A. Implementation details

We apply the proposed methods to the text-to-image diffusion model, Stable Diffusion [35] using checkpoint v1.5. We begin by inverting real images with the edit-friendly DDPM inversion [20], sampling images with 100 denoising timesteps. To find semantic correspondence during transfer, we use the feature maps input to the self-attention layer. We set the denoising step t ∈ [42, 100] and layer l ∈ [2, 3] from the up-blocks of U-net to find correspondences and rearrange features. Additionally, we apply AdaIN at denoising step t ∈ [82, 100] and use the off-the-shelf model SAM [21] to obtain object masks. And we measure dense correspondence at timestep 92 and layer 2. All of the experiments are conducted on an NVIDIA A6000 GPU and during the transfer experiments, the GPU memory usage amounted to about 15.17 GB.

B. Evaluation method details

B.1. Appearance similarity

To evaluate the success of transferring the appearance of the reference image, we conduct an experiment comparing the color histograms (A_{hist}) of the result image and the ground truth (GT) image. The comparison region is set by segmenting the object using SAM [21] for both the GT and result images. For the comparison of color histograms, we measure the Bhattacharyya distance as:

$$A_{\text{hist}}(H_G, H_O) = D_B(H_G, H_O) \tag{5}$$

where $D_B(H_G, H_O)$ is the Bhattacharyya distance between the color histograms of the masked GT image (H_G) and the masked transferred output image (H_O).

Additionally, we measure semantic similarity using CLIP score:

$$A_{\text{clip}}(G, O) = \frac{1}{N}\sum_{i=1}^{N} \text{CLIP}(G_i, O_i) \tag{6}$$

where G_i and O_i are the masked GT and masked transferred output images, respectively, and N is the total number of images.

The dataset used in the experiments is described in Appendix C.

B.2. Structure preservation

To evaluate the preservation of the target image's structure, we conduct a depth evaluation (I_{depth}), a mean Intersection over Union (mIoU, S_{miou}) and a key point evaluation (S_{key}).

For S_{depth}, we use an off-the-shelf depth estimation model [33]. We extract depth from the target image and the transferred results of each model, then measure the root mean square error (RMSE) at the object level:

$$S_{\text{depth}}(D_T, D_O) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (D_{T,i} - D_{O,i})^2} \tag{7}$$

where D_T and D_O are the depth maps of the masked target image and the transferred output image, respectively, and N is the total number of pixels.

For S_{miou}, we use SAM to obtain the masks of the ground truth (GT) and the transferred result objects. The mIoU is then measured at the object level as:

$$S_{\text{miou}}(T, O) = \frac{1}{N}\sum_{i=1}^{N} \frac{|M_{T,i} \cap M_{O,i}|}{|M_{T,i} \cup M_{O,i}|} \tag{8}$$

where T and O denote the target and output images, M represents the object mask obtained from SAM-HQ, and N is the total number of objects.

To follow the default settings of the models, ours, Cross-Image [1], DiffEditor [29], Splice VIT [39], IP-Adapter [44], and ZeST [8] are tested at an image resolution of 512^2. Swapping AE [30] and DiffuseIT [23] are tested at a resolution of 256^2.

For S_{key}, we assess structural preservation through pose estimation with ViTPose++ [42]. Following its approach, we evaluate AP-10K samples [AP-10K] from the training set and compute Average Precision (AP) using Object Keypoint Similarity (OKS) over thresholds $\tau \in [0.5, 0.95]$ with target keypoints as ground truth. Our method achieves higher AP than competitors, demonstrating superior structural retention. OKS is defined as:

$$\text{OKS} = \frac{\sum_i \exp\left(-\frac{d_i^2}{2s^2\kappa_i^2}\right)\delta(v_i > 0)}{\sum_i \delta(v_i > 0)}, \tag{9}$$

where d_i is the Euclidean distance between the detected and ground truth keypoints, s is the object scale, κ_i is a keypoint-specific constant, and v_i is the keypoint visibility.

Using OKS, the Average Precision (AP) score is computed as:

$$\text{AP} = \frac{1}{|\tau|}\sum_{\tau} \text{Precision}(\tau). \tag{10}$$

The precision at each threshold τ is given by:

$$\text{Precision}(\tau) = \frac{|\{\text{detected keypoints} \mid \text{OKS} \geq \tau\}|}{|\{\text{all detected keypoints}\}|}. \tag{11}$$

The dataset used in the experiments is described in Appendix C.

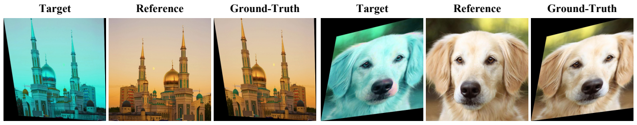


Figure S1. Examples of building and AFHQ for I_{hist}.

