Sparse Fine-Tuning of Transformers for Generative Tasks

Wei Chen, Jingxi Yu, Zichen Miao, Qiang Qiu
Purdue University, IN, USA
{chen2732, yu667, miaoz, qqiu}@purdue.edu

Abstract

Large pre-trained transformers have revolutionized artificial intelligence across various domains, and fine-tuning remains the dominant approach for adapting these models to downstream tasks due to the cost of training from scratch. However, in existing fine-tuning methods, the updated representations are formed as a dense combination of modified parameters, making it challenging to interpret their contributions and understand how the model adapts to new tasks. In this work, we introduce a fine-tuning framework inspired by sparse coding, where fine-tuned features are represented as a sparse combination of basic elements, i.e., feature dictionary atoms. The feature dictionary atoms function as fundamental building blocks of the representation, and tuning atoms allows for seamless adaptation to downstream tasks. Sparse coefficients then serve as indicators of atom importance, identifying the contribution of each atom to the updated representation. Leveraging the atom selection capability of sparse coefficients, we first demonstrate that our method enhances image editing performance by improving text alignment through the removal of unimportant feature dictionary atoms. Additionally, we validate the effectiveness of our approach in the text-to-image concept customization task, where our method efficiently constructs the target concept using a sparse combination of feature dictionary atoms, outperforming various baseline fine-tuning methods.

1. Introduction

In recent years, pre-trained large models have revolutionized artificial intelligence across domains like natural language processing and computer vision [11, 21, 43, 48]. Given the prohibitive computational resources required for training these models from scratch, fine-tuning has emerged as the dominant paradigm for adapting pre-trained models to specific downstream tasks, enabling efficient knowledge transfer while minimizing computational overhead.

Various parameter-efficient fine-tuning (PEFT) methods [16, 24, 40] have been developed to tune the large transformers by updating a small subset of parameters while keeping large pre-trained weights frozen. A representative family of these approaches [16, 24] utilizes lightweight parameterizations, such as low-rank decomposition, to adapt the weight matrices. However, despite being lightweight, these methods obtain the updated features with a dense matrix multiplication between the input and the adapted weights, making it challenging to interpret how the individual updated parameters contribute to the newly learned representations and to identify the new knowledge gained through fine-tuning.

To tackle this challenge, we take inspiration from sparse coding [9, 27, 29, 30, 34, 36, 39, 49–51], in which signals are expressed as a sparse combination of basic elements, *i.e.*, *dictionary atoms*. In this paper, we frame model fine-tuning as a sparse coding problem, where the adapted feature is constructed using a dictionary of feature atoms. These feature atoms are selectively combined with *sparse coefficients*, ensuring that only a small subset of atoms contributes to the updated representation. This sparsity not only disentangles interactions between features but also allows each atom to capture a distinct aspect of the downstream data.

We first analyze with a toy experiment that the tuned feature dictionary atoms function as building blocks of the adapted representation, and tuning these atoms allows for effective adaptation to downstream tasks. We demonstrate that with pre-trained coefficients and atoms, fine-tuning the atoms effectively captures the representation of downstream tasks. Representing features through individual atoms enhances controllability in the adaptation process.

We validate our method on generative fine-tuning tasks. We show that the target concept can be effectively represented using feature dictionary atoms after fine-tuning the pre-trained model within our framework. Compared to baseline methods, our approach allows the selection of specific atoms to refine the generated image. This adaptability improves performance on image editing tasks by enhancing editing results through the removal of unimportant atoms. We also apply our approach to customized text-to-image tasks, showing that fine-tuning atoms effectively captures

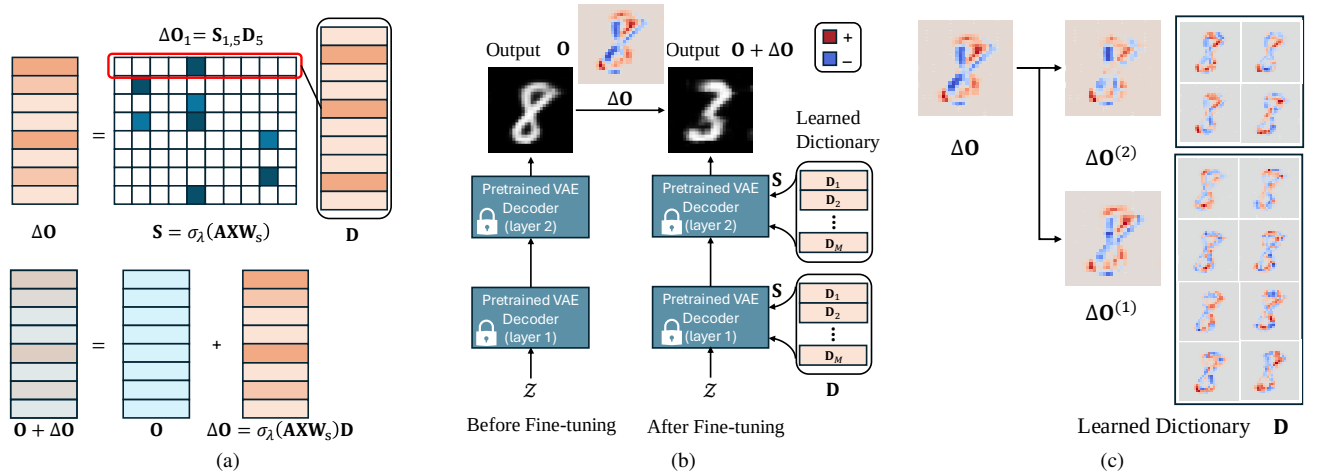

Figure 1. (a) Our method leverages sparse coding and enforces high sparsity in adapted feature $\Delta\mathbf{O}$, ensuring that the output feature is constructed using only a few dictionary atoms $\mathbf{D}$. (b) The pre-trained model initially generates "8". It generates "3" after integrating the learned dictionary $\mathbf{D}$ which is fine-tuned on "3". (c) The overall transformation of the generated digit from "8" to "3" using the dictionary atoms. The red indicates an increase in value at a specific location and blue indicates a decrease. The total influence is decomposed into contributions from atoms at different layers: $\Delta\mathbf{O}^{(1)}$ represents the combined effect of 8 atoms in the first layer, while $\Delta\mathbf{O}^{(2)}$ accounts for the 4 atoms in the second layer. The details of this experiment is provided in Section 3.3

personal concepts while preserving the capabilities of pre-trained models. Our method outperforms various baseline approaches, demonstrating its effectiveness in controlled adaptation for generative modeling.

We summarize our contributions as follows:

- We formulate large transformer fine-tuning as a sparse coding problem, where the adapted feature representation is a sparse combination of a dictionary of feature atoms.
- We introduce a fine-tuning framework that leverages a combination of dictionary atoms and demonstrates that tuning atoms enhances control over the adaptation process.
- We validate the effectiveness of our approach in generative tasks by demonstrating the improvement in image editing and concept customization.

2. Related Work

Sparse Representation and Attention Recent research has demonstrated the significant role of sparsity in enhancing both interpretability and computational efficiency in generative models through various approaches including sparse autoencoders for feature extraction [45], sparse feature circuits for improved generalization [28], transcoders for detailed circuit analysis [12], dictionary learning for monosemantic representations [5], and selective parameter tuning strategies [17]. Complementary work has addressed the quadratic computational complexity of standard attention mechanisms through sparse attention techniques that compute attention scores for only a subset of token pairs [3, 10, 46, 48], with optimizations including dynamic pattern adjustment [23], hardware-aligned implementations [55], and graph-based conceptualizations [47], collectively enabling transformers to process significantly longer sequences while maintaining model quality despite challenges in balancing efficiency with performance in complex long-range dependency scenarios. Compared to previous works, our method applies sparse coding for fine-tuning transformer-based models in generative tasks.

Representation Tuning Recent representation tuning approaches operate directly on model representations rather than weights, achieving parameter efficiency while enhancing controllability, and interpretability across various model types and tasks [1, 8, 25, 33, 52, 53]. Unlike traditional PEFT approaches that impact representations indirectly through weight adjustments, representation tuning methods preserve the original pre-trained model weights while directly modifying the output representations. In comparison, our work studies the sparsity in the updated representation and frames it as a dictionary learning problem.

Personalized Text-to-Image Generation Extensive research has been conducted on aligning text prompts and generated concepts [18, 19, 22, 31, 32, 38, 44]. Zhang et al. [57] developed an object-conditioned energy-based attention map alignment technique in diffusion models, enhancing control over specific objects in generated images. Another study by Zhang et al. [56] improved semantic fidelity in text-to-image synthesis through attention regulation. Hao et al. [15] introduces ConceptExpress, an unsupervised method for extracting personalized concepts from a single image. Ren et al. [42] addressed memorization issues in text-to-image diffusion models by analyzing cross-

attention mechanisms, proposing solutions to reduce overfitting to user-specific data. We approach this problem by adopting the ideas from sparse coding and learning the target concept as a sparse combination of coefficients and dictionary atoms.

