ConceptPrism: Concept Disentanglement in Personalized Diffusion Models via Residual Token Optimization

Supplementary Material

9. Derivation of Equation 4

The objective of our exclusion loss is to ensure that the residual token $\mathbf{t}_{\text{residual}}^{(i)}$ remains uninformative regarding the target concept when reconstructing other images $\mathbf{x}^{(j)}$ where $j \neq i$. As explained in Section 4.2, this is achieved by minimizing the KL divergence between the distribution conditioned on the residual token and the unconditional distribution:

$$\mathcal{L}_{\text{KL}} = \mathbb{E}_{i,\mathbf{z}_t^{(j\neq i)},t}\Big[D_{\text{KL}}\Big(p_\theta(\mathbf{z}_{t-1}^{(j)}|\mathbf{z}_t^{(j)},t,\varnothing) \parallel p_\theta(\mathbf{z}_{t-1}^{(j)}|\mathbf{z}_t^{(j)},t,\Gamma(c_{\text{residual}}^{(i)}))\Big)\Big]. \tag{6}$$

In diffusion models [24], the reverse process $p_\theta(\mathbf{z}_{t-1}|\mathbf{z}_t,t,\mathbf{c})$ is parameterized as a Gaussian distribution $\mathcal{N}(\mathbf{z}_{t-1};\boldsymbol{\mu}_\theta(\mathbf{z}_t,t,\mathbf{c}),\boldsymbol{\Sigma}_\theta(\mathbf{z}_t,t))$. Following standard practice [9], we fix the covariance matrix $\boldsymbol{\Sigma}_\theta$ to a time-dependent constant $\beta_t\mathbf{I}$. The cumulative signal scale $\bar{\alpha}_t$ is defined as $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$ following DDPM notation [9], based on the noise schedule β_t. Under the assumption of equal covariance, the KL divergence between two Gaussians is proportional to the squared Euclidean distance between their means:

$$D_{\text{KL}}(\mathcal{N}(\boldsymbol{\mu}_1,\boldsymbol{\Sigma}) \parallel \mathcal{N}(\boldsymbol{\mu}_2,\boldsymbol{\Sigma})) \propto \|\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\|_2^2. \tag{7}$$

Thus, minimizing $\mathcal{L}_{\text{KL}}$ is equivalent to minimizing the distance between the predicted means given the null condition $\varnothing$ and the residual condition $c_{\text{residual}}^{(i)}$. The mean $\boldsymbol{\mu}_\theta$ is related to the noise prediction $\boldsymbol{\epsilon}_\theta$ by:

$$\boldsymbol{\mu}_\theta(\mathbf{z}_t,t,\mathbf{c}) = \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t,t,\mathbf{c})\right). \tag{8}$$

Substituting this into the distance metric, the terms involving $\mathbf{z}_t$ cancel out, simplifying the objective to the difference between noise predictions:

$$\begin{aligned}
&\left\|\boldsymbol{\mu}_\theta(\mathbf{z}_t^{(j)},t,\varnothing) - \boldsymbol{\mu}_\theta(\mathbf{z}_t^{(j)},t,\Gamma(c_{\text{residual}}^{(i)}))\right\|_2^2 \\
&= \left\| \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{z}_t^{(j)} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^{(j)},t,\varnothing)\right)\right. \\
&\quad \left. - \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{z}_t^{(j)} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^{(j)},t,\Gamma(c_{\text{residual}}^{(i)}))\right)\right\|_2^2 \\
&= \frac{\beta_t^2}{(1-\bar{\alpha}_t)(1-\beta_t)}\left\|\boldsymbol{\epsilon}_\theta(\mathbf{z}_t^{(j)},t,\Gamma(c_{\text{residual}}^{(i)})) - \boldsymbol{\epsilon}_\theta(\mathbf{z}_t^{(j)},t,\varnothing)\right\|_2^2.
\end{aligned} \tag{9}$$

By absorbing the scalar coefficients $\frac{\beta_t^2}{(1-\bar{\alpha}_t)(1-\beta_t)}$ into the weighting term w_t and summing over all valid pairs (i,j) where $j \neq i$, we arrive at the final exclusion loss presented in Equation 4.

