HyperDreamBooth: HyperNetworks for Fast Personalization of Text-to-Image Models

Nataniel Ruiz Yuanzhen Li Varun Jampani Wei Wei Tingbo Hou
Yael Pritch Neal Wadhwa Michael Rubinstein Kfir Aberman
Google Research

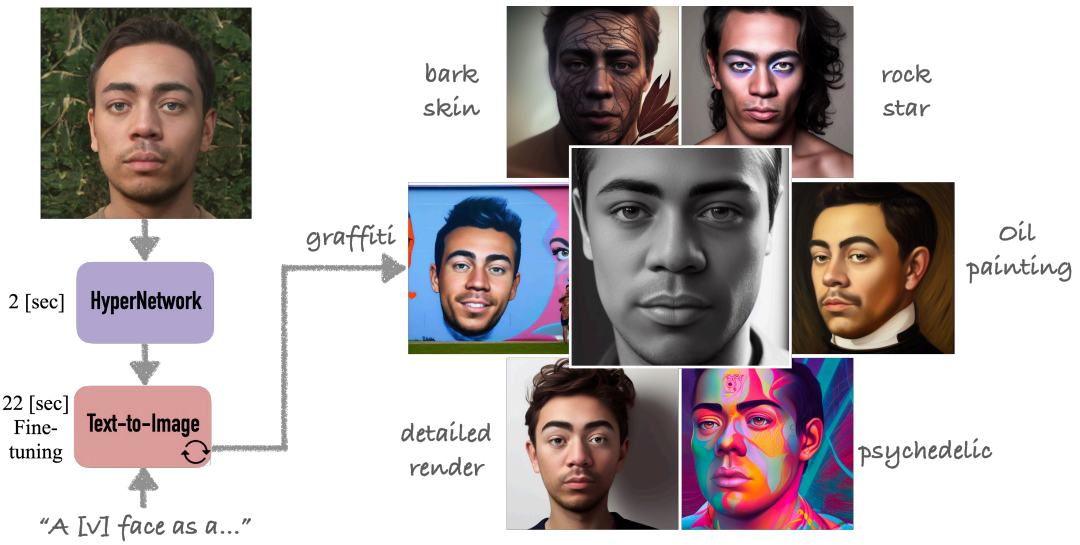

Figure 1. Using only a *single* input image, *HyperDreamBooth* is able to personalize a text-to-image diffusion model **25x** faster than DreamBooth [25], by using (1) a HyperNetwork to generate an initial prediction of a subset of network weights that are then (2) refined using fast finetuning for high fidelity to subject detail. Our method both *conserves model integrity and style diversity* while closely approximating the subject's essence and details.

Abstract

*Personalization has emerged as a prominent aspect within the field of generative AI, enabling the synthesis of individuals in diverse contexts and styles, while retaining high-fidelity to their identities. However, the process of personalization presents inherent challenges in terms of time and memory requirements. Fine-tuning each personalized model needs considerable GPU time investment, and storing a personalized model per subject can be demanding in terms of storage capacity. To overcome these challenges, we propose HyperDreamBooth—a hypernetwork capable of efficiently generating a small set of personalized weights from a single image of a person. By composing these weights into the diffusion model, coupled with fast finetuning, HyperDreamBooth can generate a person's face in various contexts and styles, with high subject details while also preserving the model's crucial knowledge of diverse styles and semantic modifications. Our method achieves personalization on faces in roughly 20 seconds, **25x** faster than DreamBooth and **125x** faster than Textual Inversion, using as few as one reference image, with the same quality and style diversity as DreamBooth. Also our method yields a model that is **10,000x** smaller than a normal DreamBooth model.*

1. Introduction

Recent work on text-to-image (T2I) personalization [25] has opened the door for a new class of creative applications. Specifically, for face personalization, it allows generation of new images of a specific face or person in different styles. The impressive diversity of styles is owed to the strong prior of pre-trained diffusion model, and one of the key properties of works such as DreamBooth [25], is the ability to implant a new subject into the model without damaging the model's prior. Another key feature of this type of method is that subject's essence and details are conserved even when applying vastly different styles. For example, when training on photographs of a person's face, one is able to generate new images of that person in animated cartoon styles,

where a part of that person's essence is preserved and represented in the animated cartoon figure - suggesting some amount of visual semantic understanding in the diffusion model. These are two core characteristics of DreamBooth and related methods, that we would like to leave untouched. Nevertheless, DreamBooth has some shortcomings: size and speed. For size, the original DreamBooth paper finetunes all of the weights of the UNet and Text Encoder of the diffusion model, which amount to more than 1GB for Stable Diffusion. In terms of speed, notwithstanding inference speed issues of diffusion models, training a DreamBooth model takes about 5 minutes for Stable Diffusion (1,000 iterations of training). This limits the potential impact of the work. In this work, we want to address these shortcomings, without altering the impressive key properties of DreamBooth, namely *style diversity* and *subject fidelity*, as depctied in Figure 1. Specifically, we want to *conserve model integrity* and *closely approximate subject essence* in a fast manner with a small model.

