DreamCache: Finetuning-Free Lightweight Personalized Image Generation via Feature Caching

Emanuele Aiello[◇] Umberto Michieli[†] Diego Valsesia[◇] Mete Ozay[†] Enrico Magli[◇]
[◇] Politecnico di Torino [†] Samsung R&D Institute UK

Abstract

Personalized image generation requires text-to-image generative models that capture the core features of a reference subject to allow for controlled generation across different contexts. Existing methods face challenges due to complex training requirements, high inference costs, limited flexibility, or a combination of these issues. In this paper, we introduce DreamCache, a scalable approach for efficient and high-quality personalized image generation. By caching a small number of reference image features from a subset of layers and a single timestep of the pretrained diffusion denoiser, DreamCache enables dynamic modulation of the generated image features through lightweight, trained conditioning adapters. DreamCache achieves state-of-the-art image and text alignment, utilizing an order of magnitude fewer extra parameters, and is both more computationally effective and versatile than existing models.[1]

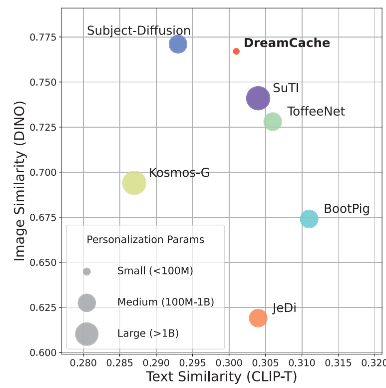

Figure 1. **DreamCache** is a finetuning-free personalized image generation method that achieves an optimal balance between subject fidelity, memory efficiency, and adherence to text prompts.

1. Introduction

Recent advancements in text-to-image generation, fueled by the development of diffusion models [10, 28], have enabled high-quality and diverse image generation from textual descriptions. Diffusion models [23, 25] gradually transform random noise into images through a sequence of denoising steps, conditioned on the input text prompt.

An active area of research is personalizing these models, enabling the generation of novel images of a reference subject in various contexts, while maintaining flexibility for text-based editing. Early personalization techniques [1, 6, 8, 24, 30, 32, 35, 35], such as the seminal DreamBooth [24] relied on fine-tuning (FT) the generative model for each reference subject. However, these approaches are often impractical for many use cases due to costly test-time FT, which can take several minutes per subject. To address this, FT-free (*i.e.*, zero-shot) personalized image generation methods have emerged to eliminate test-time optimization. These FT-free approaches can be broadly categorized into two families: encoder-based methods and reference-based methods, each with distinct drawbacks.

Encoder-based methods [7, 15, 20, 33, 36] utilize dedicated image encoders, such as CLIP [22] or DINO [2], to extract relevant features from reference images. While these encoders can produce high-quality results, they are often large, require extensive training to align text and image features, and reduce the model's flexibility [13, 15, 19, 36].

In contrast, reference-based methods [21, 37] condition the diffusion model directly on reference features drawn from the U-Net denoiser, integrating these features at each denoising step. While effective, these methods require feature extraction at every generation step, leading to higher computational costs and memory demands. Additionally, they often require an input textual caption for conditioning, which introduces variability and can decrease output precision.

Some recent works have proposed to finetune the U-Net backbone itself [21, 37, 39]. However, this hinders the model's ability to switch between personalized and non-personalized tasks and risks inducing the "language drift" phenomenon, where the personalization training degrades the model's linguistic comprehension [12, 24].

[1]Code, model, and synthetic dataset can be found at https://github.com/Emanuele97x/DreamCache

Table 1. **Methods overview.** Our DreamCache achieves state-of-the-art generation quality at reduced computational costs. *: value refers to the personalization stage for each personal subject.

Method	FT-free	Enc-free	Plug&Play	Ref-UNet-free	Extra Params	Train Params	# Dataset	Train Time
Textual Inversion [6]	✗	✓	✓	✓	768*	768*	3-5*	50 min*
DreamBooth [24]	✗	✓	✗	✓	-	0.9B*	3-5*	10 min*
Custom Diffusion [12]	✗	✓	✗	✓	-	57M*	3-5*	10 min*
ELITE [33]	✓	✗	✗	✓	457M	77M	125K	14 days
BLIP-Diffusion [13]	✓	✗	✓	✓	380M	1.5B	129M	96 days
IP-Adapter [36]	✓	✗	✓	✓	402M	22M	10M	28 days
Kosmos-G [19]	✓	✗	✓	✓	1.6B	1.6B	9M	-
JeDi [37]	✓	✓	✗	✗	-	0.9B	3M	48 days
SuTI [4]	✓	✗	✗	✓	400M	2.5B	500K	-
Subject-Diffusion [15]	✓	✗	✓	✗	700M	700M	76M	-
BootPig [21]	✓	✓	✗	✗	0.95B	0.95B	200K	18 hours
ToffeeNet [39]	✓	✗	✗	✓	632M	0.9B	5M	-
CAFE [38]	✓	✗	✗	✓	14B	1B	355K	-
DreamCache (ours)	✓	✓	✓	✓	25M	25M	400K	40 hours

