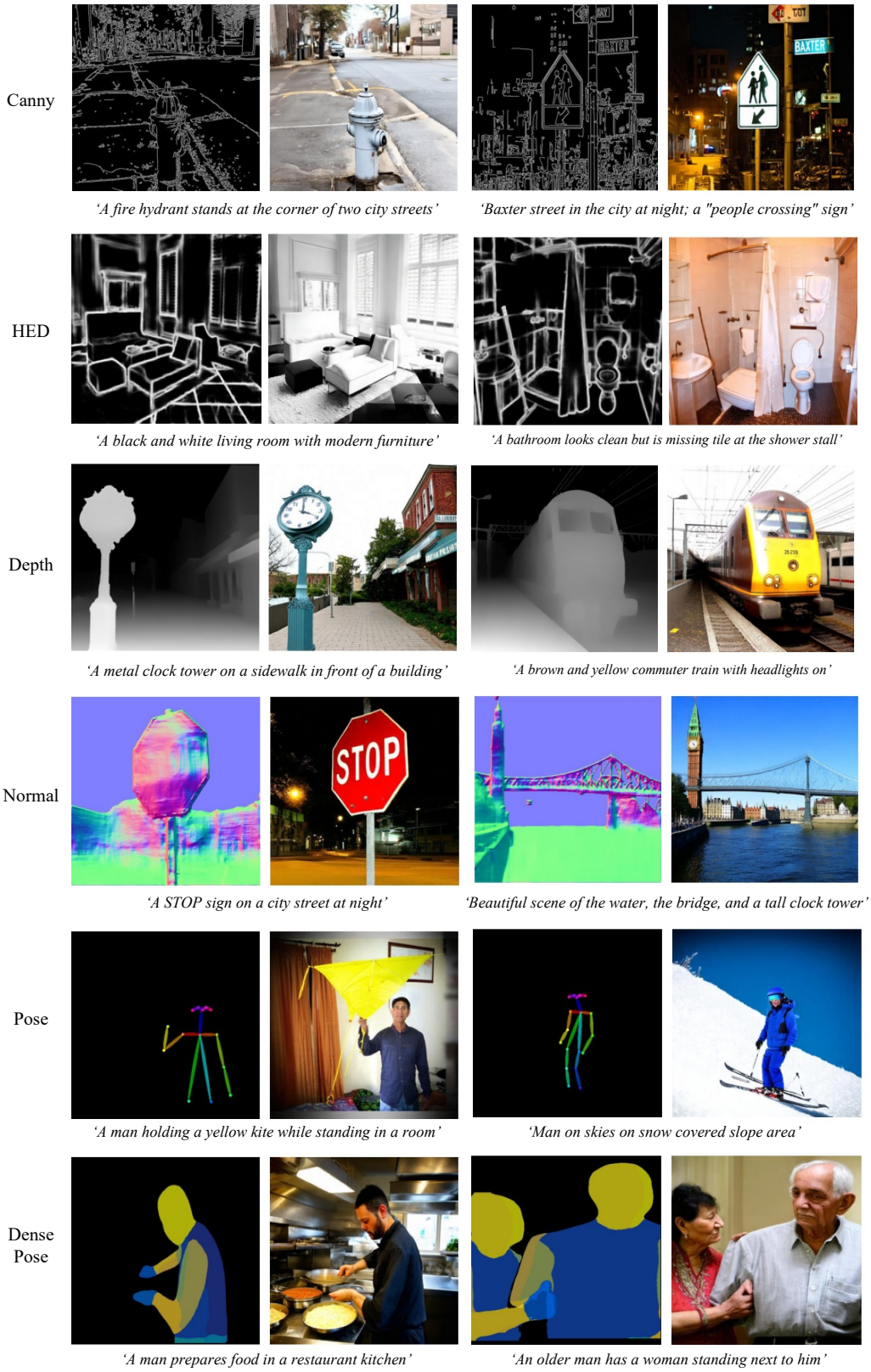

Figure 10: Generated images from UFC with DiT backbone in *30-shot* setting.