Our work proposes to tackle the problems of **size** and **speed** of DreamBooth, while preserving **model integrity**, **editability** and **subject fidelity**. We propose the following contributions:

- *Lighweight DreamBooth (LiDB)* - a personalized text-to-image model, where the customized part is roughly 100KB of size. This is achieved by training a DreamBooth model in a low-dimensional weight-space generated by a random orthogonal incomplete basis inside of a low-rank adaptation [14] weight space.
- New *HyperNetwork* architecture that leverages the Lightweight DreamBooth configuration and generates the customized part of the weights for a given subject in a text-to-image diffusion model. These provide a strong directional initialization that allows us to further finetune the model in order to achieve strong subject fidelity within a few iteration. Our method is **25x** faster than DreamBooth while achieving similar performances.
- We propose the technique of *rank-relaxed finetuning*, where the rank of a LoRA DreamBooth model is relaxed during optimization in order to achieve higher subject fidelity, allowing us to initialize the personalized model with an initial approximation using our HyperNetwork, and then approximate the high-level subject details using rank-relaxed finetuning.

One key aspect that leads us to investigate a HyperNetwork approach is the realization that in order to be able to synthesize specific subjects with high fidelity, using a given generative model, we have to "modify" its output domain, and insert knowledge about the subject into the model, namely by modifying the network weights.

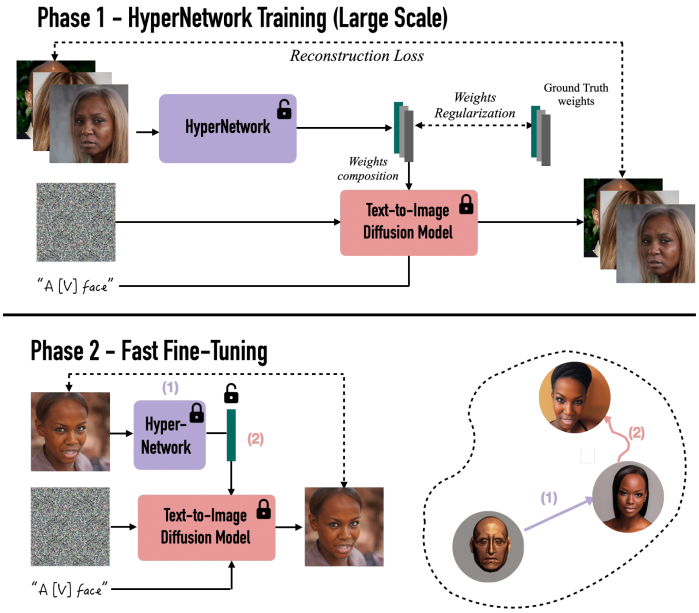

Figure 2. **HyperDreamBooth Training and Fast Fine-Tuning.** Phase-1: Training a hypernetwork to predict network weights from a face image, such that a text-to-image diffusion network outputs the person's face from the sentence *"a [v] face"* if the predicted weights are applied to it. We use pre-computed personalized weights for supervision, using an L2 loss, as well as the vanilla diffusion reconstruction loss. Phase-2: Given a face image, our hypernetwork predicts an initial guess for the network weights, which are then fine-tuned using the reconstruction loss to enhance fidelity.

2. Related Work

Text-to-Image Models Several recent models such as Imagen [26], DALL-E2 [22], Stable Diffusion (SD) [24], Muse [7], Parti [33], etc., demonstrate excellent image generation capabilities given a text prompt. Some Text-to-Image (T2I) models like SD and Muse also allow conditioning the generation with a given image via an encoder network. Techniques such as ControlNet [35] propose ways to incorporate new input conditioning such as depth. However, current text and image-based conditioning in these models do not capture sufficient subject details. For ease of experimentation, we demonstrate our HyperDreamBooth on the SD model, given its relatively small size. Yet, the proposed technique is generic and applicable to any T2I model.

Personalization of Generative Models Personalized generation aims to create varied images of a specific subject from one or a few reference images. Earlier approaches utilized GANs to manipulate subject images into new contexts. Pivotal tuning [23] fine-tunes GANs with inverted latent codes, while [20] fine-tunes StyleGAN with around 100 images for a personalized generative prior. Casanova et al. [6] condition a GAN with an input image to produce vari-

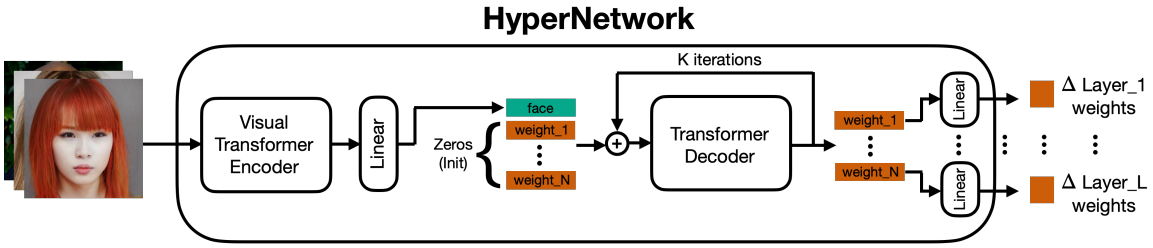

Figure 3. **HyperNetwork Architecture:** Our hypernetwork consists of a Visual Transformer (ViT) encoder that translates face images into latent face features that are then concatenated to latent layer weight features that are initiated by zeros. A Transformer Decoder receives the sequence of the concatenated features and predicts the values of the weight features in an iterative manner by refining the initial weights with delta predictions. The final layer weight deltas that will be added to the diffusion network are obtained by passing the decoder outputs through learnable linear layers.

ations. However, these GAN-based techniques often lack subject fidelity or diverse context in generated images.

