Diffusion Meta-Prompting and Steering for Generalizable Foundation Model Adaptation

Deepak Sridhar[1] **Yi Li**[2] **Kartikeya Bhardwaj**[2] **Shuangjun Liu**[2] **Taotao Jing**[2]

Yuan Li[2] **Shuai Zhang**[2] **Jiancheng Lyu**[2] **Dashan Gao**[2] **Nuno Vasconcelos**[1]

[1]**University of California, San Diego** [2]**Qualcomm AI Research***

desridha@ucsd.edu, nvasconcelos@ucsd.edu

Abstract

Prompt learning is a popular method for adapting foundation models, but learned prompts are typically task-specific and fail to generalize to new classes, domains, or compositions of tasks. In this paper, we introduce a **Diffusion Meta-Prompt (DMP)** model , a framework that models the distribution of learned prompts using diffusion models. Given a repository of previously learned prompts, DMP is trained and sampled without access to the original task examples or task losses, and synthesizes new prompts conditioned on natural language task descriptions. To improve the sampling stability, we introduce a test-time steering strategy for DMP, which uses the best training-selected prompt in the repository as a latent anchor during diffusion sampling, without retraining the DMP or accessing test classes. DMP improves generalization across classification, retrieval and text-to-image generation tasks, supports concept composition and negative prompting without explicit training. It reduces storage and inference costs by over 90% compared to prompt retrieval methods. For composite classification, DMP achieves upto **2.0%** average gain over prior meta-learning methods across 55 pairs of datasets with gains as high as **8.5%** on specific pairs such as Eurosat and Flowers. DMP also enhances cross-task generalization with $\sim$**2-9%** improvement for hierarchical classification task. We further provide a theoretical guarantee bounding the expected task loss of prompts sampled from a DMP. Code is available here: DMP

1 Introduction

Foundation models [Rombach et al., 2022, Radford et al., 2021] generalize to diverse tasks owing to their scale, and can be customized to downstream tasks with parameter-efficient methods [Hu et al., 2022, Zhang and Agrawala, 2023] such as prompt learning (PL) [Zhou et al., 2022b]. In vision-language models, prompts are learned in the visual [Jia et al., 2022], textual [Zhou et al., 2022a], or both spaces [Roy and Etemad, 2024, Hao et al., 2025] to improve fine-grained [Helber et al., 2018] and taxonomic classification [Wu et al., 2024], class discrimination [Zhou et al., 2022b,a], and robustness to domain shift [Ge et al., 2022]. In generative models, prompts customize synthesis to specific concepts, people, or objects [Gal et al., 2022, Ruiz et al., 2023], or control the strength of attributes such as age, emotion, or style [Sridhar and Vasconcelos, 2024].

Despite their power, learned prompts have key limitations (Figure 1, left). They are inherently task-specific, requiring task-specific data and losses: classification prompts generalize poorly beyond the few-shot base classes and must be retuned for each label set, while generative prompts for attributes such as age or smiling must be learned separately per entity or concept. Since different applications could share the same prompts, practical systems rely on large repositories of prompts or adaptation weights, such as HuggingFace or Civitai [Civitai, 2023], where prompts must be stored, searched, and manually composed [Luo et al., 2024], limiting reuse.

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

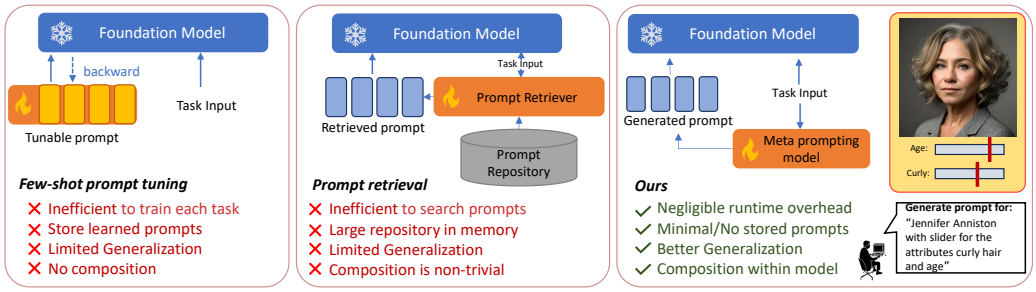

Figure 1: **DMP versus existing approaches**: DMP models provide a natural language interface for the adaptation of foundation models, while eliminating the complexities of large prompt repositories, offering better generalization and composition with a minimal runtime overhead due to the feasibility of using small DMP models and fast sampling.

Table 1: **Conceptual differences between DMP and prior meta-learning methods.**

Method	Task Data/ Loss Free	Supports Multiple Tasks	Learns across Tasks	Core Idea
Bayesian Prompt Learning (BPL)	✗	✗	✗	Bayesian uncertainty modeling over task-specific prompts.
Prompt Learning via Meta-Regularization (ProMetaR)	✗	✓	✓	Meta-regularization to improve prompt generalization across tasks; requires training data.
Gradient-Regulated Meta-Prompt (GRAM)	✗	✓	✗	Meta-learned prompt init + gradient regulator for few-shot cross-domain generalization.
PRewrite (Prompt Rewriting w/ RL)	✗	✓	✗	LLM-based prompt rewriter trained with RL to improve downstream task performance.
DMP (Ours)	✓	✓	✓	**Learns a prompt distribution; one-shot guided sampling, composition, negative prompts.**

In this work, we propose a different perspective: instead of treating prompts as isolated optimization outcomes, they should be considered as samples from a structured distribution. Across tasks and domains, learned prompts exhibit regularities that admit smooth interpolation, and support meaningful composition and negation. Capturing this structure would enable the reuse of prior prompt learning effort and improve generalization to new tasks. Rather than learning one set of prompts at a time, there is a need for meta-prompting techniques capable of learning to generate prompts across many tasks and models. However, to be scalable, the learning should be feasible *without revisiting the original task data or losses*. Instead, one would like to rely on the large prompt repositories that are already available, whose prompts were themselves obtained by standard prompt optimization on their respective source tasks. This rules out classical meta-learning techniques [Finn et al., 2017, Nichol et al., 2018, Li et al., 2017, Andrychowicz et al., 2016], based on a double optimization loop, where a prompt generator is trained jointly with the prompted models across several tasks.

To accomplish this goal we propose an approach, *Diffusion Meta-Prompting (DMP)*, where a generative model is learned over prompt embeddings, using diffusion models. A DMP is trained on a repository of previously learned prompts, and learns to synthesize new prompts conditioned on natural language task descriptions. As illustrated on the right of Figure 1, at inference, prompts are simply sampled with the DMP model, enabling one-shot adaptation of foundation models to unseen classes, composite tasks, and novel concept combinations. Table 1 compares DMP against prior meta-learning approaches, showing its uniqueness in terms of not requiring access to the original task data or losses once a prompt repository is available, while learning joint prompt structure over tasks, and requiring a minimal amount of prompt storage. The combination of these properties makes DMP much more scalable in the number of tasks and capable of prompt interpolation and generalization.

Although the ability of diffusion models to synthesize prompts is not surprising, different diffusion trajectories can produce prompts with different downstream generalization behavior. To mitigate these effects, we introduce a simple *test-time steering* mechanism, which allows DMP to reuse the prompt repository at inference time without any additional training. For each dataset, we select the prompt seed with the highest training accuracy, and use this as an anchor during DDIM sampling. At each denoising step, the predicted clean prompt is interpolated with this anchor. This biases the sampling trajectory toward a known low-risk region of the learned prompt manifold while still allowing the diffusion model to synthesize new prompts conditioned on the task specification. Importantly, the anchor is selected once, using the source-task training performance already associated with the repository, and does not use target-task examples or novel test classes. This allows task generalization.

As shown in the rightmost panel of Figure 1, DMPs eliminate the complexities of dealing with prompt repositories, offering better generalization and compositionality with a minimal runtime overhead. For example, given a subject name ("Jennifer Aniston") and attributes ("hair" and "age"), a DMP can generate a set of prompts to personalize a foundation diffusion model like Stable Diffusion XL (SDXL) to the generation of images with this combination of concepts.

We choose diffusion as the generative model for prompts since it offers better generalization than alternatives like autoregressive methods [Radford et al., 2019] (shown later in Table 19, Appendix A.6.1). This is demonstrated in two ways. First, we show that *DMPs learn to produce a range of sophisticated prompt operations*, such as concept composition, guided sampling, novel subject generation, editing and negative prompting *without explicit training*. Second, we show that *DMPs can learn multiple tasks,* by showing that a single model can be trained to produce prompts for both subject personalization and slider attributes. This replaces the combinatorial complexities of searching separate prompt repositories for the two (and potentially more) tasks into a single DMP model. DMPs are also shown applicable to a diversity of fundamentally different tasks, ranging from image synthesis to composite classification. For the latter, we show that learning the distribution of prompts across datasets and label sets generalizes to unseen classes better than prior meta-learning approaches and the standard approach of individual prompt learning per dataset. The fact that this happens *even though the DMP model is trained on the individual prompts of the baseline approach* shows that there is structure in prompt space, which DMPs learn for improved generalization.

Beyond prompt embeddings, the same principle can be applied to foundation-model representations themselves. We therefore evaluate DMP in a sparse multi-view retrieval setting, where a query view of an object must retrieve the remaining views of the same object from a gallery in which many views are missing. In this setting, DMP is trained on objects disjoint from the gallery and synthesizes embeddings for missing views. Retrieval is then performed with an aggregated gallery containing both observed and synthesized embeddings. We demonstrate that DMP can also model such structured distributions beyond learned prompts, and can improve retrieval for a different representation space and downstream objective without accessing the gallery objects during training (Table 7).

Overall, the paper makes the following key contributions

- We introduce *Diffusion Meta-Prompting*, a meta-learning framework that treats learned prompts as samples from a shared distribution, which is learned with diffusion models. Unlike prompt learning, retrieval, or refinement methods, DMP is trained once on a repository of previously learned prompts and synthesizes new task-specific prompts at inference **without revisiting the original task data or task losses, and without additional prompt optimization.**
- We introduce a test-time steering mechanism at inference without any additional training that produces prompts with better generalization (upto **2.0%**) over standard sampling.
- We provide a theoretical guarantee bounding the expected task loss of prompts sampled from a DMP in terms of the empirical loss of the prompt repository and the denoising error of the diffusion model.
- Through extensive experiments, we show that DMP improves generalization in the most challenging regimes where prompt learning typically fails, including cross-task, and composite classification settings (**up to 2.0% accuracy gains**), while also **reducing storage and inference costs by over 90%** compared to prompt retrieval approaches.
- We also demonstrate that DMP is representation-agnostic by applying it to sparse multi-view retrieval experiment which improves retrieval by **+3.44 mAP** over direct DINOv2 retrieval.

2 Related work

Foundation models. We focus on two classes of foundation models: vision-language representation models [Radford et al., 2021] and text-to-image (T2I) generation models [Rombach et al., 2022]. Contrastively trained vision-language models, such as CLIP, are widely used for open-set classification, detection, and segmentation, with prompting techniques enabling adaptation to fine-grained tasks [Zhou et al., 2022b]. In T2I models, personalization methods like Textual Inversion [Gal et al., 2022] allow generation of images of custom concepts using only a few examples. Prompt Sliders [Sridhar and Vasconcelos, 2024] and [Baumann et al., 2024] learn concepts or attributes in textual space, either globally or locally. In this work, we propose to unify the prompt learning for foundation models by training a diffusion model to synthesize the prompts conditioned on natural language across tasks.

Prompt learning. Prompt tuning learns textual [Zhou et al., 2022a], visual [Jia et al., 2022], or multi-modal [Khattak et al., 2023, Yang et al., 2024, Li et al., 2025b,a] prompts for CLIP [Radford et al., 2021]. Textual prompt learning, pioneered by CoOp [Zhou et al., 2022b] and CoCoOp [Zhou et al., 2022a], fine-tunes a CLIP model for few-shot transfer by optimizing continuous prompt vectors in the language branch. Visual prompt tuning [Jia et al., 2022] introduces task-specific learnable prompts in the visual encoder while keeping the backbone fixed. Bayesian prompt learning [Derakhshani et al., 2023] formulated prompt learning as a variational inference problem and demonstrated its ability to generalize to unseen classes at the expense of base class accuracy. Multi-modal prompt learning [Roy and Etemad, 2024, Hao et al., 2025] optimizes prompts in both vision and language encoders to improve cross-modal alignment. Table 1 and 10 in Appendix shows how DMP differs from prior prompt-learning approaches across key axes. Unlike prior approaches, which refine prompts for a single task or single example, DMP uniquely enables *multi-task prompt learning and generation, without access to task specific data or model training, simply requiring access to a repository of pre-computed prompts*.

Meta learning. Meta- learning methods aim to amortize learning across tasks, enabling rapid adaptation to new problems [Ha et al., 2017, Hospedales et al., 2021]. These methods typically have a double-loop optimization, e,g, iterating between leaning of the prompt generator and of the downstream model. Unlike DMP, this is usually not scalable to large numbers of tasks. Recently, diffusion models have been explored as generative priors for weights and representations [Zhang et al., 2024, Du et al., 2024]. [Du et al., 2024] proposed a diffusion model for *refining* CLIP prompts for classification. It is trained on a *specific example and classification task* to improve CLIP prompts for that task. Unlike this type of refinement approach, DMP is a generalist method, trained to sample prompts across tasks and downstream functionalities.

3 Diffusion Meta-Prompting

We introduce the *Diffusion Meta-Prompting* (DMP) framework for synthesizing prompts conditioned on task descriptions. A community of users first learn a set of prompts $x(c)$ for each task c in a set $\mathcal{C}$, using standard prompt learning techniques. The prompts $x(c)$ may cover multiple tasks, e.g. SDXL image synthesis or CLIP-based image classification. The learned prompts $x(c) \in \mathbb{R}^d$ are stored in a *prompt repository* $\mathcal{R}$, together with textual descriptions $y(c)$ of the task, e.g. "SDXL prompt for tattoo synthesis". The repository prompts $x(c)$, are then used to train the DMP model. This is a conditional prompt generator $p_\theta(x \mid y)$, parameterized by a diffusion model. Since the repository prompts $x(c)$ are already learned with a downstream task loss $\mathcal{L}_{\text{task}}(S; c)$, a DMP trained with the standard MSE loss suffices to synthesize prompts of low downstream risk for task c when conditioned on $y(c)$, as shown theoretically in Proposition 1.

