Continual Diffusion with STAMINA: STack-And-Mask INcremental Adapters

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

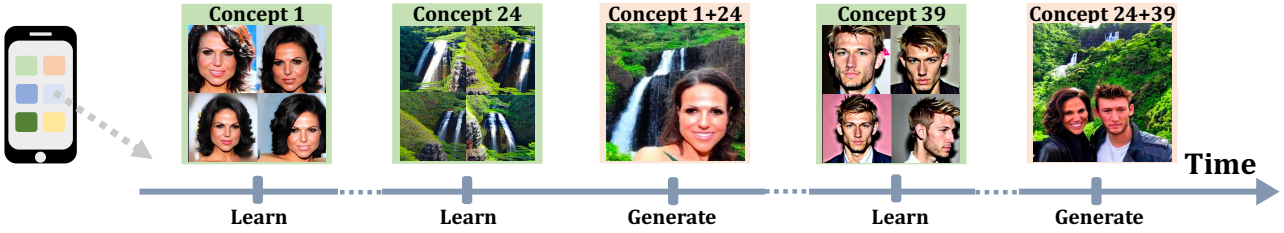

Figure 1. Our work demonstrates sequentially learning long sequences of concepts. At any time, we can generate photos of any prior learned concepts, including multiple concepts together. *Images denoted as "generate" in this figure are real results from our method.*

Abstract

Recent work has demonstrated a remarkable ability to customize text-to-image diffusion models to multiple, fine-grained concepts in a sequential (i.e., continual) manner while only providing a few example images for each concept. This setting is known as continual diffusion. *Here, we ask the question:* Can we scale these methods to longer concept sequences without forgetting? *Although prior work mitigates the forgetting of previously learned concepts, we show that its capacity to learn new tasks reaches saturation over longer sequences. We address this challenge by introducing a novel method, STack-And-Mask INcremental Adapters (STAMINA), which is composed of low-ranked attention-masked adapters and customized MLP tokens. STAMINA is designed to enhance the robust fine-tuning properties of LoRA for sequential concept learning via learnable hard-attention masks parameterized with low rank MLPs, enabling precise, scalable learning via sparse adaptation. Notably, all introduced trainable parameters can be folded back into the model after training, inducing no additional inference parameter costs. We show that STAMINA outperforms the prior SOTA for the setting of text-to-image continual customization on a 50-concept benchmark composed of landmarks and human faces, with no stored replay data. Additionally, we extended our method to the setting of continual learning for image classification, demonstrating that our gains also translate to state-of-the-art performance in this standard benchmark.*

1. Introduction

Remarkable progress in text-to-image diffusion models has ushered in an era of practical applications ranging from generating detailed product images for e-commerce and advertising platforms to facilitating creative endeavors in content generation and story-telling. A particularly compelling direction in this field is the task of customizing models to multiple, fine-grained concepts in a *continual* manner, a recently introduced paradigm known as Continual Diffusion [1]. In this setting, models are sequentially adapted to new concepts using a few example images without forgetting prior learned concepts. This is an import problem for practical use cases such as customizing a generative model for a user over time without requiring permissions to store their private and personal user data (visualized in Figure 1). While prior work has shown significant strides in mitigating catastrophic forgetting for this setting, we ask: *How does the model's performance evolve when we scale to longer concept sequences?*

In this paper, we start by addressing this question and demonstrate with quantitative analysis that, while the state-of-the-art (SOTA) method for this task (C-LoRA [1]) can significantly alleviate catastrophic forgetting, the capacity to learn new tasks rapidly saturates after a certain number of tasks. In response, we propose an innovative approach, **STack-And-Mask INcremental Adapters (STAMINA)**, which combines the robustness of low-rank adaptations with the precision of attention masking, and enhances them with learnable MLP tokens. STAMINA com-

prises two key elements: 1) Low-Rank Adapters (LoRA) with hard-attention masks parameterized by low-rank MLP modules and Gumbel softmax, and 2) learnable MLPs that replace the custom token feature embeddings used in prior works [1–3]. Importantly, all trainable parameters introduced can be seamlessly folded back into the model post-training, thereby not inducing any additional parameter costs during inference. Our method exceeds the prowess of prior C-LoRA for Continual Diffusion by strongly surpassing its plasticity to learn longer concept sequences without sacrificing its resilience to catastrophic forgetting.

We demonstrate that STAMINA significantly outperforms the prior SOTA in text-to-image continual customization without any stored replay data on 3 benchmarks, including a comprehensive 50-concept benchmark composed of human faces and landmarks. Moreover, we show that STAMINA achieves this performance while requiring significantly fewer training steps compared to the previous SOTA. **In summary, our contributions in this paper are as follows:**

1. We empirically demonstrate the limitations of existing approaches in scaling to longer task sequences in the Continual Diffusion setting.
2. We propose the STAMINA method, a novel combination of LoRA with attention masking and MLPs, which boosts the model's ability to learn and remember longer sequences of tasks. Importantly, all of the trainable parameters introduced by STAMINA can be integrated back into the model post-training, ensuring no additional inference parameter costs.
3. We present extensive experiments showing that STAMINA significantly outperforms the current state-of-the-art in the Continual Diffusion setting on a 50-concept benchmark while requiring significantly fewer training steps.
4. We demonstrate the versatility of STAMINA and extend it into the setting of continual learning for image classification, and show that our gains translate to state-of-the-art performance for a widely used 20-task benchmark.

2. Background and Related Work

Conditional Image Generation: Conditional image generation is a well-studied topic with several approaches, including Generative Adversarial Networks (GANs)[4, 5], Variational Autoencoder (VAE)[6], and diffusion models [7–9]. We focus on the popular diffusion-based models that use free-form text prompts as conditions [10, 11]. These models operate by iteratively adding noise to the original image and then removing the noise through a backward pass to produce a final generated image. To inject text conditions, a cross-attention mechanism is introduced in the transformer-based U-Net [12].