B.3. Dense correspondence

We evaluate dense correspondence using flow maps, which represent pixel displacements derived from the correspondences estimated by each method. These flow maps are computed by subtracting the difference between the target pixel coordinates from their corresponding matches. To measure deviations from the GT flow map, we calculate the L1 distance at the resolution of 512^2 as,

$$D_{\text{flow}}(F_{\text{pred}}, F_{\text{GT}}) = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum|F_{\text{pred},i} - F_{\text{GT},i}|}{|\mathcal{M}_i|} \tag{12}$$

where N is the total number of images, $F_{\text{pred},i}$ and $F_{\text{GT},i}$ are the predicted and ground truth optical flow for image i, respectively, and $\mathcal{M}_i$ is the validity mask indicating the valid pixels in the flow.

SD-DINO [48] and Telling-Left-from-Right [47] employ both 960^2 and 840^2 image resolutions to extract feature descriptors across two distinct models, and ours utilizes a resolution of 512^2. Source and target images of varying sizes are resized to the input resolution required by each method, following the padding strategy detailed in the official implementation of SD-DINO [48]. Both ours and SD-DINO [48] compute dense correspondence by upsampling feature maps to 512^2. Telling-Left-from-Right [47] derives dense correspondence with feature maps at their original resolution (60^2), using a window-soft-argmax operation, and subsequently upsamples the correspondence map to 512^2. The dataset used in the experiments is described in Appendix C.

C. Evaluation dataset

For the quantitative evaluation, we used the AFHQ [9], AP-10K [45], and TSS [38] datasets, and a Building dataset collected from the Pexels[1]. This dataset will be publicly available. Especially, as shown in Fig. S1, to evaluate appearance transfer performance, we created datasets for A_{hist} and A_{clip} with the following setup: (1) Reference: original image (2) Target: shape and color-augmented image derived from the original image (3) Ground-Truth (GT): shape augmented image derived from the original image. We perform appearance transfer on 1) Reference to (2) Target, and measure the score by comparing the result object with (3) GT object. To align with the training domain of the pre-trained Swapping AE [30], we applied flip and weak warping as augmentations. Additionally, to evaluate structure preservation, we use a building dataset comprising 30 pairs of structure and target images, as well as an AFHQ dataset with 42 pairs. We evaluate dense correspondence on the TSS dataset [38], which includes dense correspondence flows and semantic masks for 400 image pairs sampled from the FG3DCAR [25], JODS [36], and PASCAL [16] datasets.

D. Baseline settings

All experiments were conducted at a resolution of 512^2, except for the Swapping AE and DiffuseIT, which were trained at a resolution of 256^2.

D.1. For appearance transfer comparison

Cross-Image. Cross-Image [1] employs edit-friendly DDPM inversion [20] for image inversion. Images are sampled with 100 denoising timesteps. And Cross-Image does not use an object mask during transfer, so the background of the target is not preserved after the transfer. The KV injection in self-attention occurs at t ∈ [42, 100] and layer l ∈ [2, 3] from the up-blocks. The contrast strength is set to 1.65, and the swap guidance scale is set to 3.5. Additionally, for consistency with our model, experiments were conducted using Stable Diffusion v1.5.

DiffEditor. DiffEditor is experimented with under Stable Diffusion v1.5. We use the standard DDIM scheduler for 50 denoising steps. The classifier-free guidance scale was set to the default value of 5. And Diffeditor uses an object mask during transfer, so the background of the target is preserved.

DiffuseIT. DiffuseIT [23] utilizes external models to guide the denoising process. We set the denoising timestep to 200, skipping the initial 80 timesteps, and use a resampling step of N=10 (resulting in a total of 130 iterations). Images are resized to a resolution of 224^2 to compute the ViT and CLIP losses, as these models only accept this resolution. These settings are the default configuration for image-guided manipulation as specified by the authors. Additionally, other configurations, including hyperparameters, follow the default settings provided by the authors. Since the provided checkpoint is trained at a resolution of 256^2, we also conducted experiments at this resolution.

Splice ViT. Splice ViT [39] employs a pre-trained DINO ViT model [42] as a feature extractor for optimizing the model on a single image pair. We use the 12-layer pre-trained ViT-B/8 model provided in the official DINO ViT implementation. For the ViT loss, images are resized to a resolution of 224^2. Keys are derived from the deepest attention module for self-similarity, and the output of the deepest layer is used to

[1] https://www.pexels.com/

extract the appearance from the target appearance image. We optimize using an input image pair with a resolution of 512^2 for 2000 iterations. These settings follow the default settings provided by the authors, and other configurations, including hyperparameters, also follow the provided configurations.