In this work, we propose DreamCache, a novel finetuning-free approach to personalized image generation, that overcomes the limitations of existing methods (see Fig. 1) by using a feature caching mechanism that enables text-free encoding and efficient conditioning during personalization. Specifically, we first create a synthetic dataset [21] containing triplets of captions, target images, and reference subjects to capture subjects in various contexts. Next, we pretrain lightweight attention-based conditioning adapters to inject subject-specific features into the image generation process. During personalization, the reference image is processed through the pretrained denoiser of the base diffusion model without text conditioning, thus eliminating the need for user-generated captions, while caching features from a small subset of layers at a single timestep. For personalized sampling, these cached features are injected into the denoiser through the pretrained conditioning adapters.
Table 1 summarizes the key properties of existing methods and illustrates how DreamCache fits within the current landscape; further details are explored in Sec. 2. As an encoder-free approach, DreamCache introduces only a small number of additional parameters, making it significantly lightweight and suitable for deployment on resource-constrained devices. For example, methods like [33] and [13] introduce 380M parameters due to their reliance on CLIP encoders, whereas DreamCache requires only 25M additional parameters. Moreover, caching features from a few selected U-Net layers at a single preprocessing timestep bypasses the need for full U-Net reference processing during generation, leading to substantial computational and memory savings that enable real-time, high-quality personalized generation. Another key advantage of DreamCache is its *plug-and-play* nature, allowing concurrent generation of personalized and non-personalized content without altering original U-Net weights, thus preserving the integrity of the original model and enabling a wider range of deployment scenarios, especially on mobile platforms.

In summary, DreamCache represents a significant step toward practical, and scalable personalized image generation, with the following contributions:

- We propose a feature caching approach that creates multi-resolution representations of the reference image in a caption-free and efficient manner.
- We design an attention-based conditioning mechanism that leverages the cached features for personalized image generation, achieving computational- and memory- efficient personalized sampling.
- Our approach achieves state-of-the-art quality in personalized image generation at substantially lower computational and data costs compared to existing methods.

2. Background and Related Work

Personalized image generation aims at generating images containing a specific subject. This task has been widely studied, resulting in two main approaches: fine-tuning methods, which require test-time finetuning on multiple subject reference images, and finetuning-free (zero-shot) methods, which learn a generalizable conditioning mechanism to generate reference subjects without the need for further optimization.

Finetuning-based Personalization DreamBooth [24] finetunes the entire U-Net with reference images while introducing a regularization loss to mitigate overfitting. On the other hand, Custom Diffusion [12] only finetunes the K and

"A dragon..." "as street graffiti" "playing with fire" "as a plushie" "working as a barista"

"A cat..." "in Ukiyo-e style" "with a rainbow scarf" "Van Gogh painting" "wearing a diploma hat"

Figure 2. **Personalized generations by DreamCache.** The first column contains reference images. The generated images correspond to the text prompts above each column.

V projections for the cross-attention blocks of the U-Net. Text-based personalization methods optimize single (like Textual Inversion [6]) or multiple (like in P+ [32]) input token embeddings. Later methods [1, 8, 30, 35] build on these, with innovations like Perfusion [30] using dynamic rank-1 updates to prevent overfitting while keeping encodings lightweight. However, all finetuning-based methods are computationally intensive, often requiring minutes of finetuning per reference subject at test time.

Finetuning-Free Personalization To reduce computational demands, recent research has shifted toward zero-shot personalization methods that eliminate subject-specific finetuning, typically employing image encoders to condition the generation process via the features extracted from the reference images [7, 15, 20, 33, 36]. Examples include BLIP-Diffusion [13], which pretrains a Q-Former to learn image features aligned with text, and IP-Adapter, which uses a frozen CLIP encoder to extract text-aligned visual features that modulate the cross-attention layers of the generative model. Other approaches, like Kosmos-G [19] and CAFE [38], connect large language models (LLMs) with diffusion models to condition generation on personalized concepts. SuTI [4] takes a different approach by training millions of subject-specific experts and subsequently training a model via apprenticeship learning, enabling effective zero-shot personalized generation at test time.