Recent works have explored the generation of *custom* concepts, such as a unique stuffed toy or a pet. Dreambooth [13] fine-tunes the whole parameter set in a diffusion model using given images of the new concept, while Textual Inversion [2] learns custom feature embedding "words". Methods are not limited to conditioning on only text - recent work has shown that these models can be customized to effectively add conditioning with new modalities (such as semantic segmentation maps and sketches) [14]. In contrast, Custom Diffusion [3] learns multiple concepts using a combination of cross-attention fine-tuning, regularization, and closed-form weight merging. However, Custom Diffusion struggles to learn similar, fine-grained concepts in a sequential manner (i.e., Continual Diffusion), motivating the recent C-LoRA [1] work which is the first to sequentially customize Stable Diffusion in a "continual learning" manner. We mention that several recent pre-print works [15–19] have emerged that build or extend upon Dreambooth and Custom Diffusion, but none of these methods are designed for the sequential setting, and the contributions can be considered as orthogonal to our paper.

Continual Learning: Continual learning involves training a model on a sequence of tasks, each with a different data distribution, while preserving the knowledge learned from previous tasks. Existing methods to address the issue of catastrophic forgetting can be categorized into three groups [20]. Regularization-based methods [21–24] add regularization terms to the objective function during training a new task. For instance, EWC [22] estimates the significance of model parameters and applies per-parameter weight decay. Rehearsal-based methods [25–33] save or synthesize samples from past tasks in a data buffer and replay them alongside the new task's data. Nonetheless, privacy or copyright issues may prevent using this method. Architecture-based methods [34–37] separate model parameters for each task. Recent prompt-based continual learning methods for Vision Transformers such as L2P [38], DualPrompt [39], and CODA-Prompt [40] have surpassed rehearsal-based methods in classification problems without needing a replay buffer. Although they have shown success in classification problems, their utility in text-to-image generation is unclear since they infer discriminative features of data to create prompts for classification. However, we do compare to them in their original problem setting.

While most research on continual learning focuses on uni-modal problems, a few approaches have been suggested for the multimodal setting. REMIND [41] proposed continual VQA tasks with latent replay but require storing compressed training data. CLiMB [42] adapted CL to coarsely different VL tasks, including VQA, NLVR, VE, and VCR, assuming knowledge of the evaluated task-id at inference time. Construct-VL [43] concentrates on natural language visual reasoning. We formulate our problem as the continual adaptation of Stable Diffusion to multiple, fine-grained

concepts, and most of the methods reviewed in this section do not directly apply to our problem. The only relevant work, C-LoRA [1], is discussed in the next section.

Sparsity for Continual Learning: A key component of our approach is *sparse model adaptations* via attention masking to mitigate forgetting and interference in our model. Prior works have also leveraged sparsity for continual learning in other problem settings. HAT [44] also applies hard attention masks, but these are task-conditioned masks on network paths within the model. This approach require task id during inference and thus cannot be applied to multi-concept generation, excluding them from our setting. DGM [45] applies sigmoid attention masks to increase sparsity and scalability for image classification, but this approach requires dynamic parameter expansion, which would not be practical in the setting of continually adapting a *large-scale pre-trained* model. The work of Schwarz *et al.* [46] also demonstrates that sparsity can reduce catastrophic forgetting via a sparsity-inducing weight re-parameterization. It may be possible for this method to be implemented to work for adapting an existing pre-trained model, but that is out of the scope for our paper. GPM [47] demonstrates positive effects for sparsity in continual learning with k-winner activation MLPs, but it is also not clear how this could be implemented in the context of adapting our pre-trained text-to-image diffusion model. Finally, sparsity has also been applied to meta-continual learning [48], which requires additional training on a meta-dataset. While these works demonstrate the fundamental advantages for sparsity in continual image classification, it is not clear how they can be applied to our specific setting of text-to-image customization, motivating our approach.

3. A Closer Look at Continual Diffusion

In the Continual Diffusion setting, we learn N customization "tasks" $t \in \{1, 2, \ldots, N-1, N\}$, where N is the total number of concepts that will be shown to our model. The recent C-LoRA [1] was the first to propose a method for Continual Diffusion. In this section, we will first define some preliminaries and then review the C-LoRA method and discuss its shortcomings.

C-LoRA utilizes parameter-efficient fine-tuning of a transformer. Consider the context of the single-head cross-attention operation [49], given as $\mathcal{F}_{attn}(Q, K, V) = \sigma\left(\frac{QK^\top}{\sqrt{d'}}\right)V$, where σ stands for the softmax operator, $Q = \boldsymbol{W}^Q \boldsymbol{f}$ represents query features, $K = \boldsymbol{W}^K \boldsymbol{c}$ serves as key features, and $V = \boldsymbol{W}^V \boldsymbol{c}$ functions as value features. Additionally, $\boldsymbol{f}$ indicates latent image features, $\boldsymbol{c}$ denotes text features, and d' is the output dimensionality. In this equation, the matrices $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V$ map inputs $\boldsymbol{f}$ and $\boldsymbol{c}$ to the query, key, and value features, respectively.

Prior works [1, 3] only modify $\boldsymbol{W}^K, \boldsymbol{W}^V$ (referred to as $\boldsymbol{W}^{K,V}$) which project the text features. In learning a customization "task" t, the following loss is minimized in the prior C-LoRA [1]:

$$\min_{\boldsymbol{W}_t^{K,V} \in \theta} \quad L_{SD}(x, \boldsymbol{\theta}) + \lambda_f \mathcal{L}_{forget}(\boldsymbol{W}_{t-1}^{K,V}, \boldsymbol{W}_t^{K,V}) \quad (1)$$

Here, x stands for the input data of the new concept, $\mathcal{L}_{SD}$ denotes the loss function for Stable Diffusion with respect to model θ, $\mathcal{L}_{forget}$ minimizes forgetting between old task $\boldsymbol{W}_{t-1}^{K,V}$ and new task $\boldsymbol{W}_t^{K,V}$, and λ_f is a hyperparameter selected through a straightforward exponential sweep.