3. Method

3.1. Preliminary

Multi-head attention (MHA). The attention layer [48] transforms the input sequence $\mathbf{X} \in \mathbb{R}^{N\times C_i}$ to the output sequence $\mathbf{O} \in \mathbb{R}^{N\times C_o}$, where N denotes the sequence length, C_i and C_o are the dimension of input and output features. The attention layer projects the input as $\mathbf{Q} = \mathbf{X}\mathbf{W}_q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_k$, $\mathbf{V} = \mathbf{X}\mathbf{W}_v$ using the corresponding projection matrices $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{C_i\times C_o}$, and calculates the attention map,

$$\mathbf{A} = \text{attn}(\mathbf{X}) = \text{softmax}(\mathbf{X}\mathbf{W}_q\mathbf{W}_k^\intercal\mathbf{X}^\intercal). \tag{1}$$

The attention map $\mathbf{A} \in \mathbb{R}^{N\times N}$ captures the token-wise relationship by doing inner-product in a space transformed by $\mathbf{W}_q, \mathbf{W}_k$. Then the output of the attention layer is,

$$\mathbf{O} = \mathbf{A}\mathbf{X}\mathbf{W}_v. \tag{2}$$

Multi-head attention extends this by allowing multiple attention mechanisms to work in parallel, with each head independently learning attention patterns. The project matrices for each head are $\mathbf{W}_q^{(h)}, \mathbf{W}_k^{(h)}, \mathbf{W}_v^{(h)} \in \mathbb{R}^{C_i\times\frac{C_o}{H}}$, where $h = 1, \cdots, H$ and H is the number of heads. The multi-head attention represents as,

$$\mathbf{Z} = \sum_{h=1}^{H} \mathbf{A}^{(h)}\mathbf{X}\mathbf{W}_{vo}^{(h)}, \tag{3}$$

where $\mathbf{W}_o \in \mathbb{R}^{C_o\times C_o}$ and $\mathbf{W}_{vo}^{(h)} = \mathbf{W}_v^{(h)}\mathbf{W}_o^{(h)\intercal}$, $\mathbf{W}_o^{(h)} \in \mathbb{R}^{C_o\times\frac{C_o}{H}}$, $\mathbf{W}_{vo}^{(h)} \in \mathbb{R}^{C_i\times C_o}$.

Sparse coding. Sparse coding is a signal representation technique that models a signal as a linear combination of a small number of basis functions from a dictionary. Given a signal $\mathbf{X} \in \mathbb{R}^{N\times C_i}$, the goal of sparse coding is to find a dictionary $\mathbf{D} \in \mathbb{R}^{m\times C_i}$ (where $m > C_i$ is the number of atoms) and a sparse coefficient vector $\mathbf{S} \in \mathbb{R}^{N\times m}$ such that the signal can be approximated as:

$$\mathbf{X} \approx \mathbf{S}\mathbf{D}, \tag{4}$$

where $\mathbf{S}$ has only a few nonzero elements, ensuring sparsity. The sparse coding problem is often formulated as an optimization problem:

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{S}\mathbf{D}\|_2^2 + \lambda\|\mathbf{S}\|_1, \tag{5}$$

where λ is a regularization parameter controlling the trade-off between sparsity and reconstruction error. The Proximal gradient method is a common approach to solving this optimization problem, including Iterative Shrinkage-Thresholding Algorithm (ISTA) and its accelerated version (FISTA) [2], which utilize proximal operators for efficient iterative updates. For sparse coding with ℓ_1 regularization, the corresponding proximal operator is the soft-thresholding function:

$$\text{prox}_{\lambda\|\cdot\|_1}(\mathbf{S}) = \text{sign}(\mathbf{S}) \odot \max(|\mathbf{S}| - \lambda, 0), \tag{6}$$

where $\odot$ is Hadamard product. It encourages sparsity by shrinking small values of $\mathbf{S}$ toward zero. Sparse coding is widely used in signal processing, machine learning, and neuroscience, providing efficient and interpretable representations of data.

3.2. Sparse Fine-Tuning

In this work, we conceptualize model fine-tuning as a sparse coding problem, i.e., the adapted features are constructed using a dictionary of feature atoms. As our method focuses on the transformer architecture, we first propose an interpretation of the attention operation by viewing the product between the attention map and value matrices as a combination of dictionary atoms with dictionary coefficients. We then propose a fine-tuning approach by representing the adapted features as a learned sparse combination of dictionary atoms with sparse coefficients, representing the fine-tuned representations' underlying sparsity nature.

A dictionary view of attention. For single head attention (2), each row of $\mathbf{A}$ can be interpreted as coefficients for combining $\mathbf{V}$ into the output $\mathbf{O}$, *i.e.*, $\mathbf{O}_i = \sum_j \mathbf{A}_{ij}\mathbf{V}_j$. In the case of multi-head attention (3),

$$\sum_{h=1}^{H} \mathbf{A}^{(h)}\mathbf{X}\mathbf{W}_{vo}^{(h)} = \begin{bmatrix}\mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(H)}\end{bmatrix} \begin{bmatrix}\mathbf{X}\mathbf{W}_{vo}^{(1)} \\ \vdots \\ \mathbf{X}\mathbf{W}_{vo}^{(H)}\end{bmatrix}, \tag{7}$$

it can be represented as a product of a composite coefficient $\left[\mathbf{A}^{(1)}, \cdots, \mathbf{A}^{(H)}\right] \in \mathbb{R}^{N\times NH}$, and an extensive dictionary $\left[\mathbf{X}\mathbf{W}_{vo}^{(1)}, \cdots, \mathbf{X}\mathbf{W}_{vo}^{(H)}\right]^\intercal \in \mathbb{R}^{NH\times C_o}$. Each row of the output $\mathbf{O}$ is a linear combination of $\mathbf{X}\mathbf{W}_{vo}$, *i.e.*, $\mathbf{O}_i = \sum_{h,j} \mathbf{A}_{ij}^{(h)}(\mathbf{X}\mathbf{W}_{vo}^{(h)})_j$. When represented as a linear combination of dictionary elements, multi-head attention and single-head attention follow the same formulation. In the following discussion, we focus on the single-head attention formulation, which can be seamlessly extended to the multi-head attention case.

A sparse dictionary view of attention. From the perspective of sparse coding, the feature is expressed as $\mathbf{SD}$, where $\mathbf{S}$ represents the *sparse* coefficients and $\mathbf{D}$ consists of dictionary atoms that remain *data-independent* once learned. Although attention can be expressed as a linear combination of atoms, it does not inherently exhibit sparsity. To fully leverage the advantages of sparse representations, we propose to represent the attention mechanism (2) in the form of sparse coding,

$$\mathbf{O} = \mathbf{SD} = \sigma_\lambda(\mathbf{AXW}_s)\mathbf{D}, \tag{8}$$

where $\sigma_\lambda(\cdot)$ can be soft-thresholding function or $\text{ReLU}(x-\lambda)$ to ensure sparsity of $\mathbf{S} = \sigma_\lambda(\mathbf{AXW}_s) \in \mathbb{R}^{N\times C_o}$, $\mathbf{W}_s \in \mathbb{R}^{C_i\times M}$ and $\mathbf{D} \in \mathbb{R}^{M\times C_o}$ is the feature dictionary atoms, M is the number of atoms.

With this formulation, we obtain the feature dictionary atoms $\mathbf{D}$ (we use the term "atoms" for simplification in this paper), which are the building blocks for feature representation $\mathbf{O}$. We illustrate our formulation in Figure 1 (a). This formulation provides the following properties:

- The sparse coefficients $\mathbf{S} = \sigma_\lambda(\mathbf{AXW}_s)$ ensure that the feature representation $\mathbf{O}$ is combined by only a few $\mathbf{D}$. We present an extreme example to illustrate how atoms enhance interpretability: $\mathbf{O}_i = \sum_{j=1}^{M} \mathbf{S}_{ij}\mathbf{D}_j$ can be simplified as $\mathbf{O}_i = \mathbf{S}_{ij^*}\mathbf{D}_{j^*}$ if all elements of $\mathbf{S}_i$ is 0 except the element at index j^*, and only $\mathbf{D}_{j^*}$ contributes to the output feature $\mathbf{O}_i$.
- The sparse coefficients $\mathbf{S}$ are data-dependent, while the feature dictionary atoms $\mathbf{D}$ are data-independent. While different inputs generate different coefficients, the output features are always constructed using the same feature dictionary atoms.
- We can adjust the size of the feature dictionary by varying M to handle tasks with different difficulties. Large M leads to an overcomplete dictionary while small M produces an undercomplete dictionary.

Fine-tuning via sparse dictionary learning. The above formulation frames feature representation through the lens of sparse coding. In the context of large model fine-tuning, we freeze the pre-trained parameters to make feature representations $\mathbf{O}$ of the pre-trained model fixed. Instead, we introduce a sparse coding perspective to encode the adapted feature representation $\Delta\mathbf{O}$, which is learned from downstream tasks. Specifically, we have,

$$\mathbf{O} + \Delta\mathbf{O} = \mathbf{O} + \sigma_\lambda(\mathbf{AXW}_s)\mathbf{D}. \tag{9}$$

With this formulation, the pre-trained weights remain unchanged, preserving the capabilities of the pre-trained model. The adapted feature $\Delta\mathbf{O}$ is represented as a linear combination of feature dictionary atoms $\mathbf{D}$, using sparse coefficients $\mathbf{S} = \sigma_\lambda(\mathbf{AXW}_s)$. This formulation offers a framework for understanding how $\mathbf{D}$ contributes to the adapted features in the fine-tuned model. We illustrate our method in Figure 1 (a).

