Continual Diffusion with STAMINA: STack-And-Mask INcremental Adapters -Supplementary Materials (Appendix)-

James Seale Smith[1,2] **Yen-Chang Hsu**[1] **Zsolt Kira**[2] **Yilin Shen**[1] **Hongxia Jin**[1]

[1]Samsung Research America, [2]Georgia Institute of Technology

A. Analysis on Interference

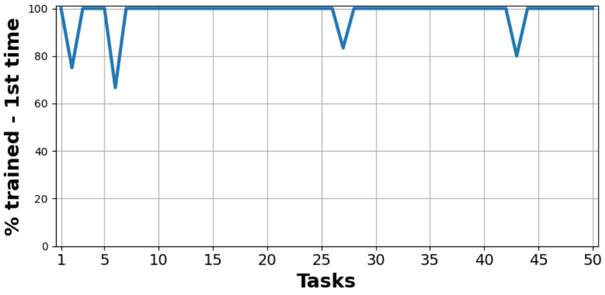


Figure A. Percentage of non-zero $\boldsymbol{W}_t^{K,V} - \boldsymbol{W}_{t-1}^{K,V}$ adaptations which are modifying the pre-trained weights $\boldsymbol{W}_{init}^{K,V}$ at the corresponding position for the first time. Here, a high number equates to low interference (good), and a low number equates to high interference (bad).

In Figure A, we show that STAMINA has *low interference* in changes to the pre-trained weights over tasks. Specifically, we plot the percentage of non-zero $\boldsymbol{W}_t^{K,V} - \boldsymbol{W}_{t-1}^{K,V}$ weight adaptations which are modifying the pre-trained weights $\boldsymbol{W}_{init}^{K,V}$ in their corresponding locations (i.e., indices in the weight matrix) for the first time. The reader should recall that our weight adaptations are *sparse* due to a hard masking mechanism (Eq. 5) and sparsity regularization loss (Eq. 7). Thus, in combination with the forgetting loss (Eq. 3), our method should naturally *avoid altering the pre-trained weights in the same index locations as previous tasks*. We show this exactly - over 50 tasks, the percentage remains high, indicating little to no interference during each task. We note that in some tasks the percentage drops below 100%, demonstrating that some interference still exists in our method.

On the contrary, this same plot for C-LoRA [12] and Custom Diffusion [7] would, by the designs of these methods, show close to or exactly 0% from tasks 2 and beyond, indicating *high interference* at each task. This high interference is likely a strong contributor to the increased catastrophic forgetting of past task concepts in these methods.

B. Additional Metrics

In the main paper tables, we provided the following metrics: A_{mmd} (↓), which gives the average MMD score ($\times 10^3$) after training on all concept tasks, F_{mmd} (↓), which gives the average forgetting, and N_{param} (↓), which gives the % number of parameters being trained. To provide additininal context to our experiments, we provide: KID (↓), which gives the Kernel Inception Distance ($\times 10^3$) between generated and dataset images, and plasticity P_{mmd} (↓), which gives the average plasticity (ability to learn new tasks) as the average MMD score ($\times 10^3$) for all concepts measured directly after after training. The new metrics can be found in Tables A,B,C.

$$P_{mmd} = \frac{1}{N} \sum_{j=1}^{N} MMD\left(\mathcal{F}_{clip}(X_{D,j}), \mathcal{F}_{clip}(X_{j,j})\right) \quad (12)$$

C. Plasticity Analysis

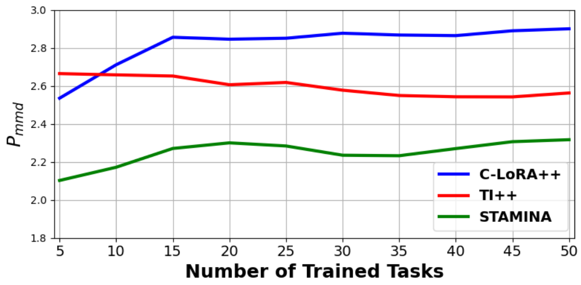


Figure B. Average plasticity P_{mmd} (↓) vs. number of trained tasks.