Alternatively, encoder-free methods such as JeDi [37] and BootPIG [21] use features from the generative model's backbone to guide the generation. JeDi creates a multi-view synthetic dataset and modifies spatial self-attention to jointly attend to images of the same concept in a batch. BootPIG retains a trainable copy of the original U-Net, adding reference self-attention layers to enable adaptation of the personalized model for reference features. While these methods remove the need for an additional encoder, they still require computationally expensive inference, as reference images must be processed in parallel during generation-—a cost that accumulates due to the iterative nature of the diffusion process.

In contrast, our method, DreamCache, caches a subset of reference features from the U-Net without text conditioning, eliminating the need for parallel inference and reducing memory overhead by avoiding separate model loading. This results in a test-time computational efficiency similar to encoder-based methods while offering the flexibility of encoder-free feature extraction and injection.

Recent works such as BootPIG [21] and Toffee-5M [39] emphasize the importance of synthetic data that explicitly decouples the subject from the background, reporting improved performance. Inspired by these approaches, we adopt a similar generative pipeline to create a synthetic dataset we use to train DreamCache.

Moreover, since our method is plug-and-play and keeps the U-Net frozen, we reduce the high cost of training faced by other approaches [15, 19, 21, 37–39]. A detailed overview of current methods, the number of trained parameters, and their training cost can be found in Table 1.

Feature Caching Feature caching has been explored to reduce generation time in diffusion models by caching intermediate activations. Some studies [17, 34] exploit temporal redundancy during the training process to cache ac-

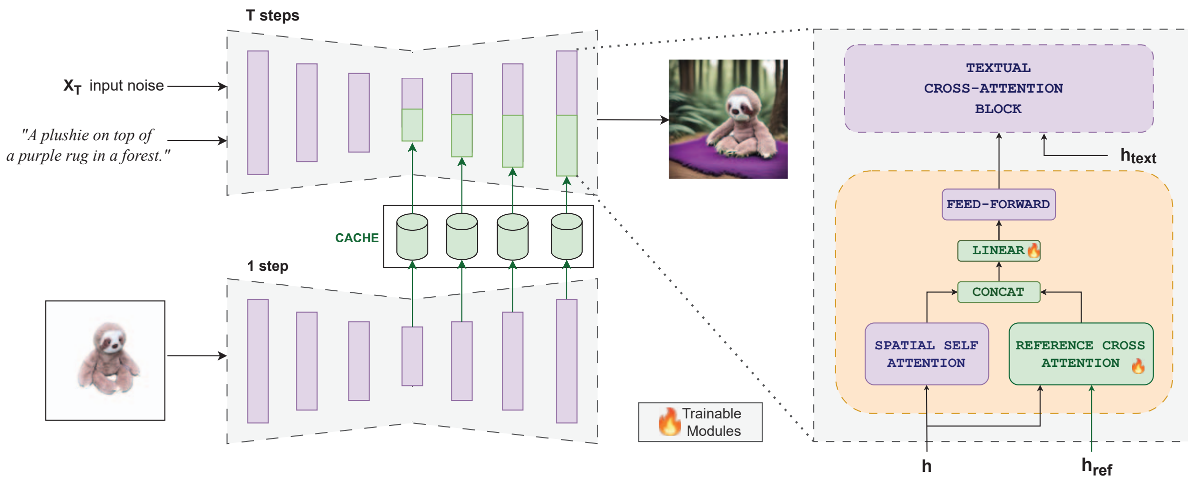

Figure 3. **Overview of DreamCache.** Original U-Net layers are shown in violet, while the novel components introduced by DreamCache are highlighted in green. During personalization, features from selected layers of the diffusion denoiser are cached from a single timestep, using a null text prompt. These cached features serve as reference-specific information. During generation, conditioning adapters inject the cached features into the denoiser, modulating the features of the generated image to create a personalized output.

tivations across timesteps, reducing computational load at later timesteps. Other works focus on caching layer activations within the diffusion framework, avoiding redundant computations. Learning-to-Cache [16] introduces a dynamic caching mechanism that learns to skip computation for selected layers of the diffusion model. In contrast to those works, which generally cache intra-model features for some layers to save computations, we utilize feature caching to encode multi-resolution features of a reference image from a few selected layers to condition the generation process of a new personalized image. Our approach recalls successful few-shot learners for discriminative problems [18, 27, 31] and extends them to personalized image generation.

3. Method

Given a pretrained text-to-image generative model ϵ_θ and an image containing a reference subject $\mathbf{I}_{\text{ref}}$, the goal of personalized sampling is to generate novel images containing the reference subject in various contexts while maintaining textual control. We propose DreamCache, a novel approach for extracting conditioning signals from $\mathbf{I}_{\text{ref}}$ and guiding of the image generation process. This method leverages a pretrained diffusion model, conditioning adapters that are pretrained with a synthetic dataset, and a feature cache from the reference image. Sample outputs generated by DreamCache are shown in Fig. 2, with a method overview in Fig. 3.