Training. DMPs are diffusion models [Sohl-Dickstein et al., 2015, Ho et al., 2020a] that synthesize prompts by iteratively denoising a noise seed. DMP training is based on a pair of forward and backward Markov chains. Given a task or concept c, an associated prompt $x(c)$ is retrieved from $\mathcal{R}$ and noised according to a forward process that progressively adds noise to $x_0 = x(c)$, according to

$$x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t}\, \epsilon_t, \tag{1}$$

where t is a timestep, $\alpha_t := \prod_{s=1}^{t}(1 - \beta_s)$, $\{\beta_t\}_{t=1}^{T}$ is a variance schedule, $\epsilon_t \sim N(0, \mathbf{I})$ and N is a Gaussian distribution. In the reverse process, a neural network ϵ_θ recurrently denoises x_t to recover x_0. The network $\epsilon_\theta(x_t, t)$ is a U-Net [Ronneberger et al., 2015] with self and cross-attention layers [Vaswani et al., 2017]. The latter are conditioned by a text prompt $y(c)$ that specifies the task c, e.g. "a personalization prompt for Jennifer Anniston", in the example of Figure 1. A text embedding τ_θ maps $y(c)$ into a conditioning vector $\tau_\theta(y)$, where we omit the argument c for brevity. The denoising network $\epsilon_\theta(x_t, \tau_\theta(y), t)$ is trained to predict noise ϵ_t, by minimizing the risk

$$\mathcal{L}_{\text{denoise}}(\theta) = \mathbb{E}_{t, x_0, y, \epsilon}\left[\|\epsilon_t - \epsilon_\theta(x_t, \tau_\theta(y), t)\|^2\right]. \tag{2}$$

In our implementation, the DMP model is trained with classifier-free guidance, where an empty text or null prompt is used 20% of the time, to allow for better guidance during sampling. After training, a user simply specifies the task text prompt $y(c)$. The diffusion model samples a prompt $x(c)$ with

$$x_{t-1} = \sqrt{\alpha_{t-1}}\hat{x}_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2}\, \epsilon_\theta(x_t, \tau_\theta(y), t) + \sigma_t \epsilon_t \tag{3}$$

where $\hat{x}_0 = (x_t - \sqrt{1-\alpha_t}\,\epsilon_\theta(x_t, \tau_\theta(y), t))/\sqrt{\alpha_t}$ is a prediction of the denoised prompt according to (1), $x_T \sim N(0, \mathbf{I})$ is a noise seed, and $x(c) = x_0$. From a meta-learning perspective, θ are meta-parameters amortizing the production of prompts across tasks. Unlike per-task optimization, DMP enables *one-shot prompt synthesis* by sampling from the learned generator.

Theoretical Guarantee. The following result provides a theoretical guarantee for DMP performance, by bounding the distance between the task loss of the prompts learned by traditional prompt learning and those sampled with a DMP. The proof is given in Appendix A.1.

Proposition 1 (DMP performance guarantee)**.** *Let θ be a DMP such that the the empirical version of the denoising loss of (2) over a repository of n i.i.d. prompts satisfies $\mathcal{L}_{\text{denoise}}(\theta) \leq \tau$ for condition y. Then with probability at least $1-\delta$,*

$$\left|\mathbb{E}_{x\sim p_\theta(x|y)}[\mathcal{L}_{\text{task}}(x)] - \mathbb{E}_{x\sim \widehat{p}_{\text{data}}(x|y)}[\mathcal{L}_{\text{task}}(x)]\right| \leq L_{\max}\sqrt{2C\,\tau} + L_{\max}\sqrt{\frac{\log(2/\delta)}{2n}} + o(1). \quad (4)$$

where $p_\theta(x \mid y)$ is probability distribution of the DMP model conditioned on y, $\hat{p}_{data}(x \mid y)$ is the empirical distribution of the data conditioned on y, $\mathcal{L}_{\text{task}}(x)$ is a task loss bounded by $[0, L_{\max}]$, and L_{max}, C are constants.

Hence, given large enough n and small enough τ, DMP is guaranteed to meet the performance of prompt learning.

Architecture. To implement the DMP model, we develop a 1D variant of the popular Stable Diffusion model, where 2D operations are replaced by their 1D counterpart (e.g., 1D-conv). Since the size of the prompt embeddings (d) is relatively small, we have found that, for most applications, the model can operate directly in the space $\mathcal{P}$ of prompts $x(c) \in \mathbb{R}^d$. However, for applications involving unusually long prompts ($d \geq 2048$), we have also trained a variational autoencoder that maps prompts from $\mathcal{P}$ into a space of lower dimensionality, for faster training and convergence. This is a 1D variant of stable diffusion autoencoder with less than 1M parameters and a latent dimension of 128. See appendix section A.10 for more details.

Concept Composition. By framing diffusion models as Energy Based Models, [Liu et al., 2022b] showed that it is possible to compose multiple tasks by conjunction or negation. Given a diffusion model $\epsilon_\theta(x_t, t)$, n tasks c_i are combined by implementing the denoising chain with

$$\hat{\epsilon}(x_t, t) = \epsilon_\theta(x_t, t) + \sum_{i=1}^{n} \eta_i \left(\epsilon_\theta(x_t, \tau_\theta(y(c_i)), t) - \epsilon_\theta(x_t, t)\right) \quad (5)$$

where η_i is a hyperparameter corresponding to a temperature scaling of task c_i. If the conditioning is just empty text, this reduces to classifier-free guidance. The standard implementation of task composition is to run the *downstream* diffusion model n times (once per task) and average noise predictions with (5) [Liu et al., 2022b]. This is significantly more complex than performing the task composition of (5) in the (much less complex) DMP, which enables the sampling of *single* prompt x_0 for the downstream model that combine *all* tasks $c_1, \ldots, c_n$.

Test-Time Steering. While DMP can synthesize prompts in one shot, different samples from $p_\theta(x \mid y(i))$ may lie in prompt regions with different downstream generalization behavior. To address this, we introduce a simple test-time steering mechanism that biases diffusion sampling toward a high-performing prompt already available in the repository, without retraining the DMP model.

For each task dataset i, let $\{x_{i,m}\}_{m=1}^{M}$ denote the prompts obtained from M different prompt-learning initializations. We evaluate prompt x, by measuring the accuracy $\text{Acc}_{\text{train}}(x)$ of the downstream model prompted with x on the training split, and select the best

$$x_i^\star = \arg\max_m \ \text{Acc}_{\text{train}}(x_{i,m}). \quad (6)$$

This prompt is then used as an anchor for DMP sampling by replacing, in eq. (3), $\hat{x}_0$ with an estimate of the denoised prompt steered by $x_i^\star$

$$\tilde{x}_0(x_t, y(i), t) = (1-\lambda_t)\hat{x}_0(x_t, y(i), t) + \lambda_t x_i^\star, \quad (7)$$

where $\lambda_t \in [0, 1]$ controls the steering strength. In practice, we use a constant value $\lambda_t = \lambda$ for all denoising steps. The DDIM transition is then computed with $\tilde{x}_0$ in place of $\hat{x}_0$ in (3) with $\sigma_t = 0$.

This procedure can be interpreted as imposing a test-time prior over the prompt manifold. The diffusion model still conditions on the dataset text $y(i)$ and samples from a random noise seed, but the synthesized prompt is gently pulled toward a known low-risk region of the repository. Note that the generated prompt is not a copy of $x_i^\star$. Instead, $x_i^\star$ acts as an anchor that stabilizes sampling around a prompt basin of empirically strong performance on the training data.

For composite tasks, e.g. classification problems where the conditioning string contains class names from K datasets of anchors $\{x_{i_j}^\star\}_{j=1}^K$, we use the average anchor $x_{\text{comp}}^\star = \frac{1}{K}\sum_{j=1}^K x_{i_j}^\star$, as $x_i^\star$ in Eq. 7. Importantly, the prompt anchor is selected only from training-set performance and does not require access to novel test data. The method is therefore a lightweight test-time scaling strategy: it reuses the repository to identify a reliable anchor and improves the stability of DMP sampling while preserving the ability to synthesize new prompts conditioned on task specifications.

4 Experiments

Implementation Details. We conduct all experiments on 24GB (NVIDIA-A10 or 3090-RTX) GPUs using pytorch. DMP models are trained with the standard hyperparameters of [Rombach et al., 2022], a learning rate of $1e^{-6}$, and batch size of 320, for $2,000$ epochs. This takes about a day to train on 4 GPUs for 3,000 SDXL personalization prompts. For classification prompts, the DMP autoencoder and diffusion is trained for $10,000$ epochs, which takes about 10 hours[2]. DMP uses 50 DDIM timesteps to sample one prompt which takes about 1 sec and it can be sped up with faster sampling methods [Liu et al., 2022a]. For fair comparisons, we use the codebase of respective prompt learning methods and run all our experiments under this setup. For test-time steering, we use $\lambda = 0.2$. More details on the architecture and models are in the appendix A.10.

4.1 DMP for classification

We trained a DMP model, DMPClass, for prompting a CLIP-style classifier. The goal is not to beat the SOTA in prompt based classifiers, but to show that DMP can synthesize prompts that generalize to multiple tasks, outperforming other meta-prompting approaches. We utilize CoOp prompts to build the prompt repository with imagenet and the 10 fine-grained datasets commonly used in the prompt learning literature [Zhou et al., 2022b,a]. We note that DMP is agnostic to the prompt repository and can be applied to other prompting methods. However, all methods other than CoOp use additional weights, projection matrices, or architectural enhancements that involve high dimensional parameter matrices. Modeling these parameters requires much larger diffusion models and compute than we used, which is sufficient to learn prompts. More importantly, most of the meta-prompting baselines that we compare to simply do not scale to the generation of such parameters. Appendix section A.3 discusses the application of DMP to other prompting methods, where it is only used to sample prompts, maintaining the additional parameters from the original methods. The appendix shows that DMP still achieves consistent gains over the original prompts, although these are smaller than for CoOp, which has no additional parameters.

Prompt Repository. Given a classification problem, a prompt set is trained for N epochs (where N follows the original paper settings), using a context vector of size K. The CoOp paper reports full results only for context length 16. We use the context length of 4 ($K = 4$) suggested in CoCoOp [Zhou et al., 2022a]. All prompt learning methods are trained under the Base2New setting, which assesses generalization to unseen (New) classes by only using half of dataset (Base) classes for training. DMP is trained with the prompts originally learned on the Base set. We evaluate the DMP-sampled prompts on training (Base), remaining (New), and all (All) classes. We use $M = 40$ different initializations per dataset and save the corresponding prompt embeddings to produce a training repository $\mathcal{R} \in \mathbb{R}^{11\times 40\times K\times d}$. The best performing prompt of (6) ($x_{\text{comp}}^\star$ for composite tasks) is used as anchor for DMP sampling and as the baseline CoOp prompt, against which DMP prompts are compared.

Training. Due to the high dimensionality of the prompt embeddings ($K \times d = 2048$), DMPClass models are trained with an autoencoder. To obtain the text inputs, we concatenate all the class labels $c_k, k \in \{1, \dots C_i\}$, of dataset i into a text string $y(i) = \{c_1, \dots, c_{C_i}\}$ whose text embedding $\tau(y(i))$ is used to condition the DMPClass model. This setup allows us to flexibly mix and match class names across datasets to enable straightforward compositionality (Table 3) without requiring handcrafted or semantically enriched descriptions. Once the model is trained, it suffices to sample conditioned on a string of mixed class labels from any of the datasets. Since the CLIP text encoder has a limit

[2]All datasets used in the paper were solely downloaded and evaluated by the university

Table 2: Zero-shot accuracy comparison on combined dataset classes.

Method	All	Base	New
CoOp	69.6	73.6	76.7
BPL	72.0	**77.5**	78.4
ProMetaR	71.2	75.8	77.6
DMP	**73.6**	76.0	**78.8**
Δ_{CoOp}	**+4.0**	**+2.4**	**+2.1**

Table 3: Accuracy comparison on combined EuroSAT and Flowers dataset pair.

Method	All	Base	New
CoOp (ZS)	35.5	50.2	59.4
BPL (ZS)	52.6	61.3	73.4
ProMetaR (ZS)	48.0	59.9	67.1
DMP (ZS)	**61.1**	**74.5**	**77.8**
BPL (Trained)	56.2	76.1	75.1

Table 4: Ablation of test-time steering on 55 composite dataset-pairs classification. Δ_{steer}, Δ_{CoOp} denotes the absolute accuracy gain over respective baselines.

Split	CoOp	BPL	ProMetaR	DMP w/o Steering	DMP w/ Steering	Δ_{steer}	Δ_{CoOp}
All	58.7	60.8	61.2	61.4	**63.1**	**+1.7**	**+4.4**
Base	65.1	66.3	67.4	67.3	**69.4**	**+2.1**	**+4.3**
New	65.1	68.7	68.9	68.6	**70.8**	**+2.2**	**+5.7**
Avg.	63.0	65.3	65.8	65.8	**67.8**	**+2.0**	**+4.8**

Table 5: Memory and complexity requirements for DMP against retrieval.

Method	Mem (GB) ↓	Time (s) ↓
Stylus	1.30	12.1
DMP	**0.12**	**1**
Δ	**(91%↓)**	**(92%↓)**

Table 6: Comparison of DMP-Variation with Textual Inversion (TI) for variation synthesis.

