DreamCatcher:
Efficient Multi-Concept Customization via Representation Finetuning

Supplementary Material

A. Overview

The overview of this supplementary material is as follows:

- Details of Implementation in Section B.
- Additional Analysis on other Positions in Section C.
- Additional Ablation study in Section D.
- Analysis of the Effect of Concept Masks. Section E.
- Details of Personalized Image Inpainting in Section F.
- Hyperparameter Search Result in Section G.
- Interaction Case with Concepts in Section H.
- Training Time / VRAM Usage Comparison in Section I.
- Limitation & Failure Cases in Section J.
- Details of Experimental Hyperparameter in Section K.
- Additional Qualitative Results in Section L.

B. Implementation Details

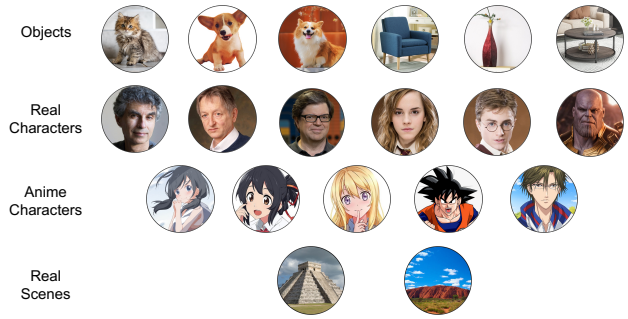


Figure A. Dataset Overview.

In this section, we detail the experimental setup. Overall, we follow Mix-of-Show [4] configuration for a fair comparison with prior work. As illustrated in Figure A, we evaluated the widely used Mix-of-Show dataset. The dataset consists of 19 items across four categories: six real objects, six real characters, five animated characters, and two scenes. The model checkpoint is based on Stable Diffusion [7]. Specifically, we utilize chilloutmix for real objects and characters and anything-v4 for animated characters. Our method employs layer-wise text embedding [8] as used in Mix-of-Show. The learning rates are set as follows: 1e-3 for text embedding, 1e-5 for the text encoder, 1e-4 for the U-Net, and 5e-5, 5e-4 for intervention scale. All image generation experiments are conducted using DPM Solver++ with 20 steps. For single-concept tuning, training is conducted with a batch size of 1 over 1,000 iterations on an A100 GPU. To evaluate multi-concept generation, we follow existing methods by employing sketch conditions for objects and pose conditions for real and anime characters, using T2I Adapter [6] to generate fixed images. Specifically, we use 50 sketches for objects, 13 poses for real characters, and 8 poses for animated characters, with each condition containing 2 to 5 concepts. To generate diverse images, we vary the sequence of concepts in pose conditions to produce multiple outputs. Subsequently, concept-specific masks are applied to crop and gather approximately 1,000 images, which are then used to compute CLIP scores for each concept. It provides numerous evaluation points for assessing merging performance.

C. Additional Analysis on other Positions

In this section, we analyze the impact of representation finetuning at different locations in the diffusion model. Figure B presents the rank analysis results of activations at various positions. Similar to self-attention, resblocks exhibit ranks between 20 and 50. In contrast, the key and value of cross-attention show low ranks, similar to the output of cross-attention. This finding suggests that the hidden representations of cross-attention, which take text as input, generally have low rank. However, as shown in Table A, applying interventions to the outputs of self-attention or the keys and values of cross-attention yields suboptimal performance. Furthermore, partial updates using concept masks are infeasible for keys and values. Thus, our approach of intervening on the output of cross-attention demonstrates superior parameter efficiency and effectiveness in multi-concept generation.

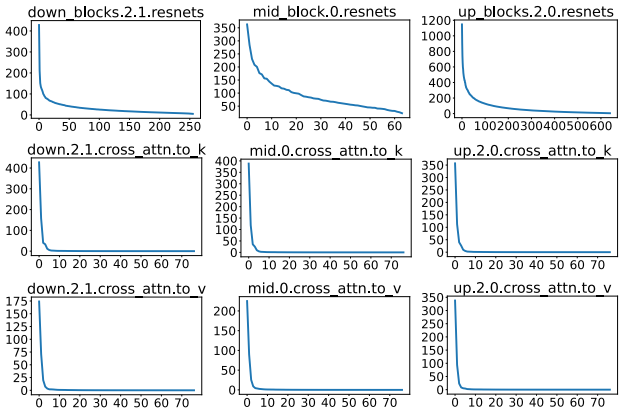


Figure B. SVD Rank Analysis of other positions. SVD top ranks (X-axis) and singular values (Y-axis).