Smith *et al.* [1] propose to parameterize the weight change between old task $\boldsymbol{W}_{t-1}^{K,V}$ and new task $\boldsymbol{W}_t^{K,V}$ using LoRA [50][1], which decompose the weight matrices into low-rank residuals, as expressed by:

$$\begin{aligned} \boldsymbol{W}_t^{K,V} &= \boldsymbol{W}_{t-1}^{K,V} + \boldsymbol{A}_t^{K,V} \boldsymbol{B}_t^{K,V} \\ &= \boldsymbol{W}_{init}^{K,V} + \sum_{t'=1}^{t-1} \boldsymbol{A}_{t'}^{K,V} \boldsymbol{B}_{t'}^{K,V} + \boldsymbol{A}_t^{K,V} \boldsymbol{B}_t^{K,V} \end{aligned} \quad (2)$$

Here, $\boldsymbol{A}_t^{K,V} \in \mathbb{R}^{D_1 \times r}$, $\boldsymbol{B}_t^{K,V} \in \mathbb{R}^{r \times D_2}$, with $\boldsymbol{W}^{K,V} \in \mathbb{R}^{D_1 \times D_2}$, and r being a hyper-parameter controlling the rank of the weight matrix update, chosen using a simple grid search. The initial values from the pre-trained model are represented as $\boldsymbol{W}_{init}^{K,V}$.

C-LoRA additionally proposed a novel regularization method which involves penalizing the LoRA parameters $\boldsymbol{A}_t^{K,V}$ and $\boldsymbol{B}_t^{K,V}$ for altering locations that have been previously modified by earlier concepts in $\boldsymbol{W}_t^{K,V}$. Specifically, C-LoRA contains the forgetting loss $\mathcal{L}_{forget}$, given as:

$$\mathcal{L}_{forget} = || \left| \sum_{t'=1}^{t-1} \boldsymbol{A}_{t'}^{K,V} \boldsymbol{B}_{t'}^{K,V} \right| \odot \boldsymbol{A}_t^{K,V} \boldsymbol{B}_t^{K,V} ||_2^2 \quad (3)$$

where $\odot$ represents the element-wise product, or the Hadamard product.

3.1. The Plasticity Problem

In Section 5, we find that C-LoRA suffers in our benchmarks which involve longer task sequences. *Why could that be?* Our intuition is that, as the weights diverge further from the pre-trained backbone, $\mathcal{L}_{forget}$ becomes more and more restrictive, thus limiting the ability to learn new tasks (plasticity). We show this in Figure 2 by plotting the average distance from the pre-trained weights, given as $||\boldsymbol{W}_t^{K,V} - \boldsymbol{W}_{init}^{K,V}||_2$, versus tasks. We see that, while the model tends to see high changes in the early tasks, this rapidly saturates, suggesting that low plasticity may be a contributor for the diminishing performance in C-LoRA on longer task sequences. These results motivate us to find a better approach for *scalable* Continual Diffusion.

[1] This was also proposed for NLVR [43] and offline customization [51].

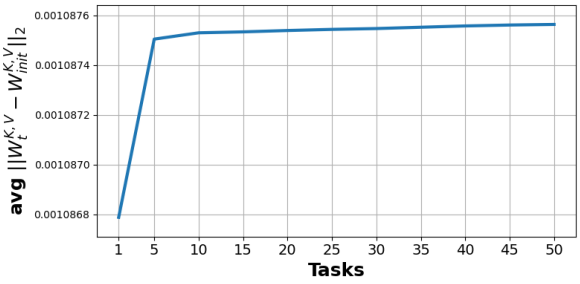

Figure 2. Average distance from pre-trained weights, given as $||\boldsymbol{W}_t^{K,V} - \boldsymbol{W}_{init}^{K,V}||_2$, vs task for C-LoRA [1].

4. The STAMINA Approach

Inspired by the analysis of the previous section, we propose an innovative approach to Continual Diffusion: **Stack-And-Mask INcremental Adapters (STAMINA)**, composed of stack-able low-rank adapter with learnable hard-attention masks to encourage precise and sparse weight residuals on the pretrained diffusion model. The intuition behind our approach is that *sparse* adaptations are less likely to interfere and, therefore, will avoid saturating plasticity as the number of tasks grow. The overview of our proposed method is illustrated in Figure 3, and the rest of this section motivates and describes each component of STAMINA.

Hard Attention Masks with Gumbel-Softmax: Consider the low rank weight decomposition $\boldsymbol{W}_t^{K,V} = \boldsymbol{W}_{t-1}^{K,V} + \boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V}$. The intuition is that low-rank adaptation is robust against over-fitting, making it a strong continual learner [1, 43]. However, there is one glaring weakness with this approach: **precision**. Specifically, the low-rank properties of the product $\boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V}$ lack the ability to target important spots in the weight matrix without incurring many unnecessary adaptations to other spots in the matrix.

To reduce interference between learned concepts, we propose to apply hard-attention masks on the $\boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V}$ product. Rather than learn masks with continuous outputs on the range $[0, 1]$, as many sparse continual learning works discussed in Section 2 do, we desire a true discrete binary mask to retain desirable robust properties of the low-rank weight residuals (a decision validated by our ablations). Thus, we leverage[2] the Gumbel-Softmax [53] operation, which provides a differentiable approximation to the discrete argmax operation, enabling the learning of true *hard attention* masks during training.