10. More Experimental Details

Algorithms. The exact implementation of ConceptPrism are summarized in Algorithm 1 and Algorithm 2. Algorithm 1 outlines the token optimization stage in Section 4.1 to 4.3, where the target and residual tokens are jointly optimized. Algorithm 2 describes the subsequent concept disentangled fine-tuning stage in Section 4.4. For computational efficiency, we approximate the exclusion loss $\mathcal{L}_{\text{excl}}$ during the token optimization. Instead of utilizing the entire set of $N-1$ negative samples, we randomly sample a subset of 3 indices $\mathcal{J} \subset \{k \mid k \neq i\}$ at each iteration.

Image Captions on Section 4.3. We employed a Vision Language Model (VLM), specifically Gemini 2.5 Flash, to generate descriptive captions for each reference image. These captions serve as the initialization for residual tokens (Section 4.3). The specific prompt used is provided below. Note that this initialization captures the overall scene, ensuring the initial tokens are not biased toward either the target concept or specific residuals.

Prompt for Scene Description

You are an AI that describes visual scenes in detail. Your mission is to describe a given image using 8-32 words of text.

Instructions

1. **Purpose of Description:** Describe the image in sufficient detail that the original scene can be visually **reconstructed** based **solely on the text description**.
2. **Elements to Describe:** You must include not only the main **subject** but also the core visual elements that constitute the scene, such as its key **attributes, background, composition, lighting, and style**.
3. All descriptions must be written in English and be between 8 and 32 words.

Algorithm 1 Token Optimization

Require: Reference images $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, Pre-trained Model ϵ_θ, Exclusion weight β, Total timesteps T
Ensure: Optimized tokens $\mathbf{t}_{\text{target}}, \{\mathbf{t}_{\text{residual}}^{(i)}\}_{i=1}^{N}$
Initialize $\mathbf{t}_{\text{target}}$ randomly
Initialize $\{\mathbf{t}_{\text{residual}}^{(i)}\}_{i=1}^{N}$ with caption embeddings of $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$
while not converged **do**
  Sample $i \sim \mathcal{U}(1, N)$ and $\mathbf{x}^{(i)} \sim \mathcal{X}$
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(\{1, \dots T\})$
  // Compute Reconstruction Loss
  $c^{(i)} \leftarrow$ "$\mathbf{t}_{\text{target}}$ with $\mathbf{t}_{\text{residual}}^{(i)}$"
  $\mathcal{L}_{\text{rec}} \leftarrow \|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{z}_t^{(i)}, t, \Gamma(c^{(i)}))\|_2^2$
  // Compute Exclusion Loss
  Sample $\mathcal{J} \subset \{k \mid k \neq i\}$ s.t. $|\mathcal{J}| = 3$
  $\mathcal{L}_{\text{excl}} \leftarrow 0$
  for $j \in \mathcal{J}$ **do**
    $\mathcal{L}_{\text{excl}} \leftarrow \mathcal{L}_{\text{excl}} + \|\epsilon_\theta(\mathbf{z}_t^{(j)}, t, \Gamma(\mathbf{t}_{\text{residual}}^{(i)})) - \epsilon_\theta(\mathbf{z}_t^{(j)}, t, \varnothing)\|_2^2$
  end for
  Update $\mathbf{t}_{\text{target}}, \mathbf{t}_{\text{residual}}^{(i)}$ via $\nabla(\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{excl}})$
end while

Algorithm 2 Concept Disentangled Fine-Tuning

Require: Reference images $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$, Optimized tokens $\mathbf{t}_{\text{target}}, \{\mathbf{t}_{\text{residual}}^{(i)}\}_{i=1}^{N}$, Model ϵ_θ, Total timesteps T
Ensure: Personalized Model ϵ_{θ^*}
Fix $\mathbf{t}_{\text{target}}$ and $\{\mathbf{t}_{\text{residual}}^{(i)}\}_{i=1}^{N}$
Inject LoRA parameters θ_{lora} into attention layers of ϵ_θ
while not converged **do**
  Sample batch $(i, \mathbf{x}^{(i)})$ from $\mathcal{X}$
  Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(\{1, \dots T\})$
  // Fine-tune with disentangled tokens
  $c \leftarrow$ "$\mathbf{t}_{\text{target}}$ with $\mathbf{t}_{\text{residual}}^{(i)}$"
  $\mathcal{L}_{\text{ft}} \leftarrow \|\boldsymbol{\epsilon} - \epsilon_\theta(\mathbf{z}_t^{(i)}, t, \Gamma(c))\|_2^2$
  Update θ_{lora} via $\nabla\mathcal{L}_{\text{ft}}$
end while

11. More Ablation Studies

Impact of Target Token Length. We investigated the impact of the target token length (n) on performance. A longer token sequence theoretically provides greater capacity to encode visual information. We evaluated lengths of $n \in \{1, 2, 4, 8\}$ as summarized in Table 5. The results indicate that $n = 1$ yields the optimal balance. Increasing n marginally improves Concept Fidelity (DINO) but significantly degrades Text Alignment (CLIP-T).