HyperNetworks, introduced as auxiliary networks predicting weights for neural networks [12], have been applied in image generation tasks akin to personalization, such as StyleGAN inversion [3], resembling methods that aim to invert an image's latent code for editing in GAN spaces [2]. They have also been used in other tasks such as language modeling [15, 19, 21].

T2I Personalization via Finetuning Recent techniques enhance T2I models for improved subject fidelity and versatile text-based recontextualization. Textual Inversion [10] optimizes text embeddings on subject images for image generation, while [30] explores a richer inversion space capturing more subject details. DreamBooth [25] adapts entire network weights for subject fidelity. Various methods, like CustomDiffusion [18], SVDiff [13], LoRa [1, 14], StyleDrop [28], and DreamArtist [9], optimize specific network parts or use specialized tuning strategies. Despite their effectiveness, most of these techniques are slow, taking several minutes per subject for high-quality results.

Fast T2I Personalization Several recent and concurrent works aim for faster T2I model personalization. Some, like E4T [11] and ELITE [31], involve encoder learning followed by complete network finetuning, while our hypernetwork directly predicts low-rank network residuals. SuTI [8] creates a dataset for training a separate network to generate personalized images, but lacks high subject fidelity and affects the original model's integrity. Concurrent work InstantBooth [27] and Taming Encoder [16] introduce conditioning branches for diffusion models, requiring training on large datasets. FastComposer [32] focuses on identity blending in multi-subject generation using image encoders. Techniques like [4], Face0 [29], and Celebbasis [34] explore different conditioning or guidance approaches for efficient T2I personalization. However, balancing diversity, fidelity, and adherence to image distribution remains challenging. Our proposed hypernetwork-based approach directly predicts low-rank network residuals for subject-specific adaptation, differing from existing techniques.

3. Preliminaries

Latent Diffusion Models (LDM). Text-to-Image (T2I) diffusion models $\mathcal{D}_\theta(\epsilon, \mathbf{c})$ iteratively denoises a given noise map $\epsilon \in \mathbb{R}^{h \times w}$ into an image I following the description of a text prompt T, which is converted into an input text embedding $\mathbf{c} = \Theta(T)$ using a text encoder Θ. In this work, we use Stable Diffusion [24], a specific instatiation of LDM [24]. Briefly, LDM consists of 3 main components: An image encoder that encodes a given image into latent code; a decoder that decodes the latent code back to image pixels; and a U-Net denoising network $\mathcal{D}$ that iteratively denoises a noisy latent code. See [24] for more details.

DreamBooth [25] provides a network fine-tuning strategy to adapt a given T2I denoising network $\mathcal{D}_\theta$ to generate images of a specific subject. At a high-level, DreamBooth optimizes all the diffusion network weights θ on a few given subject images while also retaining the generalization ability of the original model with class-specific prior preservation loss [25]. In the case of Stable Diffusion [24], this amounts to finetuning the entire denoising UNet has over 1GB of parameters. In addition, DreamBooth on a single subject takes about 5 minutes with 1K training iterations.

Low Rank Adaptation (LoRA) [1, 14] provides a memory-efficient and faster technique for DreamBooth. Specifically, LoRa proposes to finetune the network weight residuals instead of the entire weights. That is, for a layer l with weight matrix $W \in \mathbb{R}^{n \times m}$, LoRa proposes to finetune the residuals ΔW. For diffusion models, LoRa is usually applied for the cross and self-attention layers of the network [1]. A key aspect of LoRa is the decomposition of ΔW matrix into low-rank matrices $A \in \mathbb{R}^{n \times r}$ and $B \in \mathbb{R}^{r \times m}$: $\Delta W = AB$. The key idea here is that $r << n$

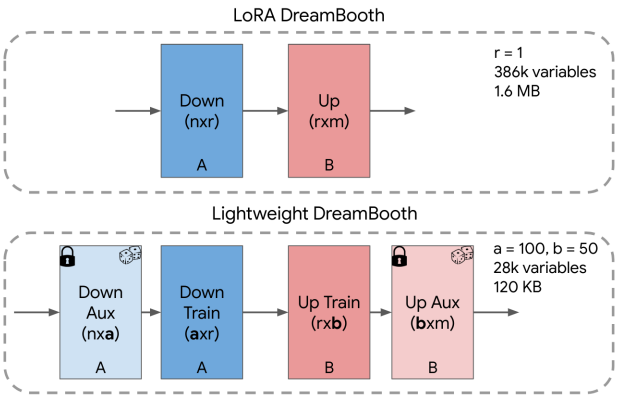

Figure 4. **Lightweight DreamBooth:** we propose a new low-dimensional weight-space for model personalization generated by a random orthogonal incomplete basis inside LoRA weight-space. This achieves models of roughly 100KB of size (**0.01%** of original DreamBooth and **7.5%** of LoRA DreamBooth size) and, surprisingly, is sufficient to achieve strong personalization results with solid editability.

and the combined number of weights in both A and B is much lower than the number of parameters in the original residual ΔW. Priors work show that this low-rank residual finetuning is an effective technique that preserves several favorable properties of the original DreamBooth while also being memory-efficient as well as fast, remarkably even when we set $r = 1$. For stable diffusion 1.5 model, LoRA-DreamBooth with $r = 1$ has approximately 386K parameters corresponding to only about 1.6MB in size.