Swapping AE. We use the pretrained checkpoints provided on the official GitHub repository. We evaluate the AFHQ dataset and the LSUN Church pretrained models, treating the Building dataset as in-domain for LSUN Church model. In all evaluations, the target image is treated as the structure image, and the reference image is treated as the texture image. Additionally, we set the texture mixing alpha to 1.0, i.e,. simple texture swapping.

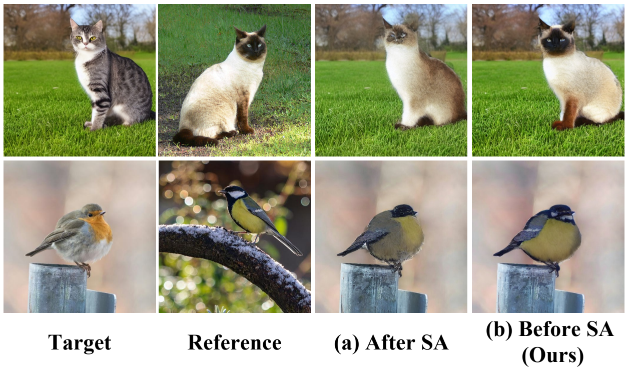
IP-Adapter. To account for target depth, we adopt the IP-Adapter + ControlNet model, using SDXL as the base model. The target image's depth map is extracted using off-the-shelf depth estimator [33], normalized, and then used as a condition for ControlNet. The reference image is provided as the input image. The ControlNet conditioning scale is set to 0.7, and the DDIM step is set to 30, following the inference settings from the official repository.

ZeST. ZeST utilizes Dense Prediction Transformers [34] for depth estimation and Rembg [13] for foreground extraction. It also employs Stable Diffusion XL Inpainting in conjunction with the corresponding version of depth-based ControlNet and IP-Adapter. Additionally, all other configurations, including hyperparameters, follow the default settings provided by the authors.

D.2. For semantic correspondence comparison

SD-DINO. SD-DINO [48] employs Stable Diffusion v1.5 with a diffusion model timestep of $t = 100$ as the visual descriptor, while integrating DINOv2 [6] as an auxiliary descriptor. Stable Diffusion features are extracted from the 2nd, 5th, and 8th layers of the U-Net decoder at timestep $t = 50$, while DINOv2 descriptors are derived from the token facet of its 11th layer. The input resolutions are 960^2 for Stable Diffusion and 840^2 for DINOv2, resulting in a feature map with a resolution of 60^2. Then, we use 512^2 upsampled feature map to find semantic correspondence.

Telling-Left-from-Right. Telling-Left-from-Right [47] adopts Stable Diffusion and DINOv2 features in a manner similar with SD-DINO. Furthermore, it incorporates the instance matching distance (IMD) to compare the target image with the horizontally flipped source image, thereby mitigating pose variation in paired images. Semantic correspondence is computed on the 60^2 resolution map using window soft argmax with a window size of 10, followed by upsampling to 512^2 for evaluation.



Target **Reference** **(a) After SA** **(b) Before SA (Ours)**

Figure S2. **Qualitative comparison between the results transferred using features after the self-attention layer and ours.** (a) Results transferred using features after the self-attention layer. (b) Results transferred using features before the self-attention layer (ours). (a) shows mismatched semantic correspondence, while (b) demonstrates accurate semantic correspondence.

Metrics	A_{hist} ↓	
Dataset	Building	AFHQ
After SA	0.478	0.581
Before SA(Ours)	**0.469**	**0.577**

Table S1. **Comparison of appearance similarity on different feature positions.** For all datasets, the appearance similarity of transferred results using features before the self-attention layer shows a lower I_{hist} compared to those transferred using features after the self-attention layer. We mark the best score in bold.

E. Ablation study for feature positions

We use the input features of the self-attention layer for correspondence measurement and feature injection. However, the output of the self-attention layer can also be used for correspondence measurement. Through experiments, we confirm that the input features to self-attention yield better performance. Fig. S2 (a), which uses the self-attention output features, shows less accurate matching compared to Fig. S2 (b), which uses the self-attention input features. And as shown in Tab. S1, the transferred results using input features better preserve the reference appearance compared to those using output features.

F. Ablation study for time steps

Our method measures dense correspondence at timestep 92. Because our method performs sparse key-point matching in the early steps and dense matching in the later steps. As shown in Tab. S2, the flow map distance is lower in the later steps compared to the early steps. It demonstrates that dense correspondence is more effectively measured in the later steps than in the early ones.

Time Step \ Dataset	$D_{\text{flow}} \downarrow$		
	FG3D CAR	JODS	PASCAL
62	10.75	32.86	28.37
77	9.71	30.16	24.72
92(Ours)	**9.43**	**28.75**	**21.83**

Table S2. **Comparison of dense correspondence on different time steps.** For all datasets, the dense correspondence measured at later time step shows a lower flow map distance compared to that measured at mid time step. We mark the best score in bold.