This trade-off suggests that a high-capacity target token captures excessive details and leads to conflicts with the text prompt. This phenomenon may stem from the transformer, which struggles to handle extended contexts, or the concept entanglement of the target token. Specifically, an over-parameterized target token risks memorizing the entire reference images including residuals, regardless of the exclusion loss applied to residual tokens. While introducing a mechanism to minimize the information overlap between target and residual tokens remains a promising direction for future work, our empirical results confirm that a single token ($n = 1$) provides sufficient capacity for effectively capturing most concepts.

n	1	2	4	8
CLIP-T	0.357	0.352	0.339	0.325
DINO	0.210	0.207	0.212	0.220

Table 5. **Impact of Target Token Length.** n denotes the length of the target token. We adopted $n = 1$ for our main experiments.

Impact of Exclusion Loss Weight. We analyzed the impact of the exclusion loss weight β on model performance. (Table 6). The results show that introducing $\mathcal{L}_{excl}$ significantly improves Concept Fidelity (DINO) compared to the baseline without exclusion loss ($\beta = 0$). This confirms that our exclusion objective effectively prevents residual tokens from absorbing the target concept.

However, an excessively large β leads to a degradation in Text Alignment (CLIP-T). A strong penalty forces residual tokens to discard necessary image-specific details and converge towards a null condition. Consequently, the target token inadvertently captures this residual information to minimize the reconstruction loss. Our experiments suggest that $\beta = 0.05$ yields the optimal trade-off.

β	0	0.01	0.05	0.1	0.5
CLIP-T	0.358	0.356	0.354	0.346	0.347
DINO	0.183	0.198	0.207	0.209	0.206

Table 6. **Impact of Exclusion Loss Weight.** We adopted the exclusion loss weight $\beta = 0.05$ for our main experiments.

12. More Comparisons with Prior Works

VLM Evaluations. Standard quantitative metrics often diverge from human perception. To assess practical performance, we conducted an evaluation using a VLM. Specifically, we employed GPT-4o on 24 random concept-prompt pairs from DreamBench. The VLM evaluated images based on three criteria: (1) *Concept Fidelity*, (2) *Text Alignment*, and (3) *Overall Preference*, providing a binary decision (satisfy/unsatisfy) for each. Figure 9 summarizes the satisfaction rates. Our method consistently achieved the highest scores across all criteria, demonstrating the effectiveness of our disentanglement framework.

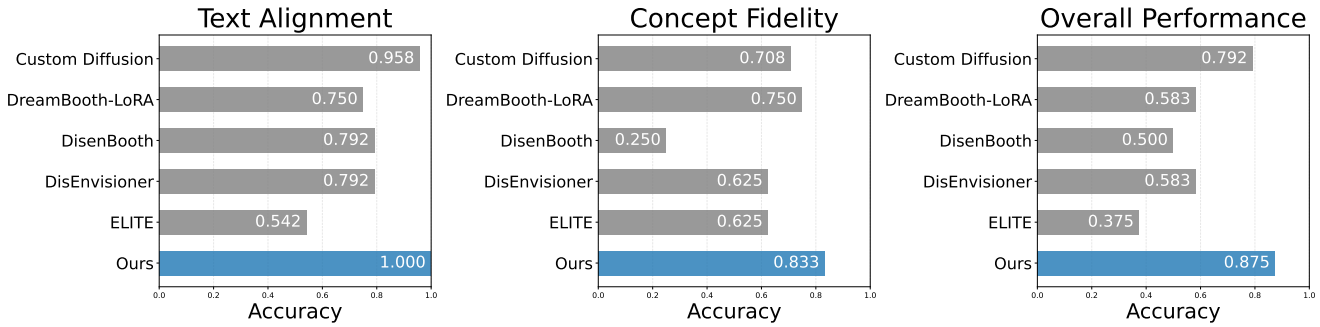

Figure 9. **VLM Evaluations.** We prompted GPT-4o to assess generated images on three criteria: Concept Fidelity, Text Alignment, and Overall Preference. The charts display the satisfaction rates for each method. Our method consistently outperforms baselines across all metrics, validating the effectiveness of our disentanglement framework.

