Continual Personalization for Diffusion Models

Supplementary Material

Figure 6. **More qualitative visualization.** Note that only Continual Diffusion [8] and CNS are capable of performing continual personalization, while Mix-of-Show [13] and Orthogonal Adaptation [27] require to keep LoRAs for each concept for personalization. It can be seen that our personalized outputs match concepts learned across different time, alleviating appearance leakage and catastrophic forgetting problems.

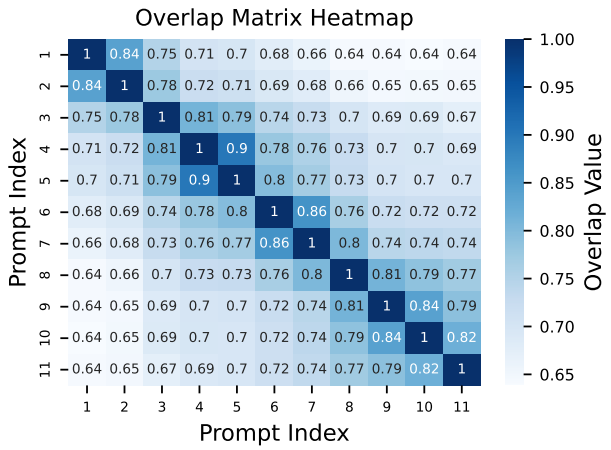

Figure 7. Confusion matrix of the similarity scores between query pairs. This figure was provided in Supplementary Sec. 2.

8. Implementation Details

8.1. Datasets

To evaluate the continual personalization, we collect 3 sets of concept images from [30, 31] and open source images on the net[0]. All set includes 7 concepts and 42 prompts for the evaluation. There are 35 prompts with single concepts, 4 prompt with double concepts and 3 prompts with triple concepts, totally 42 prompts. Concept types in the first set include three animals, two objects and two styles. Concept types in the second set include three animals, two background scenes, one object and one style. Concept types in the third set include two animals, one background scene, three objects and one style. We evaluate ours framework along with all the other baselines with the same dataset setting.

8.2. Baseline methods

Textual Inversion We clone the code base of textual inversion from official GitHub repository[1]. The learning rate of text embedding is $5e-4$ while training steps is set to 1500. For multi-concept personalization, we naively feed all the trained special tokens to the text-to-image diffusion model.

Custom Diffusion We clone the code base of custom diffusion from official GitHub repository[2]. The learning rate of both text embedding and diffusion model are $4e-5$ while training steps is set to 2500. In the experiment of multi-concept personalization presented in Tab. 1, we fuse all the learned model weights with constrained optimization to merge concepts as described in [18].

Mix-of-Show We clone the code base of custom diffusion from official GitHub repository[3]. The learning rate of text embedding is $1e-3$ and LoRA is $1e-4$ while batch size is set to 2. Training steps across each concept are different and determined by the number of images of each concept, as the setting in official code. In the experiment of multi-concept personalization presented in Tab. 1, we fuse all the learned LoRA weights with LoRA fusion approach as described in [13].

Orthogonal Adaption We implement Orthogonal Adaption by ourselves because official code is not available during our research process. We use the recommended randomized orthogonal basis, which is consistent with the paper. The learning rate of text embedding is $1e-3$ and LoRA is $1e-5$ while training steps is set to 500.

Continual Diffusion We also implement Continual Diffusion by ourself, as the official code is unavailable during our research. We follow the self-regularization loss presented in [32] to fulfill continual personalization. The learning rate for both text embedding and LoRA are $5e-4$ while the training steps is set to 700.

8.3. Threshold of the neuron selection

We define a threshold of 30% for the neuron selection process described in Sec. 4.2. Specifically, we select the top 30% of representative neurons in each row based on the importance scores computed using Eq. (3).

9. Additional Experimental Results

9.1. Use of similarity scores

To verify this, we now conduct experiments with 11 text queries, which is constructed as follows. We start with a text query with 5 adjective-noun pairs; then, we replace an adjective or a noun to produce another query, until all words are replaced (as the 11th query). We list three examples:

1. Rainy beach, orange sky, palm trees, green sand, calm ocean
2. Rainy beach, ..., wild ocean
3. Rainy beach, ..., wild horse

Note that larger similarity scores indicate a higher overlapping percentage of selected neurons. With the aforementioned 11 queries, Fig. 7 shows the associated confusion matrix (i.e., query pairs with smaller differences in words show larger overlapping/similarity scores). With the above experiments, our neuron selection scheme for identifying the described concepts can be verified.