4. Method

Our approach consists of 3 core elements which we explain in this section. We begin by introducing the concept of the Lightweight DreamBooth (LiDB) and demonstrate how the Low-Rank decomposition (LoRa) of the weights can be further decomposed to effectively minimize the number of personalized weights within the model. Next, we discuss the HyperNetwork training and the architecture the model entails, which enables us to predict the LiDB weights from a single image. Lastly, we present the concept of rank-relaxed fast fine-tuning, a technique that enables us to significantly amplify the fidelity of the output subject within a few seconds. Fig. 2 shows the overview of hypernetwork training followed by fast fine-tuning strategy in our HyperDreamBooth technique.

4.1. Lightweight DreamBooth (LiDB)

Given our objective of generating the personalized subset of weights directly using a HyperNetwork, it would be beneficial to reduce their number to a minimum while maintaining strong results for subject fidelity, editability and style diversity. To this end, we propose a new low-dimensional weight space for model personalization which allows for personalized diffusion models that are 10,000 times smaller than a DreamBooth model and more than 10 times smaller than a LoRA DreamBooth model. Our final version has only 30K variables and takes up only 120 KB of storage space.

The core idea behind Lightweight DreamBooth (LiDB) is to further decompose the weight-space of a rank-1 LoRa residuals. Specifically, we do this using a random orthogonal incomplete basis within the rank-1 LoRA weight-space. We illustrate the idea in Figure 4. The approach can also be understood as further decomposing the Down (A) and Up (B) matrices of LoRA into two matrices each: $A = A_{\text{aux}} A_{\text{train}}$ with $A_{\text{aux}} \in \mathbb{R}^{n \times a}$ and $A_{\text{train}} \in \mathbb{R}^{a \times r}$ and $B = B_{\text{train}} B_{\text{aux}}$ with $B_{\text{train}} \in \mathbb{R}^{r \times b}$ and $B_{\text{aux}} \in \mathbb{R}^{b \times m}$. where the *aux* layers are randomly initialized with row-wise orthogonal vectors and are frozen; and the train layers are learned. Two new hyperparameters are introduced: a and b, which we set experimentally. Thus the weight-residual in a LiDB linear layer is represented as:

$$\Delta W x = A_{\text{aux}} A_{\text{train}} B_{\text{train}} B_{\text{aux}}, \tag{1}$$

where $r << \min(n, m)$, $a < n$ and $b < m$. A_{aux} and B_{aux} are randomly initialized with orthogonal row vectors with constant magnitude - and frozen, and B_{train} and A_{train} are learnable. Surprisingly, we find that with $a = 100$ and $b = 50$, which yields models that have only 30K trainable variables and are 120 KB in size, personalization results are strong and maintain subject fidelity, editability and style diversity. We show results for personalization using LiDB in the experiments section.

4.2. HyperNetwork for Fast Personalization of Text-to-Image Models

We propose a HyperNetwork for fast personalization of a pre-trained T2I model. Let $\tilde{\theta}$ denote the set of all LiDB residual matrices: A_{train} and B_{train} for each of the cross-attention and self-attention layers of the T2I model. In essence, the HyperNetwork $\mathcal{H}_\eta$ with parameters η takes the given image $\mathbf{x}$ as input and predicts the LiDB low-rank residuals $\hat{\theta} = \mathcal{H}_\eta(\mathbf{x})$. The HyperNetwork is trained on a dataset of domain-specific images with a vanilla diffusion denoising loss and a weight-space loss:

$$L(\mathbf{x}) = \alpha ||\mathcal{D}(\mathbf{x} + \epsilon, \mathbf{c}) - \mathbf{x}||_2^2 + \beta ||\hat{\theta} - \theta||_2^2, \tag{2}$$

where $\mathbf{x}$ is the reference image, θ are the pre-optimized weight parameters of the personalized model for image $\mathbf{x}$, $\mathcal{D}$ is the diffusion model conditioned on the noisy image $\mathbf{x} + \epsilon$ and the supervisory text-prompt $\mathbf{c}$, and finally α and β are hyperparameters that control for the relative weight of each loss. Fig. 2 (top) illustrates the hypernetwork training.

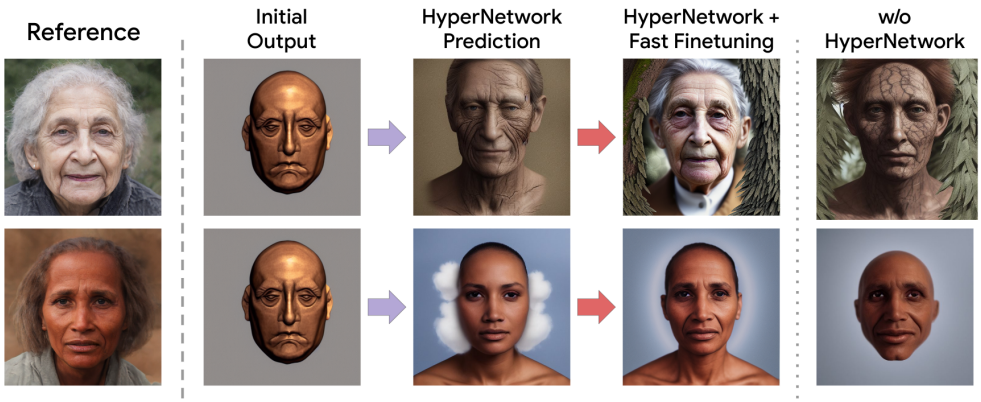

Figure 5. **HyperNetwork + Fast Finetuning** achieves strong results. Each row displays outputs from initial HyperNetwork prediction (HyperNetwork Prediction column) and after HyperNetwork prediction with fast finetuning (HyperNetwork + Fast Finetuning). Results without the HyperNetwork component highlight its importance.