We consider our mask as a *binary categorical distribution* and expand in an additional dimension of size 2, and take the operation over the expanded dimension. With this parameterization, values of 1 in the first index of the expanded dimension equates to "pass through the mask" after the Gumbel-Softmax, and the other dimension can be discarded. Specifically, we learn mask $\boldsymbol{\mathcal{M}}_t^{K,V} \in \mathbb{R}^{D_1 \times D_2}$ as:

$$\boldsymbol{\mathcal{M}}_{t,i,j}^{K,V} = \frac{\exp\left(\frac{\log(\hat{\boldsymbol{m}}_{t,i,j,1}^{K,V}) + g_{i,j,1}}{\tau}\right)}{\sum_{z=0}^{1}\exp\left(\frac{\log(\hat{\boldsymbol{m}}_{t,i,j,z}^{K,V}) + g_{i,j,z}}{\tau}\right)} \tag{4}$$

where $\hat{\boldsymbol{m}}_t^{K,V} \in \mathbb{R}^{D_1 \times D_2 \times 2}$ represents a learnable mask tensor *before* Gumbel-Softmax is taken over the expanded third dimension; i, j represent taking this operation over $i = 1, \ldots, D_1$ and $j = 1, \ldots, D_2$; τ is a temperature hyperparameter controlling smoothness[3]; and g are i.i.d samples drawn from Gumbel$(0, 1)$ [53]. Notice that $\boldsymbol{\mathcal{M}}_t^{K,V}$ denotes our final learned mask which is applied to the $\boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V}$ product, whereas $\hat{\boldsymbol{m}}_t^{K,V}$ denotes the learnable parameters before Gumbel-Softmax.

Using this masking approach, our weight residuals are now given as:

$$\begin{aligned}\boldsymbol{W}_t^{K,V} &= \boldsymbol{W}_{t-1}^{K,V} + \boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V} \odot \boldsymbol{\mathcal{M}}_t^{K,V} \\ &= \boldsymbol{W}_{init}^{K,V} + \left[\sum_{t'=1}^{t-1} \boldsymbol{A}_{t'}^{K,V}\boldsymbol{B}_{t'}^{K,V} \odot \boldsymbol{\mathcal{M}}_{t'}^{K,V}\right] \\ &\quad + \boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V} \odot \boldsymbol{\mathcal{M}}_t^{K,V}\end{aligned} \tag{5}$$

MLP Mask Parameterization: Rather than optimize a fixed tensor, we further enhance our masking capacity and flexibility with a low-rank multi-layer perception (MLP) parameterization, $\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}$, operating on a fixed input $\mathbf{1}$. The idea is to leverage the power of MLPs to learn more complex transformations between tasks, thereby mitigating the risk of catastrophic forgetting. Specifically, we propose to learn our mask as:

$$\theta_{\boldsymbol{\mathcal{M}}_{t,i,j}^{K,V}} = \frac{\exp\left(\frac{\log(\theta_{\hat{\boldsymbol{m}}_{t,i,j,1}^{K,V}}) + g_{i,j,1}}{\tau}\right)}{\sum_{z=0}^{1}\exp\left(\frac{\log(\theta_{\hat{\boldsymbol{m}}_{t,i,j,z}^{K,V}}) + g_{i,j,z}}{\tau}\right)} \tag{6}$$

where $\theta_{\hat{\boldsymbol{m}}_t^{K,V}}$ is the learnable mask MLP *before* Gumbel-Softmax is taken. As demonstrated later with our ablations, the MLP parameterization is fundamental to our method - we found simply optimizing a mask matrix directly to be ineffective as the mask would have little to no updates during learning.

In order to keep the number of learnable parameters low and not increase the search space over new hyperparameters, we leverage a very simple two layer MLP for $\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}$ which consists of two linear layers and ReLU [54] operating on the fixed input tensor $\mathbf{1}$. All dimensions before the final layer are of dimension r, the same low rank

[2] We note that Shen *et al.* [52] use a similar approach in another setting.

[3] We use a default value of $\tau = 0.5$

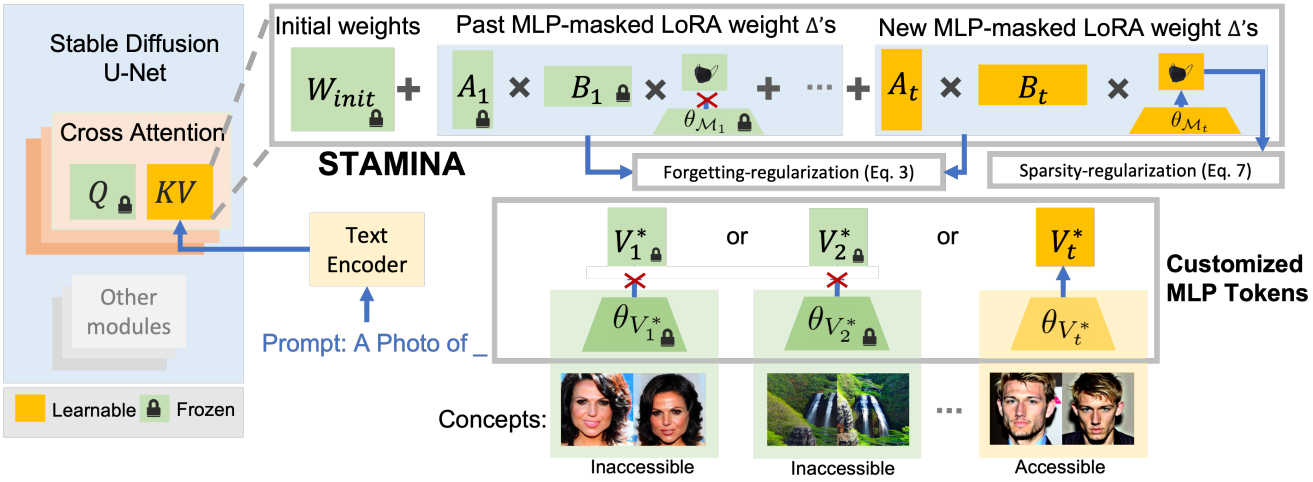

Figure 3. An overview of our approach. We learn custom tokens via MLPs operating on a fixed input. A prompt which includes the custom token is passed to the Stable Diffusion model. Our STAMINA approach modifies the key-value (K-V) projection in U-Net cross-attention modules without forgetting by using sparse, low-ranked, adaptations masked with MLP hard-attention. Importantly, trainable parameters, including the MLPs, can be reintegrated back into the original model backbone after training, incurring no cost to storage or inference.