E More Results

Results with more spatial conditions As previously discussed in Section 5.3, we evaluate UFC on more novel spatial conditions, including 3D meshes, wireframes, and point clouds. We use iso3d [10], a 3D isolated object dataset(with no background), to generate spatial conditions for image pairs. We only report qualitative results, as the absence of a pre-trained condition-prediction network prevents us from measuring controllability using the evaluation approach in Section 5.1. Moreover, the insufficient collected validation sets hinder reliable assessment of image quality with FID [15] or Inception Score [44]. Figure 11 shows the qualitative results of our method, demonstrating the effectiveness of our method in a few-shot (30 supports) setting.

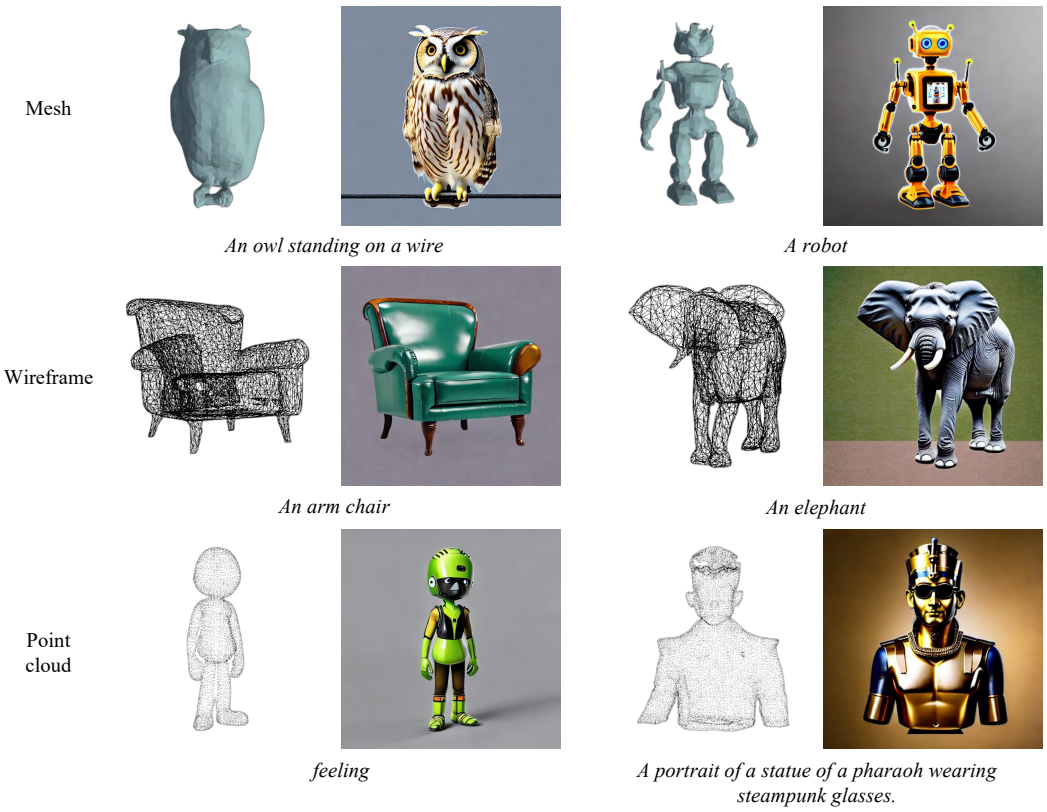

Figure 11: Generated images from UFC with spatial conditions of 3D meshes, wireframes, and point clouds in *30-shot* setting.

More qualitative results We present more qualitative results of UFC (UNet) in Figure 12 and Figure 13. All the results are generated with a support size of 30.