Supervisory Text Prompt We propose to eschew any type of learned token embedding for this task, and our hypernetwork acts solely to predict the LiDB weights of the diffusion model. We simply propose to condition the learning process "a [V] face" for all samples, where [V] is a rare identifier described in [25]. At inference time variations of this prompt can be used, to insert semantic modifications, for example "a [V] face in impressionist style".

HyperNetwork Architecture Concretely, as illustrated in Fig. 3, we separate the HyperNetwork architecture into two parts: a ViT image encoder and a transformer decoder. We use a ViT-H for the encoder architecture and a 2-hidden layer transformer decoder for the decoder architecture. The transformer decoder is a strong fit for this type of weight prediction task, since the output of a diffusion UNet or Text Encoder is sequentially dependent on the weights of the layers, thus in order to personalize a model there is interdependence of the weights from different layers. In previous work [3, 12], this dependency is not rigorously modeled in the HyperNetwork, whereas with a transformer decoder with a positional embedding, this positional dependency is modeled - similar to dependencies between words in a language model transformer. To the best of our knowledge this is the first use of a transformer decoder as a HyperNetwork.

Iterative Prediction We find that the HyperNetwork achieves better and more confident predictions given an iterative learning and prediction scenario [3], where intermediate weight predictions are fed to the HyperNetwork and the network's task is to improve that initial prediction. We only perform the image encoding once, and these extracted features $\mathbf{f}$ are then used for all rounds of iterative prediction for the HyperNetwork decoding transformer $\mathcal{T}$. This speeds up training and inference, and we find that it does not affect the quality of results. Specifically, the forward pass of $\mathcal{T}$ becomes:

$$\hat{\theta}_k = \mathcal{T}(\mathbf{f}, \hat{\theta}_{k-1}), \tag{3}$$

where k is the current iteration of weight prediction, and terminates once $k = s$, where s is a hyperparameter controlling the maximum amount of iterations. Weights θ are initialized to zero for $k = 0$. Trainable linear layers are used to convert the decoder outputs into the final layer weights. We use the CelebAHQ dataset [17] for training the HyperNetwork, and find that we only need 15K identities to achieve strong results, much less data than other concurrent methods. For example 100k identities for E4T [11] and 1.43 million identities for InstantBooth [27].

4.3. Rank-Relaxed Fast Finetuning

We find that the initial HyperNetwork prediction is in great measure directionally correct and generates faces with similar semantic attributes (gender, facial hair, hair color, skin color, etc.) as the target face consistently. Nevertheless, fine details are not sufficiently captured. We propose a final fast finetuning step in order to capture such details, which is magnitudes faster than DreamBooth, but achieves virtually identical results with strong subject fidelity, editability and style diversity. Specifically, we first predict personalized diffusion model weights $\hat{\theta} = \mathcal{H}(\mathbf{x})$ and then subsequently finetune the weights using the diffusion denoising loss $L(\mathbf{x}) = ||\mathcal{D}_{\hat{\theta}}(\mathbf{x} + \epsilon, \mathbf{c}) - \mathbf{x}||_2^2$. A key contribution of our work is the idea of *rank-relaxed* finetuning, where we relax the rank of the LoRA model from $r = 1$ to $r > 1$ before fast finetuning. Specifically, we add the predicted HyperNetwork weights to the overall weights of the model, and then perform LoRA finetuning with a new higher rank. This expands the capability of our method of approximating high-frequency details of the subject, giving higher subject fidelity than methods that are locked to lower ranks of

Table 1. **Comparisons.** We compare our method for face identity preservation (Face Rec.), subject fidelity (DINO, CLIP-I) and prompt fidelity (CLIP-T) to DreamBooth and Textual Inversion. We find that our method preserves identity and subject fidelity more closely, while achieving a higher score in prompt fidelity.

Method	Face Rec. ↑	DINO ↑	CLIP-I ↑	CLIP-T ↑
Ours	**0.655**	**0.473**	**0.577**	**0.286**
DreamBooth	0.618	0.441	0.546	0.282
Textual Inversion	0.623	0.289	0.472	0.277

Table 2. **Comparisons with DreamBooth**. We compare our method to differently tuned versions of DreamBooth that minimize optimization time. Altering hyperparameters by increasing the learning rate and decreasing iterations leads to degraded results in DreamBooth. DreamBooth-Agg-1 uses 400 iterations and DreamBooth-Agg-2 uses 40 iterations as opposed to the normal 1200 iterations used in our vanilla DreamBooth.

Method	Face Rec. ↑	DINO ↑	CLIP-I ↑	CLIP-T ↑
Ours	**0.655**	**0.473**	**0.577**	0.286
DreamBooth	0.618	0.441	0.546	0.282
DreamBooth-Agg-1	0.615	0.323	0.431	**0.313**
DreamBooth-Agg-2	0.616	0.360	0.467	0.302

weight updates. To the best of our knowledge we are the first to propose such rank-relaxed LoRA models.

We use the same supervision text prompt "a [V] face" this fast finetuning step. We find that given the HyperNetwork initialization, fast finetuning can be done in 40 iterations, which is **25x** faster than DreamBooth [25] and LoRA DreamBooth [1]. We show an example of initial, intermediate and final results in Figure 5.