In Figure B, we directly compares plasticity vs. number of trained tasks for C-LoRA, TI++, and STAMINA in the Table 2.a 50 task benchmark. This figure shows (i) a stronger decrease in plasticity for C-LoRA (compared to STAMINA) and (ii) C-LoRA converging to a much worse plasticity value.

Table A. 50-Task Full Results

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)	KID (↓)	P_{mmd} (↓)
TI++ [2]	0.00	2.52	**0.00**	38.33	2.56
CD [3]	2.23	5.99	5.67	85.08	3.17
CD+EWC [21]	2.23	5.15	3.95	64.47	3.45
C-LoRA [1]	0.09	3.09	1.41	45.37	2.79
Ours	0.19	**2.29**	0.01	**25.73**	**2.32**

Table B. 20-Task Results on Google Landmarks dataset v2 [59]

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)	KID (↓)	P_{mmd} (↓)
TI++ [2]	0.00	2.91	**0.00**	33.69	3.03
CD [3]	2.23	5.20	5.10	114.55	3.25
CD [3] (Merge)	2.23	14.83	8.43	331.21	10.19
CD+EWC [21]	2.23	5.10	3.56	80.58	3.23
C-LoRA [1]	0.09	3.09	0.38	53.24	3.15
Ours	0.19	**2.42**	0.01	**31.73**	**2.44**

Table C. 20-Task Results on Celeb-A HQ [57,58]

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)	KID (↓)	P_{mmd} (↓)
TI++ [2]	0.00	2.37	**0.00**	35.49	2.35
CD [3]	2.23	7.58	6.56	104.54	3.43
CD [3] (Merge)	2.23	13.84	8.61	353.40	7.83
CD+EWC [21]	2.23	7.39	5.81	91.61	3.45
C-LoRA [1]	0.09	2.25	0.33	37.41	2.15
Ours	0.19	**2.18**	0.03	**28.63**	**2.07**