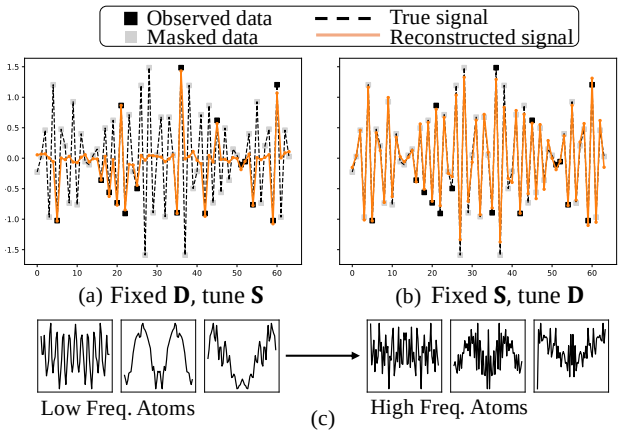

Figure 2. (a) Adjusting only the coefficients fails to adapt the model to the downstream task, whereas (b) modifying only the atoms enables a perfect fit to the new task composed of high-frequency signals. (c) Fine-tuning the atoms effectively transitions the model from low-frequency to high-frequency representations. They encode the representation of the downstream task.

3.3. Analysis of Dictionary Atoms

In this section, we demonstrate distinct properties of atoms $\mathbf{D}$ and sparse coefficients $\mathbf{S}$.

Features atoms are key to fine-tuning. We analyze the distinct behaviors of coefficients and atoms, and discover that atoms play a crucial role in adapting to new tasks. To explore this within the context of an attention block, we perform a toy experiment where masked one-dimensional signals are reconstructed using an attention layer. The signals, with a length of 64, are synthesized by selecting 5 random Fourier bases, with a low frequency ranging from $\frac{0}{64}$ to $\frac{24}{64}$. The objective is to reconstruct the signal after masking out 75% of its values with only 16 randomly sampled observations. After we have a pre-trained model on the low-frequency signals, we fine-tune the model to a new task where signals are constructed with high-frequency Fourier bases ranging from $\frac{25}{64}$ to $\frac{32}{64}$. In this experiment, we test two different cases: (1) tuning only coefficients on the high-frequency data with atoms fixed, and (2) tuning only atoms on the high-frequency data with coefficients fixed. As illustrated in Figure 2, when adapting to high-frequency signals, modifying only the atoms while keeping the coefficients fixed allows the model to effectively generalize to the new task. In contrast, updating only the coefficients is insufficient for proper adaptation. Building on this observation, we hypothesize that adjusting atoms is more suitable than adjusting coefficients for adapting to a new task. We ob-

serve from Figure 2 (c) that atoms are capable of encoding the representation of the downstream task. The experiment details are in Appendix B.

Identifying the importance of feature atoms. We analyze the impact of atoms on the newly learned concept by visualizing both the influence of individual atoms and their combined effect. In this experiment, we adopt a transformer-based variational autoencoder (VAE) [20] with a two-layer decoder. The experiment process is illustrated in Figure 1 (b). We first pre-train the model on an image-denoising task using a subset of the MNIST dataset containing digits 5 to 9. We then fine-tune the model on a downstream denoising task with only digit 3 by learning a dictionary for each layer with $M = 100$. In this scenario, the digit "3" is the newly learned concept. Given the same noisy latent, the decoder outputs an "8" without incorporating fine-tuned feature atoms, whereas it generates a "3" when the fine-tuned feature atoms are applied. The model requires only 12 atoms to shift the generated digits from 8 to 3. The influence of each atom is visualized in Figure 1 (c), with the overall impact represented as $\Delta\mathbf{O}$, where the red area indicates an increase in pixel value at a specific location and the blue area represents a decrease. It is straightforward to see that these 12 feature atoms enhance the regions corresponding to "3" while suppressing certain portions of "8". Additionally, Figure 1 (c) illustrates the individual behavior of each atom, showing how different atoms affect distinct parts of the generated digit. The experiment details are in Appendix B.

3.4. Design Variations

In this section, we outline the various design choices implemented in our experiments. Our method achieves sparse coefficient $\mathbf{S} = \sigma_\lambda(\mathbf{A}\mathbf{X}\mathbf{W}_s)$ by utilizing non-linear function $\sigma_\lambda(\cdot)$. In the literature on sparse coding [4], there are three different methods: (i) soft-thresholding function, $\sigma_\lambda(\mathbf{x}) = \text{prox}_{\lambda\|\cdot\|_1}(\mathbf{x}) = \text{sign}(\mathbf{x}) \odot \max(|\mathbf{x}| - \lambda, 0)$, (ii) ReLU, $\sigma_\lambda(\mathbf{x}) = \text{ReLU}(\mathbf{x} - \lambda) = \max(\mathbf{x} - \lambda, 0)$, and (iii) top-$k$ activation selection, which selects the most significant features of top k contributing to the learned representation. In practice, we observe that top-k activation selection maintains a consistent level of sparsity across different inputs. In contrast, methods (i) soft-thresholding function and (ii) ReLU exhibit sensitivity to input values, leading to variations in sparsity. However, methods (i) and (ii) offer control over the coefficient values, ensuring they stay above a specified threshold.

Furthermore, we find that when $\sigma_\lambda(\mathbf{X}\mathbf{W}_s)$ is highly sparse, the resulting product $\mathbf{A}\sigma_\lambda(\mathbf{X}\mathbf{W}_s)$ also retains sparsity. In modern transformers, computation of $\mathbf{A}$ and $\mathbf{X}\mathbf{W}_v$ are performed separately. For simplicity in implementation, we adopt the formulation $\mathbf{A}\sigma_\lambda(\mathbf{X}\mathbf{W}_s)$.

3.5. Analysis of Our Design

Comparison with LoRA. When representing attention mechanism (1-3) as a linear combination of dictionary atoms, updating $\mathbf{W}_q, \mathbf{W}_k$ is equivalent to updating the coefficients while updating $\mathbf{W}_v, \mathbf{W}_o$ is equivalent to updating the atoms. LoRA applies adaptation on $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ with a low-rank matrices $\mathbf{W}_A \in \mathbb{R}^{C_i \times r}, \mathbf{W}_B \in \mathbb{R}^{r \times C_o}$, where r represents the rank. LoRA adapts both coefficients and atoms in a low-rank way. In comparison, our method adapts the coefficients with $\mathbf{W}_s$ and obtains sparse coefficients $\sigma_\lambda(\mathbf{A}\mathbf{X}\mathbf{W}_s)$, while representing atoms with $\mathbf{D}$.

Parameters and computational cost. To adapt the representation at each layer, we need additional parameters for $\mathbf{W}_s$ and $\mathbf{D}$, which are $(C_i + C_o)M$. After the fine-tuning, considering the density of the coefficients ρ, i.e., $1 - \rho$ is sparsity, the storage required for the parameters is $C_iM + \rho C_oM$. In comparison, LoRA requires $4(C_i + C_o)r$ additional parameters. The FLOPs required by our method are $2NMC_i + 2\rho NMC_o$. LoRA requires $8C_iC_or$ FLOPs. For example, if $C_i = C_o = 1024$, $M = 256$, $r = 16$, $N = 1024$, $\rho = 0.01$, our method requires 264k parameters for storage and 0.54GFLOPs for compututation, while LoRA requires 131k parameters and 0.13GFLops. Our method involves slightly higher computational costs than LoRA, as it accounts for feature interactions, whereas LoRA only updates the weights. Next, we demonstrate that by leveraging sparse feature representations, our approach achieves greater robustness compared to LoRA.

Our method provides more stable adaptation. We compare an extreme case between LoRA and our method, by considering the attention in a dictionary form $\Delta\mathbf{O} = \mathbf{A}\mathbf{V}$ or $\Delta\mathbf{O} = \mathbf{S}\mathbf{D}$. In this experiment, we examine the difference between updating coefficients using a low-rank approach versus a sparse approach. Both LoRA and our method produce the updated feature $\Delta\mathbf{O}$ with rank 1, where LoRA achieves this by setting $r = 1$ and our method enforces only one active atom in $\mathbf{D}$. The key difference lies in how the coefficients are updated—LoRA adapts them in a low-rank manner, whereas our method updates them sparsely. As shown in Figure 4, by fine-tuning the model on a single concept, both LoRA and our method successfully reconstruct the concept using the same prompt. LoRA fails to produce meaningful images with a slightly modified prompt, while our method remains robust, consistently generating a concept that aligns with the updated text prompt. We attribute the effectiveness of our approach to the sparse coefficients produced by $\sigma_\lambda(\mathbf{A}\mathbf{X}\mathbf{W}_s)$. With a different text prompt, the values in $\mathbf{S}$ either remain zero or take on a significant value when receiving the proper input, ensuring that the generated result remains coherent. In contrast, LoRA produces noisy coefficients, leading to completely disrupted outputs.