5. Experiments

We implement our HyperDreamBooth on the Stable Diffusion v1.5 diffusion model and we predict the LoRa weights for all cross and self-attention layers of the diffusion UNet as well as the CLIP text encoder. For privacy reasons, all face images used for visuals are synthetic, from the SFHQ dataset [5]. For training, we use 15K images from CelebA-HQ [17].

5.1. Subject Personalization Results

Our method achieves strong personalization results for widely diverse faces, with performance that is identically or surpasses that of the state-of-the art optimization driven methods [10, 11, 25]. Moreover, we achieve very strong editability, with semantic transformations of face identities into highly different domains such as figurines and animated characters, and we conserve the strong style prior of the model which allows for a wide variety of style generations. We show results in Figure 6.

Table 3. **HyperNetwork Ablation**. We ablate components: No Hyper (without hypernetwork at test-time), Only Hyper (using hypernetwork prediction without fast finetuning), and our full method without iterative prediction (k=1). Our full method performs best for all fidelity metrics, with No Hyper achieving slightly better prompt following.

Method	Face Rec. ↑	DINO ↑	CLIP-I ↑	CLIP-T ↑
Ours	**0.655**	**0.473**	**0.577**	0.286
No Hyper	0.647	0.392	0.498	**0.299**
Only Hyper	0.631	0.414	0.501	0.298
Ours (k=1)	0.648	0.464	0.570	0.288

5.2. Comparisons

Qualitative Comparisons We compare our method to Textual Inversion [10], DreamBooth [25] and E4T [11]. Results are shown in Figure 7. We observe that our method strongly outperforms both Textual Inversion and DreamBooth generally, in the one-input-image regime - and obtains strong results compared to E4T, especially in cases where E4T overfits to the reference face pose and realistic appearance, even though the output should be highly stylized.

Quantitative Comparisons and Ablations We compare our method to Textual Inversion and DreamBooth using face recognition metrics ("Face Rec." from a VGGFace2 Inception ResNet), along with DINO, CLIP-I, and CLIP-T metrics [25]. Using 100 CelebAHQ identities and 30 prompts (style modification and recontextualization), totaling 30,000 samples, Table 1 illustrates our approach outperforming in all metrics. However, face recognition metrics are relatively weak here due to network training limitations (realistic face bias). To compensate, we conduct a user study (details below).

We also conduct comparisons with more aggressive

Table 4. **User Study**. Given limitations of face recognition networks (stylized faces are OOD), we conduct an identity fidelity user study comparing our stylized generations against DB and TI. Our approach generally receives higher user preference.

	Ours	DB	Undecided	Ours	TI	Undecided
Pref. ↑	**64.8%**	23.3%	11.9%	**70.6%**	21.6%	7.8%

Table 5. **User Stylization and Identity Preference**. We compare the user preference of stylization and identity between our approach and the SoTA approach E4T. Users generally prefer our method.