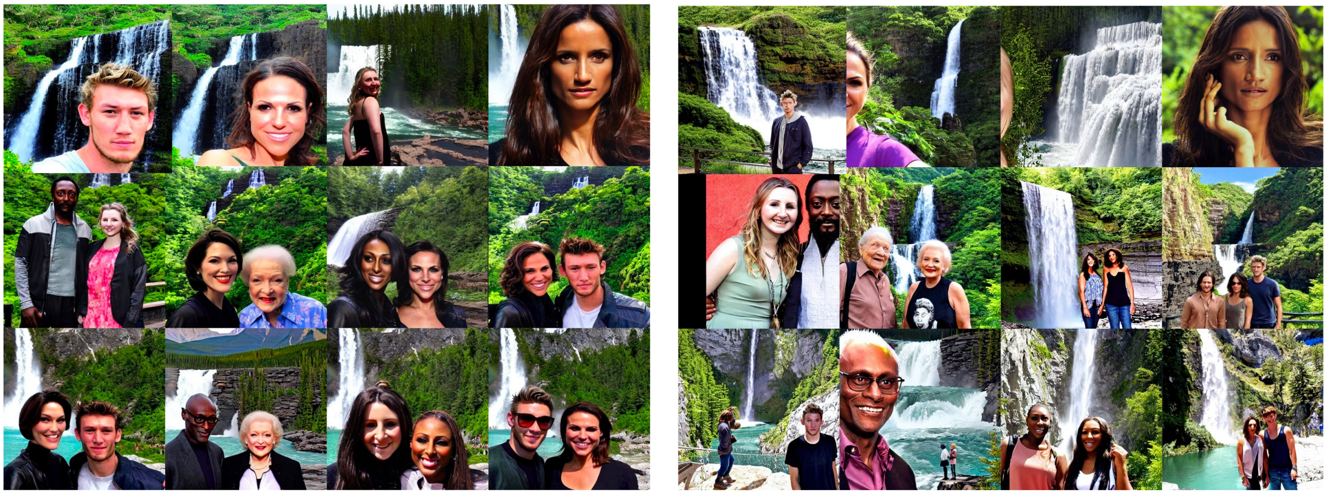
D. Additional Implementation Details

We use 2 A100 GPUs to generate all results. All hyperparameters were searched with an exponential search (for example, learning rates were chosen in the range $5e-2, 5e-3, 5e-4, 5e-5, 5e-6, 5e-7, 5e-8$). We found a learning rate of $5e-6$ worked best for the Custom Diffusion [7] methods, and a learning rate of $5e-4$ worked best for the LoRA-based methods and Textual Inversion [3]. Following Smith *et al.* [12], we use a loss weight of $1e6$ and $1e8$ for EWC [6] and C-LoRA, respectively. For our method, we found a loss weight of $1e-3$ and $1e3$ worked best for the sparsity penalty (Eq.7) and forgetting loss (Eq.3), respectively. We found a rank of 16 was sufficient for LoRA for the text-to-image experiments and 64 for the image classification experiments. These were chosen from a range of $8, 16, 32, 64, 128$. We use 500 training steps (twice as many as reported in Kumari *et al.* [7] due to our data being fine-grain concepts rather than simple objects) except for C-LoRA, which requires longer training steps (we use 2000 as reported in Smith *et al.* [12]). We regularize training with generated auxiliary data (as done in Smith *et al.* [12]) for *all* methods.

The simple MLPs used in our paper are composed of two linear layers and a ReLU [1] layer in between. For the mask MLPs, $\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}$, the dimension of linear layers 1 and 2 are $r \times r$ and $r \times D_1 \cdot D_2 \cdot 2$, where r is the same low rank as the LoRA parameters $\boldsymbol{A}_t^{K,V}$ and $\boldsymbol{B}_t^{K,V}$, and $D1, D2$ are the dimensions of the weight $\boldsymbol{W}^{K,V}$. For the custom token MLPs $\theta_{V_t^*}$, the dimension of linear layers 1 and 2 are both $D_{token} \times D_{token}$, where D_{token} is the dimension of the token embedding.

E. Benchmark Dataset Details

Given the datasets Celeb-A HQ [5, 8] and Google Landmarks v2 [15], we sample concepts at random which have at least 10 individual training images each. Specifically, we iterate randomly over the fine-grained identities of each dataset (person for Celeb-A HQ and waterfall location for



(a) Successes

(b) Failures

Figure C. STAMINA multi-concept generations after training on 50 tasks.

Google Landmarks V2) and check whether the identity has sufficient unique examples in the dataset; we do this until we reached the number of desired concepts for each dataset. Each concept customization is considered a "task", and the tasks are shown to the model sequentially.

F. Additional Details for Image Classification Setting

In Section 5.2, we benchmark our approach using ImageNet-R [4, 13] which is composed of 200 object classes with a wide collection of image styles, including cartoon, graffiti, and hard examples from the original ImageNet dataset [10]. This benchmark is chosen because the distribution of training data has significant distance to the pre-training data (ImageNet), thus providing a problem setting which is both fair and challenging.

We use the same experimental settings as those used in the recent CODA-Prompt [11] paper. We implement our method and all baselines in PyTorch[9] using the ViT-B/16 backbone [2] pre-trained on ImageNet-1K [10]. All methods are trained with a batch size of 128 for 50 epochs; the prompting-based methods use a learning rate of $5e-3$, whereas the LoRA based methods use a learning rate of $5e-4$. We compare to the following methods (the same rehearsal-free comparisons of CODA-Prompt): CODA-Prompt [11], Learning to Prompt (L2P) [14], DualPrompt [13], and C-LoRA [12]. We use the same classification head as L2P, DualPrompt, and CODA-Prompt. For additional details, we refer the reader to original CODA-Prompt [11] paper. For our method, we add STAMINA to the QKV projection matrices of self-attention blocks throughout the ViT model, and use the same 64 rank as used in C-LoRA [12].