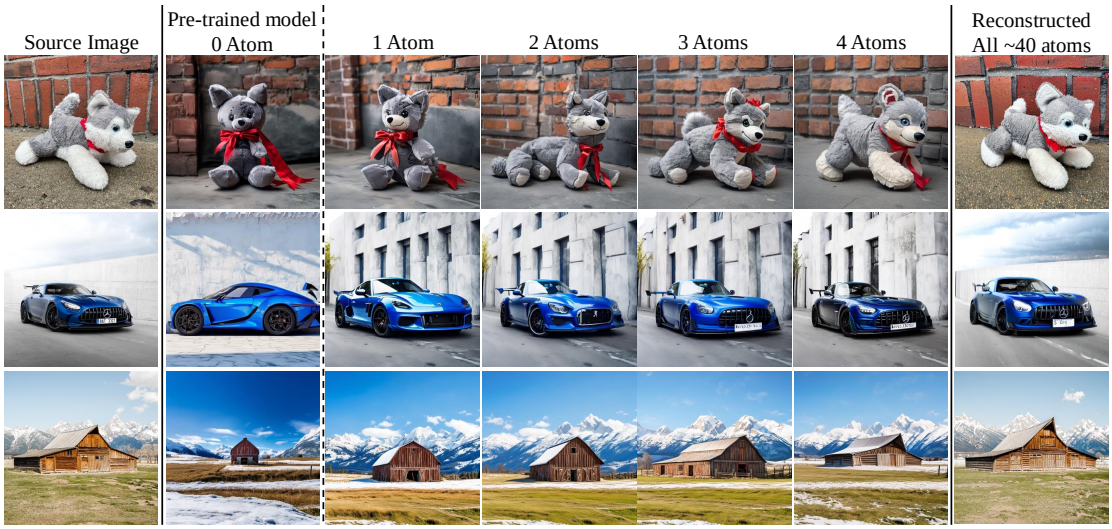

Figure 3. The pre-trained model can generate the target concept when integrated with the learned dictionary, even with about 40 atoms. With just four atoms, the core structure of the concept can still be generated. Adjusting the number of selected atoms influences the generated results, offering a controllable approach to image editing. Additional discussions of these figures are in Section 4.1.

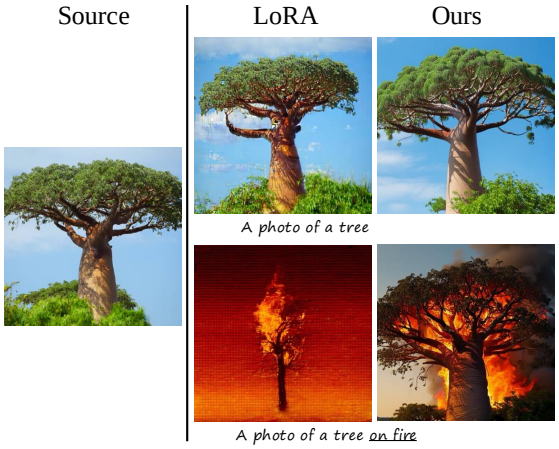

Figure 4. The role of sparsity: Our method utilizes the same number of atoms as LoRA, yet LoRA struggles to adapt to the target prompt effectively. Sparse coding ensures that atoms are either activated or inactive, providing a structured and interpretable adaptation process. In contrast, LoRA produces noisy coefficients, leading to completely disrupted outputs.

4. Experiments

In this section, we evaluate the effectiveness of our method in image generation tasks. We demonstrate that feature dictionary atoms effectively represent the target concept after fine-tuning the pre-trained model. Compared to baselines, our approach enables selective atom adjustment, improving image editing by refining results and enhancing text-to-image concept customization while preserving pre-trained model capabilities.

4.1. Representing Concepts Using Atoms

Experimental settings. In this experiment, we utilize PixArt–Σ [7], a Diffusion Transformer (DiT) [37] model consisting of 28 transformer blocks. We fine-tune the model on a single image for 60 epochs using our method and reconstruct the corresponding concept using the same prompt used during fine-tuning. The selected images are drawn from DreamBooth [44], Custom Diffusion [22]. We control the number of active atoms and visualize the variation of the generated results with different atoms.

The concept is represented by a set of atoms. As shown in Figure 3, after fine-tuning our method on the target concept, its representation can be effectively captured using a small set of atoms, such as 40 atoms. Among these, just 4 atoms are sufficient to construct the core structure of the concept, including key attributes like pose and color. We provide details descriptions of these images in Appendix C.

4.2. Image Editing

Building on our previous observation that adjusting the number of atoms influences the generated results, we explore leveraging this property for image editing. Image editing tasks aim to modify an image using a generative model while retaining as many details as possible from the original [58]. However, certain editing tasks pose challenges in maintaining most of the original details while achieving effective modifications. For example, altering the style can significantly change the overall appearance of the image. We argue that successful image editing should preserve the main components of the image while applying modifications selectively. Since our method represents the core

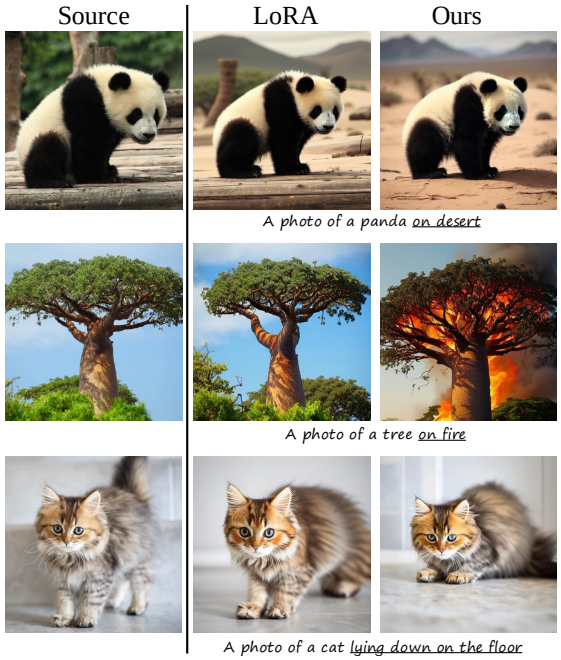

Figure 5. Results on image editing. Our method shows improved results by better following editing prompts while preserving the key components of the source images.

components of a concept using only a few atoms, it is well-suited for image editing tasks, ensuring that key structural elements remain intact while allowing for controlled transformations.

Experimental settings. In our experiment, we select data from previous work [58] and evaluate different editing tasks, including background modification, scene alteration, and pose adaptation. Previous works [14, 58] fine-tune models to learn a concept and modify its appearance by adjusting the prompt. However, these approaches have not yet been applied to DiT. Following their setup, we compare our method with fine-tuning approaches, such as LoRA [16], using them as baselines for evaluation. We adopt the DiT architecture and fine-tune both LoRA and our method on a single image for 60 epochs to achieve full reconstruction. After fine-tuning, we modify the image using a different prompt for editing.

Image editing results. The results are shown in Figure 5. Our method applies edits based on the test prompt while maintaining the core structure of the source images. In addtion, we control the number of active atoms by adjusting the coefficient density ρ, where $1-\rho$ represents the sparsity level. In Figure 6, as sparsity increases, only the most essential atoms remain, preserving the core concept while allowing the pre-trained model greater flexibility to integrate editing effects based on the modified text prompt.

Figure 6. Adjusting the density, *i.e.*, the number of activated atoms, can change the influence on the edited image.

4.3. Concept Customization

In the previous section, we demonstrated that our method can efficiently represent the target concept using only a small number of atoms. In this section, we further show that even with just a few atoms, our method achieves good performance in concept customization.

Experimental settings. We evaluated our approach on the DreamBooth [44] dataset, which consists of 30 subjects across 15 distinct classes. Among these, 9 are live subjects, while the remaining 21 are objects. Each subject has between 4 to 6 images, captured under varying conditions, in diverse environments, and from multiple angles. All 30 subjects were trained for 15 epochs and validated on 25 different prompts. For our experiments, we adopted the Pixart-Σ, a Diffusion Transformer (DiT) [7], as the base architecture and fine-tuned it using our proposed method. To verify the effectiveness of our approach, we compared its performance against several PEFT baselines for personalized text-to-image generation, including LoRA [16], DoRA [24], and OFT [40]. We use the AdamW optimizer with a learning rate of 1×10^{-4} to fine-tune the Pixart-Σ [7] for 15 steps. For the baseline methods, LoRA and DoRA are assigned a rank of $r = 16$, while OFT is set to $r = 4$. We generated five images for each prompt at a resolution of 1024×1024.

Evaluation metrics. We measure concept alignment by computing the mean cosine similarity between the generated image's CLIP embeddings [41] and those of the original concept images. We quantify image diversity using the Vendi score [13] with DINOv2 embeddings [35]. For text alignment, we calculate the average cosine similarity in the CLIP feature space [41]. ImageReward [54] is a general-purpose text-to-image human preference reward model that evaluates and ranks AI-generated images by scoring them based on human preferences.