as $\boldsymbol{A}_t^{K,V}$ and $\boldsymbol{B}_t^{K,V}$. While Eq. (5) decomposes how the pre-trained weights are adapted, **we emphasize that all of our learned parameters can be directly folded back into the original pre-trained weight, incurring no additional storage or computation costs at inference.**

Sparsity Regularization: In order to achieve desirable *sparsity* properties of the attention masks, we introduce a sparsity regularization on the positive outputs of $\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}$. The simple regularization encourages the mask to produce a 0 for each spot in the weight matrix residual rather than a 1. Thus, outputs of the low rank product $\boldsymbol{A}_t^{K,V}\boldsymbol{B}_t^{K,V}$ which are less important to learning the new task are zeroed out, leading to precise and minimal changes to the pre-trained weights. Furthermore, since the mask is truly binary (rather than sigmoid), the mask will not learn complex high-rank features (which could potentially interfere with the robust, low-rank fine-tuning properties of $\boldsymbol{A}_t^{K,V}$ and $\boldsymbol{B}_t^{K,V}$) and instead provides the model a clear delineation of which parameters are deemed important for the task at hand, as demonstrated in our ablation experiments. The formal loss is given as:

$$\mathcal{L}_{sparse} = ||\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}(\mathbf{1})||_1 \tag{7}$$

MLP for Custom Token Feature Embedding: To further improve the model's customization capability, we replace the custom token feature embeddings V_t^* from the previous work with learnable MLP modules $\theta_{V_t^*}$, parameterized in the same manner as $\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}$. This approach allows the model to adapt the token embeddings based on the specific characteristics of each task, providing a more efficient and flexible way of incorporating task-specific information. While not a key contribution of our work, we found that this custom token parameterization increases the amount of knowledge that can be "learned" by the custom tokens, requiring fewer changes to the model and thus less catastrophic interference and forgetting.

Putting it all together: Our final optimization is described as:

$$\min_{\left(\boldsymbol{W}_t^{K,V}\in\theta,\ \theta_{V_t^*}\right)} L_{SD}(x,\boldsymbol{\theta}) + \lambda_s\mathcal{L}_{sparse}(\theta_{\boldsymbol{\mathcal{M}}_t^{K,V}}(\mathbf{1})) + \lambda_f\mathcal{L}_{forget}(\boldsymbol{W}_{t-1}^{K,V}, \boldsymbol{W}_t^{K,V}) \tag{8}$$

where λ_s is a hyperparameter chosen with a simple exponential sweep. We re-emphasize that, by folding all learned parameters back into the original pre-trained weights, we ensure no additional storage or computational costs at inference, making our approach both storage and compute efficient.

5. Continual Text-To-Image Experiments

Implementation Details: For the most part, we use the same implementation details as Custom Diffusion [3] and C-LoRA [1]. We use 500 training steps (twice as many as reported in Kumari *et al.* [3] 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.* [1]). We use the prompt "a photo of a $V*$ X", where $V*$ is a learnable custom token, and X is the object category (e.g., *person*), which is removed [1] for C-LoRA and STAMINA. For LoRA, we searched for the rank using a simple exponential sweep and found that a rank of 16 sufficiently learns all concepts. Additional training details, including our chosen loss weighting hyperparameter values, are located in Appendix D.

Metrics: We report the same metrics which were originally proposed for Continual Diffusion [1]: (i) N_{param}, the number of parameters *trained* (i.e., unlocked during training a task) in % of the U-Net backbone model, (ii) A_{mmd}, the average MMD score ($\times 10^3$) after training on all concept tasks, and (iii) F_{mmd}, average forgetting. Consider N customization "tasks" $t \in \{1, 2, \ldots, N-1, N\}$, where N is the total number of concepts that will be shown to our model. We denote $X_{i,j}$ as task j images generated by the model after training task i. Furthermore, we denote $X_{D,j}$ as original dataset images for task j. Using these terms, we calculate the A_{mmd} metric (where lower is better) as:

$$A_{mmd} = \frac{1}{N} \sum_{j=1}^{N} MMD\left(\mathcal{F}_{clip}(X_{D,j}), \mathcal{F}_{clip}(X_{N,j})\right) \quad (9)$$

where $\mathcal{F}_{clip}$ denotes a function embedding into a rich semantic space using CLIP [55], and MMD denotes the Maximum Mean Discrepancy [56] with a polynomial kernel. To calculate the forgetting metric F_{mmd}, we calculate the average distance the images have changed over training, or:

$$F_{mmd} = \frac{1}{N-1} \sum_{j=1}^{N-1} MMD\left(\mathcal{F}_{clip}(X_{j,j}), \mathcal{F}_{clip}(X_{N,j})\right) \quad (10)$$

Baselines: In addition to the prior SOTA C-LoRA [1], we compare to recent customization methods Textual Inversion [2] and Custom Diffusion [3]. For a better comparison, we compare to a version of Textual Inversion [2] which leverages our same MLP custom tokens $\theta_{V_t^*}$ rather than V_t^*, TI++. For Custom Diffusion, we compare to both sequential training (denoted as *sequential*) as well as the constrained merging optimization variant (denoted as *merged*) which stores separate KV parameters for each concept individually and then merges the weights together into a single model using a closed-form optimization (see Kumari *et al.* [3] for more details). We also compare to the continual learning method EWC [22] combined with Custom Diffusion. We note here that Generative Replay [57] and DreamBooth [13] were found to be non-competitive in this setting [1], and thus we exclude the results from our experiments.