At the core of DreamCache, we utilize the denoiser in the pretrained diffusion model to extract multi-resolution features from $\mathbf{I}_{\text{ref}}$ by caching the activations of a few selected layers. To improve generalization, we cache features using a forward pass with a null text prompt. When personalized sampling is performed, the cached features are processed by adapters to act as conditioning signals, modulating the denoiser features of the image under generation at corresponding layers. These adapters, once pretrained on a synthetic dataset, enable zero-shot personalized generation with any new reference image, requiring no further finetuning once its features are cached.

In the following, we detail the three main aspects of DreamCache, namely i) how to cache reference features (Sec. 3.1); ii) how to condition the diffusion model on the cached features for personalized sampling (Sec. 3.2); iii) how to train the adapters used for model conditioning (Sec. 3.3).

3.1. Caching Reference Features

To extract information from the reference image for personalized sampling, we perform a forward pass through the denoiser of the diffusion model at a single timestep. We select $t = 1$, the least noisy timestep, to obtain clean features that are optimal for conditioning the personalized generation process. Additionally, we remove the text conditioning to decouple visual content of the reference image from the text caption, thus also eliminating the need for user-provided captions for reference images. This contrasts with methods such as JeDi [37], which are sensitive to caption content. During the forward pass, activations are computed for all layers of the denoiser, but only a subset is cached. Based on our experiments with the Stable Diffusion U-Net, we find that caching features from a middle bottleneck layer and every second layer in the decoder offers the best balance between generation quality and caching efficiency (see Sec. 4.3).

Formally, the feature cache $\mathcal{H}_{\text{FC}}$ consists of the activations of the denoiser ϵ_θ from the selected layers $\mathcal{L}$ at timestep $t = 1$, using a null text prompt $\varnothing$ and noisy reference image $\mathbf{I}_{\text{ref}} + \mathbf{n}_t$, with noise realization $\mathbf{n}_t$, expressed as:

$$\mathcal{H}_{\text{FC}} = \{\mathbf{h}_{\text{ref},L} : L \in \mathcal{L}\} \tag{1}$$

$$\mathbf{h}_{\text{ref},L} = \epsilon_\theta(\mathbf{I}_{\text{ref}} + \mathbf{n}_t, \varnothing, t; l)|_{t=1,l=L}. \tag{2}$$

Notice that the cached features have different spatial resolutions, from the low-resolution bottleneck layer to the higher-resolution decoder layers, allowing a multi-resolution representation of the reference image. This is particularly useful for enabling both global semantics and fine-grained detail guidance. As in prior work [15, 21, 33], we foreground-segment the reference image to isolate the subject from the background before caching its features.

3.2. Conditioning on Cached Reference Features

We propose a novel conditioning adapter mechanism composed of: i) a cross-attention block between the cached features and the features of the image under generation; ii) a concatenation operation between the features from the self-attention block of the original U-Net denoising backbone and those of the cross-attention block; and iii) a projection layer. A block diagram is shown in Fig. 3 (right).

Omitting the layer subscript for clarity, the conditioning adapter mechanism is expressed mathematically as follows:

$$\mathbf{q} = \mathbf{W}_Q\mathbf{h}, \quad \mathbf{k}_c = \mathbf{W}_K\mathbf{h}_{\text{ref}}, \quad \mathbf{v}_c = \mathbf{W}_V\mathbf{h}_{\text{ref}}, \tag{3}$$

$$\mathbf{a}_c = \text{softmax}\left(\frac{\mathbf{q}\mathbf{k}_c^T}{\sqrt{d}}\right)\mathbf{v}_c, \tag{4}$$

$$\mathbf{a} = \mathbf{W}_{\text{proj}}\left([\mathbf{a}; \mathbf{a}_c]\right), \tag{5}$$

where $\mathbf{h} \in \mathbb{R}^{N\times d}$ are the current d-dimensional features of the N-pixel image under generation, and $\mathbf{h}_{\text{ref}}$ are the cached reference features from Eq. (2). $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$, and $\mathbf{W}_{\text{proj}}$ are learnable projection matrices, whose training is described in Sec. 3.3. The concatenation $[\mathbf{a}; \mathbf{a}_c] \in \mathbb{R}^{N\times 2d}$ combines the output of self-attention $\mathbf{a} \in \mathbb{R}^{N\times d}$ and cross-attention $\mathbf{a}_c \in \mathbb{R}^{N\times d}$. The concatenation operation allows a flexible information fusion without explicit alignment constraints, compared to other approaches in similar works (see Sec. 4.3). The learnable projection matrix $\mathbf{W}_{\text{proj}}$ reduces the dimensionality of the concatenated features back to $\mathbb{R}^{N\times d}$ to maintain compatibility with the original backbone.