Customization results. As shown in Table 1, compared with baseline methods, our approach not only represents the concept with a minimal set of learned atoms but also generalizes robustly to diverse scenarios. In particular, our

Table 1. Comparison with baseline methods on concept customization.

	Alignment	Diversity	Fidelity	ImageReward
LoRA [16]	0.218	4.707	0.711	<u>0.220</u>
DoRA [24]	0.213	4.540	<u>0.712</u>	0.142
OFT [40]	0.190	3.926	0.663	0.137
Ours ($\rho = 0.02$)	**0.235**	**5.205**	0.702	0.182
Ours ($\rho = 0.04$)	<u>0.231</u>	<u>5.172</u>	**0.713**	**0.431**

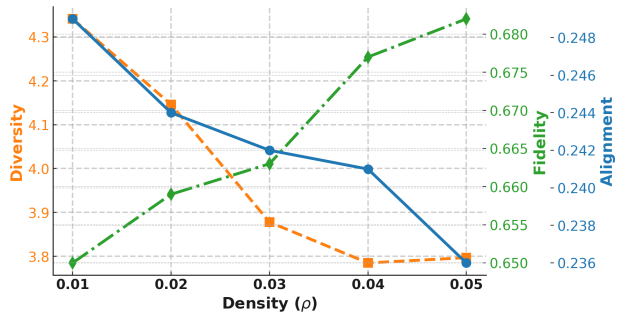

Figure 7. The influence of different density ρ on the image diversity, fidelity and text-to-image alignment.

method achieves higher text-to-image alignment and diversity scores than most other PEFT baselines, while still preserving a relatively high fidelity score. This indicates that our approach effectively captures essential features of each concept without sacrificing the quality of customization text-to-image generation.

Number of parameters in different methods. We compare the trainable parameters of our method and the baselines. The pre-trained PixArt model has around 616M parameters in total. Among the fine-tuning methods, both LoRA and DoRA have around 4M trainable parameters. OFT uses around 37M trainable parameters, and our method trains around 17M parameters.

4.4. Abalation Study

The impact of sparsity in the coefficients. We analyze the impact of varying sparsity levels in the coefficients through a concept customization experiment, evaluating text alignment, image diversity, and image fidelity. As shown in Figure 7, higher sparsity (low density) enhances text alignment and image diversity but results in slightly lower image fidelity. In contrast, lower sparsity (high density) improves fidelity, indicating that using more atoms allows for a more precise representation of the concept from the source image. In our experiment, we adopt both $\rho = 0.02$ and $\rho = 0.04$ for either better diversity or better fidelity.

Table 2. The influence of different member numbers M. As the number of members increases, alignment and diversity scores decrease, while the fidelity score improves.

$\rho = 0.01$	Alignment	Diversity	Fidelity
M = 16	<u>0.248</u>	**5.867**	0.547
M = 256	**0.249**	<u>4.341</u>	0.650
M = 1024	0.241	3.733	<u>0.672</u>
M = 2048	0.238	3.525	**0.692**

The impact of member numbers M. We also examine the impact of the number of members, *i.e.*, the dictionary size, on the generated results while maintaining the same level of sparsity. As shown in Table 2, increasing the number of members improves image fidelity, whereas a smaller dictionary size reduces fidelity but enhances text alignment and image diversity. In our experiment, we choose $M = 256$ to provide balanced fidelity and diversity.

Sparsity coefficients provide more stable adaptation. In this experiment, we compare our method with LoRA on the image editing task. As we discussed in Section 3.5, LoRA adapts the coefficients in a low-rank manner, whereas our method updates them sparsely. Both LoRA and our method produce the updated feature $\Delta\mathbf{O}$ with rank 1, where LoRA achieves this by setting $r = 1$ and our method enforces only one active atom in $\mathbf{D}$. As shown in Figure 4, by fine-tuning the model on a single concept, both LoRA and our method successfully reconstruct the concept using the same prompt. However, when applying the fine-tuned weights to generate images with a slightly modified prompt, LoRA becomes unstable, failing to produce meaningful images. In contrast, our method remains robust, consistently generating a concept that aligns with the updated text prompt.

5. Conclusion

In this paper, we introduce to frame model fine-tuning as a sparse coding problem, where the adapted feature representation is constructed using a dictionary of feature atoms. We propose a fine-tuning framework that leverages a combination of feature dictionary atoms, demonstrating that tuning them enhances control over the adaptation process. We validate the effectiveness of our approach across various generative tasks. By representing concepts with only a few essential atoms, our method proves to be highly adaptable and can be effectively applied to image editing, style mixing, and concept personalization. Our study provides a new perspective on the mechanics of fine-tuning, paving the way for more controllable adaptation strategies in large-scale pre-trained models.

References

[1] Sriram Balasubramanian, Samyadeep Basu, and Soheil Feizi. Decomposing and interpreting image representations via text in vits beyond CLIP. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2009. 3

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 2

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 5

[5] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. Towards monosemanticity: Decomposing language models with dictionary learning. 2023. Transformer Circuits Thread. 2

[6] Emmanuel J Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, pages 1433–1452, 2006. 3

[7] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91, 2024. 6, 7, 2

[8] Wei Chen, Zichen Miao, and Qiang Qiu. Large convolutional model tuning via filter subspace. *International Conference on Learning Representations*, 2025. 2

[9] Xiuyuan Cheng, Zichen Miao, and Qiang Qiu. Graph convolution with low-rank learnable local filters. *arXiv preprint arXiv:2008.01818*, 2020. 1

[10] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 2

[11] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1

[12] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders enable fine-grained interpretable circuit analysis for language models. 2024. AI Alignment Forum. 2

[13] Dan Friedman and Adji Bousso Dieng. The vendi score: A diversity evaluation metric for machine learning. *arXiv preprint arXiv:2210.02410*, 2022. 7

[14] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *CVPR*, 2023. 7

[15] Shaozhe Hao, Kai Han, Zhengyao Lv, Shihao Zhao, and Kwan-Yee K Wong. Conceptexpress: Harnessing diffusion models for single-image unsupervised concept extraction. In *European Conference on Computer Vision*, 2024. 2

[16] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 1, 7, 8

[17] Teng Hu, Jiangning Zhang, Ran Yi, Hongrui Huang, Yabiao Wang, and Lizhuang Ma. Sara: High-efficient diffusion model fine-tuning with progressive sparse low-rank adaptation. *arXiv preprint arXiv:2409.06633*, 2024. 2

[18] Jeeyung Kim, Erfan Esmaeili, and Qiang Qiu. Text embedding is not all you need: Attention control for text-to-image semantic alignment with text self-attention maps. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8031–8040, 2025. 2

[19] Taewook Kim, Wei Chen, and Qiang Qiu. Learning to customize text-to-image diffusion in diverse context. *arXiv preprint arXiv:2410.10058*, 2024. 2

[20] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 2019. 5

[21] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[22] 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*, 2023. 2, 6

[23] Xunhao Lai, Jianqiao Lu, Yao Luo, Yiyuan Ma, and Xun Zhou. Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. *arXiv preprint arXiv:2502.20766*, 2025. 2

[24] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024. 1, 7, 8

[25] Yiyang Liu, James Chenhao Liang, Ruixiang Tang, Yugyung Lee, Majid Rabbani, Sohail Dianat, Raghuveer Rao, Lifu Huang, Dongfang Liu, Qifan Wang, et al. Re-imagining multimodal instruction tuning: A representation view. *arXiv preprint arXiv:2503.00723*, 2025. 2

[26] Yang Luo, Xiaozhe Ren, Zangwei Zheng, Zhuo Jiang, Xin Jiang, and Yang You. Came: Confidence-guided adaptive memory efficient optimization. *arXiv preprint arXiv:2307.02047*, 2023. 2

[27] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis Bach. Supervised dictionary learning. *Advances in neural information processing systems*, 2008. 1

[28] Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*, 2024. 2

[29] Zichen Miao, Ze Wang, Wei Chen, and Qiang Qiu. Continual learning with filter atom swapping. In *International Conference on Learning Representations*, 2021. 1

[30] Zichen Miao, Ze Wang, Xiuyuan Cheng, and Qiang Qiu. Spatiotemporal joint filter decomposition in 3d convolutional neural networks. *Advances in Neural Information Processing Systems*, 34:3376–3388, 2021. 1

[31] Zichen Miao, Jiang Wang, Ze Wang, Zhengyuan Yang, Lijuan Wang, Qiang Qiu, and Zicheng Liu. Training diffusion models towards diverse image generation with reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10844–10853, 2024. 2

[32] Zichen Miao, Zhengyuan Yang, Kevin Lin, Ze Wang, Zicheng Liu, Lijuan Wang, and Qiang Qiu. Tuning timestep-distilled diffusion model using pairwise sample optimization. *arXiv preprint arXiv:2410.03190*, 2024. 2

[33] Zichen Miao, Wei Chen, and Qiang Qiu. Coeff-tuning: A graph filter subspace view for tuning attention-based large models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20146–20157, 2025. 2

[34] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, pages 607–609, 1996. 1

[35] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 7

[36] Gaurav Patel, Christopher Sandino, Behrooz Mahasseni, Ellen L Zippi, Erdrin Azemi, Ali Moin, and Juri Minxha. Efficient source-free time-series adaptation via parameter subspace disentanglement. *arXiv preprint arXiv:2410.02147*, 2024. 1