Method	$S_{\text{miou}} \uparrow$		$A_{\text{hist}} \downarrow$	
Component \Dataset	Building	AFHQ	Building	AFHQ
Baseline(KV injection)	0.833	0.926	0.495	0.603
+Feature rearrange	0.942 **(+0.109)**	0.968 **(+0.038)**	0.484 (-0.009)	0.582 **(-0.021)**
+ AdaIN(Ours)	0.939 (-0.003)	0.972 (+0.004)	0.469 **(-0.015)**	0.577 (-0.005)

Table S3. **Quantitative ablation results.** We mark the greatest difference in scores between the components in bold.

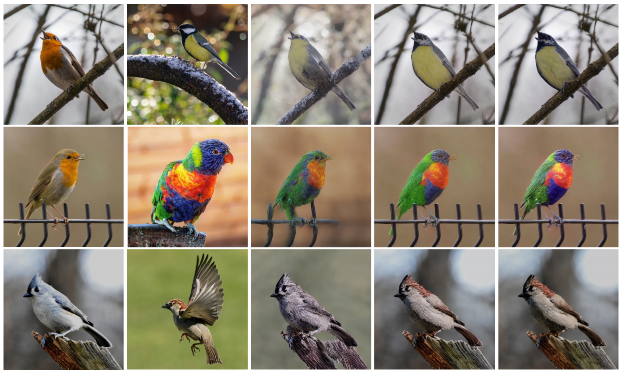

(a) Target image (b) Reference image (c) Using *K/V* injection (d) Using Semantic-based feature rearranging (e) + AdaIN [ours]

Figure S3. Additional samples of ablation study.

G. Quantitative ablation results for each component

We add the below table to provide quantitative ablation for Fig. H. Feature rearrange is our core component and mainly helps structure S_{miou}). AdaIN adjusts color distribution (A_{hist}).

H. Additional examples on ablation study

We present additional samples from the ablation study analyzing the effects of each component of our model in Fig. S3.

	Ours	DiffEditor	ZeST	Others
For mask	SAM-HQ [22]	EfficientSAM [41]	Rembg [13]	X

Table S4. **Approaches for obtaining object masks.** The table compares the approaches used to obtain object masks in Ours, DiffEditor, and ZeST. Others refer to other baselines, including Cross-Image, DiffEditor, DiffuseIT, SpliceViT, Swapping AE, and IP-Adapter. These baselines do not utilize object masks.

Metrics	$A_{\text{hist}} \downarrow$		$S_{\text{miou}} \uparrow$	
Dataset	Building	AFHQ	Building	AFHQ
Ours *	0.469	0.577	0.939	0.972
Ours *w/o mask*	0.467	0.579	0.858	0.943
Cross-Image	0.491	0.608	0.758	0.915
DiffEditor *	0.478	0.608	0.863	0.943
DiffuseIT	0.477	0.607	0.855	0.951
Splice ViT	0.472	0.580	0.842	0.943
Swapping AE	0.481	0.629	0.821	0.942
IP-Adapter	0.487	0.616	0.642	0.950
ZeST *	0.497	0.602	0.925	0.980

Table S5. **Comparison of appearance similarity and structure preservation for our model without a mask.** Ours *w/o mask* refers to our method without using an object mask. * indicates a model using a mask. We mark the best score in red and the second-best score in yellow.

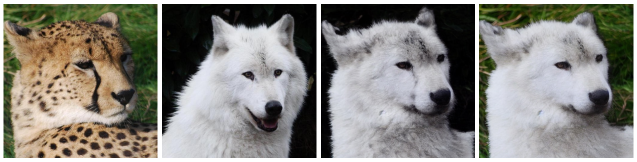

Target Reference (a) Ours *w/o mask* (b) Ours

Figure S4. **Qualitative comparison between our model without an object mask and the our model.** (a), which does not apply the object mask during transfer, fails to preserve the background of the target image, whereas (b), with the mask applied, successfully retains the background.

I. Ablation study for object mask

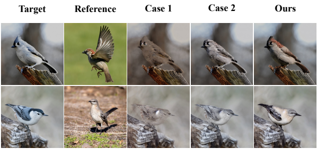
This section evaluates the role and effectiveness of object masks in appearance transfer tasks. Tab. S4 summarizes the approaches for obtaining object masks adopted by Ours and each baseline. Ours, DiffEditor, and ZeST utilize object masks during the transfer process, while other competitors do not incorporate object masks in their transfer processes.