Method	CLIP-I ↑	CLIP-T ↑
Self Attention Output	0.772	0.644
Cross Attention K	0.739	0.647
Cross Attention V	0.764	0.645
Cross Attention K, V	0.760	0.646
Ours	0.791	0.634

Table A. Comparison with other positions in diffusion model.

D. Additional Ablation Study

In this section, we conduct an ablation study to evaluate the impact of our Concept-Aware Intervention and Softmax Scheduling on multi-concept generation. First, we compare the performance of our approach against a baseline that applies interventions without Concept-Aware Intervention, using a method similar to FedAvg [1] as defined in Equation (A). This comparison allows us to assess how targeted interventions influence the quality and consistency of multi-concept generation.

$$\hat{h}^{(l)} = h^{(l)} + \frac{1}{N}\Sigma_{i=1}^{N}(\Phi_{c_i}^{(l)}(h^{(l)})) \tag{A}$$

As shown in Table B, Prompt-Aware Intervention provides modest improvements over FedAvg [1], while Region-Aware Intervention results in significantly better performance. These results demonstrate that Region-Aware Intervention, which updates only the concept mask regions, plays a crucial role in maintaining the integrity of individual concepts. When both interventions are used together, the best performance is achieved. Overall, the results demonstrate that our Concept-Aware Intervention outperform the baseline by preserving individual concepts more effectively while enabling seamless integration of multiple concepts into the generated images. This underscores the critical role of tailored interventions in achieving high-quality and coherent multi-concept outputs. Furthermore, integrating Softmax Scheduling further enhances performance, demonstrating the effectiveness of our approach.

Method	CLIP-I ↑	CLIP-T ↑
FedAvg	0.744	0.558
+ Prompt-aware Intervention	0.748	0.564
+ Region-aware Intervention	0.757	0.570
Ours (Both)	0.762	0.572
Ours (Both w/ Softmax Scheduling)	**0.764**	**0.574**

Table B. Ablation Study on Concept-aware intervention.

E. Effect of Concept Mask

In this section, we investigate the effects of concept masking within region-aware intervention by varying the concept mask. In Figure C, we adjusted the mask ratios to evaluate how intervention influences concept generation. At a balanced 5:5 ratio, both cat and dog concepts were clearly delineated. However, as the ratio became more skewed towards one concept, the generated details of that concept increasingly interfered with the other, which highlights the functionality of region-aware intervention.

(10:0) **(7:3)** **(5:5)** **(3:7)** **(0:10)**

Figure C. **Effect of Concept Mask.** The purple and yellow rectangles represent concept masks for the cat and dog, respectively.

F. Personalized Image Inpainting Algorithm

This section outlines the detailed algorithms for extending our method to personalized image inpainting. Similar to [2], our approach first acquires latent representations of the image using DDIM inversion. These latents are then utilized for resampling, during which concept-aware intervention is applied at the inpainting mask locations. This updates the hidden representations inside the masked region. However, since adjacent regions outside the mask may also experience minor changes via other layers, we ensure that updates are confined to the inpainting mask region by reapplying the latent obtained from DDIM inversion at each timestep. Additionally, applying this process from the initial steps risks altering the overall image structure. To mitigate this, the intervention process is applied only after some timesteps out of the full sampling schedule. Our inpainting algorithm is provided below:

Algorithm 1: Personalized Image Inpainting

Input: Source Image I, Inpaint Mask M, Model ϵ_θ, Intervention Start Timestep S
Output: Inpainted sample x_0
$\hat{x}_{0:T} = \text{ddim\_inversion}(I)$
$x_T = \hat{x}_T$
for $t = T, T-1, ..., 1$ **do**
  $x_t \leftarrow M \otimes x_t + (1 - M) \otimes \hat{x}_t$
  if $t > S$ **then**
    $\epsilon_t \leftarrow \epsilon_\theta(x_t, t)$
  else
    $\epsilon_t \leftarrow \epsilon_\theta^{\text{Intervention}}(x_t, t, M)$
  $x_{t-1} \leftarrow \text{Update}(x_t, \epsilon_t)$
return x_0

G. Hyperparameter Search

This section presents experiments on the selection of hyperparameters used in our method.