	Ours	E4T	Undecided
Preference ↑	**60.0%**	37.5%	2.5%

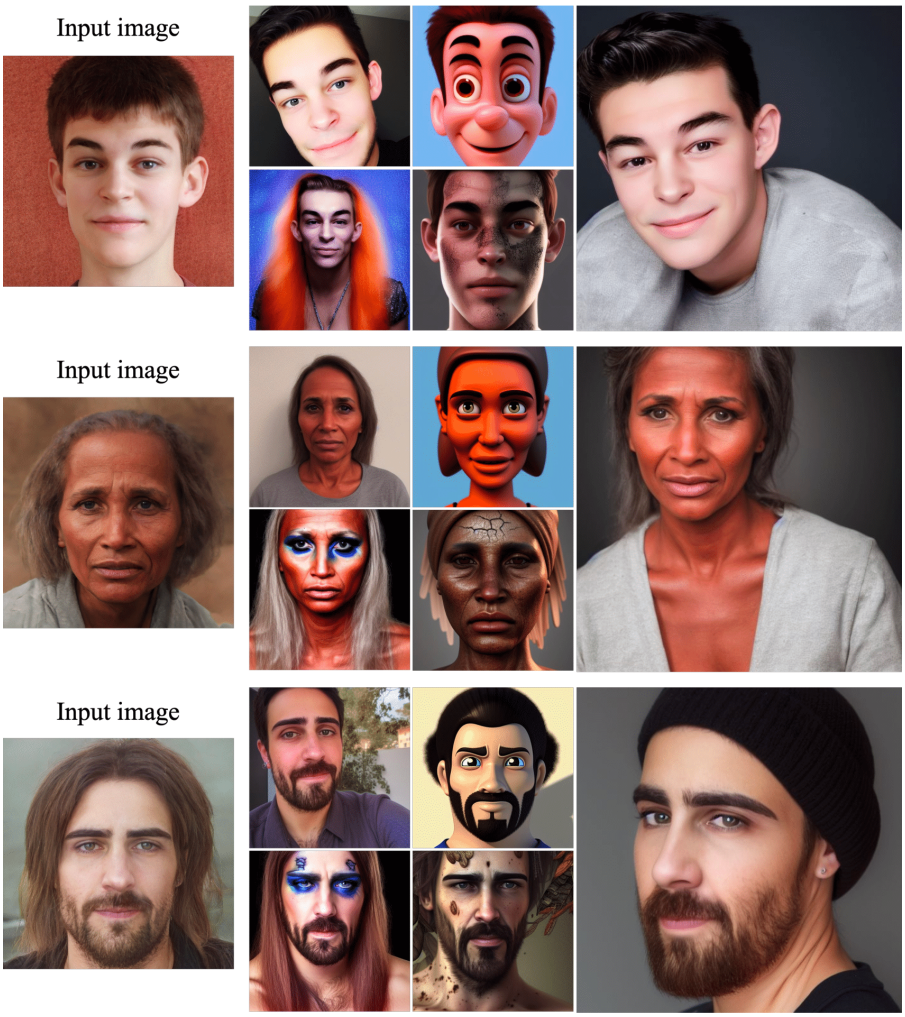

Figure 6. **Results Gallery:** Our method can generate novel artistic and stylized results of diverse subjects (depicted in an input image, left) with considerable editability while maintaining the integrity to the subject's key facial characteristics. The output images were generated with the following captions (top-left to bottom-right): *"An Instagram selfie of a [V] face", "A Pixar character of a [V] face", "A [V] face with bark skin", "A [V] face as a rock star"*. Rightmost: *"A professional shot of a [V] face"*.

DreamBooth training with altered iterations and learning rates. Specifically, DreamBooth-Agg-1 (400 iterations) and DreamBooth-Agg-2 (40 iterations) differ from our 1200-iteration vanilla DreamBooth. Table 2 reveals that aggressive DreamBooth training without our HyperNetwork initialization generally degrades results.

Additionally, we show an ablation study that explores our method's components: removing the HyperNetwork (No Hyper), utilizing only the HyperNetwork without fine-tuning (Only Hyper), and our full setup without iterative predictions (k=1). Table 3 demonstrates that our complete setup achieves superior subject fidelity, albeit with a slightly lower prompt following metric.

User Study We conduct a user study for face identity preservation of outputs and compare our method to DreamBooth and Textual Inversion. Specifically, we present the reference face image and two random generations using the same prompt from our method and the baseline, and ask the user to rate which one has most similar face identity to the reference face image. We test a total of 25 identities, and query 5 users per question, with a total of 1,000 sample pairs evaluated. We take the majority vote for each pair. We present our results in Table 4, where we show a strong preference for face identity preservation of our method.

Finally, we present a user study for overall preference of both subject fidelity and style fidelity and compare our approach to the published state-of-the-art E4T method [11] on a set of identities from the SFHQ dataset, with E4T kindly run by the authors. We present both the reference subject image as well as a reference style image and ask users which output they prefer with respect to both identity preservation