Method (SD-RV) ↓	Face-ID ↑	CLIP-T ↑	CSD ↑
TI	0.428	27.0	50.4
DMP	**0.299**	**28.4**	**61.8**
Δ	**-0.13**	**+1.4**	**+11.4**

Table 7: Sparse multi-view retrieval on MVImgNet. The gallery contains 5,447 objects, with only one to four available views per object. DMP is trained on objects disjoint from the gallery and synthesizes embeddings for missing views.

Method	NDCG	MRR	Top-1	mAP
DINOv2	91.64	94.36	91.44	86.20
DMP	**93.47**	**94.42**	**91.74**	**89.64**
Δ	**+1.83**	**+0.06**	**+0.30**	**+3.44**

of 77 tokens (around 50 words), these strings can be too long for many classes (e.g. the 1,000 Imagenet classes). To overcome this, we consider each class name independently and obtain the CLIP embedding for each class, resulting in C embeddings for C classes, which are concatenated into a vector that is used to prompt the DMPClass model. See A.6.3 for ablation on text inputs.

Composite classification. To evaluate the generalization ability of DMP, we introduce a composite classification task. We consider two settings under this task: (1) classify over the set of all classes of all datasets and (2) classify over pairs of class label sets from the fine-grained datasets we considered. For setting (2), we created 55 datasets $\mathcal{T}_i, i \in \{1, \ldots, 55\}$ containing pairs of All, Base and New classes from the original 11 datasets. These settings test out-of-domain generalization, since prompts trained on one dataset are used to classify a mix of their classes and classes from other datasets. For example, while Oxford Flowers classes require modeling flower attributes, the composition with FGVC Aircrafts requires the modeling of both flower *and* airplane attributes, on which the original prompts were never trained *simultaneously*. For each dataset $\mathcal{T}_i$, we performed classification with (1) the prompt sampled from DMP (trained on the repository of individual dataset prompts), and (2) the average of two anchor prompts $x^\star$ learned from the two datasets that compose $\mathcal{T}_i$.

For the first setting, we compare DMP against the baseline CoOp prompts as well as prior meta-prompt learning methods such as BPL [Derakhshani et al., 2023] and ProMetaR [Park et al., 2023]. Table 2 shows the results under All, Base and New Classes. For evaluation over all classes, DMP improves over CoOp by 4% and outperforms the best meta-prompting baseline (BPL) by 1.6%. Table 3 shows, that, in setting (2), the DMP gains can be more even pronounced for specific dataset class pairs, such as EuroSAT and Flowers. For this pair, DMP exceeds BPL's zero-shot baseline by **+8.5/+4.4%** and CoOp by **+25.6/18.4%** for all/new classes, respectively. We further compare DMP with BPL (Trained), which is trained on the entire dataset pair. As shown in Table 3, DMP without any task-specific training even outperforms this approach by **+2.7%** on new classes. We also note that BPL is computationally expensive, requiring approximately 10 hours on a 40GB GPU even for a single dataset pair, which limits its scalability to composite tasks as compared to DMP.

Table 4 (left) compares the performance of the DMP prompts to others, showing the average accuracy across all 55 dataset combination pairs. DMP obtains gains of **+2.0%** for Base,**+1.9%** for New, and **+1.9%** for All classes over ProMetaR, the best baseline for the pairwise setting. DMP further outperforms CoOp by **+4.3%** for Base,**+5.7%** for New, and **+4.4%** for All classes.

Effect of test-time steering. Table 4 also isolates the contribution of the test-time steering mechanism of (7). Compared with DMP sampling by the same model without steering, test-time steering consistently improves accuracy across all evaluation splits. The gain is **+1.7** points on the All split, **+2.1** on the Base split, and **+2.2** on the New split, yielding an average improvement of **+2.0** points. This suggests that steering the DDIM sampling trajectory toward a training-selected high-performing prompt is a useful test-time regularization towards high-performing regions of the prompt manifold. Importantly, the steering target is selected using only training-split performance and does not require access to base or novel test classes, so the improvement is obtained without additional prompt training or test-label supervision. Ablation on the steering strength is shown in Appendix A.3

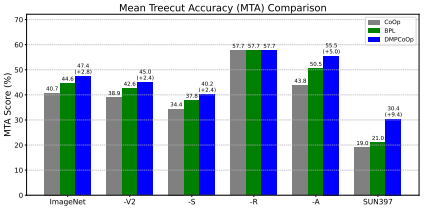

Figure 2: **Taxonomic classification generalization**: Comparison of DMP against CoOp and BPL prompts for hierarchical classification. DMP prompts generalize better than BPL prompts with $\approx$ **2-9%** accuracy gains.



Figure 3: **Generalization of DMPVariation model**. Images generated using prompts synthesized by TI (top) and DMPVariation (bottom) for various downstream model text prompts (shown on top). DMPVariation prompts are more robust than TI prompts, which impair the ability of the downstream model to generalize. See Fig. 16 for additional results.

Taxonomic. We next consider the more challenging cross-task generalization setting of taxonomic classification [Wu et al., 2024]. This tests the ability of the downstream models to classify images with respect to different class subsets in a class hierarchy, using the metric of Mean Treecut Accuracy (MTA) over 25 treecuts. Since the training classes are the leaves of taxonomy, hierarchical classification requires generalization from finer to coarser classes. For example, while 'cats' and 'dolphins' are two very different classes, they both belong to 'mammals'. [Wu et al., 2024] showed that standard prompt learning methods perform poorly on this task, where effective prompt learning requires hierarchical training and sampling of label sets across the entire taxonomy.

Figure 2 shows the MTA performance of CoOp, BPL, and DMP prompts. Since only ImageNet and SUN provide labels with respect to the entire class taxonomy, we used these two datasets, plus the OOD variants of ImageNet on this experiment. Table 21 presents the full results of the experiment. DMP prompts obtain an average gain between **2.4** and **9.4%** over BPL and a larger gain between **5.8** and **11.7%** over CoOp, even though *no prompts are ever trained for hierarchical classification*. These results are consistent with the previous findings and show that DMP excels as the generalization challenge increases, learning to produce prompts that are much more robust than those of the CoOp prompt learning method in which it was trained.

4.2 DMPVariation

Prompt learning is an approach to personalize diffusion models to the synthesis of images of specific concepts. DMPVariation is a DMP model that synthesizes personalization prompts learned via textual inversion (TI) [Gal et al., 2022], to prompt a diffusion model to synthesize variations of a subject.

Prompt Repository. To produce a prompt repository $\mathcal{R}$, we split the CelebA dataset into unique identities using the groundtruth labels, and use 3,000 of these as concepts c, for which we train personalized prompts $x(c)$ with Textual Inversion using 1,000 steps of gradient descent. To obtain prompts for variations, we note that earlier optimization steps do not fully encode face attributes, as compared to steps later in the optimization. We save the prompts $x(c)$ from 40 gradient steps (between $200 - 400$ steps in intervals of 5), per identity. This produces a repository $\mathcal{R}$ of 40 prompt variations per identity to obtain a training repository $\mathcal{R} \in \mathbb{R}^{3000\times 40\times d}$. The text string $y(c)$ associated with task c is "a photo of <identity-c>".

Training. The DMPVariation model was trained directly in prompt space $\mathcal{P}$, i.e. without autoencoder. Each personalized prompt produced by TI is used as conditional input to DMPVariation, by concatenating it with the noise vector. During training, the model learns to denoise the 40 different variations of the conditional prompt input. Hence, sampling from the DMP produces a diversity of prompts that induce the downstream model to synthesize variations of the subject face (see Fig. 15 for an example). At inference, prompts that induce variations of novel subjects can be sampled with DMP by simply conditioning on the TI prompt for the new subject.

DMPVariation demonstrates the effectiveness of DMP for generative tasks, namely the task of synthesizing images with variations of a subject or concept. To evaluate prompt generalization in this context, we compare DMPVariation prompts to the prompts originally learned by TI on the CelebA dataset. Performance is measured with Face-ID (VGGFace2), CLIP-Score across 28 diverse prompts (listed on Table 25), and the HPSv2 metric on 100 prompt concepts downloaded from the Huggingface repository. Note that none of these concepts are on the CelebA dataset. Table 6 shows that DMPVariation achieves lower Face-ID similarity, indicating diverse yet semantically similar prompts, and improves CLIP-Text similarity by **1.4%** and the CSD style metric by **+11.4%**, demonstrating better generalization to novel prompts compared to TI embeddings. However, as is

typical for generative modeling, these metrics do not capture the substantial qualitative gap between the two approaches. While TI prompts produce images that *do not fully comply* with the instruction, DMP prompts consistently produce images that satisfy the instructions.

Figure 3 illustrates this gap by presenting images synthesized with DMPVariation and Textual Inversion (TI) prompts. The downstream model, personalized to a subject with prompts produced by TI or DMPVariation, is asked to generate an image of the subject in a new context, specified as a text prompt atop each image. The personalization prompts produced by DMPVariation demonstrate greater robustness and generalization, effectively adapting to different contexts. This is unlike TI prompts, which tend to overfit to the subject, leading to poor generalization across contexts. In result, TI prompts fail to generate suitable images of the dog for all contexts other than *"in front of a house"*. The figure also shows that DMPVariation produces successful variation prompts for general objects like dogs *despite being trained only on CelebA-face identities*. Fig. 13 presents images for other objects, such as bird and statue. Fig. 16 shows additional results where TI completely fails as compared to DMPVariation.

DMP for sparse multi-view retrieval. We next evaluate whether DMP can generalize beyond prompt synthesis by applying it to a representation-level retrieval task. We consider sparse multi-view retrieval, where the query is one view of an object and the goal is to retrieve all other views of the same object from a gallery. Unlike densely populated multi-view retrieval settings, we assume that gallery coverage is incomplete: each object has only between one and four available gallery views. This creates a missing-view problem, where direct retrieval from the observed gallery may fail when the query view differs substantially from the available views of the same object.

We construct this task from a subset of MVImgNet [Yu et al., 2023] containing 5,447 gallery objects. The baseline is a retrieval operation using DINOv2 embeddings of the available gallery views. DMP is trained to synthesize missing views on a set of objects disjoint from those in the gallery, to guarantee that *all objects are novel at test time*. DMP-based retrieval complements the observed DINOv2 embeddings with DMP-synthesized embeddings of the missing views of each object. Retrieval is then performed against an aggregated gallery containing both the observed and DMP-synthesized embeddings.

Table 7 shows that DMP consistently improves retrieval performance over direct DINOv2 embeddings. The largest gain is observed for mAP ($+3.44$), indicating that synthesized missing-view embeddings improve the ranking of all relevant views, rather than only the top prediction. DMP also improves NDCG by $+1.83$ and Top-1 accuracy by $+0.30$, while maintaining comparable MRR. These results show that DMP learns useful structure in the space of foundation-model embeddings and can synthesize representations that compensate for sparse gallery coverage. Importantly, this experiment demonstrates that the benefits of DMP are not limited to learned prompt spaces: the same meta-generative framework can provide gains for different representations and downstream tasks.

Storage and Runtime Efficiency. One of the benefits of the DMP framework is its high efficiency. DMP eliminates the need to manage a repository of prompts and associated *metadata*, such as text descriptions. This contrasts with methods like Stylus [Luo et al., 2024], that automatically search, retrieve, and compose LoRAs from a repository. Table 5 compares the storage and processing requirements of DMP and Stylus, when used for the tasks that we consider in this work. DMP is significantly more efficient, reducing storage needs by **91%** and improving inference speed by **92%**.

5 Ablation Studies

Robustness to anchor perturbation. To assess whether the gains of test-time steering depend on the exact anchor $x_i^\star$ of (6), we perturb it in random directions at increasing L_2 distances before using it in (7). Table 8 shows that accuracy remains essentially unchanged under moderate to large perturbations, varying by at most 0.27 points across splits. This suggests that steering does not rely on a single privileged prompt, but instead guides sampling toward a broad high-performing neighborhood of the prompt manifold. Since the anchor is fixed throughout sampling, however, these results alone do not distinguish task-specific guidance from the stabilizing effect of a fixed target.

To separate the two, we replace the training-selected anchor with a fixed, randomly selected training prompt. Accuracy decreases from 63.1/69.4/70.8 to 62.4/69.1/70.2 on the All/Base/New splits, i.e., by 0.7/0.3/0.6 points. Since this alternative anchor also remains fixed during sampling, the degradation cannot be attributed to a loss of stabilization alone, indicating that the training-selected anchor provides task-relevant guidance beyond the text condition.

Table 8: Robustness of test-time steering to perturbations of the anchor prompt. Results are averaged over dataset pairs, and over repeated noise samples within each pair. Δ is relative to the unperturbed anchor. *Fixed anchor*: a randomly selected training prompt is used as the anchor. *Resampled*: the perturbation direction is resampled at every denoising step.

Setting		All		Base		New	
		Acc.	Δ	Acc.	Δ	Acc.	Δ
L_2	0.00	**63.10**	–	**69.40**	–	70.80	–
L_2	0.25	63.01	-0.09	69.13	-0.27	70.77	-0.03
L_2	0.50	62.94	-0.16	69.13	-0.27	70.73	-0.07
L_2	1.00	63.09	-0.01	69.28	-0.12	**70.90**	$+0.10$
Fixed anchor		62.36	-0.74	69.12	-0.28	70.21	-0.59
Resampled		61.27	-1.83	67.47	-1.93	68.56	-2.24

Table 9: Sensitivity of DMP to the class-name condition, with and without test-time steering. A fraction of the Caltech101 class names in the condition is replaced with ImageNet class names. L_2 distances of the condition embedding and of the sampled prompt are measured with respect to the original (0%) condition.