1) ReFT Rank

Table C presents the results of rank selection for the ReFT adapter. In this table, TxUy denotes the ranks for the Text encoder (x) and U-Net (y). Our setting, T4U8, represents a sweet spot in terms of both memory efficiency and output quality.

	T4U4	T8U4	**T4U8 (Ours)**	T8U8	T16U16
CLIP-I	0.7512	0.7621	**0.7643**	0.7650	0.7625
CLIP-T	0.5763	0.5774	**0.5765**	0.5761	0.5768
Memory	0.9MB	1.2MB	**1.3MB**	1.6MB	2.9MB

Table C. Ablation Study on rank for Text Encoder and U-Net.

2) Softmax Scheduling β

Table D shows the results of varying the Softmax Scheduling parameter β, which controls the magnitude of attention scores. Our setting, $\beta = 0.6$, achieves the best performance, demonstrating that appropriately scaling attention scores is effective for generation quality.

	β=0.0	β=0.2	β=0.4	β=0.6 **(Ours)**	β=0.8
CLIP-I	0.7562	0.7588	0.7622	**0.7643**	0.7633
CLIP-T	0.5793	0.5781	0.5776	**0.5765**	0.5722

Table D. Ablation Study on Softmax Scheduling parameter β.

H. Interaction with other concepts

This section presents examples where generated subjects interact with each other. Like the handshaking case in Figure D, using region masks to separate areas enables high-quality outputs even with subject interactions. In more complex cases, bounding boxes may overlap, but incorporating fine-grained masks (e.g., SAM [5]) helps better mitigate this limitation, as shown in our inpainting experiments. These results further validate the effectiveness of our region-aware control and feature-level modularity.

Figure D. Example of Interaction with Multiple Concepts.

I. Training Time / VRAM Usage Comparison

In our method, training takes 8–10 minutes on a single A100 GPU, with VRAM usage of 24GB for training and 10GB for inference. This is practical on customer-grade GPUs and comparable to Mix-of-Show and Orthogonal Adaptation. Furthermore, when combined with quantization-aware finetuning (e.g., QLoRA [3]), the computational cost can be further reduced to meet edge-device constraints. In contrast, Mix-of-Show requires slow and memory-intensive weight fusion at inference, and LoraComposer demands 51GB of VRAM, limiting its use to high-end GPUs. In contrast, Our method is free from such constraints and thus more suitable for practical deployment.

Method	**Training**		**Inference**	
	Time	VRAM	Time	VRAM
Mix-of-Show	8-10m	24GB	5s (+15m)	10GB
LoRAComposer	8-10m	24GB	25s	51GB
Ours	8-10m	24GB	5.4s	10GB

Table E. Comparison of Training and Inference Resources.

J. Limitation & Failure Cases

Our method addresses the limitations of conventional modular customization by enabling highly efficient and rapid multi-concept customization through representation finetuning. Nonetheless, several limitations remain and suggest directions for future work. First, our approach depends on Region-aware cross-attention [4] and Region-aware Intervention, which utilize predefined region masks. As shown in Figure E, this occasionally leads to artifacts such as line separations between regions in the generated images. Additionally, overlapping masks may cause the blending of concepts, posing a risk to concept integrity. Both issues stem from the use of bounding box-like masks to define concepts. These challenges could potentially be resolved by replacing bounding box masks with fine-grained masks (e.g, SAM [5]) allowing for more precise delineation of concepts and reducing the likelihood of artifacts or concept mixing. However, in cases where bounding boxes completely overlap and mask interference arises, these approaches remain inherently limited, underscoring the need for further research.

Additionally, our feature-space modulation is theoretically agnostic to the number of concepts. However, when more than five concepts are added to a single image, the reduced size of each region may weaken identity representation. While increasing the resolution can alleviate this issue, it comes at the cost of higher inference overhead.