To analyze the impact of object masks, we conduct experiments with our method without using an object mask. As shown in Tab. S5, the performance of Ours *w/o mask* decreases in structure preservation compared to ZeST and DiffEditor, which use object masks. This result demonstrates that object masks are effective in maintaining the structure of the target object. Among competitors that do not use object masks, Ours *w/o mask* achieves the best structure preservation. Regarding appearance similarity, our model maintains strong performance even without a mask, owing to its semantic matching capability during the transfer process.

Fig. S4 illustrates the appearance transfer results without using an object mask. Without an object mask, the back-

	Matching level	Matching rule	Rearrange target
Ours	Feature map	One-to-One	Feature map
Case1	Feature map	One-to-Many	Feature map
Case2	Query, Key	One-to-Many	Query

(a) Matching levels, Rules, and Targets by Case



(b) Comparison between Cases and Ours

Figure S5. Comparison of matching component combinations.

ground of the target image is not preserved after the transfer. This observation highlights that object masks ensure object-aware appearance transfer. Competitors that do not use object masks, such as Cross-Image, DiffEditor, DiffuseIT, Splice-ViT, Swapping AE, and IP-Adapter, fail to preserve the background.

J. Ablation study for matching rule

Rationale: As we aim to transfer appearances according to semantic matches (e.g., beak-to-beak), it is natural to employ one-to-one winner-takes-all matches rather than softmax aggregation.

In Case 1, implicit alignments like softmax aggregation fail to preserve reference feature values. And in Case 2, the injection based on the matching between the query and key with the attention mechanism also produces similar failure results. There are no scenarios where one-to-many or many-to-one matching outperforms one-to-one. Features from similar regions inherently share similar values, so there are no cases where top-1 similarity is incorrect while top-2 to N is correct. If cosine similarity fails in one-to-one matching, cosine similarity-based attention mechanisms would also fail.

K. User study

Method	Ours	Cross-Image	Diffeditor	DiffuseIT	Splice VIT	IP-Adapter	Zest
U_{app}	**0.661**	0.059	0.033	0.026	0.096	0.062	0.064
U_{str}	**0.462**	0.014	0.042	0.121	0.030	0.062	0.276

Table S6. User study results. The bold is the best score.

We conducted a user study with 53 participants, evaluating 15 randomly selected samples for appearance similarity (U_{app}) and structure preservation (U_{str}).

L. Additional qualitative results

In Fig. S6 and Fig. S7, we provide more additional qualitative comparisons with competitors. In particular, Fig. S6 illustrates the results when the reference and target are aligned but the reference object has complex patterns, as well as when the reference and target are unaligned. And Fig. S8 and Fig. S9 showcase our transferred results across various domains. Additionally, Fig. S10 shows the results of appearance transfer from each object from two different reference images to multiple objects in a single target image. Each appearance transfer process occurs simultaneously.