Replaced names	**Condition** L_2	**w/o Steering**			**w/ Steering**		
		Prompt L_2	**Caltech101**	**ImageNet**	**Prompt** L_2	**Caltech101**	**ImageNet**
0%	0.0	0.000	92.0	64.7	0.000	93.0	64.9
40%	24.2	1.623	92.1	64.4	0.025	93.0	64.8
60%	29.8	1.606	92.6	64.3	0.049	93.1	64.5
80%	34.0	1.600	92.5	64.5	0.049	93.2	64.5

Finally, we resample the perturbation direction at every denoising step, which removes the consistency of the steering target across the trajectory. This leads to a larger reduction, 1.8/1.9/2.2 points below the unperturbed anchor, showing that a consistent steering target is also important. Overall, the anchor plays two complementary roles: its location in prompt space guides sampling toward a task-relevant region, and its consistency across denoising steps stabilizes the sampling trajectory.

Sensitivity to the class-name condition. We next examine how DMP responds to changes in its text condition, with and without steering. Starting from the Caltech101 class-name condition, we progressively replace 40–80% of the class names with distinct ImageNet class names. Table 9 reports the L_2 distance of the resulting condition embedding and of the sampled prompt from those of the original condition, together with the downstream accuracy. As more names are replaced, the condition embedding moves away from the original, with its L_2 distance increasing from 17.4 to 34.0.

Without steering, these changes propagate to the sampled prompts, whose L_2 distance from the original prompt reaches 1.42–1.62, and replacing 40% of the names changes ImageNet accuracy by 0.3 points. Hence, neither the text encoder nor DMP collapses related class vocabularies into the same representation or prompt. With steering, the prompt distances decrease to 0.025–0.049, and accuracy remains stable at 93.0–93.2% on Caltech101 and 64.5–64.9% on ImageNet. These results separate the two effects: DMP remains sensitive to the class-name condition, while the training-selected anchor limits how strongly this variation changes the sampled prompt and its downstream behavior, stabilizing generation within a locally high-performing region of prompt space.

See Appendix A.4 for application of DMP for personalization and image editing, Sec. A.6 for additional ablations and Sec. A.9 for a discussion on the limitations and scope for future works.

6 Conclusion

We introduced a *Diffusion Meta-Prompting* (DMP) framework that learns a distribution over prompts and enables synthesis of task-specific prompts directly from natural language. Across recognition, retrieval, and generation settings, DMP produces representations and prompts that generalize beyond the tasks, label sets, views, and concepts seen during training, while avoiding the overhead of storing and searching large repositories. Empirically, DMP yields consistent gains in the most challenging generalization regimes improving cross-task and composite classification, supports compositional control and semantic negation, and improves prompt compliance for subject variations. Finally, we provided a theoretical bound linking the expected downstream task loss of DMP-sampled prompts to repository quality and diffusion denoising error, offering a principled perspective on when meta-prompt generation succeeds.

Acknowledgements

This work was partially funded by Qualcomm Innovation Fellowship 2025, NSF awards IIS-2303153 and NAIRR-240300. We also acknowledge and thank the use of the Nautilus platform for some of the experiments.

Broader Impact

We introduce a new meta-learning framework for generating prompts for foundation models. It carries the risks associated with generative foundation models. While the proposed method offers benefits of better generalization with storage and runtime efficiency, it uses existing pretrained models which are shown to contain harmful biases that maybe elicited by the prompts. It can also be potentially misused to propagate harmful, unlawful or unethical information with the personalization of celebrities. Since, the framework is meta-learning, any harmful prompts can be identified before the image generation step where the embeddings can be inspected with nearest neighbor tokens in the text space. Additionally, advancements in image watermarking [Luo et al., 2022, Cao et al., 2025, Kohli and Gowal, 2023, Gowal et al., 2025] can help to identify generated image contents to protect against these risks.

References

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

Stefan Andreas Baumann, Felix Krause, Michael Neumayr, Nick Stracke, Vincent Tao Hu, and Björn Ommer. Continuous, Subject-Specific Attribute Control in T2I Models by Identifying Semantic Directions, 2024.

Jie Cao, Qi Li, Zelin Zhang, and Jianbing Ni. Secure and robust watermarking for ai-generated images: A comprehensive survey, 2025. URL `https://arxiv.org/abs/2510.02384`. A broad survey on watermarking for AI-generated visual content.

Civitai. https://civitai.com/, 2023. Webpage.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2 edition, 2006.

Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. *ICCV*, 2023.

Yingjun Du, Gaowen Liu, Yuzhang Shang, Yuguang Yao, Ramana Kompella, and Cees G. M. Snoek. Prompt diffusion robustifies any-modality prompt learning, 2024. URL `https://arxiv.org/abs/2410.20164`.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv*, 2022. doi: 10.48550/ARXIV.2208.01618. URL `https://arxiv.org/abs/2208.01618`.

Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *arXiv preprint arXiv:2202.06687*, 2022.

Sven Gowal, Rudy Bunel, Florian Stimberg, David Stutz, Guillermo Ortiz-Jimenez, Christina Kouridi, Mel Vecerik, Jamie Hayes, Sylvestre-Alvise Rebuffi, Paul Bernard, Chris Gamble, Miklós Z. Horváth, Fabian Kaczmarczyck, Alex Kaskasoli, Aleksandar Petrov, Ilia Shumailov, Meghana Thotakuri, Olivia Wiles, Jessica Yung, Zahra Ahmed, Victor Martin, Simon Rosen, Christopher

Savčak, Armin Senoner, Nidhi Vyas, and Pushmeet Kohli. Synthid-image: Image watermarking at internet scale, 2025. URL https://arxiv.org/abs/2510.09263. A deep learning–based system for invisibly watermarking AI-generated imagery at internet scale.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Fusheng Hao, Fengxiang He, Fuxiang Wu, Tichao Wang, Chengqun Song, and Jun Cheng. Task-aware clustering for prompting vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Introducing eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 204–207, 2018. doi: 10.1109/IGARSS.2018.8519248.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

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

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5149–5169, 2021. doi: 10.1109/TPAMI.2021.3069109.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

HuggingFace. https://huggingface.co/blog/lora-adapters-dynamic-loading, 2023. Webpage.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.

Muhammad Uzair khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15190–15200, October 2023.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Pushmeet Kohli and Sven Gowal. Identifying ai-generated images with synthid. Google DeepMind Blog, 2023. URL https://deepmind.google/blog/identifying-ai-generated-images-with-synthid. SynthID watermarking technology for AI-generated images.

Haoyang Li, Liang Wang, Chao Wang, Jing Jiang, Yan Peng, and Guodong Long. Dpc: Dual-prompt collaboration for tuning vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025a. URL https://arxiv.org/abs/2503.13443. arXiv preprint arXiv:2503.13443.

Zheng Li, Yibing Song, Ming-Ming Cheng, Xiang Li, and Jian Yang. Advancing textual prompt learning with anchored attributes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b. URL https://iccv.thecvf.com/Conferences/2025/AcceptedPapers. Accepted (Poster).

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=PlKWVd2yBkY.

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

Michael Luo, Justin Wong, Brandon Trabucco, Yanping Huang, Joseph E. Gonzalez, Zhifeng Chen, Ruslan Salakhutdinov, and Ion Stoica. Stylus: Automatic adapter selection for diffusion models, 2024.

Xiyang Luo, Michael Goebel, Elnaz Barshan, and Feng Yang. Leca: A learned approach for efficient cover-agnostic watermarking, 2022.

Colin McDiarmid. On the method of bounded differences. In Johannes Siemons, editor, *Surveys in Combinatorics*, volume 141, pages 148–188. Cambridge University Press, 1989.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Jinyoung Park, Juyeon Ko, and Hyunwoo J. Kim. Prompt learning via meta-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

A. Radford, J. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763, 2021.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019.

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.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

Shuvendu Roy and Ali Etemad. Consistency-guided prompt learning for vision-language models. In *ICLR*, 2024.

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. *CVPR*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Deepak Sridhar and Nuno Vasconcelos. Prompt sliders for fine-grained control, editing and erasing of concepts in diffusion models. In *In Proceedings of the IEEE/CVF European Conference on Computer Vision Workshops*, 2024.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer, 2008.

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

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

Tz-Ying Wu, Chih-Hui Ho, and Nuno Vasconcelos. ProTeCt: Prompt Tuning for Taxonomic Open Set Classification . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16531–16540, Los Alamitos, CA, USA, June 2024. IEEE Computer Society. doi: 10.1109/CVPR52733.2024.01564. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01564.

Lingxiao Yang, Ruyuan Zhang, Yanchen Wang, and Xiaohua Xie. Mma: Multi-modal adapter for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23826–23837. IEEE/CVF, 2024.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Mvimgnet: A large-scale dataset of multi-view images. In *CVPR*, 2023.

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

Baoquan Zhang, Chuyao Luo, Demin Yu, Huiwei Lin, Xutao Li, Yunming Ye, and Bowen Zhang. Metadiff: Meta-learning with conditional diffusion for few-shot learning, 2024. URL https://arxiv.org/abs/2307.16424.

Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *ICCV*, 2023.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022b.

A Appendix

A.1 Theoretical Analysis

Let $\mathcal{C}$ denote a distribution over tasks or concepts $c \sim \mathcal{C}$. For each concept c, we assume a textual description $y(c)$ (e.g., "a personalization prompt for Jennifer Aniston"), a downstream task with loss $\mathcal{L}_{\text{task}}(S; c)$ that evaluates the performance of a candidate prompt S in prompt space $\mathcal{P}$ when applied to a frozen foundation model F, and a repository $\mathcal{R}$ of exemplar prompts $x(c)$ obtained from existing prompt-learning techniques.

Our objective in this section is to bound the *expected downstream loss* when sampling prompts from the pretrained generator:

$$\mathcal{J}_{\text{pre}}(\theta) := \mathbb{E}_{c\sim\mathcal{C}}\Big[\mathbb{E}_{S\sim p_\theta(\cdot|y(c))}\big[\mathcal{L}_{\text{task}}(S; c)\big]\Big].$$

High-level intuition. When the trained diffusion model $p_\theta(\cdot \mid y)$ closely matches the repository/data distribution $p_{\text{data}}(\cdot \mid y)$ in distributional distance, the expected task loss under p_θ is close to the expected *repository* task loss under p_{data}. If the repository was constructed to contain useful prompts for downstream tasks (i.e., p_{data} has low expected task loss), then a small distributional discrepancy implies low expected task loss for p_θ as well. We make this statement precise with the following bounds.

A.1.1 Distributional discrepancy bound

We begin with a straightforward decomposition and use standard total-variation and Pinsker inequalities [Cover and Thomas, 2006, Tsybakov, 2008].

Proposition 1 (Distributional discrepancy bound)**.** *Let $p_{\text{data}}(\cdot \mid y)$ and $p_\theta(\cdot \mid y)$ be two distributions on prompt space $\mathcal{P}$ for a fixed condition y. Assume the task loss $\mathcal{L}_{\text{task}}(x)$ is bounded in $[0, L_{\max}]$. Then*

$$\begin{aligned}\Big|\mathbb{E}_{S\sim p_\theta}\big[\mathcal{L}_{\text{task}}(x)\big] - \mathbb{E}_{S\sim p_{\text{data}}}\big[\mathcal{L}_{\text{task}}(x)\big]\Big| \\ \leq 2L_{\max}\,\text{TV}\big(p_\theta, p_{\text{data}}\big).\end{aligned} \tag{8}$$

and by Pinsker's inequality,

$$\text{TV}(p_\theta, p_{\text{data}}) \leq \sqrt{\tfrac{1}{2}\,\text{KL}(p_{\text{data}}\|p_\theta)}. \tag{9}$$

Consequently,

$$\begin{aligned}\mathbb{E}_{S\sim p_\theta}\big[\mathcal{L}_{\text{task}}(x)\big] \leq \mathbb{E}_{S\sim p_{\text{data}}}\big[\mathcal{L}_{\text{task}}(x)\big] \\ + L_{\max}\sqrt{2\,\text{KL}(p_{\text{data}}\|p_\theta)}.\end{aligned} \tag{10}$$

Proof. Let $\ell(x) = \mathcal{L}_{\text{task}}(x)$ and denote $\Delta := \mathbb{E}_{p_\theta}[\ell] - \mathbb{E}_{p_{\text{data}}}[\ell]$. By the definition of total variation (and the fact that $0 \leq \ell \leq L_{\max}$),

$$\begin{aligned}|\Delta| &= \Big|\int \ell(x)\,(p_\theta - p_{\text{data}})(dx)\Big| \\ &\leq \int |\ell(x)|\,|p_\theta - p_{\text{data}}|(dx) \\ &\leq L_{\max}\int |p_\theta - p_{\text{data}}|(dx).\end{aligned} \tag{11}$$

By definition $\text{TV}(p_\theta, p_{\text{data}}) = \frac{1}{2}\int |p_\theta - p_{\text{data}}|$, hence (8) holds.

Pinsker's inequality (see e.g. [Cover and Thomas, 2006]) gives (9). Combining the two inequalities yields (10). □

Remarks. Inequality (10) reduces expected downstream loss under the model to two terms: the expected loss of the repository distribution (which is a function of data collection quality) and the KL divergence between the dataset distribution and the pretrained generator. The latter is controlled by how well the denoiser is trained (see next subsection).

A.1.2 Relating denoising loss to model-data KL

We now recall a standard link between the denoising objective used in DDPMs and a divergence between the model and data distributions (see, e.g., [Vincent, 2011, Song et al., 2020]). Under standard DDPM assumptions and appropriate variance schedule, minimizing the simplified denoising loss (2) is equivalent (up to constants and time discretization effects) to score matching / denoising score-matching which estimates the score function $\nabla_x \log p_{\text{data}}(x)$. A well-trained denoiser implies an accurate score estimator, which in turn implies a small KL divergence between the model and data distributions in x_0-space (the space of clean prompts). Formally, one can show:

Lemma 1 (Denoising loss controls KL (informal))**.** *Under the standard DDPM/score-matching correspondence and mild regularity conditions, if the denoising risk satisfies*

$$\mathcal{L}_{\text{denoise}}(\theta) \le \varepsilon,$$

then the KL divergence between $p_{\text{data}}(x_0 \mid y)$ and $p_\theta(x_0 \mid y)$ admits the bound

$$\mathrm{KL}\big(p_{\text{data}}(\cdot \mid y) \big\| p_\theta(\cdot \mid y)\big) \le C\,\varepsilon + o(1),$$

where $C > 0$ is a constant that depends on the variance schedule, the discretization, and model parametrization; the $o(1)$ term vanishes as the diffusion discretization becomes finer.