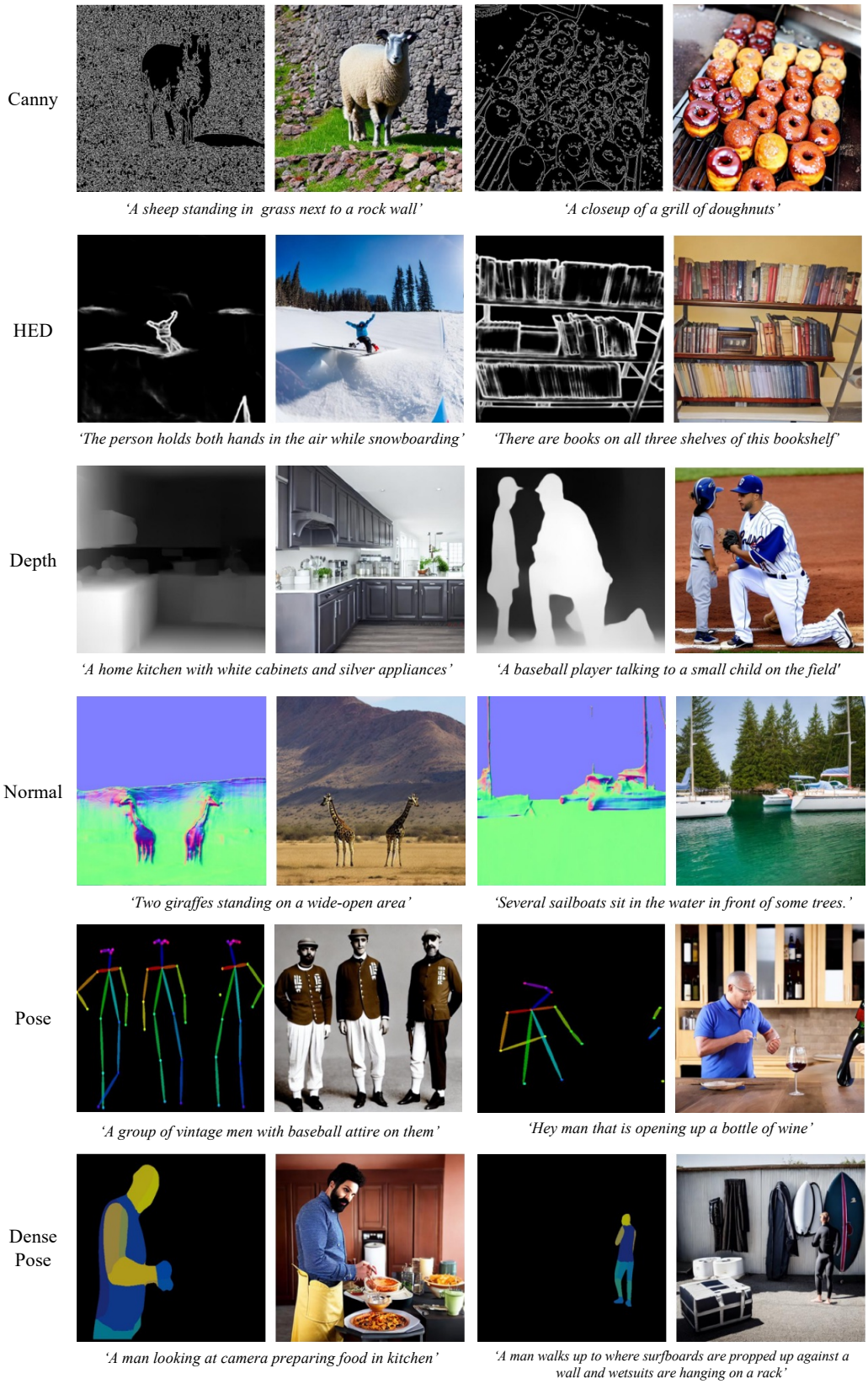

Figure 12: More qualitative results of UFC (UNet) in *30-shot* setting.