The specific prompts used for the evaluation are detailed below.

Evaluation Prompts

- **Text Alignment:**

  "You are an impartial and unambiguous judge who must evaluate the quality of the given picture. You must assign either 1 (satisfies the condition) or 0 (does not). The condition is: 'Does the picture reflect the following text prompt? - Text Prompt: {text_prompt}' Please output a single digit (1 or 0)."

- **Concept Fidelity:**

  "You are an impartial and unambiguous judge who must evaluate the quality of the given picture. You must assign either 1 (satisfies the condition) or 0 (does not). The condition is: 'Does the picture faithfully follow the visual details of the source images?' Please output a single digit (1 or 0)."

- **Overall Preference:**

  "You are an impartial and unambiguous judge who must evaluate the quality of the given picture. You must assign either 1 (satisfies the condition) or 0 (does not). The condition is: 'Considering both the text prompt and the source images, does the given picture follow both the text prompt and source visual details?' Please output a single digit (1 or 0)."

Comparisons with Recent Methods on DiT Backbone. Recent text-to-image generative models rely on Diffusion Transformer (DiT) architectures to achieve high performance [15, 32]. ConceptPrism is universally applicable to various generative model architectures as it focuses on token optimization without requiring additional networks. Figure 10 and Table 7 compare our method with personalization techniques specialized for DiT structures using a 4-bit quantized FLUX.1 model. Our method outperforms these specialized baselines by achieving balanced, high performance across concept fidelity and text alignment. Such results demonstrate our superior disentanglement capabilities in capturing shared concepts regardless of the backbone architecture. Further examples of generations on the DiT backbone are presented in Figure 14.

Table 7. **Quantitative Evaluations on DiT Backbone.** ConceptPrism achieves balanced, high performance compared to personalization techniques specialized for DiT architectures, which demonstrates the practical utility of our method.

Method	CLIP-T ↑	DINO ↑
SynCD	0.300	**0.269**
OminiCtrl	<u>0.346</u>	0.174
Ours	**0.362**	<u>0.213</u>

Table 8. **Time Comparison with Encoder-based Methods.** We measure the training and inference time for personalized T2I generation using a 4-bit quantized FLUX.1 model. Our method incurs overhead compared to encoder-based methods due to the requirement of concept-wise training. However, it enables lighter inference by removing the need for reference information after training is complete.

Method	Inference Time (s)	Training Time (s)
Ours	18.186	507.431
SynCD	38.366	-
OminiCtrl	21.093	-

Figure 10. **Comparison with recent methods using a DiT backbone.** Being architecture-agnostic, ConceptPrism is effectively applicable to the latest DiT-based generative models. Our disentanglement strategy enables a deep understanding of the target concept, excelling in handling prompts that require attribute variations of the target.

Comparisons with Encoder-based Methods. Comparisons with encoder-based personalization methods [15, 32] further verify the practical efficiency of ConceptPrism. Optimization-based approaches, including our method, incur overhead by requiring concept-wise training. As shown in Table 8, our method takes approximately 8 minutes to train a personalized concept on a 4-bit quantized FLUX.1 model using a single H100 GPU. This duration remains within a practical range for applications involving repeated inference. In return, our method eliminates the need for reference images during the inference stage. Encoder-based methods must inject reference information into the input context for every generation, leading to high computational costs. Furthermore, these methods often suffer from performance degradation as the context length increases with multiple reference images.

13. Additional Experiments

Multi-concept Composition. Our method is effectively applicable to multi-concept composition. We learn each personalized concept independently according to our protocol. These concepts are then combined through LoRA merge, requiring no joint training. Our residual tokens absorb irrelevant information, enabling target tokens to retain only essential features. Such well-disentangled tokens can be merged without interference. Including these target tokens in the text condition allows for the composition of multiple concepts in diverse forms. Figure 11 illustrates successful compositions between subjects and styles, utilizing FLUX.1 as the backbone model.