Remarks. The lemma is qualitative: precise constants follow from score-matching and likelihood bounds in the DDPM literature (e.g., [Ho et al., 2020b, Song et al., 2020]). The key message is that a small denoising loss implies a small divergence between the learned and data distributions.

Combining Lemma 1 with Proposition 1 yields a bound of the form

$$\begin{aligned}\mathbb{E}_{S\sim p_\theta}\big[\mathcal{L}_{\text{task}}(x)\big] \le \mathbb{E}_{S\sim p_{\text{data}}}\big[\mathcal{L}_{\text{task}}(x)\big] \\ + L_{\max}\sqrt{2C\,\varepsilon} + o(1).\end{aligned} \tag{12}$$

A.1.3 Concentration from finite repository

So far we have related the model expectation to the (population) data distribution. In practice we only train on finite $\mathcal{R}$ with n samples per condition y. Let $\widehat{p}_{\text{data}}$ denote the empirical distribution formed by the repository. By Hoeffding's inequality (or McDiarmid) [Hoeffding, 1963, McDiarmid, 1989], with probability at least $1-\delta$ over the draw of the repository,

$$\begin{aligned}\Big|\mathbb{E}_{S\sim \widehat{p}_{\text{data}}}\big[\mathcal{L}_{\text{task}}(x)\big] - \mathbb{E}_{S\sim p_{\text{data}}}\big[\mathcal{L}_{\text{task}}(x)\big]\Big| \\ \le L_{\max}\sqrt{\tfrac{\log(2/\delta)}{2n}}.\end{aligned} \tag{13}$$

A.1.4 DMP performance guarantee

Combining the above pieces yields the following performance guarantee for DMP.

Proposition 2 (DMP performance guarantee)**.** *Assume the denoising loss satisfies $\mathcal{L}_{\text{denoise}}(\theta) \le \varepsilon$ and the repository contains n i.i.d. prompts for the condition y. Then with probability at least $1-\delta$ over the repository sample,*

$$\begin{aligned}\mathbb{E}_{S\sim p_\theta(\cdot|y)}\big[\mathcal{L}_{\text{task}}(x)\big] \le \mathbb{E}_{S\sim \widehat{p}_{\text{data}}(\cdot|y)}\big[\mathcal{L}_{\text{task}}(x)\big] \\ + L_{\max}\sqrt{2C\,\varepsilon} \\ + L_{\max}\sqrt{\tfrac{\log(2/\delta)}{2n}} + o(1),\end{aligned} \tag{14}$$

where C is the constant from Lemma 1 and the $o(1)$ term accounts for discretization error in the diffusion approximation.

Proof. Start from Proposition 1 applied to p_θ and p_{data}:

$$\mathbb{E}_{p_\theta}[\ell] \le \mathbb{E}_{p_{\text{data}}}[\ell] + 2L_{\max}\sqrt{\tfrac{1}{2}\,\mathrm{KL}(p_{\text{data}}\|p_\theta)}.$$

Apply Lemma 1 to bound the KL by $C\varepsilon + o(1)$; hence

$$\mathbb{E}_{p_\theta}[\ell] \le \mathbb{E}_{p_{\text{data}}}[\ell] + L_{\max}\sqrt{2C\,\varepsilon} + o(1).$$

Now replace the population expectation $\mathbb{E}_{p_{\text{data}}}[\ell]$ by the empirical expectation $\mathbb{E}_{\widehat{p}_{\text{data}}}[\ell]$ and apply Proposition 13 (Hoeffding) which with probability at least $1-\delta$ yields the stated sampling error term $L_{\max}\sqrt{\frac{\log(2/\delta)}{2n}}$. Combining these terms gives (14). □

Interpretation. Bound (14) decomposes the expected task loss under the pretrained generator into (i) the empirical repository loss (quality of collected prompts), (ii) an approximation term controlled by the denoising risk (how well DMP models the repository), and (iii) a sampling term that vanishes as the repository size n grows.

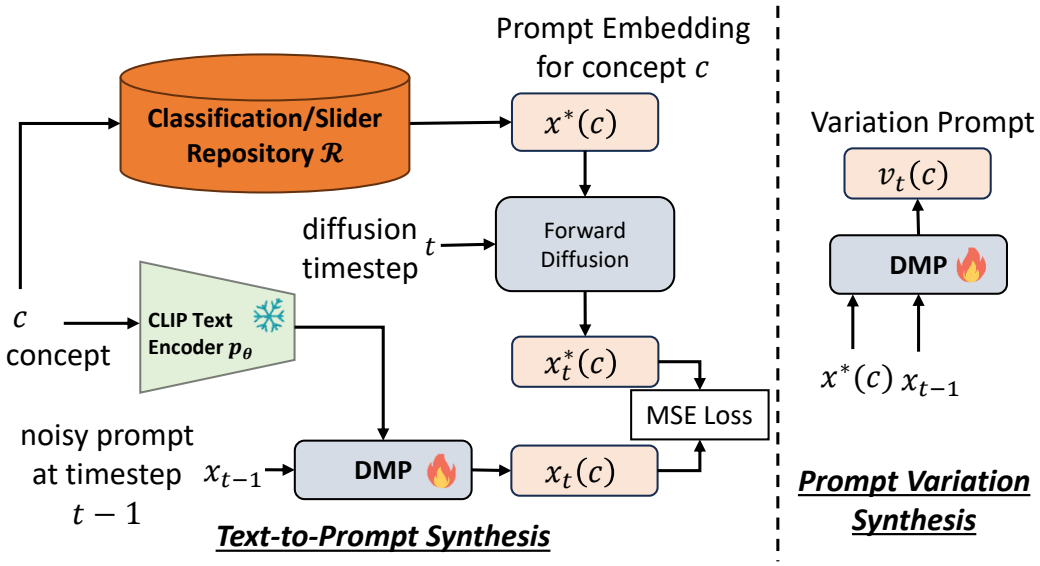

Figure 4: **Left:** Diffusion Meta-Prompt framework for Text-to-Prompt synthesis. **Right:** Prompt Variation synthesis conditioned on learned Textual Inversion Prompts.

Figure 4 shows the DMP framework where the left part denotes the training of text-to-prompt diffusion model and right part denotes the diffusion model for generating prompt variations conditioned on the original prompts.

A.2 Relation to Prior Methods

Table 10 illustrates how DMP differs from prior prompt-learning approaches across key axes such as requiring task-specific data or loss, supporting multiple tasks in a zero-shot manner, learning across multiple tasks and the need for storage of prompts. Classical prompt learning techniques are *task and model specific*, and require *task specific data and losses.* DMP instead learns the distribution of prompts from a prompt repository (produced by these prompt learning techniques). It can then be used to sample prompts for *many tasks*. Note that *DMP is not task specific and does not require task specific losses or data, just a prompt repository.* This makes DMP as a general prompt generator, free from the data and optimization constraints that characterize prior prompt-learning techniques. We show that, in many cases, DMP sampled prompts even outperform prompt learning prompts. For example, the classification results of Table 19 shows that DMP outperforms the base method on the unseen classes of the very same dataset, despite never accessing a single image.

A.3 DMP for Classification

Ablation on Steering Strength Table 11 shows the effect of the steering strength on composite classification accuracy. The trend suggests that moderate steering provides the best trade-off between stability and diversity during diffusion sampling.

Table 10: **Conceptual differences between DMP and prior prompt learning methods.** Unlike prior works that refine prompts using data for a specific task or example, DMP treats prompts as a *distribution* and trains a diffusion meta-model that amortizes prompt generation across many tasks and repositories. DMP does not require access to the original data used to train the prompts (L159–L161) and supports one-shot sampling, composition, and negative prompting without per-task optimization.

Method	**Task Data/ Loss Free**	**Supports Multiple Tasks**	**Learns across Tasks**	**Minimal Storage**	**Core Idea**
Stylus / Retrieval-based	✓	✗	✓	✗	Searches large repositories; limited generalization.
Diffusion-based Methods					
Prompt Diffusion	✗	✗	✗	✗	Refines prompts for a specific task using data.
Diff-Prompt	✗	✗	✗	✗	Mask-supervised diffusion for a single task.
Neural Network Diffusion	✗	✗	✗	✓	Diffusion to generate model weights for a particular task/dataset (weight-generation).
Conditional LoRA Param Gen.	✗	✓	✓	✓	Diffusion/hypernetwork that generates LoRA adapters conditioned on task (Cond P-Diff / CondLoRA).
DiffLoRA	✗	✓	✓	✓	Diffusion model predicts personalized low-rank (LoRA) weights at inference (zero-shot personalization).
Prompt Learning Methods					
Hierarchical Variational TTP	✗	✗	✗	✗	Test-time variational prompt generator relying on task-specific data features.
Language-Aware Soft Prompting (LASP)	✗	✗	✗	✗	Text-conditioned optimization for soft prompts, task-specific.
Consistency-guided PL (CoPrompt)	✗	✗	✗	✗	Consistency loss improves prompt robustness; task-specific.
PromptKD	✗	✗	✗	✗	Distills prompts from teacher to student in unsupervised setting; task-specific.
Dual-Prompt Collaboration (DPC)	✗	✗	✗	✗	Dual-prompt collaboration requiring tuned prompt parameters; task-specific.
Bayesian Prompt Learning (BPL)	✗	✗	✗	✗	Bayesian uncertainty modeling over task-specific prompts.
Patch-Prompt Aligned Bayesian Prompt Tuning	✗	✗	✗	✗	Bayesian hierarchical prompt generation (label-specific stochastic prompts); per-task tuning.
Meta-Learning Methods					
Prompt Learning via Meta-Regularization (ProMetaR)	✗	✓	✓	✗	Meta-regularization to improve prompt generalization across tasks; requires training data.
Gradient-Regulated Meta-Prompt (GRAM)	✗	✓	✓	✗	Meta-learned prompt init + gradient regulator for few-shot cross-domain generalization.
PRewrite (Prompt Rewriting w/ RL)	✗	✓	✗	✗	LLM-based prompt rewriter trained with RL to improve downstream task performance.
DMP (Ours)	✓	✓	✓	✓	**Learns a prompt distribution; one-shot sampling, composition, negative prompts.**

Table 11: Ablation of steering strength λ for DMP on 55 composite dataset-pairs classification. Moderate steering gives the strongest average accuracy, while overly aggressive steering begins to over-constrain the sampling trajectory.

λ	All	Base	New	Avg.
0.1	62.5	68.5	69.8	66.9
0.2	**63.1**	**69.4**	**70.8**	**67.8**
0.3	62.8	69.1	70.4	67.4
0.4	62.2	68.4	69.6	66.7
0.5	61.7	67.8	68.9	66.1
0.6	61.6	67.9	69.1	66.2
0.7	61.5	67.6	68.7	65.9
0.8	61.3	67.2	68.3	65.6
0.9	61.0	66.8	67.9	65.2
1.0	60.7	66.4	67.5	64.9

Composite classification with other prompt-learning methods. We further evaluate DMP on the composite classification task using recent prompt-learning methods, including MaPLe [khattak et al., 2023] and TAC [Hao et al., 2025]. The results are reported as the average accuracy over the 55 pairwise dataset compositions defined in Sec. 4.1.

Table 18 shows that DMP yields consistent improvements across prompt-learning backbones. The gains are largest for CoOp since we learn the full prompt distribution using a diffusion model. For other prompt learning methods such as MaPLe and TAC, the gains are smaller but remain consistent, reaching up to $+1.2/+0.6$ points for unseen classes for MaPLe/TAC respectively. This is expected because these methods learn additional parameters beyond the soft prompts, such as projection matrices or task-aware pre-context modules, which are too high-dimensional to synthesize with the current DMP implementation and available training resources. In these cases, DMP only synthesizes the prompt parameters while keeping the method-specific auxiliary parameters fixed. Extending

Table 12: Domain generalization accuracy (%) for DMPCoOp prompts sampled on ImageNet.

Method	Source	Target				
	ImageNet	-V2	-S	-A	-R	Average
CoOp	68.5	61.2	45.2	48.2	74.1	59.4
DMPCoOp	**68.7**	**62.1**	**46.5**	**50.6**	**75.1**	**60.6**

Table 13: Domain generalization accuracy (%) for DMPTAC prompts sampled on ImageNet.

Method	Source	Target				
	ImageNet	-V2	-S	-A	-R	Average
TAC	71.3	64.6	48.3	48.6	76.6	61.9
DMPTAC	**71.4**	**64.7**	**48.5**	**48.8**	**76.7**	**62.0**

Table 14: Cross-task generalization per dataset: performance for Mean Treecut Accuracy (MTA).

Method	**ImageNet**	**-V2**	**-S**	**-R**	**-A**	**SUN397**
TAC	71.3	34.1	28.3	**5.5**	36.5	30.0
DMPTAC	**71.4**	**39.6**	**37.3**	**5.7**	**38.9**	**40.0**
Δ	+0.1	+5.5	+9.0	+0.2	+2.4	+10.0

Table 15: Comparison of DMPMulti with TI prompts.

Method (SD-RV)	**Face-ID↑**	**DINO ↓**	**CLIP-I ↓**	**CLIP-T ↑**
Textual Inversion	0.428	0.627	0.696	0.244
Stylus (Top-1)	**0.434**	0.645	0.706	<u>0.246</u>
DMPMulti	**0.434**	**0.558**	<u>0.653</u>	0.245
DMPMulti(20)	<u>0.429</u>	<u>0.595</u>	**0.599**	**0.285**

Table 16: Comparison of slider prompts generated by a separate DMP (DMPSlider) against DMPMulti.