5.1. Continual Customization Results

We first benchmark on two 20-length task sequences: 512x512 resolution celebrity faces dataset, CelebFaces Attributes (Celeb-A) HQ [58, 59] (Table 1a), and waterfall landmarks of various resolutions from the Google Landmarks dataset v2 [60] (Table 1b). We refer the reader to Appendix E for additional details on benchmark dataset sampling. We first observe that the Custom Diffusion (CD) methods suffer in this setting, as previously reported by Smith *et al.* [1]. The only two competitive methods in this setting are the prior SOTA C-LoRA [1] and our MLP variant of Textual Inversion (TI) [2], TI++. While TI++ has no forgetting, tokens alone, even when learned using a MLP, do not have the capacity to capture fine-grain details in datasets such as faces and landmarks. We also see that, quantitatively, C-LoRA performs worse than TI in these benchmarks, as explained in Section 3.1. On the other hand, we see that our STAMINA approach establishes a clear SOTA performance on both benchmarks, while requiring only 500 training steps (compared to 2000 training steps for C-LoRA).

Table 1. **20-task Continual Diffusion Results:** A_{mmd} (↓) gives the average MMD score ($\times 10^3$) after training on all concept tasks, and F_{mmd} (↓) gives the average forgetting. N_{param} (↓) gives the number of parameters being trained as a % of the unmodified U-Net backbone size.

(a) Celeb-A HQ [58, 59]

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)
TI++ [2]	0.00	2.37	**0.00**
CD [3]	2.23	7.58	6.56
CD [3] (Merge)	2.23	13.84	8.61
CD+EWC [22]	2.23	7.39	5.81
C-LoRA [1]	0.09	2.25	0.33
Ours	0.19	**2.18**	0.03

(b) Google Landmarks dataset v2 [60]

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)
TI++ [2]	0.00	2.91	**0.00**
CD [3]	2.23	5.20	5.10
CD [3] (Merge)	2.23	14.83	8.43
CD+EWC [22]	2.23	5.10	3.56
C-LoRA [1]	0.09	3.09	0.38
Ours	0.19	**2.42**	0.01

What about longer task sequences? In Table 2a, we scale to a 50 length sequence containing *both* celebrity faces [58, 59] and landmarks [60]. We note that merge variant of CD is excluded, as it runs into memory errors in the merging process for such a long task sequence. Here, we notice two significant observations: (1) C-LoRA is now performing much worse than TI, while STAMINA is still retaining its SOTA performance. In Figure 4, we show qualitative results of images generated for all 50 tasks *after the full task sequence has been seen*. For example, the images in row 1 corresponding to "task 1" are being generated from the model after training on "task 50". Here, we see quite clearly that our method has gains over the existing techniques. While TI continues to learn decent im-

(a) **Target** (b) **CD** [3] (c) **TI++** [2] (d) **C-LoRA** [1] (e) **STAMINA**

Figure 4. Qualitative results of Continual Diffusion using celebrity faces from Celeb-A HQ [58, 59] and waterfalls from Google Landmarks [60]. Results are shown for 10 samples from all 50 concepts (↓) and are **generated from the model *after* training on all 50 concepts**. We sample for a variety of early (prone to forgetting) and late (prone to low plasticity) tasks. See I for source of target images.

Table 2. **50-task Continual Diffusion Results:** A_{mmd} (↓) gives the average MMD score ($\times 10^3$) after training on all concept tasks, and F_{mmd} (↓) gives the average forgetting. N_{param} (↓) gives the number of parameters being trained as a % of the unmodified U-Net backbone size.

(a) Full results

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)
TI++ [2]	0.00	2.52	**0.00**
CD [3]	2.23	5.99	5.67
CD+EWC [22]	2.23	5.15	3.95
C-LoRA [1]	0.09	3.09	1.41
Ours	0.19	**2.29**	0.01

(b) Ablation results

Ablation	A_{mmd} (↓)	F_{mmd} (↓)
Full Method	**2.29**	**0.01**
Ablate Mask	2.91	0.20
Ablate MLP Tokens	3.89	0.29
Ablate Mask MLP	2.82	0.58
Ablate Gumbel-Softmax	3.39	0.82
Ablate Sparsity	2.90	0.56

ages of high quality for each task, it misses important identifying details for most tasks. On the other hand, methods such as CD and C-LoRA suffer from catastrophic forgetting and saturated plasticity (which seems to lead to interference and forgetting in early tasks), respectively. We see that our STAMINA method has the best overall results for the full task sequence.

Ablations: We include an ablation study using the 50-concept benchmark in Table 2b. First, we ablate our mask, and see an increase in both forgetting F_{mmd} and MMD score A_{mmd}. We observe similar trends when we ablate the sparsity loss (7) and mask MLP parameterization, though with even higher forgetting in these two ablations. This implies it would be better to have no mask then have a ablated versions of our mask. We see a large gap in performance for "Ablate Gumbel-Softmax", which is where we use sigmoid activations instead of binary masks. Finally, we see the worst performance when we ablate the MLP Tokens and use V^* tokens instead. We found that increasing the steps can mitigate this to some degree, yet still under-performs our full method. *We have additional analysis in our Appendix, including a detailed analysis on weight interference (Appendix A), plasticity and KID [61] score metrics (Appendix B), a plot of plasticity vs number of trained tasks (C), and variance across runs (Appendix H).*

Multi-Concept Generations: In Figure 5, we provide some results demonstrating the ability to generate photos of multiple concepts in the same picture. We found that using the prompt styles "a photo of V* person posing next to V* waterfall" worked best. We also provide negative results in Appendix G and emphasize that there is much room for improvement in future work.

Figure 5. Our multi-concept generations after training on 50 tasks.

Table 3. **Impact of our Results (50-Task Sequence)**. UB stands for Upper Bound performance.

Method	TI++ [2]	Ours	UB
A_{mmd} (↓)	2.52	**2.29**	2.16
Δ from UB	0.36	**0.13**	0
% ↑ from UB	16.67%	**6.02%**	0%

Performance vs Training Time: Our method shows a 9.6% increase in training time over TI++, due to complex gradient propagation into the token embedding space. Despite this, our approach significantly improves A_{mmd} by 9.1%, justifying the trade-off, especially as there's no extra cost during inference. When compared to an **Upper Bound (UB)** set by CD[3], **our method more than halves the performance differential with TI++**, reducing it from 16.67% to 6.01%. This highlights our method's efficacy in continual learning, effectively balancing compute efficiency with performance enhancements.