Overall, the approach proposed for the adapter enriches feature representations used in the diffusion process of the image under generation by allowing the model to leverage both primary and conditioning-based contextual information from the cache.

3.3. Training the Conditioning Adapters

The additional parameters introduced in Sec. 3.2 to process the cached features must be trained on a large and varied dataset to ensure they generalize for any reference subject.

Collecting paired data for this training process would be prohibitively expensive, as it requires multiple images of the same subject in different contexts. To address this, we draw inspiration from the recently proposed synthetic data generation pipeline in BootPIG [21] to construct our training data. First, we utilize a large language model (Llama 3.2 [5]) to generate captions for potential target images. Each caption is used to generate an image via Stable Diffusion [23]. We then use the Segment Anything Model (SAM) [11] and Grounding DINO [14] to accurately segment the reference subject based on the text caption and generate a foreground mask of the main object in the caption.

We treat the Stable Diffusion-generated image as the target image, the foreground object pasted on a white background as the reference image, and the LLM-generated caption as the textual prompt during our training pipeline. Compared to BootPIG, our pipeline employs open-source models, making it more accessible. We will release our synthetic dataset to facilitate reproducibility and further research, since similar datasets, including BootPIG's [21], have not been publicly released. Additional details on the dataset and its statistics can be found in the Supp. Mat.

We train the newly introduced adapters' parameters ($\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$, and $\mathbf{W}_{\text{proj}}$) with the standard score matching loss [29] using both the text-conditioned noisy input and the cached reference features:

$$\mathcal{L}_{\text{diffusion}} = \mathbb{E}_{\mathbf{x}_0,\epsilon,\mathbf{c}_T,\mathbf{I}_{\text{ref}},t}\left[||\epsilon - \epsilon'_\theta(\mathbf{x}_t, \mathbf{c}_T, \mathcal{H}_{\text{FC}}, t)||_2^2\right], \tag{6}$$

where $\mathbf{x}_0$ is the target image, $\mathbf{c}_T$ is the text prompt generated by the large language model, ϵ is Gaussian noise, and t is the diffusion timestep sampled uniformly from $1, \dots, T$. The noisy image at timestep t, $\mathbf{x}_t$, is obtained by gradually adding noise to $\mathbf{x}_0$ during the forward diffusion process. The function ϵ'_θ represents the modified denoising model that incorporates the conditioning adapters.

4. Experimental Results

In this section, we present our experimental results, including quantitative and qualitative comparisons, an ablation study, and an analysis section that visualizes the behavior of the newly introduced cross-attention mechanism in the adapters.

Implementation Details We evaluate our method on two versions of Stable Diffusion (SD) [23], specifically versions 1.5 and 2.1, to ensure fair comparison with state-of-the-art methods across different backbones. As described in the ablation study, our caching and conditioning mechanism is applied to the middle layer and every second layer of the

Table 2. **Quantitative results on DreamBooth.** DreamCache obtains a better balance between DINO score and CLIP-T compared to all baselines, while also offering a more efficient computational tradeoff (see Table 1).

	Method	Backbone	#Ref	DINO (↑)	CLIP-I (↑)	CLIP-T (↑)
test-time finetuning	DreamBooth [24]	Imagen	3-5	0.696	0.812	0.306
	DreamBooth [24]	SD 1.5	3-5	0.668	0.803	0.305
	Textual Inversion [6]	SD 1.5	3-5	0.569	0.780	0.255
	Custom Diffusion [12]	SD 1.5	3-5	0.643	0.790	0.305
	BLIP-Diffusion (FT) [13]	SD 1.5	3-5	0.670	0.805	0.302
finetuning free	ELITE [33]	SD 1.5	1	0.621	0.771	0.293
	BLIP-Diffusion [13]	SD 1.5	1	0.594	0.779	0.300
	IP-Adapter [36]	SD 1.5	1	0.667	0.813	0.289
	Kosmos-G [19]	SD 1.5	1	0.694	0.847	0.287
	Jedi [37]	SD 1.5	1	0.619	0.782	0.304
	DreamCache (ours)	SD 1.5	1	0.713	0.810	0.298
	Re-Imagen [3]	Imagen	1-3	0.600	0.740	0.270
	SuTI [4]	Imagen	1-3	0.741	0.819	0.304
	Subject-Diffusion [15]	SD 2.1	1	0.771	0.779	0.293
	BootPig [21]	SD 2.1	3	0.674	0.797	0.311
	ToffeeNet [39]	SD 2.1	1	0.728	0.817	0.306
	CAFE [38]	SD 2.1	1	0.715	0.827	0.294
	DreamCache (ours)	SD 2.1	1	0.767	0.816	0.301