Method	**CLIP-s ↑**	**LPIPS↓**
Prompt Slider	**30.00**	0.219
DMPMulti	**29.86**	**0.126**
DMPSlider (separate)	**29.88**	**0.121**

Table 17: Comparison of separately trained DMP (DMPIdentity) with Textual Inversion and DMPMulti for identity synthesis.

Method (SD-RV)	**Face-ID↑**	**DINO ↓**	**CLIP-I ↓**	**CLIP-T ↑**
Textual Inversion	0.428	0.627	0.696	0.244
DMPMulti	**0.434**	**0.558**	**0.653**	**0.245**
DMPIdentity (separate)	**0.435**	**0.550**	**0.659**	**0.246**

DMP to jointly synthesize these higher-dimensional components would likely require compression or low-rank parameterization strategies, which we leave for future work.

Same dataset (Base to New generalization). In this setting, prompts are trained in the Base classes of each dataset and evaluated on all classes of the same dataset. This is the least demanding setting, since the prompts have to generalize only to unseen classes of the dataset where they were trained.

Table 19 summarizes the performance of PL and DMP prompts for four prompt learning methods, CoOp, CoPrompt, Maple, and TAC over the baselines across 11 datasets for Base-to-novel class generalization. The reported results are the average over three seed runs. As expected, DMP prompts have slightly lower performance for the base classes, since they are inherently bounded by the accuracy of the PL prompts used to train the DMP. Further, Proposition 1 suggests that the gap could be bridge by using more than 40 prompts per dataset to train the DMP model, we show that this is true by ablating the number of prompts used for training the DMP in Table 27. The table shows that, even under these conditions, the DMP prompts generalize better than the baseline, achieving an average accuracy gain between **+0.4%** and **4.8%** for New (unseen) classes, across prompting methods. For new classes, DMP outperforms CoOp on **11/11** datasets for all methods listed on the table. The table details the results for three datasets for brevity.

With regards to prompting methods, the table shows that the DMP gains are larger for CoOp, then Co Prompt, MaPLe and finally TAC. This is expected because DMP only synthesizes prompts. However, all methods other than CoOp, complement these prompts with additional learned weights or projection matrices. For these methods, we used the original learned weights together with the prompts synthesized by DMP. Despite only modifying prompts, DMP achieves gains of **+0.4-0.8%** over these methods for unseen classes. While a DMP-type technique could be used to sample the additional weights, this is a more complex endeavor due to the high dimensionality of the latter and compute requirements. We leave the synthesis of large parameter matrices for future work.

In any case, it can be concluded that, even on the same dataset setting, *meta-prompting with a diffusion model outperforms the existing prompting techniques that are used to train it*.

Table 18: Composite classification accuracy (%) averaged over 55 dataset pairs. DMP yields consistent improvements across prompt-learning methods. Gains are largest for CoOp, which only learns soft prompts, and smaller for MaPLe and TAC, which include additional high-dimensional learned components that are kept fixed in our current implementation.

Method	All	Base	New
CoOp	58.7	65.1	65.1
DMPCoOp	63.1	69.4	70.8
Δ	+4.4	+4.3	+5.7
MaPLe	57.2	64.3	63.0
DMPMaPLe	57.9	64.7	64.2
Δ	+0.7	+0.4	+1.2
TAC	63.2	69.6	69.7
DMPTAC	63.5	70.0	70.4
Δ	+0.3	+0.4	+0.6

These findings suggest that the original learned prompts may be somewhat overfitted to the base classes, whereas DMP prompts induce a classifier of stronger generalization. Over all classes (both seen and unseen) and datasets, DMP has an accuracy gain of upto **+3.0%**. It can be concluded that *meta-prompting with a diffusion model outperforms the existing prompting techniques that are used to train it*.

Table 19: Base2new generalization per dataset: performance for All, Base, and New classes, and HM (Harmonic Mean). The results reported are the average over three seed runs. The Average column reports the average over 11 datasets. * denotes our implementation as the code is not publicly available.

	(a) Average				**(b) ImageNet**				**(c) Caltech101**				**(d) OxfordPets**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
VAE [Kingma and Welling, 2014]	58.6	62.7	68.3	65.0	57.0	64.1	57.8	60.8	89.1	91.7	94.3	92.9	90.1	92.0	97.1	94.5
Top-1 Retrieval [Luo et al., 2024]	67.5	80.7	72.8	76.5	50.5	78.4	47.9	59.5	37.3	76.9	46.4	57.9	69.9	85.8	66.8	75.1
Transformer [Vaswani et al., 2017]	68.3	82.3	69.4	75.3	68.1	76.4	66.8	71.3	94.8	98.2	94.8	96.5	89.9	94.6	95.4	95.0
GPT-2 [Radford et al., 2019]	68.1	81.8	69.1	74.9	68.0	76.2	66.8	71.2	94.8	98.3	95.0	96.6	90.1	94.7	95.9	95.3
Prompt Diffusion* [Du et al., 2024]	67.0	69.9	73.0	71.4	54.7	60.6	56.4	58.4	91.3	94.4	93.2	93.8	92.3	94.3	**97.5**	95.9
CoOp [Zhou et al., 2022b]	**67.1**	**80.9**	68.6	**74.2**	68.0	**76.2**	66.8	71.2	93.8	98.1	93.0	95.5	90.6	95.2	96.5	95.8
DMPCoOp	**70.1**	80.3	**73.4**	**76.5**	**68.7**	75.4	**68.8**	**72.0**	**94.8**	**98.3**	**95.3**	**96.8**	**91.7**	**95.4**	**97.3**	**96.3**
Δ	+3.0	-0.6	+4.8	+2.3	+0.7	-0.8	+2.0	+0.8	+1.0	+0.2	+2.3	+1.3	+1.1	+0.2	+0.8	+0.5
CoPrompt [Roy and Etemad, 2024]	72.3	**83.1**	74.6	78.3	**70.7**	**76.7**	71.4	**73.9**	**95.8**	**98.7**	95.3	**97.0**	91.1	**95.3**	97.0	96.1
DMPCoPrompt	**72.9**	82.5	**75.4**	**78.5**	**70.7**	76.6	**71.5**	**73.9**	95.7	**98.7**	**95.4**	**97.0**	**91.6**	95.2	**97.2**	**96.2**
Δ	+0.6	-0.6	+0.8	+0.2	0.0	-0.1	+0.1	0.0	-0.1	0.0	+0.1	0.0	+0.5	-0.1	+0.2	+0.1
Maple [khattak et al., 2023]	72.0	**82.2**	75.1	78.2	70.2	76.7	70.5	73.5	94.5	**98.0**	94.3	96.1	**92.3**	**95.4**	**97.8**	**96.6**
DMPMaple	**72.4**	82.0	**75.9**	**78.6**	**70.3**	**76.8**	**70.6**	**73.6**	**94.8**	**98.0**	**95.5**	**96.7**	**92.3**	**95.4**	97.6	96.5
Δ	+0.4	-0.2	+0.8	+0.4	+0.1	+0.1	+0.1	+0.1	+0.3	0.0	+1.2	+0.6	0.0	0.0	-0.2	-0.1
TAC [Hao et al., 2025]	74.6	**85.2**	77.1	80.8	71.3	**78.5**	71.0	**74.6**	**95.2**	**98.6**	**95.0**	96.7	**93.1**	**96.0**	98.0	**97.0**
DMPTAC	**75.0**	85.1	**77.5**	**80.9**	**71.4**	**78.5**	**71.2**	**74.6**	**95.2**	**98.6**	**95.0**	**96.8**	**93.1**	95.9	**98.2**	**97.0**
Δ	+0.4	-0.1	+0.4	+0.1	+0.1	0.0	+0.2	0.0	0.0	0.0	0.0	+0.1	0.0	-0.1	+0.2	0.0

Table 21 shows the full results of cross-task generalization experiment described in section 4.1. We note that DMPCoOp obtains higher accuracies consistently across both the Mean Treecut Accuracy and Hierarchical Consistency Accuracy metrics on all the datasets considered. Notably, DMPCoOp obtains **+11.4%** and **+5%** improvement on SUN dataset for MTA and HCA respectively. This shows that DMPCoOp prompts are more robust and generalize well beyond the task on which it was trained.

We further provide the results of cross-domain experiments for TAC method in Table 13, and cross-task generalization in Table 14. The gains over baseline TAC are larger for cross-task generalization where DMPTAC prompts obtains **+10% on SUN and +9% on Imagenet-Sketch** datasets for hierarchical classification.

t-SNE visualization. To assess the fidelity and diversity of prompts synthesized by the DMP framework, we conducted an embedding space analysis comparing real prompts from the repository $\mathcal{R}$ with DMP-generated prompts sampled using different random noise seeds. Figure 5 presents a two-dimensional t-SNE projection of both sets of embeddings, with real prompts in blue and generated prompts in orange for SUN397, FGVC, Oxford Flowers and Stanford Cars datasets respectively from left to right.

The visualization reveals that generated prompts broadly overlap with the real prompt manifold, while also exhibiting a greater spread, indicative of higher variability. This suggests that the model captures

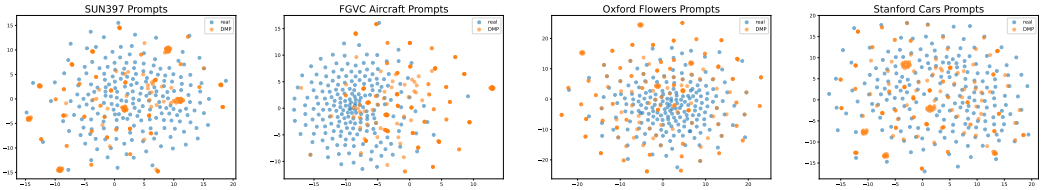

Figure 5: **t-SNE visualizations of prompt embeddings.** Each figure shows real prompts (blue) and DMP-generated prompts (orange) with different noise seeds for different datasets: (a) SUN397, (b) FGVC-Aircraft, (c) Oxford Flowers, (d) Stanford Cars. Generated prompts broadly overlap with the real prompt manifolds while exhibiting greater spread, demonstrating both fidelity and diversity across domains.

the underlying structure of prompt space without resorting to memorization, while also producing novel variations.

To quantify these observations, we computed the fraction of a prompt's 5 nearest neighbors (in embedding space) that share the same class label between real and generated prompts across 11 datasets. On average, 70.7% of generated prompts share the same nearest-neighbor labels as their real counterparts, showing strong semantic alignment between real and generated embeddings while ensuring diversity.

The clusters that are seen in the plots show that the prompts sampled by DMP are more consistent than those originally learned by the prompt learning method. This could explain the improved generalization of these prompts and the lower variance of the performance of the prompted model to different noise seeds. We observed a standard deviation of 0.7% from the mean accuracy for prompts sampled by DMPCoOp with 200 different noise seeds. In contrast, the standard deviation of the baseline prompt learning method used to build the repository from different initialization seeds show a higher variation of 4.7%.

We additionally evaluated five prompts generated from different random seeds on the combined Caltech101–ImageNet task and measured prediction disagreement, per-class accuracy correlation, and error-set overlap. Table 20 shows that, across the five prompts, the standard deviation of accuracy is only 0.09 percentage points (range: 0.2 points). The per-class accuracy correlation of 0.999, and the error set overlap of 98.66% indicate highly consistent decision boundaries across random seeds. On the other hand, the fact that different sampled prompts disagree on approximately 1.08% of individual predictions shows that they are not identical. Overall, these results indicate that diffusion sampling yields slight local variations while preserving essentially the same functional behavior and generalization performance.

Table 20: Agreement statistics across different random seeds.

Prompt pair	**Accuracy (%)**	**Prediction disagreement (%)**	**Per-class accuracy correlation**	**Error-set overlap (%)**
Seed 1 vs. Seed 2	59.4 / 59.5	1.01	0.9992	98.81
Seed 1 vs. Seed 3	59.4 / 59.5	1.12	0.9991	98.68
Seed 1 vs. Seed 4	59.4 / 59.3	1.52	0.9982	97.97
Seed 1 vs. Seed 5	59.4 / 59.5	0.97	0.9992	98.78
Seed 2 vs. Seed 3	59.5 / 59.5	0.79	0.9994	99.07
Seed 2 vs. Seed 4	59.5 / 59.3	1.43	0.9985	98.19
Seed 2 vs. Seed 5	59.5 / 59.5	0.71	0.9995	99.18
Seed 3 vs. Seed 4	59.5 / 59.3	1.21	0.9986	98.44
Seed 3 vs. Seed 5	59.5 / 59.5	0.79	0.9994	99.05
Seed 4 vs. Seed 5	59.3 / 59.5	1.19	0.9987	98.43
Mean	**59.44**	**1.08**	**0.9990**	**98.66**

Impact of Classifier-Free Guidance (CFG) scales. The performance of DMP with different guidance values is shown in Table 22. The trends show that the average accuracy across all datasets decreases with increasing guidance scales.

Variational Autoencoder Reconstruction. Table 23 shows the reconstruction accuracy of the CoOp prompts for the trained Variational Autoencoder (VAE) across different datasets. The autoencoder reconstructs the prompts almost perfectly with only 0.1% difference on average across all the datasets.

Table 21: Cross-task generalization per dataset: performance for Mean Treecut Accuracy (MTA) [Wu et al., 2024], and Hierarchical Consistency Accuracy (HCA) [Wu et al., 2024].

	ImageNet		-V2		-S		-R		-A		SUN397	
Method	MTA	HCA	MTA	HCA	MTA	HCA	MTA	HCA	MTA	HCA	MTA	HCA
CoOp	40.7	0.8	38.9	0.8	34.4	0.5	**57.7**	9.9	43.8	4.1	19.0	31.3
DMPCoOp	**47.4**	**2.6**	**45.0**	**2.1**	**40.2**	**2.0**	**57.7**	**18.7**	**55.5**	**5.5**	**30.4**	**36.3**
Δ	+6.7	+1.8	+6.1	+1.3	+5.8	+1.5	0.0	+8.8	+11.7	+1.3	+11.4	+5.0

Table 22: Comparison with different classifier free guidance scales. Base2new generalization per dataset: performance for All, Base, and New classes, and HM (Harmonic Mean).