5.2. STAMINA for Image Classification

We also demonstrate that STAMINA achieves SOTA performance for a long-task sequence in the well-established setting of *rehearsal-free continual learning for image classification*. We benchmark our approach using the 20 task ImageNet-R [39, 62] benchmark, which is composed of 200 total object classes with a wide collection of image styles, including cartoon, graffiti, and hard examples from the original ImageNet dataset [63]. We use the exact same experiment setting as the CODA-Prompt [40] paper with the ViT-B/16 backbone [64] pre-trained on ImageNet-1K [63] (additional details are available in Appendix F). We compare to Learning to Prompt (L2P) [38], DualPrompt [39], CODA-Prompt [40] (CODA-P), and C-LoRA [1]. For STAMINA and C-LoRA, we modify only the QKV projection matrices of self-attention blocks throughout the ViT model.

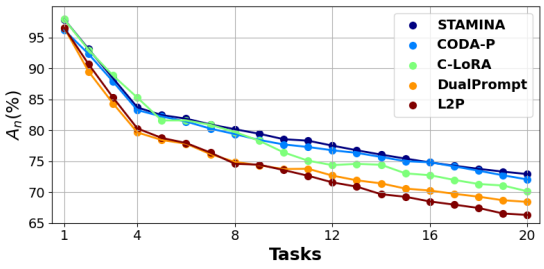

Figure 6. Image classification results on ImageNet-R [39, 62]. A_n gives the accuracy averaged over all seen tasks.

In Figure 6, we plot average accuracy A_n, or the accuracy with respect to all past classes averaged over all seen n tasks, for all 20 tasks. We choose the 20 task benchmark over the 5 and 10 task benchmarks because C-LoRA performs *under SOTA* in this longer task sequence, as reported in Smith *et al.* [1] (and is consistent with our Continual Diffusion findings). Our results demonstrate that STAMINA performs much better than C-LoRA, and *establishes a new SOTA performance on the benchmark* by outperforming all three prompting-based methods.

6. Conclusion & Limitations

In conclusion, our work addresses the challenge of scaling text-to-image diffusion models for continual customization across long concept sequences. We showed that the the SOTA C-LoRA's performance saturates with increasing tasks, and proposed STAMINA as a novel and efficient solution. STAMINA is composed of low-rank adapters with sparse, hard-attention masking and learnable MLP tokens, and can reintegrate all trainable parameters back into the original model backbone post-training. Our comprehensive evaluations on 3 different Continual Diffusion benchmarks not only highlight STAMINA's superior performance over the current SOTA, but also its efficiency, requiring significantly fewer training steps. Furthermore, we show that STAMINA is also SOTA for image classification, demonstrating its flexibility to achieve high performance in multiple continual learning settings.

Despite the success of our approach in generating long concept sequences, we acknowledge key limitations and cautions that must be addressed. **First, we strongly advocate for the responsible usage of our approach**, particularly with regards to generating faces of individuals. Consent, in our view, is paramount. Furthermore, given ethical concerns over using artists' and designers' images, we avoid artistic creativity in our work, and urge others to know and respect the sources of the data in which they customize their model with. In spite of these ethical considerations, we remain optimistic about the potential of our work to contribute positively to society in use cases such as mobile app entertainment.

References

[1] 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, 5, 6, 7, 8, 12, 13, 14

[2] 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, 6, 7, 8, 12, 14

[3] 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. 2, 3, 5, 6, 7, 8, 12, 14, 15

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[9] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[10] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[11] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B Tenenbaum. Compositional visual generation with composable diffusion models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 423–439. Springer, 2022. 2

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 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, 2015. 2

[13] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022. 2, 6

[14] Cusuh Ham, James Hays, Jingwan Lu, Krishna Kumar Singh, Zhifei Zhang, and Tobias Hinz. Modulating pretrained diffusion models for multimodal image synthesis. *arXiv preprint arXiv:2302.12764*, 2023. 2

[15] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Designing an encoder for fast personalization of text-to-image models. *arXiv preprint arXiv:2302.12228*, 2023. 2

[16] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. *arXiv preprint arXiv:2303.11305*, 2023.

[17] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept neurons in diffusion models for customized generation. *arXiv preprint arXiv:2303.05125*, 2023.

[18] Jing Shi, Wei Xiong, Zhe Lin, and Hyun Joon Jung. Instantbooth: Personalized text-to-image generation without test-time finetuning. *arXiv preprint arXiv:2304.03411*, 2023.

[19] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. *arXiv preprint arXiv:2302.13848*, 2023. 2

[20] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 2

[21] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020. 2

[22] 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, 6, 7, 12, 14

[23] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[24] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017. 2

[25] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. 2

[26] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019.

[27] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019.

[28] Nitin Kamra, Umang Gupta, and Yan Liu. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2017.

[29] Quang Pham, Chenghao Liu, and Steven Hoi. Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, 34:16131–16144, 2021.

[30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'17, pages 5533–5542, 2017.

[31] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 348–358, 2019.

[32] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9374–9384, October 2021.

[33] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020. 2

[34] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3366–3375, 2017. 2

[35] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019.