More Flexible Modifications on Learned Concepts. Most personalization methods bind the learned concept to a specific coarse class noun (e.g., "dog" or "book") during the optimization process. In contrast, ConceptPrism extracts the target concept without such semantic priors. This independence allows for fundamental semantic shifts during generation. As shown in Figure 13, our model can transfer learned visual attributes onto entirely different object categories (e.g., transforming a specific book design into a box). This capability demonstrates that our token captures intrinsic visual structures rather than being tied to a linguistic class. We conducted these experiments using the CustomConcept101 [14] dataset.

More Results on Abstract Concepts. We present further examples demonstrating our method's capability to capture abstract concepts that are difficult to describe verbally. We utilized datasets from Freepik [1] and ActionBench [11] for these experiments. The results in Figure 12 demonstrate that our learned token effectively captures abstract semantics, color palettes, and artistic styles from the given images.

Figure 11. **Multi Concept Composition.** Each personalized concept is learned independently following our protocol and merged via LoRA merge to generate multiple concepts simultaneously. Results are based on the FLUX.1 backbone model.

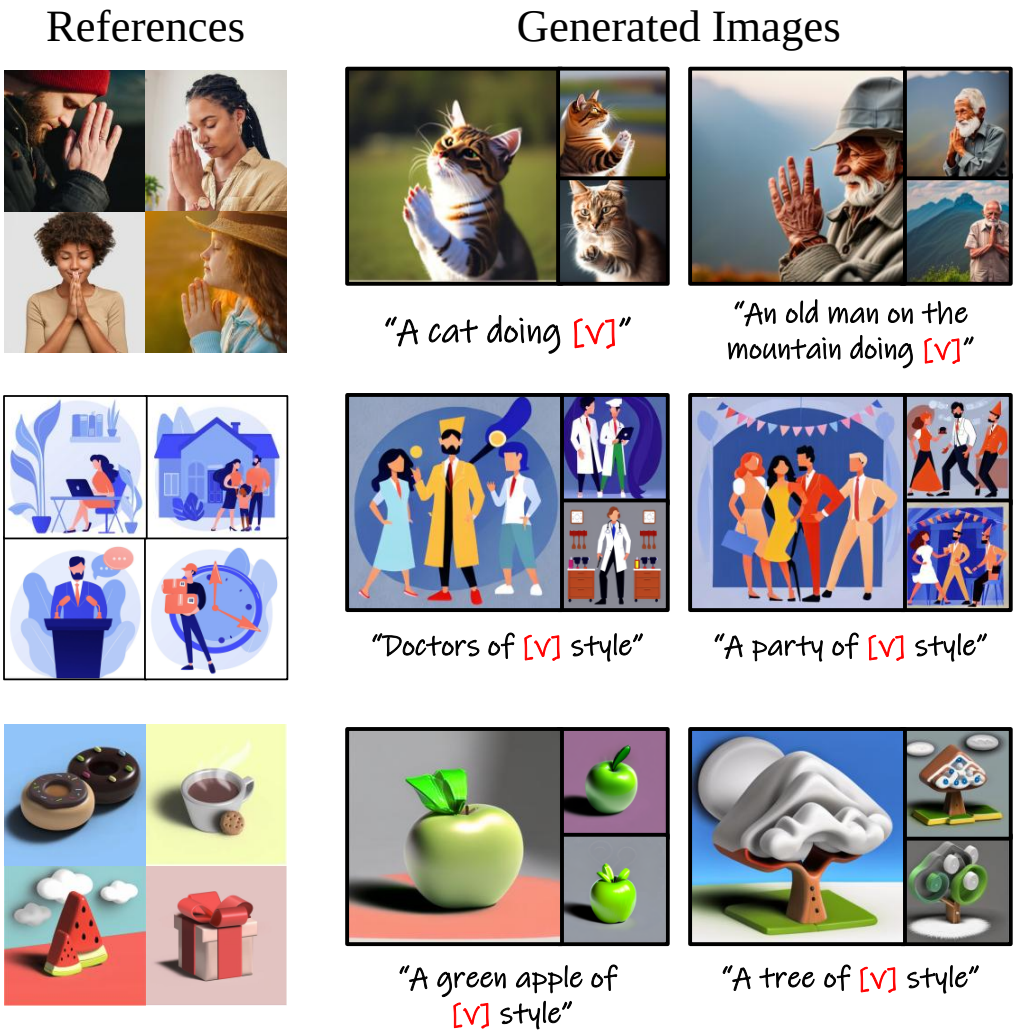

Figure 12. **Additional results on abstract concepts.** Our method effectively disentangles and learns abstract concepts from various datasets, including styles and actions.