	(a) Average (scale 4.5)				(b) Average (scale 5.5)				(c) Average (scale 6.5)				(d) Average (scale 7.5)			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
DMPCoOp	**70.1**	**80.3**	**73.4**	**76.5**	**69.6**	**80.3**	**70.6**	**76.3**	**69.6**	**79.3**	**70.4**	**75.0**	**68.1**	**79.4**	**70.1**	**75.0**

A.4 Multi-task DMP

DMPMulti is a multi-task DMP model trained to synthesize prompts for personalized subjects and prompt sliders. Both tasks are unified within a single framework through the shared CLIP-ViT L/16 text encoder.

Prompt Slider Repository. Prompt Sliders is a technique to learn text prompts in the CLIP text embedding space that allow control of particular attributes of a designated concept. The prompt is learned through the optimization

$$S^* = \arg\min_S \mathbb{E}_{x\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t} \left\|\epsilon_t - \epsilon_\theta^d(x_t, \tau_\theta(y, S), t)\right\|_2^2 \tag{15}$$

where x_t is an image, and y is any additional prompt text, such as "a photo of a". Both τ_θ and ϵ_θ are fixed during the optimization. Given a target concept c_t, a prompt slider S^* is learned to encourage the distribution of images of c_t to exhibit more positive attributes c^+ and fewer negative attributes c^-. This is implemented by replacing ϵ_t with

$$\epsilon_t(\alpha) = \epsilon_\theta^d(x_t, \tau_\theta(y(c_t)), t) + \alpha\eta \sum_{p\in P}(\epsilon_\theta^d(x_t, \tau_\theta(y(c^+, p)), t) - \epsilon_\theta^d(x_t, \tau_\theta(y(c^-, p)), t)) \tag{16}$$

and S by αS in (15), where η is a guidance scale, α a scaling parameter, $y(c_t)$ the concept name, and P a set of concepts that the attribute manipulation should preserve (for example, race or gender). The positive c^+, and negative c^- attributes are sampled from a template predefined for concept c_t. To create a slider repository $\mathcal{R}_S$, we trained prompt sliders for 20 different concepts using the SD-XL model and $3,000$ backpropagation steps. For each concept c, we save 40 prompts $S(c)$ (from the last 200 steps in intervals of 5) to obtain a training tensor $\mathcal{R}_S \in \mathbb{R}^{20\times40\times d}$.

Prompt Identity Repository. We use a random subset of 20 identities from the 3000 celebrity faces in the DMPVariation prompt repository to create a prompt identity repository $\mathcal{R}_I$ with each identity containing 40 prompts obtained from the last 200 steps in intervals of 5 to obtain a training tensor $\mathcal{R}_I \in \mathbb{R}^{20\times40\times d}$.

Training. The DMPMulti model was trained directly in prompt space $\mathcal{P}$, i.e. without autoencoder, using the full set of prompts from both repositories, i.e. $\mathcal{R} = \mathcal{R}_S \cup \mathcal{R}_I$. For each slider concept c (e.g., age, smiling etc.) we use the concept name as the text condition for DMPMulti. For identities, we use "identity-c" as the text condition where $c = 1, \ldots, 20$ is associated with each identity c in $\mathcal{R}$.

Inference. At inference, DMPMulti can be conditioned with the slider concept name, to generate prompt sliders, or with "identity-c" to generate identity prompts. Novel identities can be sampled by specifying a new identity "identity-c" with $c > 20$. Slider and identity prompts can then be fed to the downstream model in isolation or together.

Table 15 compares Identity prompts from DMPMulti with Stylus and Textual Inversion across face recognition accuracy, image-to-image similarity, and prompt fidelity. DMPMulti achieves higher identity scores, maintains prompt compliance comparable to TI/Stylus, and shows lower similarity to training images, indicating stronger generalization. In contrast, TI tends to overfit, consistent with prior findings [Yuan et al., 2023]. The last row of Table 15 shows an ablation study of using fewer samples (20 vs 40) per identity for training DMP. Notably, DMPMulti (20) achieves similar identity fidelity and even higher prompt compliance (↑17%) than the 40-sample variant.

Figure 7 shows qualitative results of slider prompts generated by DMP for the attributes "long hair" and "chubby".

Table 23: Quantitative results of VAE reconstruction: Comparison against CoOp prompts across various datasets. The VAE reconstructs the CoOp prompts baseline almost perfectly with only 0.1% difference on average.

Model	ImageNet	Oxford flowers	Oxford pets	Stanford cars	Caltech 101	Food101	FGVC Aircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp	**68.0**	**72.6**	**90.1**	**68.7**	**94.8**	84.9	**25.1**	**67.6**	51.9	58.9	**66.2**	**68.1**
Autoencoder	**68.0**	72.0	89.6	68.6	94.6	**85.1**	25.0	67.0	**52.0**	**59.5**	66.1	68.0
Δ	0.0	-0.6	-0.5	-0.1	-0.2	+0.2	-0.1	-0.6	+0.1	+0.6	-0.1	-0.1

Table 24: Ablation study on Cross-dataset generalization of DMPCoOp Imagenet prompts: Comparison of classnames against dataset names as prompt inputs to the DMPCoOp model.

Model	ImageNet	Oxford flowers	Oxford pets	Stanford cars	Caltech 101	Food101	FGVC Aircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp	68.0	66.5	88.5	61.9	**92.7**	84.8	15.2	60.7	40.9	**46.9**	65.3	62.9
DMPCoOp (Dataset Names)	67.8	65.2	88.0	**63.4**	**92.7**	83.9	16.8	62.6	**41.0**	46.3	66.6	63.2
Δ (Dataset)	-0.2	-1.3	-0.5	+1.5	0.0	-0.9	+1.6	+1.9	+0.1	-0.6	+1.3	+0.3
DMPCoOp (Class Names)	**68.7**	**69.0**	**89.2**	62.4	91.4	**85.7**	**20.5**	**62.9**	40.4	46.2	**66.7**	**63.9**
Δ (Class)	+0.7	+2.5	+0.7	+0.5	-1.3	+0.9	+5.3	+2.2	-0.5	-0.7	+1.4	+1.0

Figure 10 presents a qualitative comparison of images generated by Concept Sliders (using the author-provided models), Prompt Sliders, and DMP. The results show that Concept Sliders struggle to induce the intended attributes, even at higher scales, due to their sensitivity to training hyperparameters, which requires careful tuning as noted in [Sridhar and Vasconcelos, 2024]. We observe that DMP-generated prompts produce images that are qualitatively similar to those from Prompt Sliders. However, the original Prompt Sliders method has limitations in maintaining subject identity at higher scales, as discussed in [Sridhar and Vasconcelos, 2024]. In contrast, DMP—despite being trained with the Prompt Sliders embeddings—demonstrates greater robustness, effectively preserving subject identity even at relatively higher scales. These result corroborate the results observed in Table 2 of the paper.

Figure 18 illustrates the results of pure negative prompts, where the sliders yield images with attributes opposite to the specified concepts, such as shorter hair or a neutral (non-smiling) expression.

A.5 DMP for Variations

Generalization. Figure 12 shows images synthesized for the variation prompts generated by DMP-Variation model, for a common seed. Note that the variation prompts are conditioned by the Textual inversion embedding for the target concept, which is illustrated by a GT image in the figure. The figure shows that DMPVariation produces successful variation prompts for general objects as diverse as birds and statues *despite being trained only on CelebA-face identities*.

Qualitative Results. Figure 13 shows additional qualitative results for variations generated by DMPVariation model.

Figure 15 shows the synthesized prompts that produce variations of a person, conditioned on the textual inversion embedding of the person. Note that, while all images are synthesized with the same SDXL seed, they exhibit a diversity of background scenes, hair patterns, clothing, etc. This is an additional benefit of the natural prompt variability of DMP-based personalization: to increase the diversity of the synthesized images.

Figure 16 presents qualitative results comparing DMP and TI in generating personalized subject images across diverse contexts. The variation prompts produced by DMP demonstrate greater robustness and generalization, effectively adapting to different contexts while preserving subject identity. In contrast, Textual Inversion tends to overfit to the subject, leading to poor generalization. For instance, it completely fails to generate correct images for the Buddha statue and succeeds in only a single scenario for the duck and dog subjects. Table 29 shows the detailed HPSv2 scores for the results presented in Table 6 of the paper.

Identity Composition. Figure 17 demonstrates results of combining two celebrities, displayed on the left, using DMPVariation model to synthesize identity prompts with Eq. 8. The synthesized identity clearly incorporates prominent features from both original faces, such as the nose and chin, resulting in a cohesive blend of attributes.

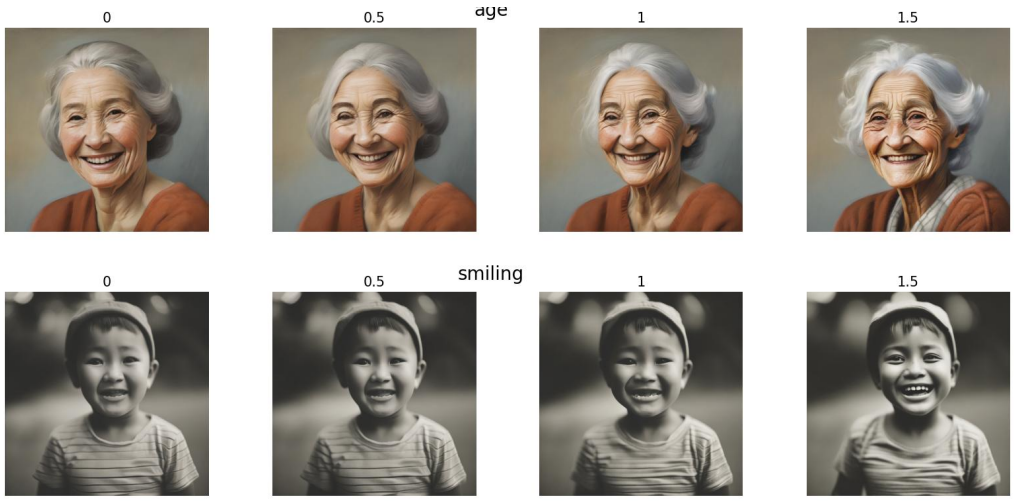

Figure 6: **Prompt Sliders** images synthesized with sliders sampled by DMPSlider when prompted for age and smiling. The prompt used for the SD-XL model is *"A photo of a beautiful man"*.



Figure 7: **Qualitative results of DMPSlider prompts** depicting the concepts “long hair" and “chubby". The prompts used for the SD-XL model for the images shown from left to right are as follows. *"A closeup photo of a person", "Professional headshot of a person"*.

Subject Composition. Figure 19 illustrates the ability of DMPVariation to generate prompts for combined concepts. In this examples, the prompts elicit the downstream model to produce images that combine the two subjects displayed on the left. This is done by sampling subject prompts using the DMPVariation model and Eq. (5). These are then fed to the SDXL model to produce the images on the right, for increasing guidance scale η. The synthesized subject clearly incorporates prominent features from both, such as the nose and chin, resulting in a cohesive blend of attributes.

Interpreting Variation Prompts. We computed the top-5 nearest-neighbor tokens in the CLIP vocabulary for the prompts sampled by DMPVariation model conditioned on a Textual Inversion embedding of a subject. For a random subject not in the training dataset, the TI prompt embedding returns `<w>karanjohar</w>`, `<w>conclude</w>`, `<w>leaked</w>`, `<w>prohibition</w>`, `<w>vijaysethu</w>`. The DMPVariation prompt returns `<w>karanjohar</w>`, `<w>pandoramusic</w>`, `episo`, `<w>leaked</w>`, `<w>refriger</w>`. It overlaps with two words out of five showing that the prompt is indeed a variation of the conditioned Textual Inversion prompt. Extending this analysis over 66 unseen identities, we found an average of **2.7 common words in the top-5 and 5.6 in the top-10** nearest-neighbor tokens, demonstrating that the DMP model effectively generalizes while retaining some of the subject-specific characteristics.

Table 25: **Prompts used for evaluating generalization.** Prompts were designed to explore style and concept variations of the subject `sks` and borrowed from DreamBooth [Ruiz et al., 2023].