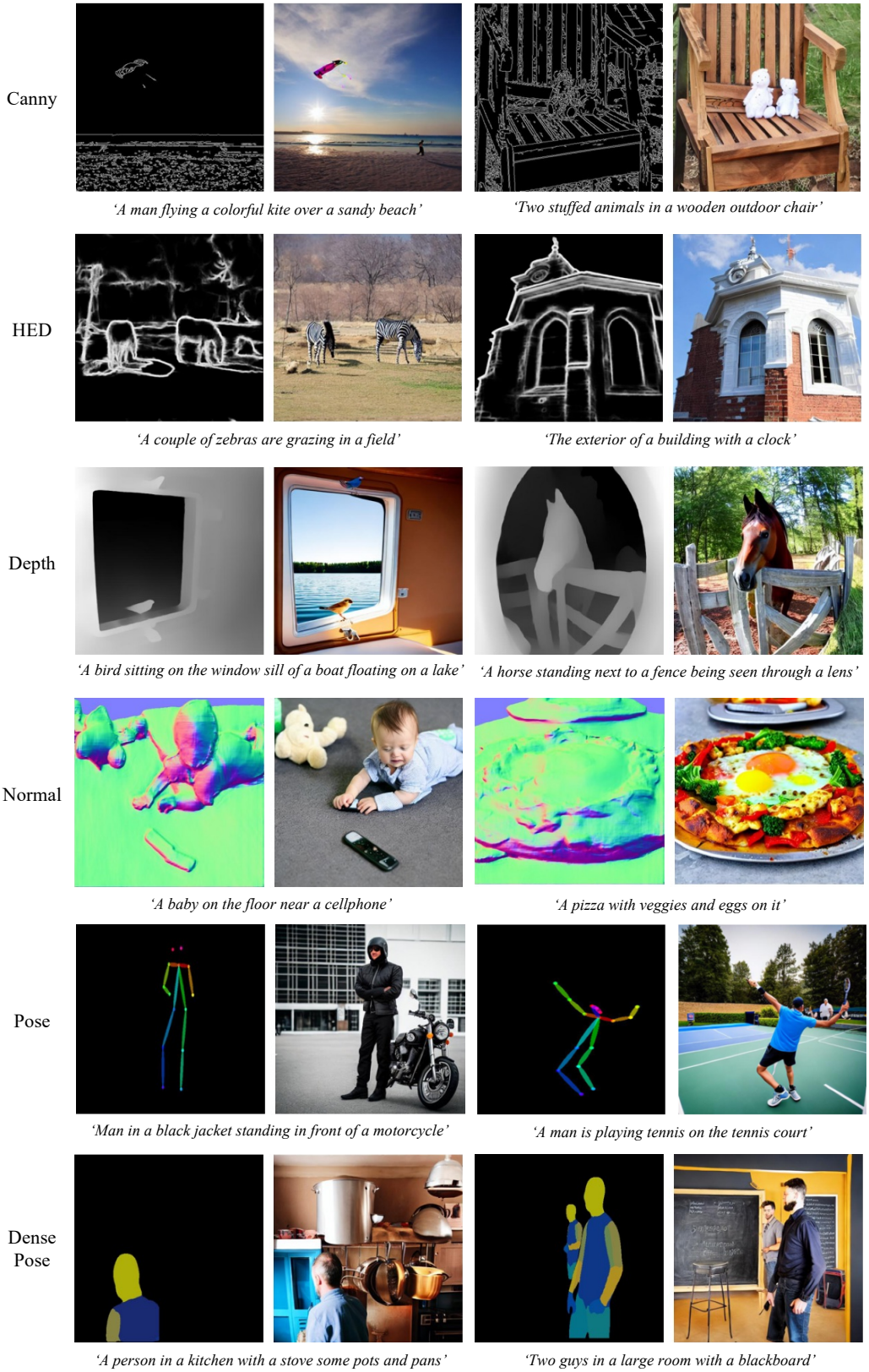

Figure 13: More qualitative results of UFC (UNet) in *30-shot* setting.

F Support Image-Condition Pairs

As described in implementation detail in Section 5.1, after few-shot fine-tuning, 5 image–condition pairs are used for inference.