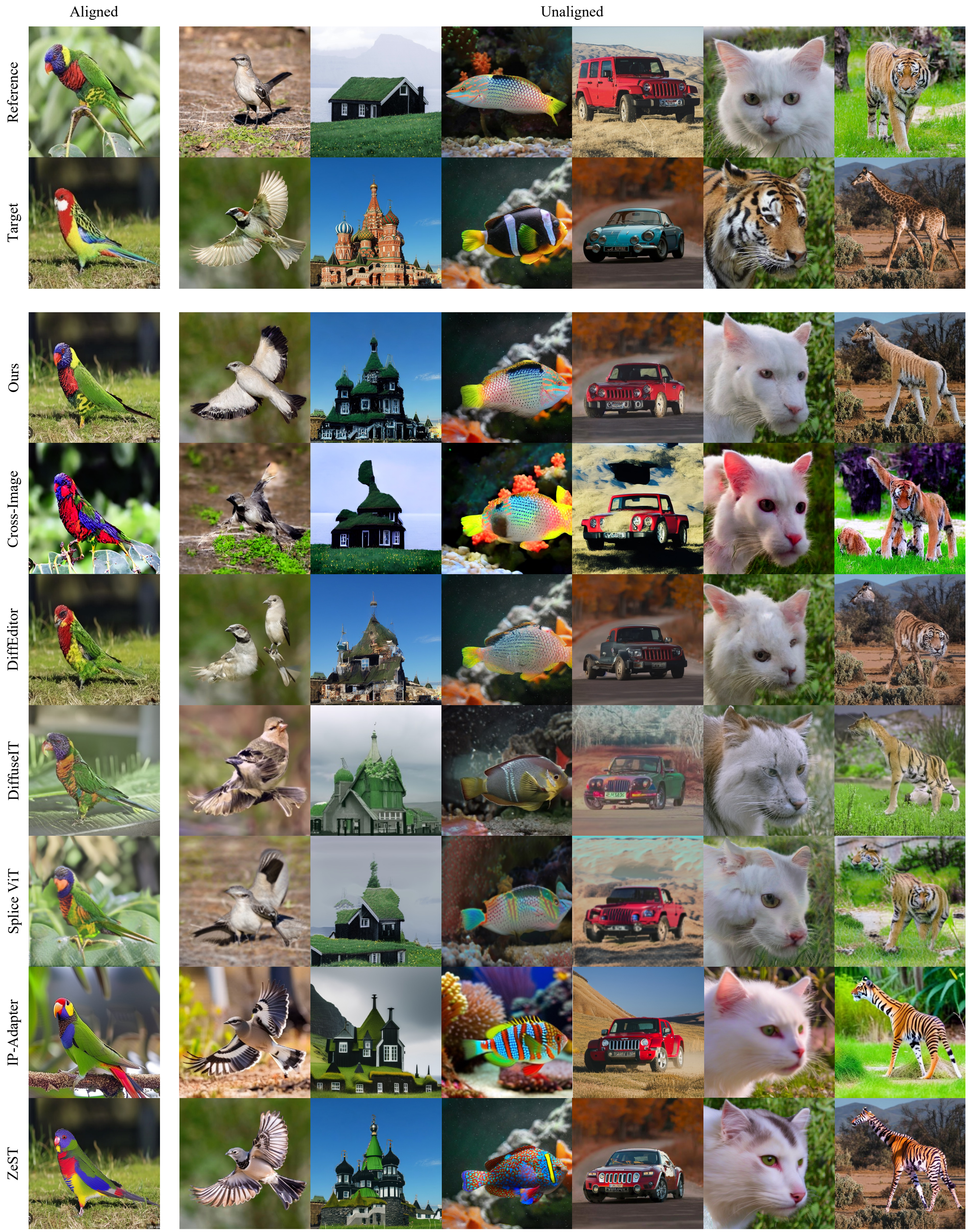


Figure S6. Our results on samples where the reference and target are aligned but the reference has complex patterns, as well as on various samples where the reference and target are misaligned.