[0] https://yuhhuey1.pixnet.net/blog/post/221501409
[1] https://github.com/rinongal/textual_inversion
[2] https://github.com/adobe-research/custom-diffusion
[3] https://github.com/TencentARC/Mix-of-Show

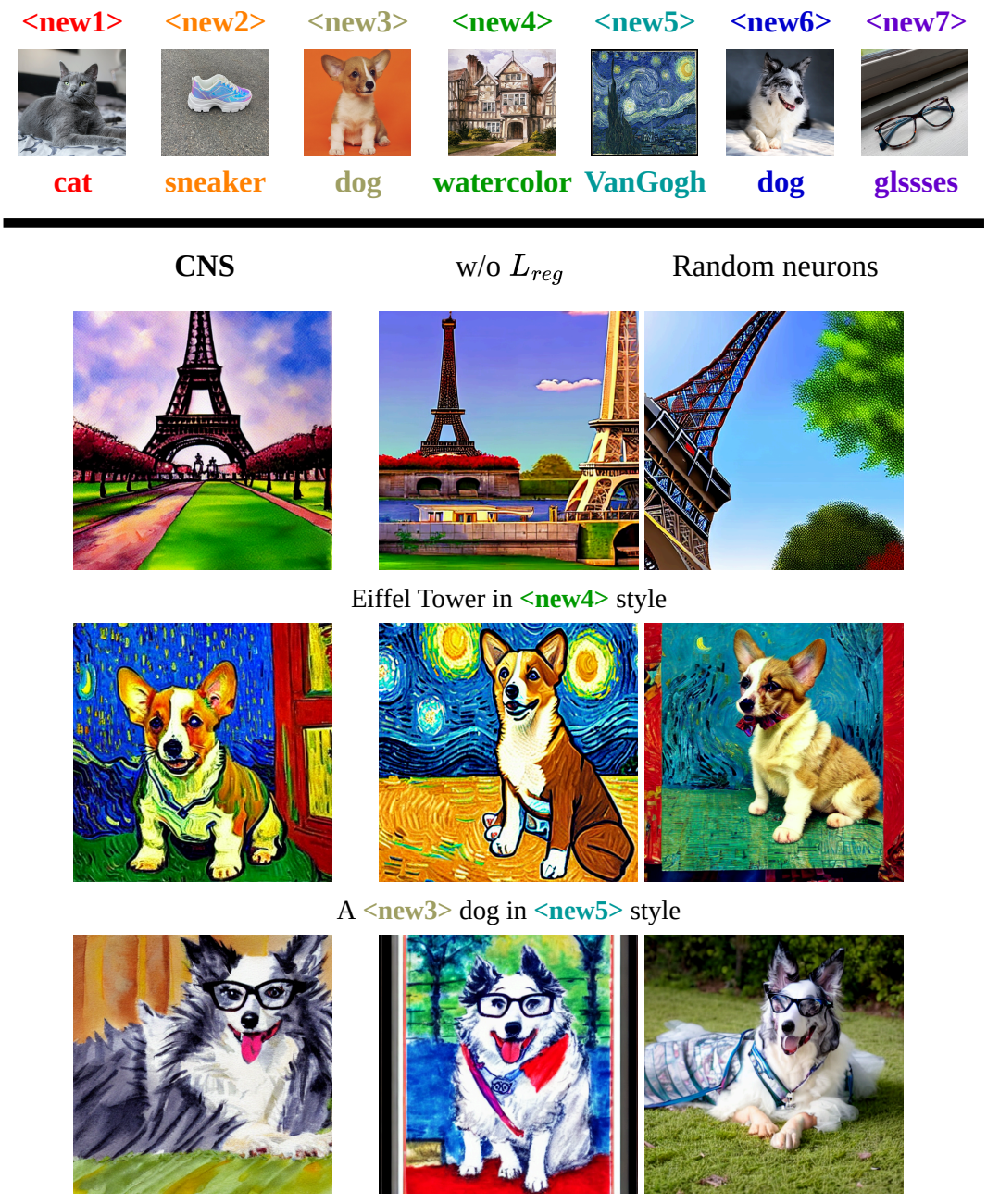

Figure 8. **Qualitative result of ablation study.** It was observed that without the assistance of L_{reg}, the objects in the images lost detail due to catastrophic forgetting. Additionally, when the same number of neurons as the concept neurons are randomly selected, the model struggles to learn the target concepts effectively, leading to noticeable performance degradation.