Figure E. Failure Cases for Multi Concept Generation.

K. Experiment Details

	Training Iterations	Optimizer	batch size	LR (Text embeds)	LR (Text encoder)	LR (U-Net)	weight decay	LR Scheduler
P+	500	AdamW	1	1e-3	-	-	0.01	Linear
Custom Diffusion	250	AdamW	1	-	-	1e-4	0.01	Constant
LoRA	500	AdamW	1	5e-4	1e-5	1e-4	0.01	Linear
Mix-of-Show	500	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear
Orthogonal Adaptation	500	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear
DreamCatcher (Ours)	500	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear

Table F. Training Hyperparameters for objects Experiments.

	Training Iterations	Optimizer	batch size	LR (Text embeds)	LR (Text encoder)	LR (U-Net)	weight decay	LR Scheduler
P+	1000	AdamW	1	1e-3	-	-	0.01	Linear
Custom Diffusion	500	AdamW	1	-	-	5e-5	0.01	Constant
LoRA	1000	AdamW	1	5e-4	1e-5	1e-4	0.01	Linear
Mix-of-Show	1000	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear
Orthogonal Adaptation	1000	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear
DreamCatcher (Ours)	1000	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear

Table G. Training Hyperparameters for Real-Character Experiments.

	Training Iterations	Optimizer	batch size	LR (Text embeds)	LR (Text encoder)	LR (U-Net)	weight decay	LR Scheduler
P+	1000	AdamW	1	1e-3	-	-	0.01	Linear
Custom Diffusion	500	AdamW	1	-	-	5e-5	0.01	Constant
LoRA	1000	AdamW	1	5e-4	1e-5	1e-4	0.01	Linear
Mix-of-Show	1000	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear
Orthogonal Adaptation	1000	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear
DreamCatcher (Ours)	1000	AdamW	1	1e-3	1e-5	1e-4	0.01	Linear

Table H. Training Hyperparameters for Anime-Character Experiments.

L. Additional Qualitative Results

Prompt: "<V_1> and <V_2> at the <V_3>, 4K, high quality, high resolution, best quality"

Prompt: "<V_1> and <V_2> at <V_3>, 4K, high quality, high resolution, best quality"

Prompt: "<V_1>, <V_2>, and <V_3> in the office, 4K, high quality, high resolution, best quality"

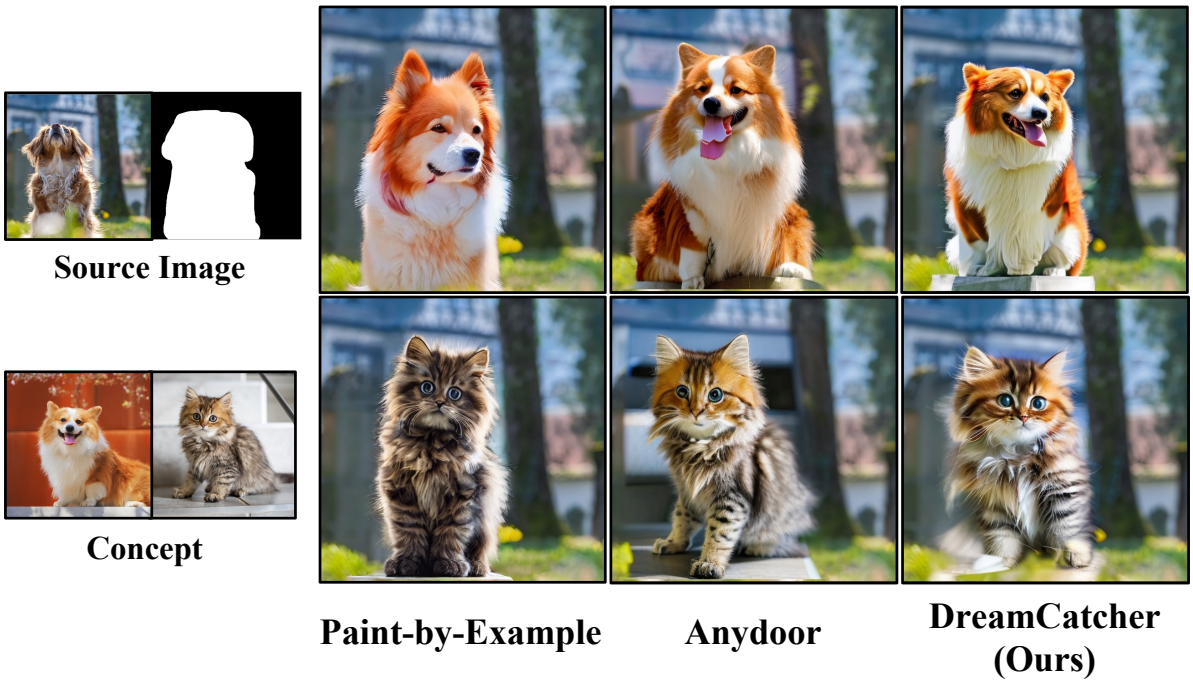
Figure F. Multi-Concept Generation Results (Real Characters)