G. Negative Multi-Concept Results

We extend our results demonstrating the ability to generate photos of multiple concepts in the same picture by showing both successful attempts (Figure Ca) and failing attempts (Figure Cb). We use the prompt style "a photo of V* person posing next to V* waterfall" for the top row (single person and single landmark) and "a photo of V* person, standing next to V* person, posing in front of V* waterfall" for rows 2 and 3 (two people and a single landmark). Unlike most results in our paper, which diffuse for 200 steps (as done in [7]), we allow the multi-concept results to diffuse for 500 steps.

Each generated image in Figure Cb used the same prompt as the corresponding image in Figure Ca. In general, we found a success rate of roughly 50% for two concept generations and 20% for the challenging 3 concept generations. The failures in row 1 (single person with single landmark) each have a blurred or occluded concept. In rows 2 and 3 (two people with single landmark), we see failures such as the landmark disappearing (row 2, column 1), imagined people (row 2, column 4), merged people (row 3, column 2), or one concept taking on characteristics of another person, such as skin tone (row 3, column 3) or age (row 2, column 2), *which could be explained by bias and is a limitation that users of this work should pay close attention to.* We hope to address these sources of failures in future work.

H. Variance Across Runs

In Table E, we provide the mean and standard deviation for each method across all 3 Continual Diffusion benchmarks (Tables 1.a, 1.b, and 2.a). We see that our method not only has the best metric performance, but also has the lowest

Table D. **Mean and standard deviation across 3 runs:** A_{mmd} ($\downarrow$) gives the average MMD score ($\times 10^3$) after training on all concept tasks, and F_{mmd} ($\downarrow$) gives the average forgetting. N_{param} ($\downarrow$) gives the number of parameters being trained as a % of the unmodified U-Net backbone size.

Table E. Celeb-A HQ [5, 8]

Method	N_{param} Train (%)	A_{mmd} ($\downarrow$)	F_{mmd} ($\downarrow$)
TI++ [3]	0.00	2.60 ± 0.23	**0.00 ± 0.00**
CD [7]	2.23	6.26 ± 0.99	5.78 ± 0.60
CD [7] (Merge)	2.23	14.34 ± 0.50	8.52 ± 0.09
CD+EWC [6]	2.23	5.88 ± 1.07	4.44 ± 0.98
C-LoRA [12]	0.09	2.81 ± 0.40	0.71 ± 0.50
Ours	0.19	**2.30 ± 0.10**	0.02 ± 0.01

standard deviation for both A_{mmd} and F_{mmd}.

I. Figure Image Sources

In our figures, we replace dataset images with generated similar images due to licensing constraints. Specifically, we generate "target data" using offline (i.e., no *continual* learning) single-concept Custom Diffusion [7], which we refer to as *pseudo figure images*. We note that all training and evaluations were completed using the original datasets, and all result images were obtained through models trained directly on the original datasets. For Figure 1, the images captioned "learn" are *pseudo figure images*, and the multi-concept images are results produced with our method. For Figure 3, all concept images are *pseudo figure images*. For Figure 4, the images labeled "target data" are *pseudo figure images*, and the rest are results from models we trained. Finally, Figures 5 and A only contain results produced from models we trained.

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 2

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[3] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2, 4

[4] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 3

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2, 4

[6] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017. 2, 4

[7] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488*, 2022. 1, 2, 3, 4

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 2, 4

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 3

[10] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 3

[11] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2211.13218*, 2022. 3

[12] James Seale Smith, Yen-Chang Hsu, Lingyu Zhang, Ting Hua, Zsolt Kira, Yilin Shen, and Hongxia Jin. Continual diffusion: Continual customization of text-to-image diffusion with c-lora. *arXiv preprint arXiv:2304.06027*, 2023. 1, 2, 3, 4

[13] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022. 3

[14] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 3

[15] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2575–2584, 2020. 2