Figure 13. **Flexible modifications on learned concepts.** Since our method learns without class noun guidance, the learned token can be flexibly applied to different subject classes. To generate these images, we used prompts specifying a new subject class, such as “A photo of [target] box” or “A photo of [target] wolf”.

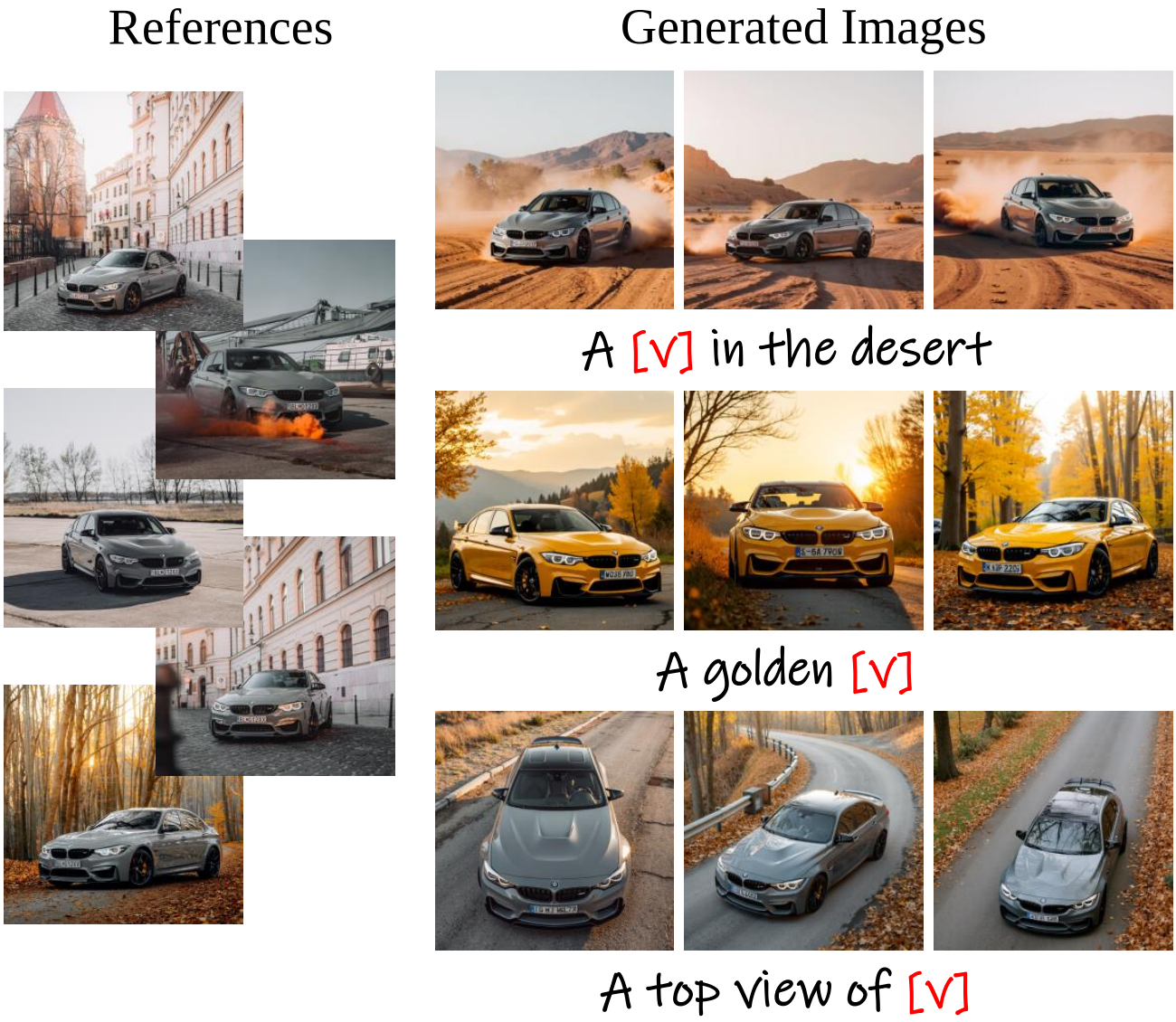

Figure 14. **Application on FLUX.1.** Since our framework operates on the text embedding space, it is agnostic to the backbone architecture of the diffusion model. We demonstrate this broad applicability by successfully generating personalized images using the FLUX.1-dev model.