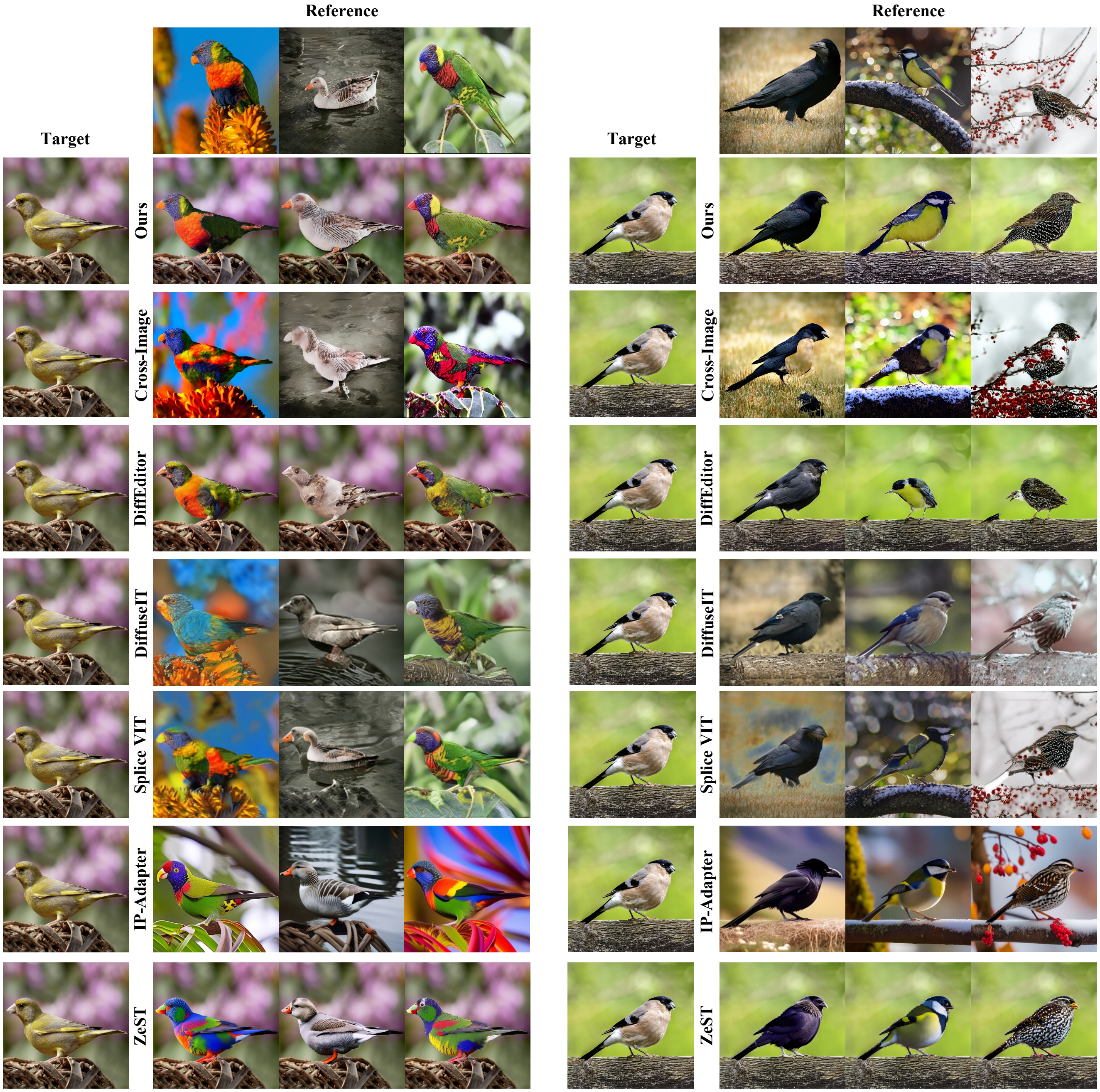


Figure S7. Qualitative comparison of appearance transfer for bird samples.

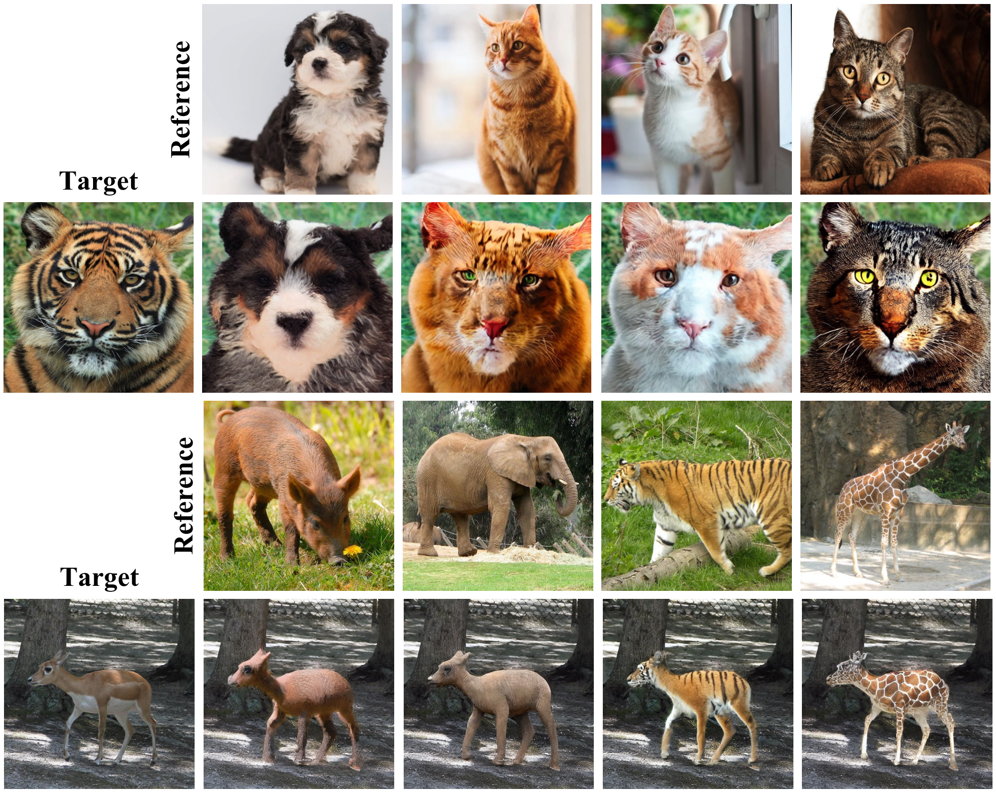


Figure S8. Our results on samples where the reference and target differ in size or are misaligned.