[36] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[37] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 2

[38] 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. 2, 8, 14

[39] 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. 2, 8, 13, 14

[40] 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. 2, 8, 13, 14

[41] Tyler L Hayes, Kushal Kafle, Robik Shrestha, Manoj Acharya, and Christopher Kanan. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pages 466–483. Springer, 2020. 2

[42] Tejas Srinivasan, Ting-Yun Chang, Leticia Leonor Pinto Alva, Georgios Chochlakis, Mohammad Rostami, and Jesse Thomason. Climb: A continual learning benchmark for vision-and-language tasks, 2022. 2

[43] James Seale Smith, Paola Cascante-Bonilla, Assaf Arbelle, Donghyun Kim, Rameswar Panda, David Cox, Diyi Yang, Zsolt Kira, Rogerio Feris, and Leonid Karlinsky. Construct-vl: Data-free continual structured vl concepts learning. *arXiv preprint arXiv:2211.09790*, 2022. 2, 3, 4

[44] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forget-

ting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018. 3

[45] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11321–11329, 2019. 3

[46] Jonathan Schwarz, Siddhant Jayakumar, Razvan Pascanu, Peter E Latham, and Yee Teh. Powerpropagation: A sparsity inducing weight reparameterisation. *Advances in neural information processing systems*, 34:28889–28903, 2021. 3

[47] Ali Abbasi, Parsa Nooralinejad, Vladimir Braverman, Hamed Pirsiavash, and Soheil Kolouri. Sparsity and heterogeneous dropout for continual learning in the null space of neural activations. In *Conference on Lifelong Learning Agents*, pages 617–628. PMLR, 2022. 3

[48] Johannes Von Oswald, Dominic Zhao, Seijin Kobayashi, Simon Schug, Massimo Caccia, Nicolas Zucchet, and João Sacramento. Learning where to learn: Gradient sparsity in meta and continual learning. *Advances in Neural Information Processing Systems*, 34:5250–5263, 2021. 3

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[50] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 3

[51] Simo Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning. https://github.com/cloneofsimo/lora. 3

[52] Chen Shen, Guo-Jun Qi, Rongxin Jiang, Zhongming Jin, Hongwei Yong, Yaowu Chen, and Xian-Sheng Hua. Sharp attention network via adaptive sampling for person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(10):3016–3027, 2018. 4

[53] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 4

[54] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 4, 13

[55] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 6

[56] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 6

[57] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2990–2999. Curran Associates, Inc., 2017. 6

[58] 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. 6, 7, 13, 14

[59] 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. 6, 7, 13, 14

[60] 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. 6, 7, 13

[61] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 7

[62] 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. 8, 13

[63] 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. 8, 13, 14

[64] 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. 8, 14

[65] 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. 14

appendix

Appendix

A. Analysis on Interference

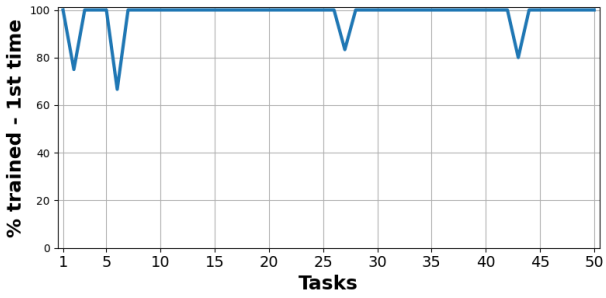

Figure A. Percentage of non-zero $\boldsymbol{W}_t^{K,V} - \boldsymbol{W}_{t-1}^{K,V}$ adaptations which are modifying the pre-trained weights $\boldsymbol{W}_{init}^{K,V}$ at the corresponding position for the first time. Here, a high number equates to low interference (good), and a low number equates 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 [1] and Custom Diffusion [3] 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 additinal 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.

$$P_{mmd} = \frac{1}{N}\sum_{j=1}^{N} MMD\left(\mathcal{F}_{clip}(X_{D,j}), \mathcal{F}_{clip}(X_{j,j})\right) \quad (11)$$

C. Plasticity Analysis

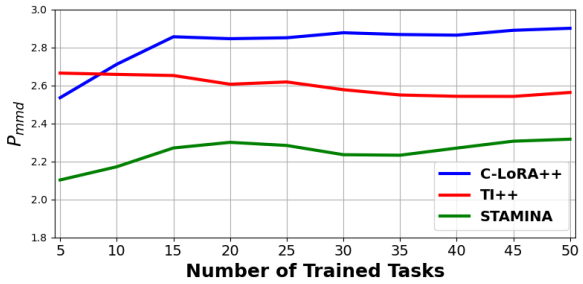

Figure B. Average plasticity P_{mmd} (↓) vs. number of trained tasks.

In Figure B, we directly compares plasticity vs. number of trained tasks for C-LoRA, TI++, and STAMINA in the Table 2a 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.

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 [3] methods, and a learning rate of $5e-4$ worked best for the LoRA-based methods and Textual Inversion [2]. Following Smith *et al.* [1], we use a loss weight of $1e6$ and $1e8$ for EWC [22] 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.* [3] due to our data being fine-grain concepts rather than simple objects)

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**

except for C-LoRA, which requires longer training steps (we use 2000 as reported in Smith *et al.* [1]). We regularize training with generated auxiliary data (as done in Smith *et al.* [1]) for *all* methods.

The simple MLPs used in our paper are composed of two linear layers and a ReLU [54] 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 [58, 59] and Google Landmarks v2 [60], 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 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 [39, 62] 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 [63]. 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 [40] paper. We implement

(a) Successes (b) Failures

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

our method and all baselines in PyTorch[65] using the ViT-B/16 backbone [64] pre-trained on ImageNet-1K [63]. 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 [40], Learning to Prompt (L2P) [38], DualPrompt [39], and C-LoRA [1]. We use the same classification head as L2P, DualPrompt, and CODA-Prompt. For additional details, we refer the reader to original CODA-Prompt [40] 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 [1].

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 [3]), 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

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

Table E. Celeb-A HQ [58, 59]

Method	N_{param} Train (%)	A_{mmd} (↓)	F_{mmd} (↓)
TI++ [2]	0.00	2.60 ± 0.23	$\mathbf{0.00 \pm 0.00}$
CD [3]	2.23	6.26 ± 0.99	5.78 ± 0.60
CD [3] (Merge)	2.23	14.34 ± 0.50	8.52 ± 0.09
CD+EWC [22]	2.23	5.88 ± 1.07	4.44 ± 0.98
C-LoRA [1]	0.09	2.81 ± 0.40	0.71 ± 0.50
Ours	0.19	$\mathbf{2.30 \pm 0.10}$	0.02 ± 0.01

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