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *CVPR*, 2023. 6

[38] Vishal Purohit, Matthew Repasky, Jianfeng Lu, Qiang Qiu, Yao Xie, and Xiuyuan Cheng. Posterior sampling via langevin dynamics based on generative priors. *arXiv preprint arXiv:2410.02078*, 2024. 2

[39] Qiang Qiu, Xiuyuan Cheng, Guillermo Sapiro, et al. Dcfnet: Deep neural network with decomposed convolutional filters. In *International Conference on Machine Learning*, pages 4198–4207. PMLR, 2018. 1

[40] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. *Advances in Neural Information Processing Systems*, 36:79320–79362, 2023. 1, 7, 8

[41] 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*, 2021. 7

[42] Jie Ren, Yaxin Li, Shenglai Zeng, Han Xu, Lingjuan Lyu, Yue Xing, and Jiliang Tang. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention. In *European Conference on Computer Vision*, 2024. 2

[43] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1

[44] 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. 2, 6, 7, 3

[45] Viacheslav Surkov, Chris Wendler, Mikhail Terekhov, Justin Deschenaux, Robert West, and Caglar Gulcehre. Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders. *arXiv preprint arXiv:2410.22366*, 2024. 2

[46] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *International conference on machine learning*, 2020. 2

[47] Nathaniel Tomczak and Sanmukh Kuppannagari. Longer attention span: Increasing transformer context length with sparse graph processing techniques. *arXiv preprint arXiv:2502.01659*, 2025. 2

[48] 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*, 2017. 1, 2, 3

[49] Ze Wang, Xiuyuan Cheng, Guillermo Sapiro, and Qiang Qiu. Stochastic conditional generative networks with basis decomposition. *arXiv preprint arXiv:1909.11286*, 2019. 1

[50] Ze Wang, Seunghyun Hwang, Zichen Miao, and Qiang Qiu. Image generation using continuous filter atoms. *Advances in Neural Information Processing Systems*, 34:17826–17838, 2021.

[51] Ze Wang, Zichen Miao, Jun Hu, and Qiang Qiu. Adaptive convolutions with per-pixel dynamic filter atom. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12302–12311, 2021. 1

[52] Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Advancing parameter efficiency in fine-tuning via representation editing. 2024. 2

[53] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. 2024. 2

[54] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 2023. 7

[55] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. Native sparse attention: Hardware-aligned and natively trainable sparse attention, 2025. 2

[56] Yang Zhang, Teoh Tze Tzun, Lim Wei Hern, and Kenji Kawaguchi. Enhancing semantic fidelity in text-to-image

synthesis: Attention regulation in diffusion models. In *European Conference on Computer Vision*, 2024. 2

[57] Yasi Zhang, Peiyu Yu, and Ying Nian Wu. Object-conditioned energy-based attention map alignment in text-to-image diffusion models. In *European Conference on Computer Vision*, 2024. 2

[58] Zhixing Zhang, Ligong Han, Arnab Ghosh, Dimitris N Metaxas, and Jian Ren. Sine: Single image editing with text-to-image diffusion models. In *CVPR*, 2023. 6, 7

Sparse Fine-Tuning of Transformers for Generative Tasks

Supplementary Material

A. Analysis

A.1. Analysis of Adapted Feature Representation

With our formulation, the adapted feature representation $\Delta\mathbf{O}$ at each layer can be expressed as a linear combination of learned atoms $\mathbf{D}$. However, deep neural networks represent features non-linearly. Although $\Delta\mathbf{O}$ can be expressed linearly in terms of $\mathbf{D}$, its influence becomes non-linear as it interacts with activation functions and transformations in subsequent layers. In this section, we introduce an approximation method to address this non-linearity, enabling us to trace and quantify the contribution of each atom to the final outputs throughout the entire model.

Formulation of the pre-trained model. We represent each layer of the pre-trained model as $f^{(l)}(\mathbf{x})$. For an L-layer model, it writes as $f^{(L)} \circ \cdots \circ f^{(1)}(\mathbf{x})$, where $l = 1, \cdots, L$ is the index of the layer. Based on the functional approximation theory, we can approximate the function in an appropriate basis. Given a basis $\{g_k\}_{k=1}^{K}$, the coefficients of basis expansion are $\mu_k = \langle f, g_k \rangle$, such that $f(\mathbf{x}) = \sum_{k=1}^{K} \mu_k g_k(\mathbf{x})$, where $\langle \cdot, \cdot \rangle$ is inner product. For simplified analysis, we utilize the polynomial basis, and represent $f(\mathbf{x})$ as:

$$f(\mathbf{x}) = \sum_{k=1}^{K} c_k \mathbf{x}^k. \tag{10}$$

The influence of atoms on one layer. Fine-tuning introduces adapted feature representation at layer $l-1$, *i.e.*, $\Delta\mathbf{O}^{(l-1)} = \mathbf{SD}$, which become the perturbation of the input $\mathbf{X}^{(l)}$ at layer l. For each input, we have $\mathbf{x} + \sum_{m=1}^{M} s_m \mathbf{d}_m$, where $\mathbf{d}_m \in \mathbb{R}^{C_o}$ is the single atom in $\mathbf{D}$ and s_m is the corresponding sparse coefficient. With the perturbation, we have

$$f(\mathbf{x} + \sum_{m=1}^{M} s_m \mathbf{d}_m) = f(\mathbf{x}) + \sum_{m=1}^{M} \mathbf{B}_K(\mathbf{d}_m), \tag{11}$$

where $\mathbf{B}_K(\mathbf{d}_m) = \sum_{k=1}^{K} \beta_{k,m} \mathbf{d}_m^k$ and $\beta_{k,m}$ depends on $\mathbf{x}$, s_m, and c_k. We assume $\langle \mathbf{d}_i, \mathbf{d}_j \rangle = 0, \forall i \neq j$ to avoid correlated influence from different atoms. It can be achieved with a simple regularization term. With the influence of linear perturbations based on atoms at layer $l-1$, the different in the output at layer l is determined by the linear combination of higher-order terms of these atoms $\mathbf{d}_m^k$.

The accumulated influence of atoms on multiple layers. Applying the same formulation to layer $l+1$ results in a similar expression as above, but incorporates the combined influence of atoms from both layer $l-1$ and layer l. Specifically, the input of layer $l+1$ is written as,

$$\mathbf{x}^{(l+1)} + \sum_{m=1}^{M} s_m^{(l)} \mathbf{d}_m^{(l)} + \sum_{m=1}^{M} \mathbf{B}_{K^{(l-1)}}(\mathbf{d}_m^{(l-1)}), \tag{12}$$

where $\mathbf{x}^{(l+1)}$ is the original input at layer $l+1$, $\sum_{m=1}^{M} s_m^{(l)} \mathbf{d}_m^{(l)}$ is the linear perturbations based on the atoms from layer l, $\sum_{m=1}^{M} \mathbf{B}_{K^{(l-1)}}(\mathbf{d}_m^{(l-1)})$ is the influence of the atoms from layer $l-1$, which is a combination of higher-order terms. The output of layer $l+1$ is,

$$\begin{aligned} &f^{(l+1)}(\mathbf{x}^{(l+1)} + \sum_{m=1}^{M} s_m^{(l)} \mathbf{d}_m^{(l)} + \sum_{m=1}^{M} \mathbf{B}_{K^{(l-1)}}(\mathbf{d}_m^{(l-1)})) \\ =&f(\mathbf{x}) + \sum_{m=1}^{M} \mathbf{B}_{K^{(l)}}(\mathbf{d}_m^{(l)}) + \sum_{m=1}^{M} \mathbf{B}_{K^{(l-1)}+K^{(l)}}(\mathbf{d}_m^{(l-1)}). \end{aligned} \tag{13}$$

This formulation can be naturally extended to the whole model. We provide detailed analysis in Appendix A.

After incorporating atoms into each layer to linearly capture the adapted feature representations, the model output can be interpreted as the original pre-trained output with perturbations introduced by the atoms at each layer. The atoms from the shallow layer (e.g., layer 1), introduce more higher-order terms to the final output, i.e., $\mathbf{B}_{\sum_{l=1}^{L} K^{(l)}}(\mathbf{d}_m^{(1)})$. It contributes more to the overall structure of the output. The atoms from the deep layer ((e.g., layer L)), introduce fewer higher-order terms, or only linear terms to the final output. It contributes more to the details of the output.

A.2. Proof

The proof of (11). To prove (11), we assume $\langle \mathbf{d}_i, \mathbf{d}_j \rangle = 0, \forall i \neq j$ to avoid correlated influence from different atoms. It can be achieved with a simple regularization term.