decoder. The total number of trainable parameters for DreamCache is 25M. We use the original SD codebase and train the model on 4× 80GB A100 GPUs for 25k steps with a batch size of 128, using the AdamW optimizer with a learning rate of 10^{-5}. Input images are resized to 512×512, with scale, shift, and resize augmentations applied to reference images to enhance model robustness to perturbations. Ablations are conducted on SD 1.5. We generate images with 50 sampling steps, employing classifier-free guidance for image and text conditioning, using a guidance scale of 7.5.

Evaluation Quantitative evaluations are conducted on the DreamBooth dataset [24], following prior approaches. DreamBooth consists of 30 subjects, each with 25 text prompts. We use a single input image per subject and generate 4 images per subject-prompt combination, resulting in 3,000 generated images. We use pretrained DINO ViT-S/16 and CLIP ViT-B/32 models to calculate the average cosine similarity of global image embeddings between generated and reference images, with metrics denoted as DINO and CLIP-I, respectively. To assess text alignment, we calculate the cosine similarity between embeddings from generated images and text prompts using CLIP's image and text encoders [9], with the corresponding score denoted as CLIP-T.

4.1. Zero-Shot Personalization

We compare DreamCache against state-of-the-art methods for finetuning-based and zero-shot personalization. Table 2 presents quantitative results, indicating the diffusion backbone and the number of reference images for each method. Our approach achieves competitive or superior performance compared to other computationally-intensive state-of-the-art methods, which are trained on larger datasets and with significantly more parameters. We refer the reader to Table 1 for the data requirements, training time, and parameter count of the various methods. We remark that generally DINO is a preferred metric for image similarity with respect to CLIP-I, as it is more sensitive to the appearance and fine-grained details of the subjects.

We also present qualitative comparisons with Kosmos-G [19] and BLIP-Diffusion [13]. We remark that several other methods are not reproducible due to the lack of code, datasets, or trained checkpoints. As seen in Fig. 4, our method excels in subject preservation and textual alignment, producing visually superior results. We also notice that Kosmos-G reports a high CLIP-I score, but after inspecting the generated images, it is clear that the score does not entirely reflect the preservation of the reference subject in generated images. In fact, Kosmos-G presents a high degree of background interference, where the partial replication of the reference background boosts the alignment score. For this reason, we also report foreground-masked metrics on the subjects, like MCLIP-I and MDINO, in the Supp. Mat.

4.2. Inference Time Evaluation

The computational efficiency of our method is compared to the reference-based method BootPIG [21] and encoder-based approaches such as Kosmos-G [19] and Subject-

Figure 4. **Visual comparison.** Personalized generations on sample concepts. DreamCache preserves reference concept appearance and does not suffer from background interference. BLIP-D [13] and Kosmos-G [19] cannot faithfully preserve visual details from the reference.

Table 3. **Computational comparison.** *: time to generate an image with 100 timesteps, evaluated on a single NVIDIA A100 GPU.

Method	Inference Time*	Extra Params Size
ELITE [33]	6.24 s	914 MB
BLIP-Diffusion [13]	3.92 s	760 MB
BootPig [21]	7.55 s	1900 MB
DreamCache (ours)	3.88 s	42 MB

Table 4. **Reference feature integration.** DreamCache uses the best tradeoff between accuracy and complexity.

Method	CLIP-I (↑)	CLIP-T (↑)	Params
Textual Sum [36]	0.788	0,282	19M
Spatial Sum	0.812	0.293	16M
Decoupled Blocks [8]	0.808	0.300	61M
Spatial Concat (ours)	0.810	0.298	25M

Diffusion [15]. Table 3 provides a detailed comparison of inference time, accounting for both personalization time (*e.g.*, the time to generate the cache for DreamCache) and the time to sample the personalized image. We also report the increase in model size, *i.e.*, the storage (in FP16 precision) required for the extra parameters to allow for personalization, showing that DreamCache is one order of magnitude more compact than the state of the art. Overall, DreamCache offers a lightweight solution that achieves state-of-the-art performance with faster inference and reduced computational overhead.

4.3. Ablation Studies

We validate our design choices through a series of studies, examining different conditioning mechanisms, evaluating our feature caching approach, and analyzing the impact of synthetic dataset scaling.