Prompt: "<V_1>, <V_2>, and <V_3> walking near a lake."

Prompt: "<V_1>, <V_2>, and <V_3> walking near a lake."

Prompt: "<V_1>, <V_2>, <V_3>, <V_4>, and <V_5> near a lake."

Figure G. Multi-Concept Generation Results (Anime Characters).



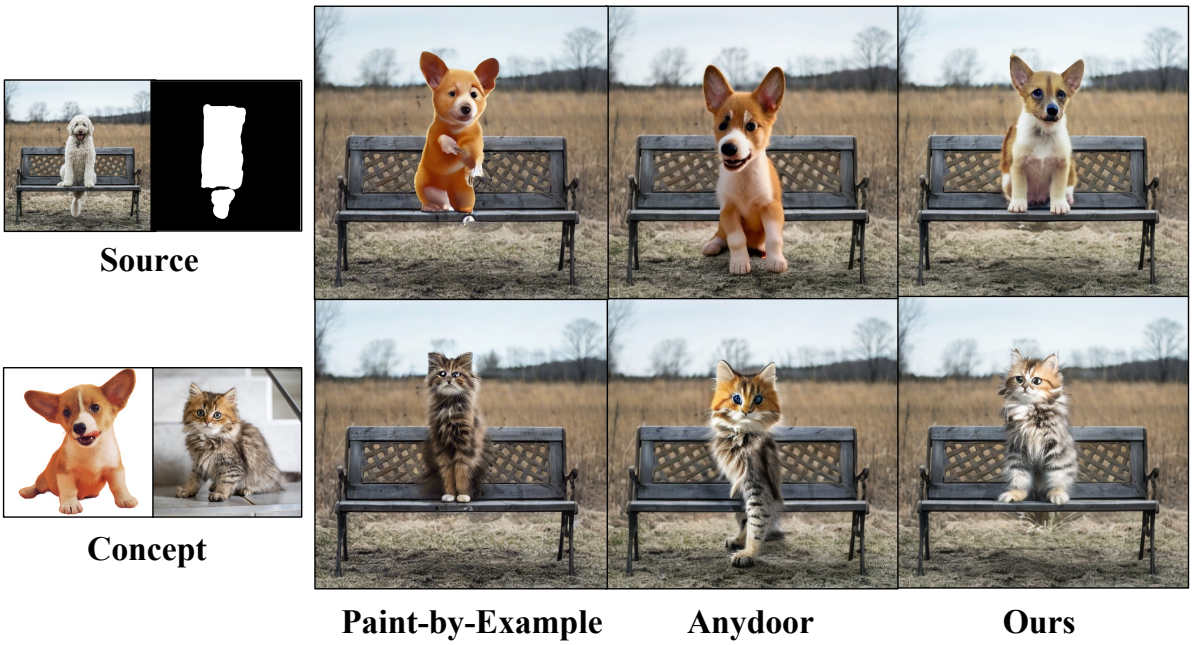


Figure H. **Personalized Image Inpainting Result.** The qualitative comparison between ours and existing inpainting methods.