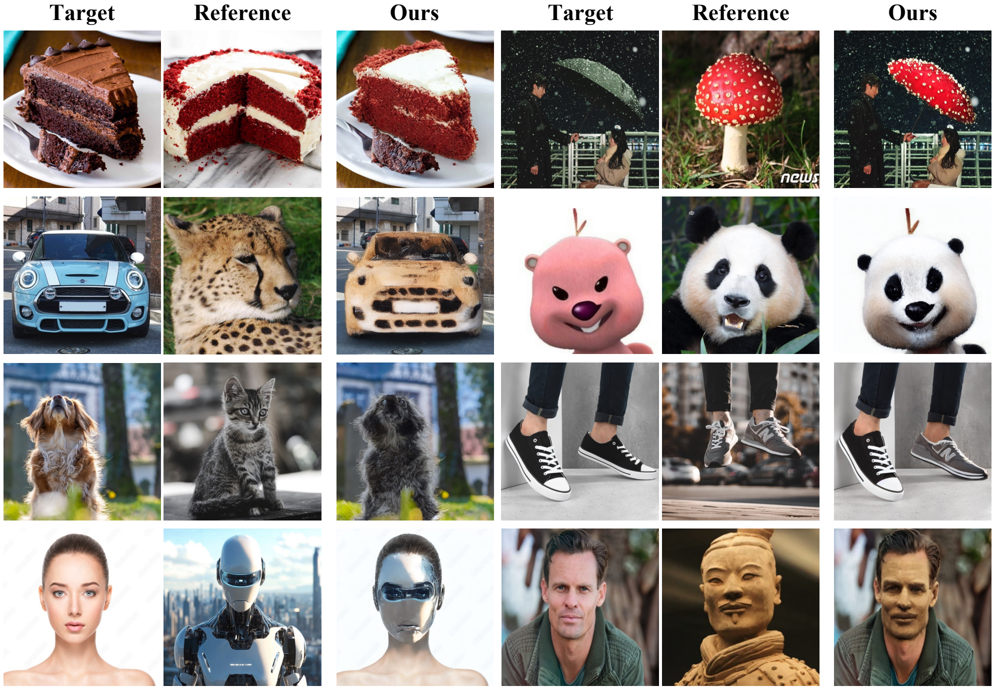


Figure S9. Our results of various domain.

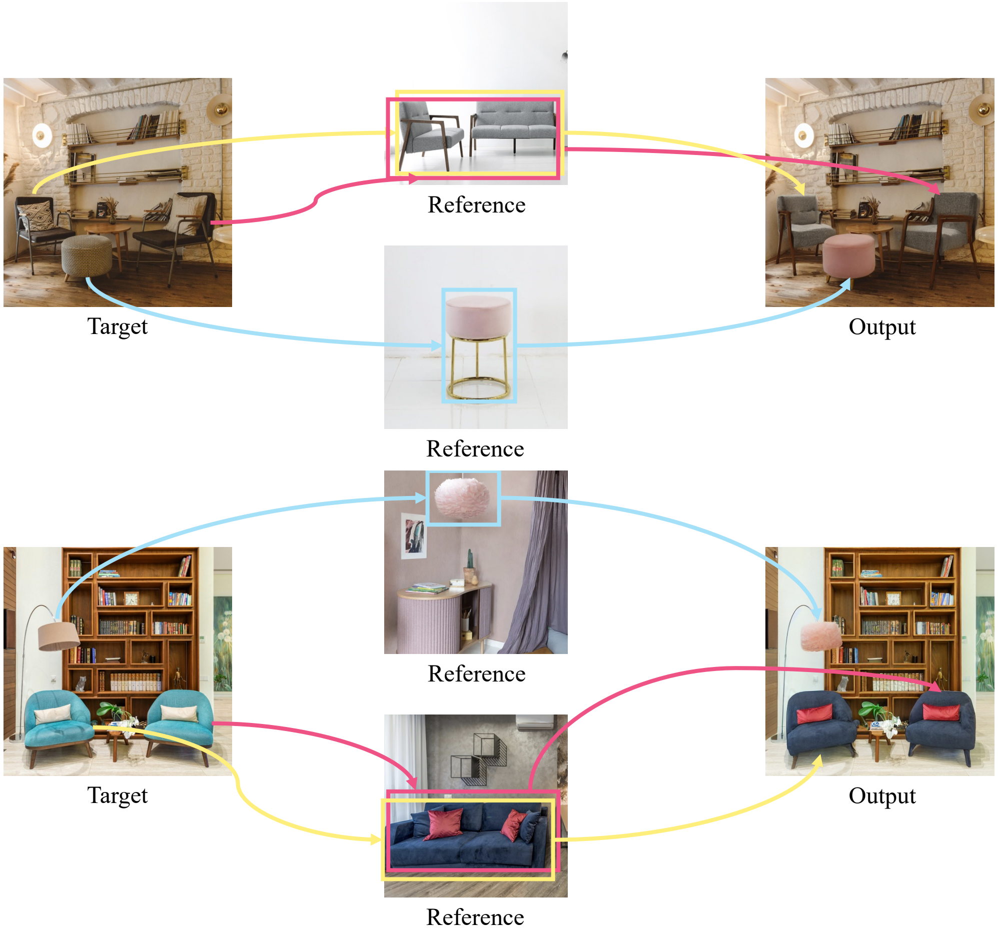


Figure S10. Results of appearance transfer between multiple objects.