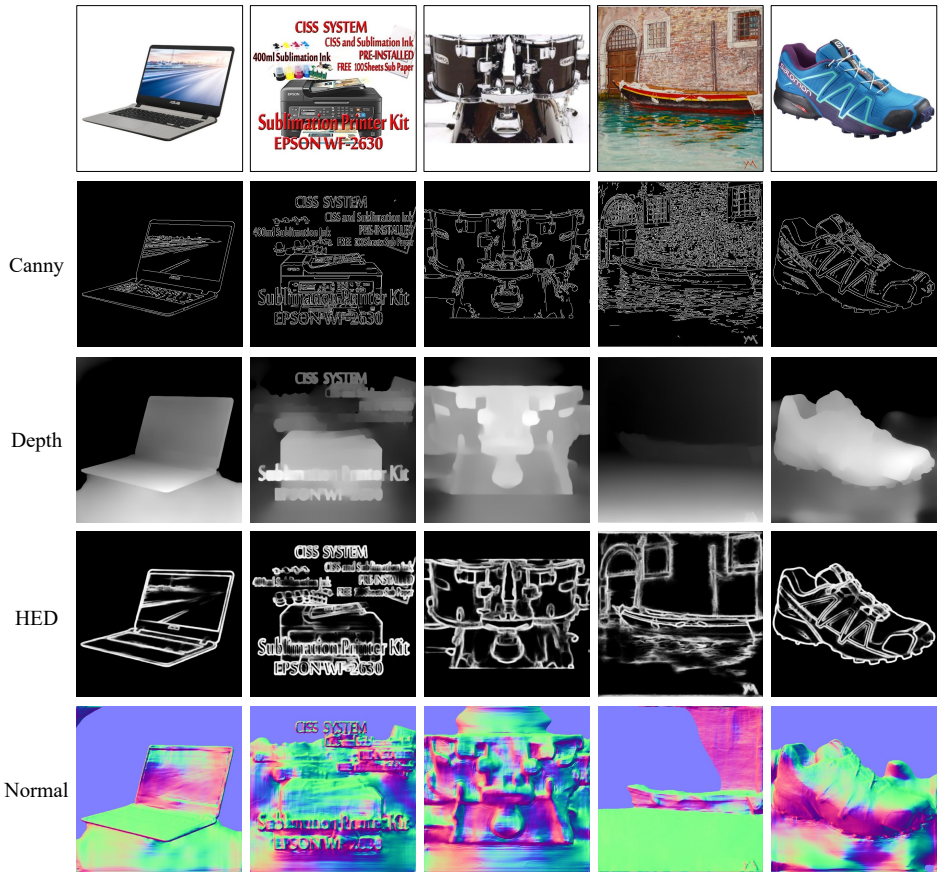

Figure 14: Five support image-label pairs used for evaluating Canny/Depth/HED/Normal tasks.

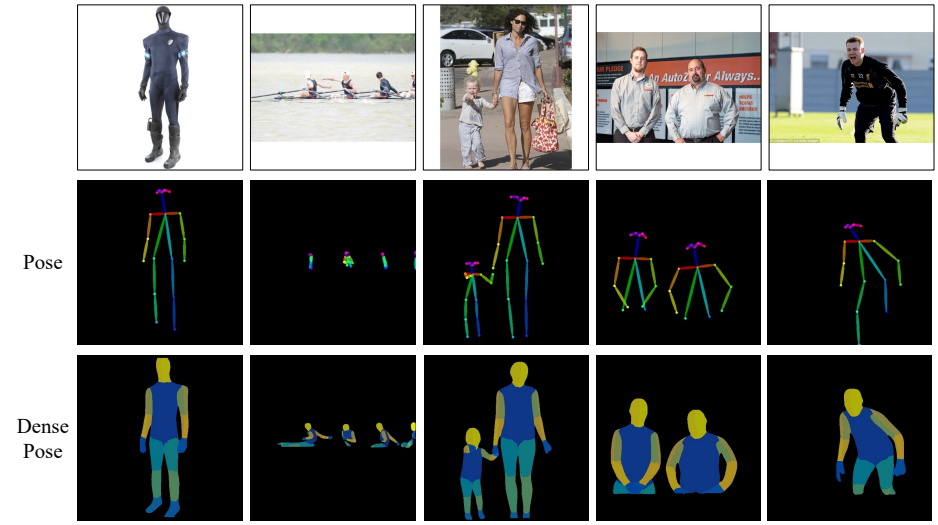

Figure 15: Five support image-label pairs used for evaluating Pose/DensePose tasks.