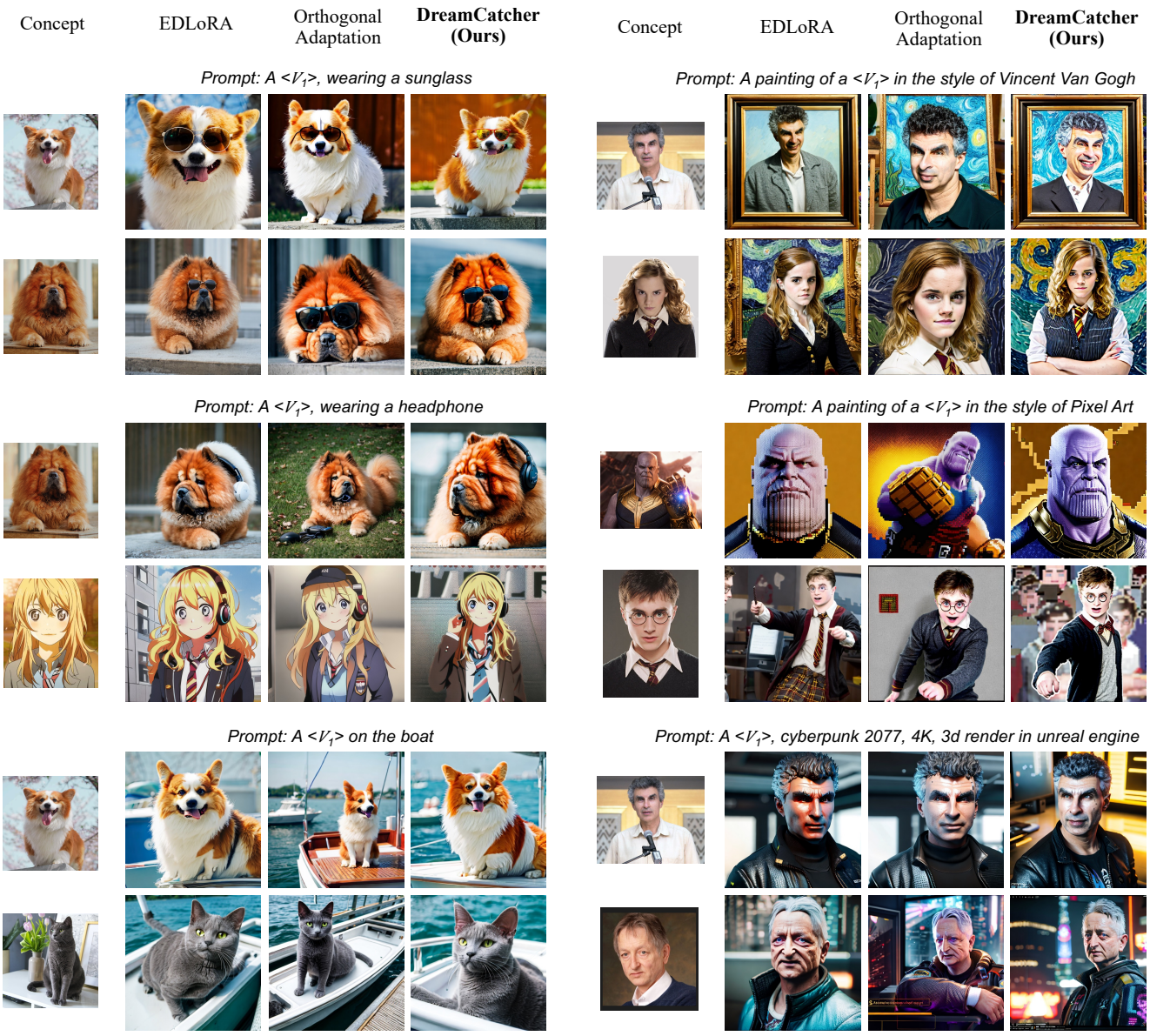


Figure I. **Results on various, complex concepts.** These results include abstract, compositional, or stylistic prompts.