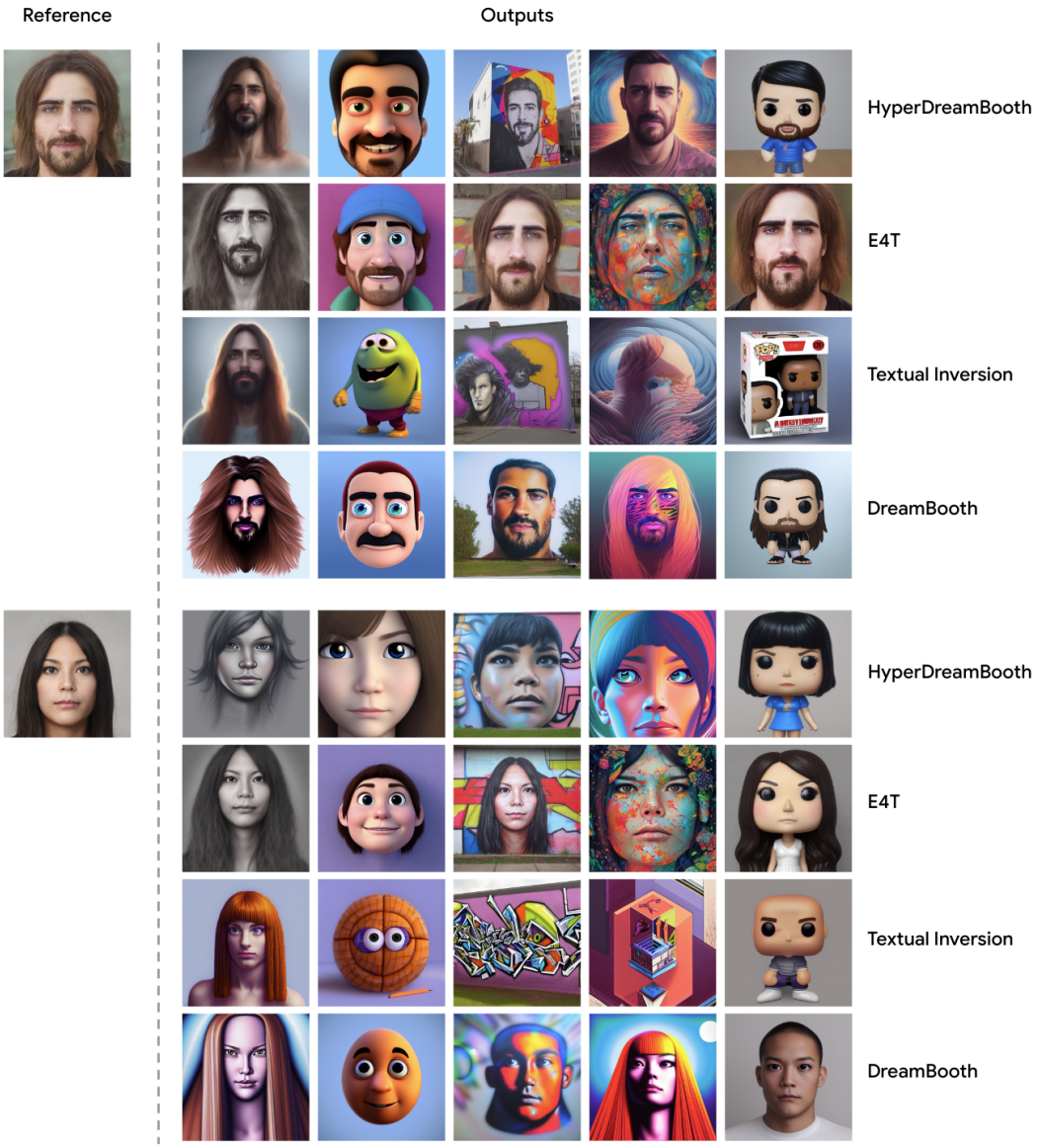

Figure 7. **Qualitative Comparison:** We compare random samples from our method (HyperDreamBooth), E4T [11], DreamBooth [25] and Textual Inversion [10] for two different identities and five different stylistic prompts. We observe that our method generally achieves very strong editability while preserving identity, generally surpassing competing methods in the single-reference regime. E4T shows strong performance but can tend to overfit to the reference head pose and realistic appearance, even when the image should be strongly stylized.

and style preservation. We test 10 identities, 4 prompts per identity, and query 15 users per question, totaling 600 samples. Results are shown in Table 5, where we observe a preference of users for our method. Although E4T is a method that achieves strong results and preserves identity well, we observe slightly less qualitative editability as well as some consistency errors with hard prompts. Note our method is trained on 15k identities vs. 100k identities for E4T.

6. Conclusion

In this work, we presented *HyperDreamBooth* a new method for fast and lightweight subject personalization of diffusion models. It leverages a HyperNetwork to generate Lightweight DreamBooth (LiDB) parameters for a diffusion model with a subsequent fast rank-relaxed finetuning that achieves a sharp reduction in size and speed compared to DreamBooth and other optimization-based personalization work. We showed that it produces high-quality and diverse images of faces with different styles and semantic modifications, while preserving subject details and model integrity.

References

[1] Low-rank adaptation for fast text-to-image diffusion fine-tuning. https://github.com/cloneofsimo/lora, 2022. 3, 6

[2] Yuval Alaluf, Or Patashnik, and Daniel Cohen-Or. Restyle: A residual-based stylegan encoder via iterative refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6711–6720, 2021. 3

[3] Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022. 3, 5

[4] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 843–852, 2023. 3

[5] David Beniaguev. Synthetic faces high quality (sfhq) dataset. https://github.com/SelfishGene/SFHQ-dataset, 2022. 6

[6] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. *Advances in Neural Information Processing Systems*, 34:27517–27529, 2021. 2

[7] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 2

[8] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Ruiz, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186*, 2023. 3

[9] Ziyi Dong, Pengxu Wei, and Liang Lin. Dreamartist: Towards controllable one-shot text-to-image generation via contrastive prompt-tuning. *arXiv preprint arXiv:2211.11337*, 2022. 3

[10] 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. 3, 6, 8

[11] 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. 3, 5, 6, 7, 8

[12] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 3, 5

[13] 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. 3

[14] 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. 2, 3

[15] Hamish Ivison, Akshita Bhagia, Yizhong Wang, Hannaneh Hajishirzi, and Matthew Peters. Hint: Hypernetwork instruction tuning for efficient zero-shot generalisation. *arXiv preprint arXiv:2212.10315*, 2022. 3

[16] Xuhui Jia, Yang Zhao, Kelvin CK Chan, Yandong Li, Han Zhang, Boqing Gong, Tingbo Hou, Huisheng Wang, and Yu-Chuan Su. Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. *arXiv preprint arXiv:2304.02642*, 2023. 3

[17] 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. 5, 6

[18] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 3

[19] Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. *arXiv preprint arXiv:2304.08467*, 2023. 3

[20] Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. *ACM Transactions on Graphics (TOG)*, 41(6):1–10, 2022. 2

[21] Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. Hypertuning: Toward adapting large language models without back-propagation. In *International Conference on Machine Learning*, pages 27854–27875. PMLR, 2023. 3

[22] 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

[23] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on graphics (TOG)*, 42(1):1–13, 2022. 2

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[25] 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. 2022. 1, 3, 5, 6, 8

[26] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[27] 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. 3, 5

[28] Kihyuk Sohn, Nataniel Ruiz, Kimin Lee, Daniel Castro Chin, Irina Blok, Huiwen Chang, Jarred Barber, Lu Jiang, Glenn Entis, Yuanzhen Li, Yuan Hao, Irfan Essa, Michael Rubinstein, and Dilip Krishnan. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023. 3

[29] Dani Valevski, Danny Wasserman, Yossi Matias, and Yaniv Leviathan. Face0: Instantaneously conditioning a text-to-image model on a face. *arXiv preprint arXiv:2306.06638*, 2023. 3

[30] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. $p+$: Extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 3

[31] 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. 3

[32] Guangxuan Xiao, Tianwei Yin, William T Freeman, Frédo Durand, and Song Han. Fastcomposer: Tuning-free multi-subject image generation with localized attention. *arXiv preprint arXiv:2305.10431*, 2023. 3

[33] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022. 2

[34] Ge Yuan, Xiaodong Cun, Yong Zhang, Maomao Li, Chenyang Qi, Xintao Wang, Ying Shan, and Huicheng Zheng. Inserting anybody in diffusion models via celeb basis. *arXiv preprint arXiv:2306.00926*, 2023. 3

[35] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2