Proof. Inserting (10) to the LHS of (11), we have

$$f(\mathbf{x} + \sum_{m=1}^{M} s_m \mathbf{d}_m) = \sum_{k=1}^{K} c_k (\mathbf{x} + \sum_{m=1}^{M} s_m \mathbf{d}_m)^k. \tag{14}$$

After expanding this equation, we have

$$\begin{aligned}&\sum_{k=1}^{K} c_k(\mathbf{x}+\sum_{m=1}^{M} s_m\mathbf{d}_m)^k \\ =&\sum_{k=1}^{K} c_k\left[\sum_{j=0}^{k}\binom{k}{j}\mathbf{x}^{k-j}\left(\sum_{m=1}^{M} s_m\mathbf{d}_m\right)^j\right].\end{aligned} \tag{15}$$

Considering $\langle \mathbf{d}_i, \mathbf{d}_j\rangle = 0, \forall i \neq j$, we have,

$$\left(\sum_{m=1}^{M} s_m\mathbf{d}_m\right)^j = \sum_{m=1}^{M}(s_m\mathbf{d}_m)^j. \tag{16}$$

Thus,

$$\begin{aligned}&f(\mathbf{x}+\sum_{m=1}^{M} s_m\mathbf{d}_m) \\ =&\sum_{k=1}^{K} c_k\left[\mathbf{x}^k+\sum_{j=1}^{k}\binom{k}{j}\mathbf{x}^{k-j}\sum_{m=1}^{M}(s_m\mathbf{d}_m)^j\right] \\ =&f(\mathbf{x})+\sum_{m=1}^{M}\sum_{k=1}^{K} c_k\sum_{j=1}^{k}\binom{k}{j}\mathbf{x}^{k-j}(s_m\mathbf{d}_m)^j.\end{aligned} \tag{17}$$

We define

$$\mathbf{B}_K(\mathbf{d}_m) = \sum_{k=1}^{K} c_k\sum_{j=1}^{k}\binom{k}{j}\mathbf{x}^{k-j}(s_m\mathbf{d}_m)^j, \tag{18}$$

which represents the higher-order terms of $\mathbf{d}_m$ up to order K. □

The proof of (13). Here we only provide a sketch of proof of (13). Compared with (11), (13) contains an additional term $\sum_{m=1}^{M}\mathbf{B}_{K^{(l-1)}}(\mathbf{d}_m^{(l-1)})$. After applying the expansion (10), we will have

$$\sum_{j=0}^{k}\binom{k}{j}(\mathbf{B}_{K^{(l-1)}}(\mathbf{d}_m^{(l-1)}))^{k-j}\left(\sum_{m=1}^{M} s_m\mathbf{d}_m\right)^j, \tag{19}$$

which produces the higher-order terms of $\mathbf{d}_m^{(l-1)}$ up to order $K^{(l-1)}+K^{(l)}$, thus we have $\mathbf{B}_{K^{(l-1)}+K^{(l)}}(\mathbf{d}_m^{(l-1)})$ in (13).

B. Experimental Settings

MNIST generation with VAE. In this experiment, the input of MNIST has the size of 28×28. The VAE consists with an encoder and decoder, each with two layers, a feature dimension of 128, and four attention heads. It first reshape the input with a patch size of 7. The latent space is mapped to a dimension of 32. We train the model using the Adam optimizer with a learning rate of 0.001 for 20 epochs.

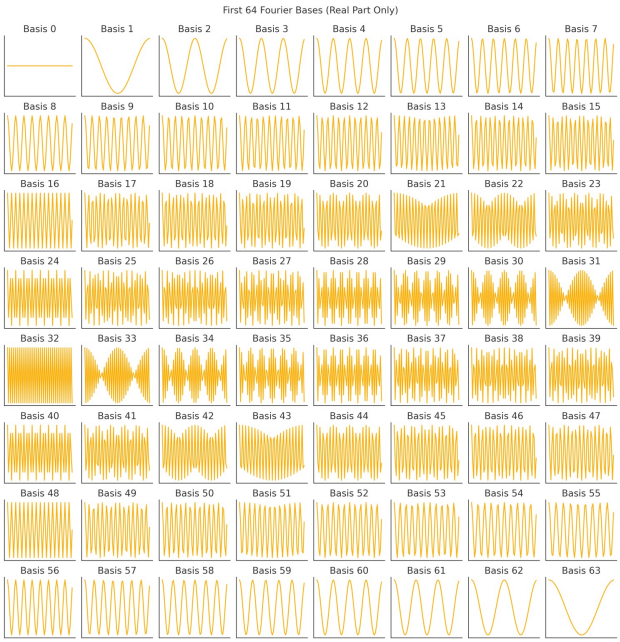

Figure 8. The illustration of Fourier basis.

Generative tasks with DiT. In this experiment, we use the CAME [26] optimizer with a learning rate of 1×10^{-4} to fine-tune the Pixart-Σ [7]. For the baseline methods, LoRA and DoRA are assigned a rank of $r = 16$, while OFT is set to $r = 4$. We generated five images for each prompt at a resolution of 1024×1024. We run the experiment on Nvidia A5000 with 24GB memory. For the image editing task, we train the model on 1 image for 60 epochs, which takes about 5min. For the concept customization task, we train the model on 4-6 images for 15 epochs, which takes about 5min.

C. Additional Experimental Results

C.1. Toy Experiment

Experimental setting. In this experiment, we only use the attention block to transform the signal, the feature dimension is 128, and the sequence length is 64. The synthetic signals are generated by randomly combining 5 Fourier bases, which is shown in Figure 8. To leverage the transformer, we first project the 1D signal into a 64D space using a randomly initialized projection matrix, which remains frozen during training. The output signal is then mapped from 64D back to 1D by summing across all 64 dimensions.

Sparse coefficients provide interpretability. The sparse coefficients $\mathbf{S}$ enhance interpretability by establishing a direct connection between a small subset of atoms and the output feature, *i.e.*, $\mathbf{O}_i = \sum_{j=1}^{M}\mathbf{S}_{ij}\mathbf{D}_j$. This behavior

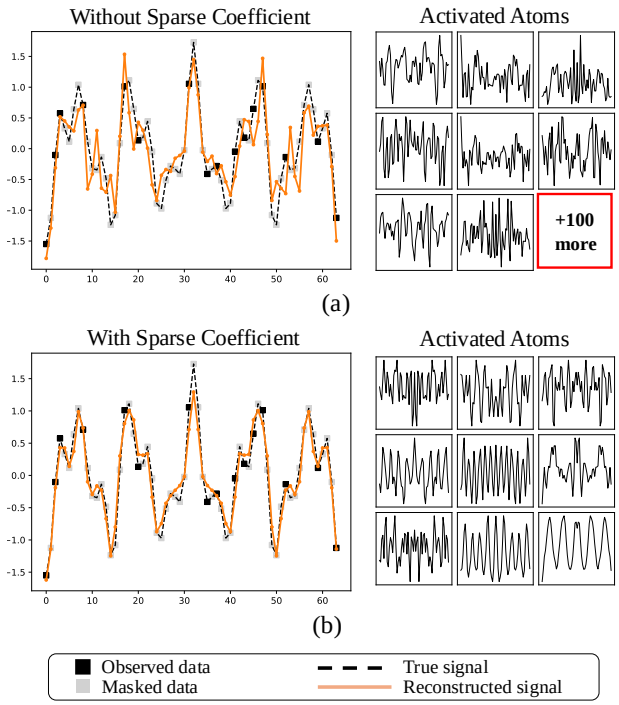

Figure 9. Compared to the performance of (a) standard attention $\mathbf{AXW}_v$, (b) the sparse combination of atoms $\sigma_\lambda(\mathbf{AXW}_s)\mathbf{D}$ provides more interpretability. The reconstructed signal is a linear combination of basic signals. Without sparse coefficients, constructing the signal requires 100 atoms. However, with sparse coefficients, the reconstructed signal is effectively represented using only 9 atoms, demonstrating a more compact and interpretable composition.

is also validated in the literature on sparse coding, compressed sensing, and dictionary learning [6]. To explore this within the context of an attention block, we perform an experiment where masked one-dimensional signals are reconstructed using an attention mechanism. The signals, with a length of 64, are synthesized by selecting 5 random Fourier bases, with a low frequency ranging from $\frac{0}{64}$ to $\frac{24}{64}$. The objective is to reconstruct the signal after masking out 75% of its values with only 16 randomly sampled observations. As illustrated in Figure 9, for both $\mathbf{O} = \mathbf{AXW}_v$ and $\mathbf{O} = \sigma_\lambda(\mathbf{AXW}_s)\mathbf{D}$, the attention block successfully reconstructs signals after jointly learning the coefficients and atoms. $\sigma_\lambda(\mathbf{AXW}_s)\mathbf{D}$ takes advantage of sparse coding, and requires only 9 atoms to fully reconstruct the signal. In comparison, $\mathbf{AXW}_v$ requires 100 atoms to construct the signal. Interestingly, the atoms $\mathbf{D}$ in $\sigma_\lambda(\mathbf{AXW}_s)\mathbf{D}$ naturally resemble certain Fourier basis functions from the ones used to synthesize real data.

Description of reconstructed images. Figure 3 also shows that the number of selected atoms impacts the generated results. For example, when learning the concept described in the top row, "*A grey* $\langle V \rangle$ *wolf plushie ...*", by adjusting the number of active atoms we can see that some atoms influence the texture of the fur, while others shape the posture of the object or determine the position of the bow tie. For the concept in the second row, "*A blue* $\langle V \rangle$ *sports car ...*", we observe that certain atoms influence the orientation of the sports car, while others affect the position of the rear wing and the shape of the grille. For the concept in the third row, "*A* $\langle V \rangle$ *wooden barn ...*", we observe that certain atoms influence the number of lean-tos on the barn, while others affect the slope of the roof.