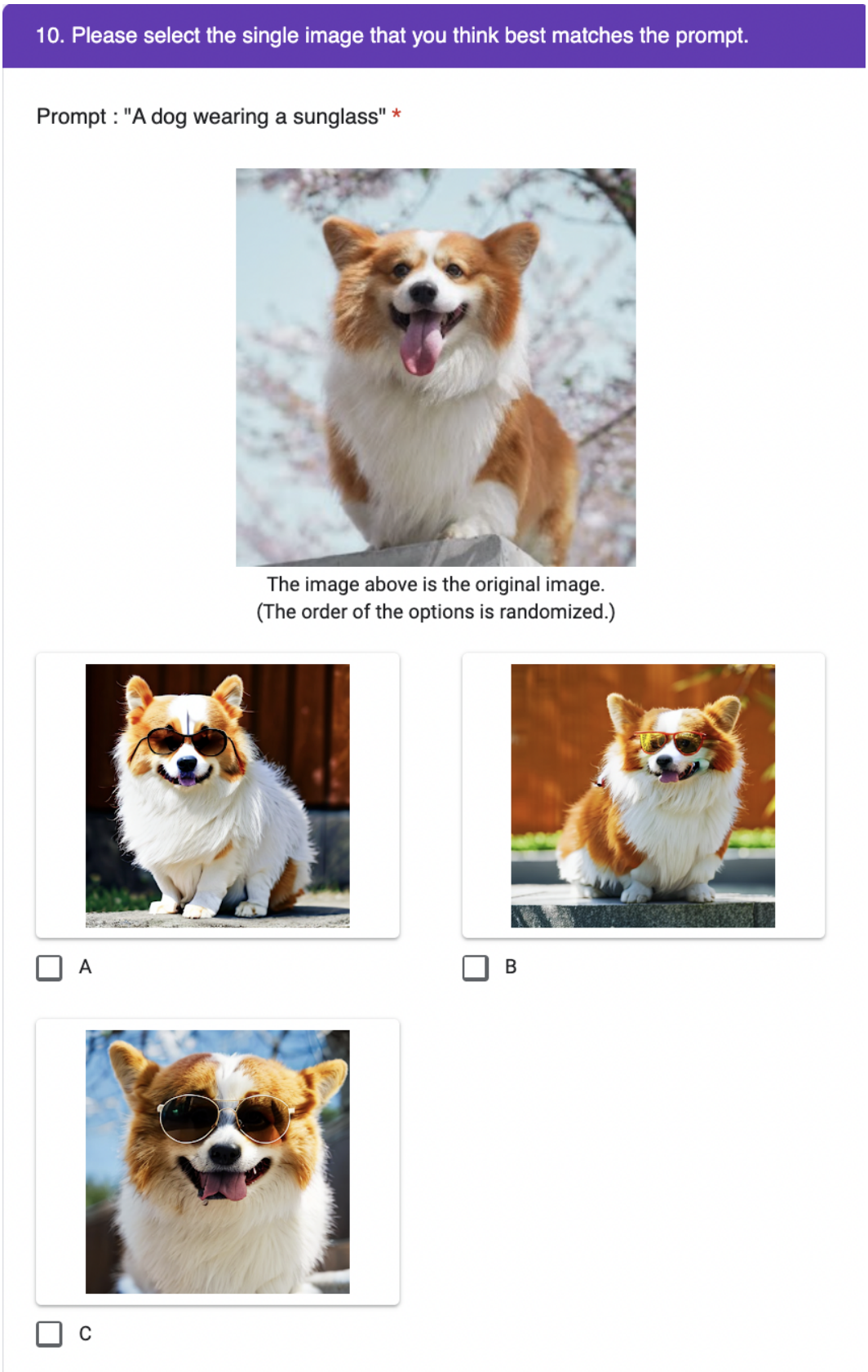


Figure J. **Example of the User Survey.**

References

[1] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021. 2

[2] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022. 2

[3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36: 10088–10115, 2023. 3

[4] Yuchao Gu, Xintao Wang, Jay Zhangjie Wu, Yujun Shi, Yunpeng Chen, Zihan Fan, Wuyou Xiao, Rui Zhao, Shuning Chang, Weijia Wu, et al. Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[5] Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36: 29914–29934, 2023. 3

[6] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4296–4304, 2024. 1

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[8] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. p+: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 1