Reference Feature Integration We compare various conditioning strategies to integrate reference features in Table 4. Our spatial cross-attention block with concatenation between the output of self- and cross-attention ("Spatial Concat") was evaluated against different alternatives, including IP-Adapter's conditioning mechanism [36] ("Textual Sum"), which sums the decoupled cross-attention output with that from textual cross-attention. We also tested a variant ("Spatial Sum") where self- and cross-attention conditioning outputs are summed. Additionally, we also assessed an alternative conditioning procedure inspired by ViCo [8] ("Decoupled Blocks"), involving independent and interleaved cross-attention blocks. Results in Table 4 indicate that the proposed "Spatial Concat" offers the best balance of text alignment and parameter efficiency.

We further explored optimal conditioning insertion within the U-Net backbone in Table 5, determining that applying

Table 5. **Cache positioning** in the U-Net backbone offers a further tradeoff between accuracy and complexity.

Encoder	Middle	Decoder	CLIP-I (↑)	CLIP-T (↑)	Params
✓	✗	✗	0.721	0.303	11M
✓	✓	✗	0.749	0.306	19M
✗	✓	✗	0.716	0.302	8M
✗	✗	✓	0.799	0.296	17M
✗	✓	✓	0.810	0.298	25M
✓	✓	✓	0.813	0.297	36M

Table 6. **Caching with text** is not influential and adds complexity.

Text-Free	CLIP-I (↑)	CLIP-T (↑)
✗	0.811	0.295
✓	0.810	0.298

Table 7. **Dataset impact** for both synthetic and real data.

Dataset	CLIP-I (↑)	CLIP-T (↑)
Synthetic-50K	0.781	0.304
Synthetic-200K	0.797	0,301
Synthetic-400K	0.810	0.298
LAION-5M	0.814	0.242

conditioning (and therefore feature caching) at the middle layer and every second layer of the decoder achieved the best tradeoff between performance and parameter count.

Text Input for Cached Features Our feature caching procedure is designed to be text-free, leveraging the classifier-free guidance used during pretraining where captions were occasionally omitted. We compare this approach with a version including textual inputs during caching (*e.g.*, '*A photo of* ..."). Table 6 shows that adding text conditioning slightly reduces text alignment while increasing complexity and potentially introducing noise in cases of inaccurate captions.

Dataset Impact We demonstrate the importance of our synthetic dataset to train the conditioning adapters and the effect of scaling its size. For this purpose, we created synthetic datasets according to the procedure in Sec. 3.3 of sizes 50K, 200K, and 400K samples. We also tested the real-world 5M samples from the LAION [26] dataset, which, lacking target-caption-reference triplets, required reuse of target images as reference images too. Table 7 shows that increasing dataset size improves image alignment, though slightly reduces textual alignment. Notably, LAION improves image alignment but struggles with textual alignment. This highlights the importance of triplet data (target image, reference image, and caption) for effective zero-shot personalization, ensuring both subject preservation and textual editability.

Figure 5. **Visualization of reference image impact.** Cross-attention maps between cached reference features and features of the image under generation. *Left:* attention map at layers at 16×16 resolution (left reference, right generated). *Right:* 32×32. Attention values are highly localized in the region of interest.

4.4. Visualizing Reference Impact

Finally, we analyze how the cross-attention mechanism in DreamCache impacts image generation by visualizing cached reference feature influence. Attention map visualizations at different resolutions are provided in Fig. 5. Specifically, attention maps between the query from the current generation and the key derived from reference features reveal a highly localized focus on the subject, without interference from background elements. This mechanism models correspondences effectively, integrating reference information into the generated image.

5. Discussion and Conclusions

In this paper, we proposed DreamCache, a novel approach to personalized text-to-image generation that uses feature caching to overcome the limitations of existing methods. By caching reference features from a small subset of layers of the U-Net only once, our method significantly reduces both computational and memory demands, enabling efficient, real-time personalized image generation. Unlike previous approaches, DreamCache avoids the need for costly finetuning, external image encoders, or parallel reference processing, making it lightweight and suitable for plug-and-play deployment. Our experiments demonstrate that DreamCache achieves state-of-the-art zero-shot personalization with only 25M additional parameters and a fast training process. While DreamCache is a promising direction towards more efficient personalized generation, it has some limitations. Although effective for single-subject personalization, our approach may require adaptation for complex multi-subject generation where feature interference can occur. Additionally, certain edge cases, such as highly abstract or stylistic images, may challenge the caching mechanism's capacity to accurately preserve subject details. To address these challenges, future work may explore adaptive caching techniques or multi-reference feature integration.