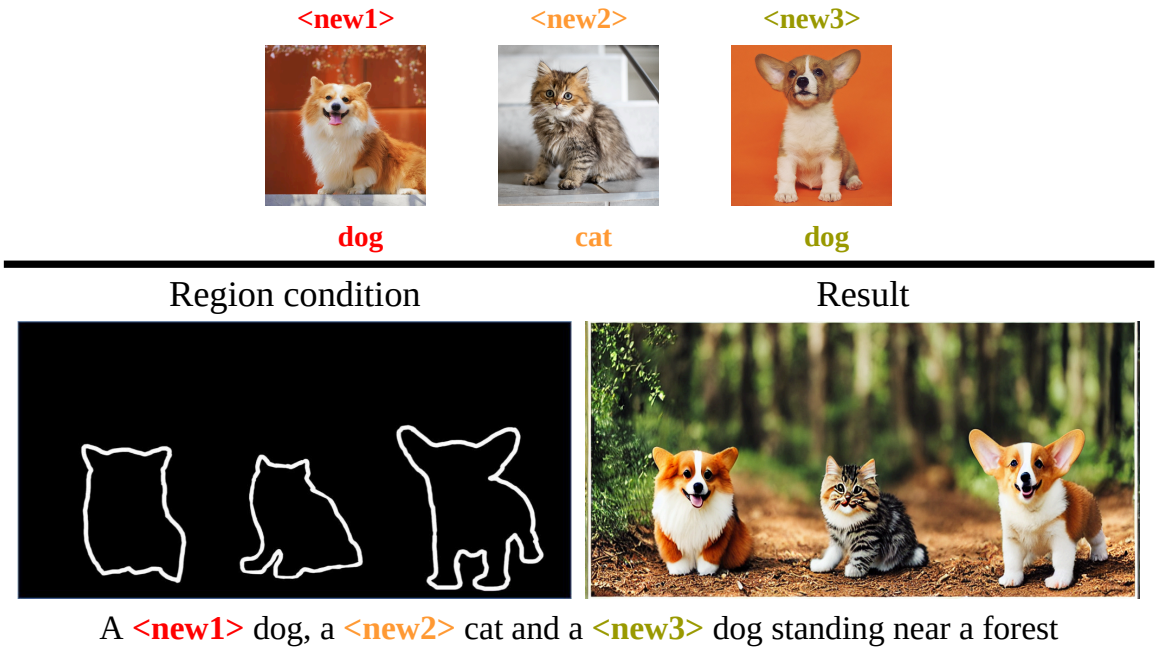

Figure 9. **Region control experiment.** In this figure, we shows the result of applying region control with CNS. Aside from fusing different concepts into an image, CNS can easily apply any test-time region control to specify where each concept should be placed.

9.2. Percentage of the updated neurons

In this section, we provide a detailed calculation of the updated neuron percentage within our framework. In CNS, we target only the linear layers in the cross-attention components $W^{k,v}$, as described in Sec. 3. From these, we identify the top 30% of the using the importance scores calculated by Eq. (3). Additionally, we perform the *concept neurons* selection process to isolate the *concept neurons* from the , resulting in about 5% of the neurons remaining in the cross-attention layers. Ultimately, this process updates only around 0.13% of the total parameters for a single concept personalization.

9.3. Percentage of the overlapping between concept neurons

In this section, we present an overall estimation of the overlapping percentage among *concept neurons* for different concepts. Specifically, we compute the pairwise overlapping mIoU between concepts, finding an average overlapping mIoU of 0.24 across *concept neurons*. Across all 7 concepts in one set, the overlapping mIoU is 0.02.

9.4. Empirical experiment of general neurons

In this section, we provide the details of how we calculate *general neurons*. We prompt the GPT [1] to generate 100 diverse image descriptions and collect them to identify the *general neurons* as detailed in Sec. 4.2.

9.5. Visualization of ablation study

Figure 8 shows the qualitative result of our ablation experiments mentioned in Sec. 5.4. Noted that without the help of L_{reg}, the objects in the images lost details as the result of catastrophic forgetting. In the meantime, if we randomly pick the same amount of neurons as concept neurons, model fails to learn target concepts properly and degradation of performance can be easily observed.

9.6. Region control

We have also observed the notorious attribute binding issue in CNS. However, CNS can easily leverage any test-time region control method to solve attribute binding. In Fig. 9, we illustrate the result of combining CNS with the region control method proposed in [13] to mitigate the issue of attribute binding.