C.2. Personalization Results Comparison

In this section, we showcase the comparison of personalization results for various concepts selected from the DreamBooth [44] dataset.

As shown in Figures 10–15, our approach produces outputs that not only align more accurately with the text prompts but also preserve the fine details of each learned concept more effectively than the baselines.

Figure 10. Personalization generated results comparison

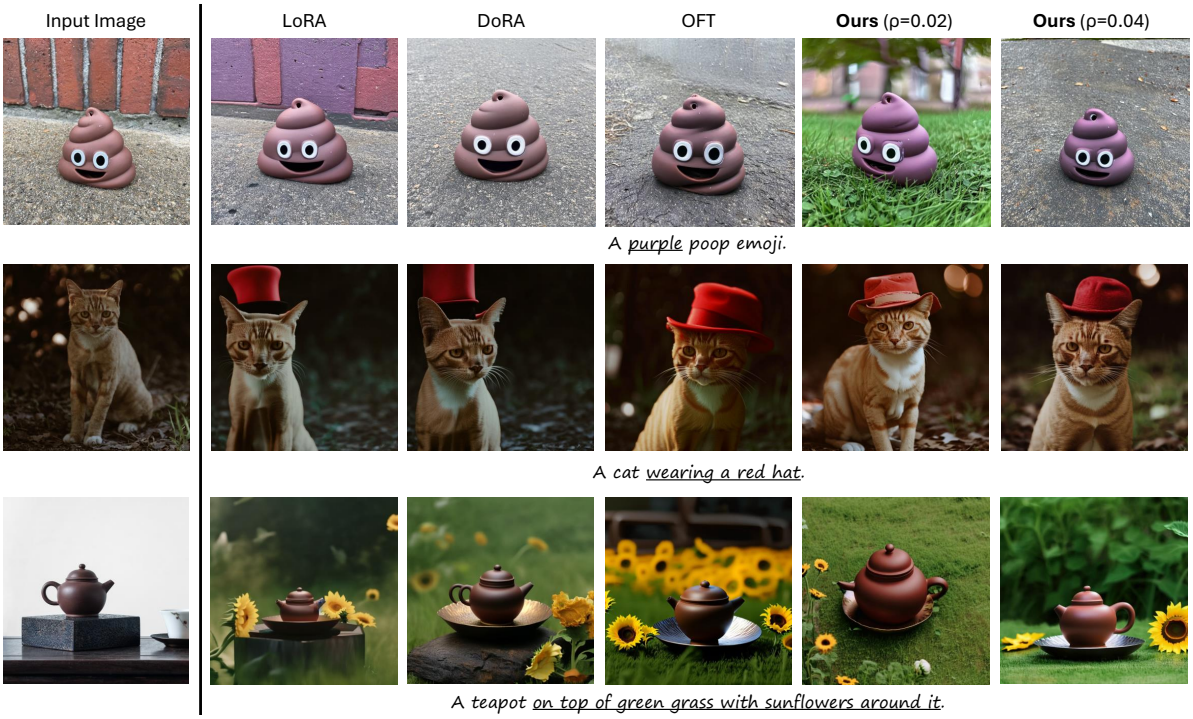

Figure 11. Personalization generated results comparison

Figure 12. Personalization generated results comparison

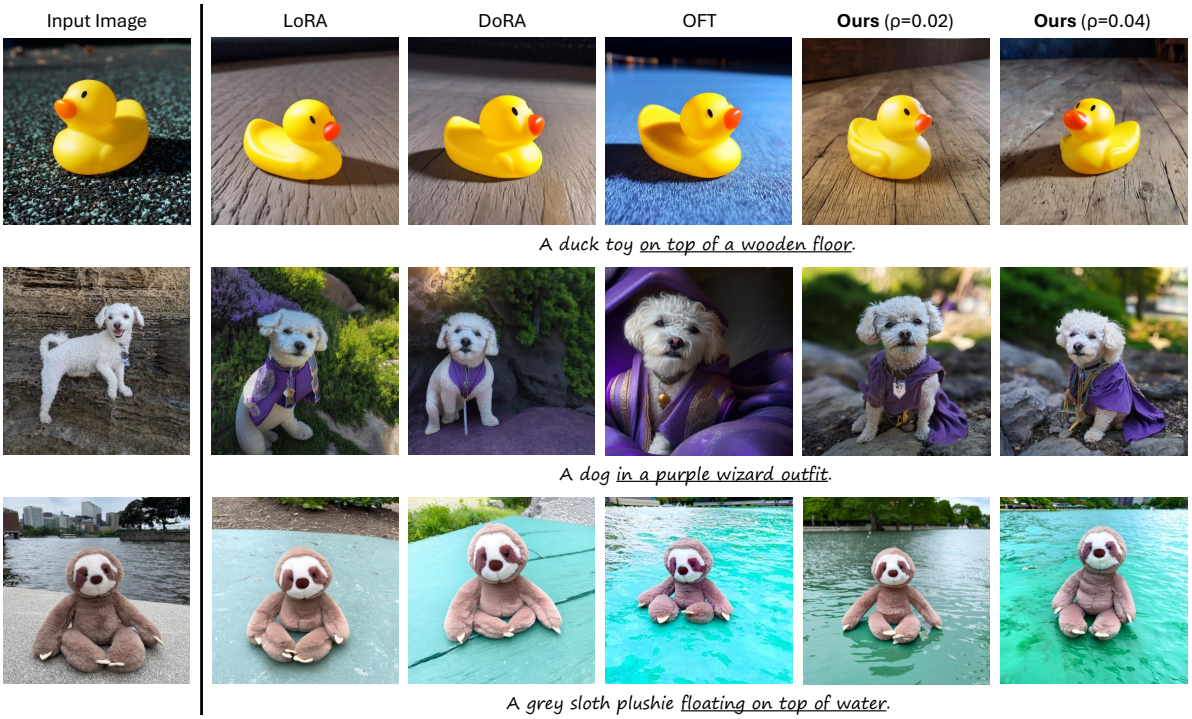

Figure 13. Personalization generated results comparison

Figure 14. Personalization generated results comparison

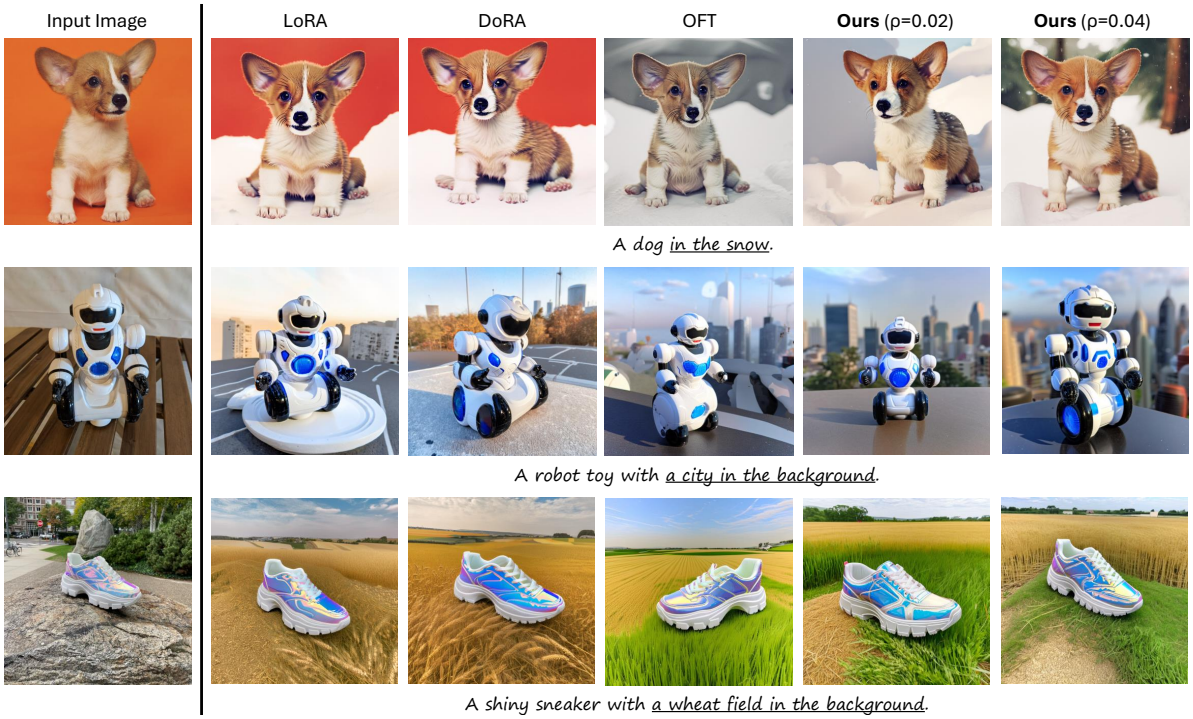

Figure 15. Personalization generated results comparison