References

[1] Moab Arar, Rinon Gal, Yuval Atzmon, Gal Chechik, Daniel Cohen-Or, Ariel Shamir, and Amit H. Bermano. Domain-agnostic tuning-encoder for fast personalization of text-to-image models. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–10, 2023. 1, 3

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 1

[3] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W Cohen. Re-imagen: Retrieval-augmented text-to-image generator. In *The Eleventh International Conference on Learning Representations*, 2022. 6

[4] Wenhu Chen, Hexiang Hu, Yandong Li, Nataniel Rui, Xuhui Jia, Ming-Wei Chang, and William W Cohen. Subject-driven text-to-image generation via apprenticeship learning. *arXiv preprint arXiv:2304.00186*, 2023. 2, 3, 6

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 5

[6] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 3, 6

[7] Rinon Gal, Moab Arar, Yuval Atzmon, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Encoder-based domain tuning for fast personalization of text-to-image models. *ACM Transactions on Graphics (TOG)*, 42(4):1–13, 2023. 1, 3

[8] Shaozhe Hao, Kai Han, Shihao Zhao, and Kwan-Yee K Wong. Vico: Plug-and-play visual condition for personalized text-to-image generation. *arXiv preprint arXiv:2306.00971*, 2023. 1, 3, 7

[9] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7514–7528, 2021. 6

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 5

[12] 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. 1, 2, 6

[13] Dongxu Li, Junnan Li, and Steven CH Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *arXiv preprint arXiv:2305.14720*, 2023. 1, 2, 3, 6, 7

[14] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 5

[15] Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning. *arXiv preprint arXiv:2307.11410*, 2023. 1, 2, 3, 5, 6, 7

[16] Xinyin Ma, Gongfan Fang, Michael Bi Mi, and Xinchao Wang. Learning-to-cache: Accelerating diffusion transformer via layer caching. *arXiv preprint arXiv:2406.01733*, 2024. 4

[17] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15762–15772, 2024. 3

[18] Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. *Advances in neural information processing systems*, 31, 2018. 4

[19] Xichen Pan, Li Dong, Shaohan Huang, Zhiliang Peng, Wenhu Chen, and Furu Wei. Kosmos-g: Generating images in context with multimodal large language models. *arXiv preprint arXiv:2310.02992*, 2023. 1, 2, 3, 6, 7

[20] Maitreya Patel, Sangmin Jung, Chitta Baral, and Yezhou Yang. λ-eclipse: Multi-concept personalized text-to-image diffusion models by leveraging clip latent space. *arXiv preprint arXiv:2402.05195*, 2024. 1, 3

[21] Senthil Purushwalkam, Akash Gokul, Shafiq Joty, and Nikhil Naik. Bootpig: Bootstrapping zero-shot personalized image generation capabilities in pretrained diffusion models. *arXiv preprint arXiv:2401.13974*, 2024. 1, 2, 3, 5, 6, 7

[22] 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. 1

[23] 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. 1, 5

[24] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. 1, 2, 6

[25] 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. 1

[26] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 8

[27] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 4

[28] 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. 1

[29] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 5

[30] Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-locked rank one editing for text-to-image personalization. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023. 1, 3

[31] Eleni Triantafillou, Hugo Larochelle, Jake Snell, Josh Tenenbaum, Kevin Jordan Swersky, Mengye Ren, Richard Zemel, and Sachin Ravi. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018. 4

[32] 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. 1, 3

[33] 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. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15897–15907, 2023. 1, 2, 3, 5, 6, 7

[34] Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, et al. Cache me if you can: Accelerating diffusion models through block caching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6211–6220, 2024. 3

[35] Jianan Yang, Haobo Wang, Yanming Zhang, Ruixuan Xiao, Sai Wu, Gang Chen, and Junbo Zhao. Controllable textual inversion for personalized text-to-image generation. *arXiv preprint arXiv:2304.05265*, 2023. 1, 3

[36] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 1, 2, 3, 6, 7

[37] Yu Zeng, Vishal M Patel, Haochen Wang, Xun Huang, Ting-Chun Wang, Ming-Yu Liu, and Yogesh Balaji. Jedi: Joint-image diffusion models for finetuning-free personalized text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6786–6795, 2024. 1, 2, 3, 4, 6

[38] Yufan Zhou, Ruiyi Zhang, Jiuxiang Gu, and Tong Sun. Customization assistant for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9182–9191, 2024. 2, 3, 6

[39] Yufan Zhou, Ruiyi Zhang, Kaizhi Zheng, Nanxuan Zhao, Jiuxiang Gu, Zichao Wang, Xin Eric Wang, and Tong Sun. Toffee: Efficient million-scale dataset construction for subject-driven text-to-image generation. *arXiv preprint arXiv:2406.09305*, 2024. 1, 2, 3, 6