Prompts			
a sks on the beach	sks flower arrangement	sks stained glass window	sks as a witcher
A photo of two sks on a boat	sks Funko Pop	sks latte art	A cubism painting of sks person
Manga drawing of sks	Pointillism painting of sks	Ukiyo-e painting of sks	A sks as a knight in plate armor
sks as a knight in plate	Banksy art of sks	sks piloting a fighter jet	Greek sculpture of sks
Fauvism painting of sks	Cave mural depicting sks	sks by Andy Warhol	sks in the style of Archer
Colorful graffiti of sks	sks as Ziggy Stardust	sks in a comic book	Watercolor painting of sks
a sand sculpture of sks	sks in a Santa hat	sks as a wizard	a photo of sks

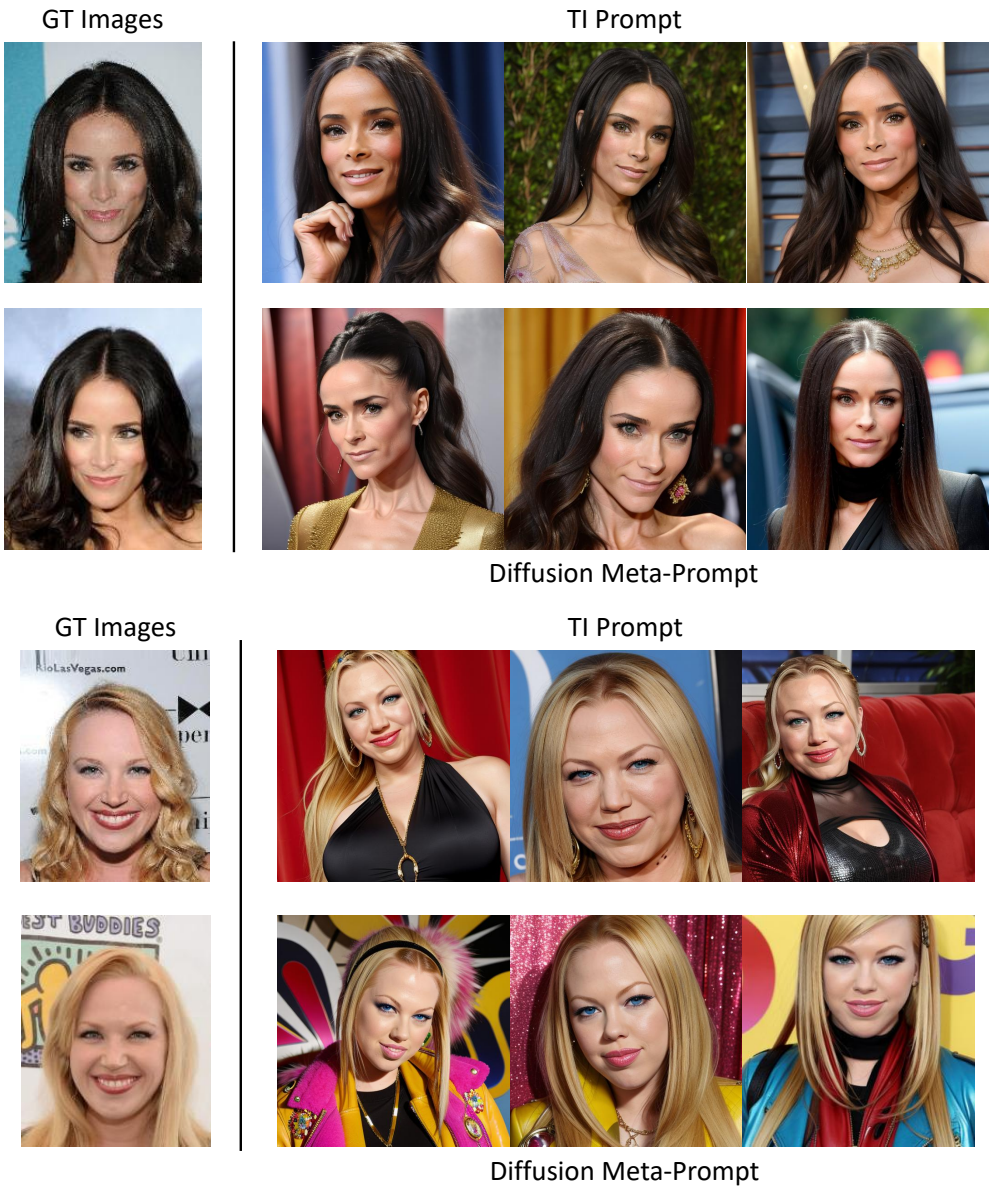

Figure 8: **Qualitative comparison** of image synthesis with TI and DMPMulti prompts. Left: groundtruth images for Identity-38 and Identity-96 in the training set. Right: images synthesized by SD for the original TI prompts (first row) and prompts sampled from the DMPMulti model (second row):

A.6 Ablation Studies

The following section lists the setup for the ablation study on sensitivity of DMP to class-name conditions in Table 9. Table 26 shows the corresponding class names provided to DMP at each replacement level.

Original 0% Condition. The original Caltech101 condition is:

```
accordion; bonsai; chandelier; euphonium; menorah;
saxophone; stapler; yin_yang; anchor; camera
```

It contains objects, instruments, and symbols drawn from Caltech101.

Replacement Pool. The replacement names are different ImageNet classes:

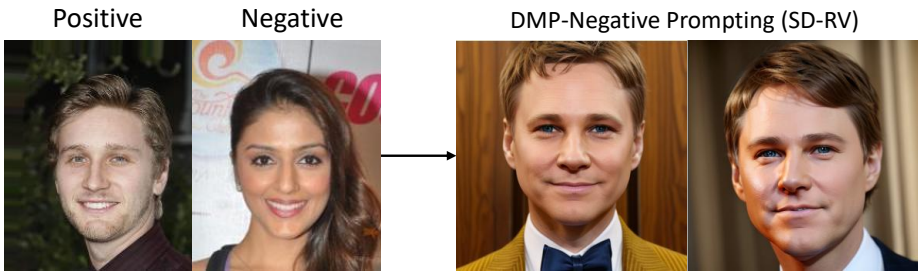

Figure 9: **Negative prompting** samples from the SD Realistic Vision model when prompted by the DMPMulti, itself prompted with id-21 (leftmost) as positive and id-24 (second from left) as negative prompt. The generated identity has features opposing to id-24 (chubby cheeks, small eyes, wide nose, etc.)

```
brambling; goldfinch; house finch; junco;
indigo bunting; American robin; bulbul; jay
```

These names are inserted cumulatively and in a fixed order. This controls which names change and ensures that the only systematic variable is the replacement percentage.

Table 26: Class-name conditions supplied to DMP at each replacement level.

Replacement	Class-name condition supplied to DMP
0%	accordion, bonsai, chandelier, euphonium, menorah, saxophone, stapler, yin_yang, anchor, camera
20%	brambling, goldfinch, chandelier, euphonium, menorah, saxophone, stapler, yin_yang, anchor, camera
40%	brambling, goldfinch, house finch, junco, menorah, saxophone, stapler, yin_yang, anchor, camera
60%	brambling, goldfinch, house finch, junco, indigo bunting, American robin, stapler, yin_yang, anchor, camera
80%	brambling, goldfinch, house finch, junco, indigo bunting, American robin, bulbul, jay, anchor, camera

A.6.1 Ablation Study on Alternative methods for Meta-Prompting.

The first two rows of Table 19 show the ablation study of using Transformer and GPT-2 for modeling the distribution of CoOp prompts across the 11 datasets. We include the Transformer as a non-generative baseline and GPT-2 as an autoregressive generative model. For the Transformer, we use a pretrained RoBERTa-Base model, which is finetuned using LoRA to predict prompt embeddings from text conditions. A linear layer is added on top of the final layer to produce output embeddings of the target dimension (2048 for CoOp/CoPrompt), and training is done with MSE loss. The table shows that transformer tends to overfit to the base classes as it is able to closely match the performance of the baseline CoOp on the base classes while under-performing on the novel classes leading to a decrease in the overall performance. For GPT-2, we discretize the continuous prompt embeddings by identifying their top-5 nearest tokens in the CLIP embedding space. These tokens are then converted back to text and used to finetune a pretrained GPT-2 model with LoRA, trained to predict the top-5 tokens using standard cross-entropy loss. The model is conditioned on the same text inputs as used in the diffusion counterpart. Although each text condition has 40 associated prompts from different initializations (as described in Section 4.1), the resulting tokens after discretization are almost identical across seeds. This indicates that the variation captured in the continuous embedding space is lost during the discretization process-a known limitation, as discretization inherently reduces information. The table reflects this observation and shows that GPT-2 based modeling is inferior to diffusion since diffusion is much better for modeling continuous distribution of prompts. Moreover, unlike autoregressive methods, diffusion offers multiple benefits such as classifier-free guidance, negative prompting, inversion, editing and composition that are challenging or infeasible with models like GPT-2. These results show that modeling the prompt distribution with diffusion is more effective and flexible than using autoregressive methods.



Figure 10: **Qualitative comparison** of images synthesized with baseline prompt sliders and concept sliders against DMPSlider. The prompt used for the SD-XL model is *"A photo of a girl"* and the slider trained to control the concept **"curlyhair"**.

A.6.2 Ablation on the number of prompts per concept

We already include the ablation study when using only 20 prompts per concept for identity synthesis in Table 15. Here, we include an additional ablation study on the number of prompts for classification tasks in Table 27. It shows that DMP works well even with as few as 5 or 10 prompts per task/concept. For 5 prompts per class, DMPCoOp obtains +1.2% gain on new classes and it increases as n increases (+1.6/+2.5/+3.0 for n =10/20/40 prompts respectively). The results are consistent with our theoretical guarantee discussed in Appendix A.1 where higher n corresponds to lower downstream risk and better generalization.

A.6.3 Ablation on the text inputs to DMPCoOp model

We conducted an ablation study using the dataset names as the prompt or text condition to the DMPCoOp model instead of the classnames. Table 24 shows that the performance of the model using dataset names is better than the baseline CoOp prompts by 0.3% on average while it is 0.7% lower as compared to the model using classnames. This shows that using class-specific names generates prompts with robust generalization than just using a single dataset name as the text condition.

A.6.4 Ablation on the quality of training data

To investigate this, we assessed the impact of the quality of training data by training DMP with noisy prompts. We apply additive Gaussian noise (n_i) with a standard deviation of $\sigma = 0.01$ to the ith

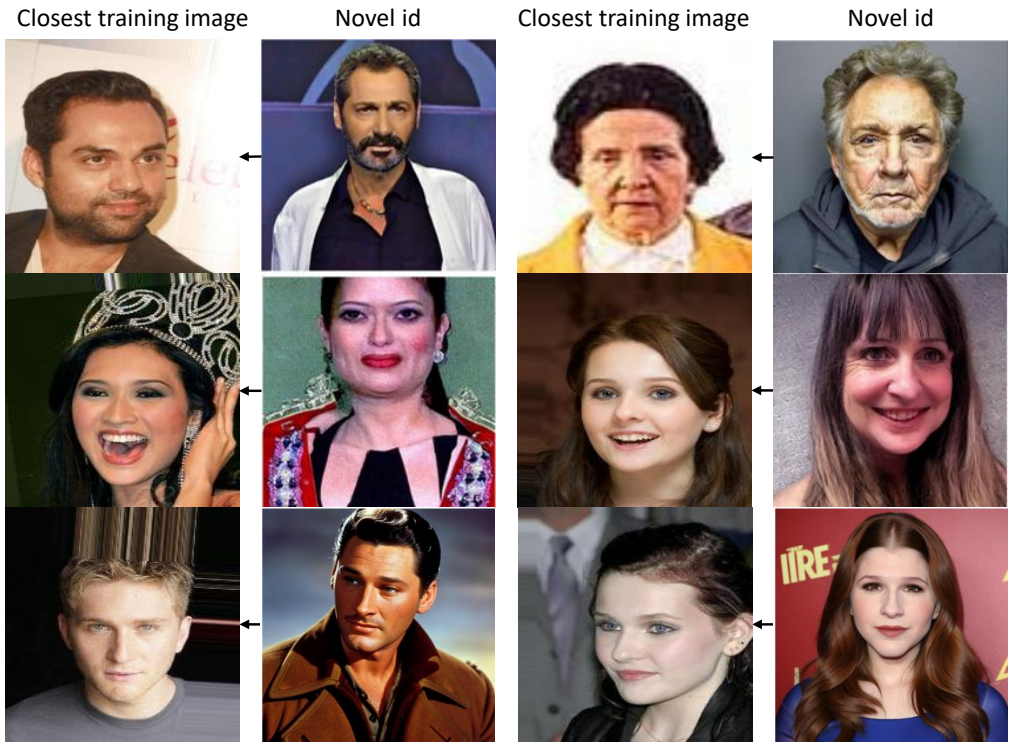

Figure 11: **Additional Qualitative results:** Nearest neighbors in the training set shown on the left for novel ids sampled by DMPMulti model to the right.

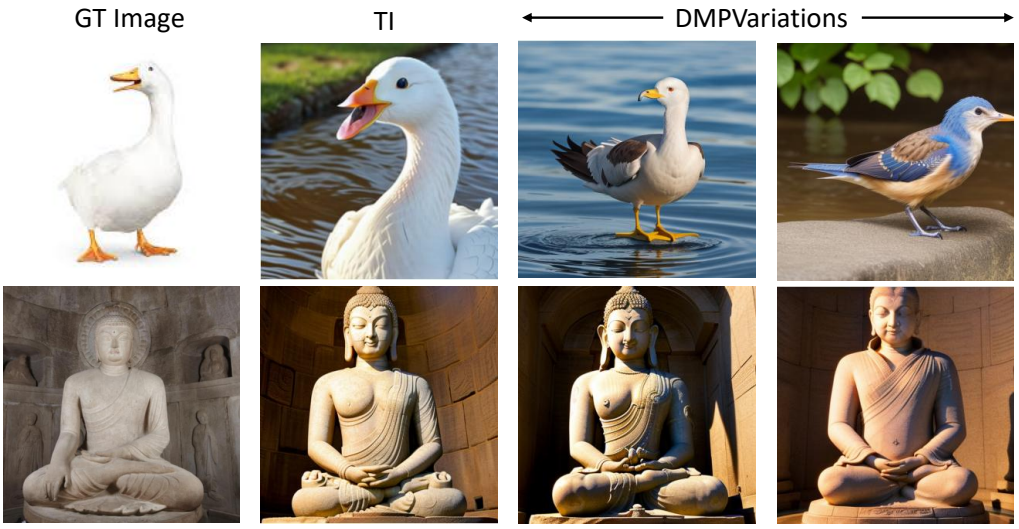

Figure 12: **Generalization of DMPVariation model**. Left: groundtruth images. Second: images generated by SD for the original TI prompts and the last two columns are the images for variation prompts sampled by DMP. See Fig. 13 for additional results.

prompt $x(c)_i$ as

$$x(c)_i = (1 - \sigma)x(c)_i + \sigma n_i.$$

The noisy prompt is applied to $x\%$ of the training dataset, where $x \in \{10, 40\}$.

Table 28 summarizes the experiment where random noise is added to the prompts in the training repository. Meta-prompting is observed to be robust to noise levels ranging from 10% to 40% of the training prompts. DMPCoOp trained with 10% noisy prompts still obtains +1.9% improvement over the baseline. Further, the average H.M for the DMP model with 10% noisy prompts is slightly better (76.7 vs 76.5 for DMPCoOp) than the DMPCoOp model trained on clean prompts suggesting that a

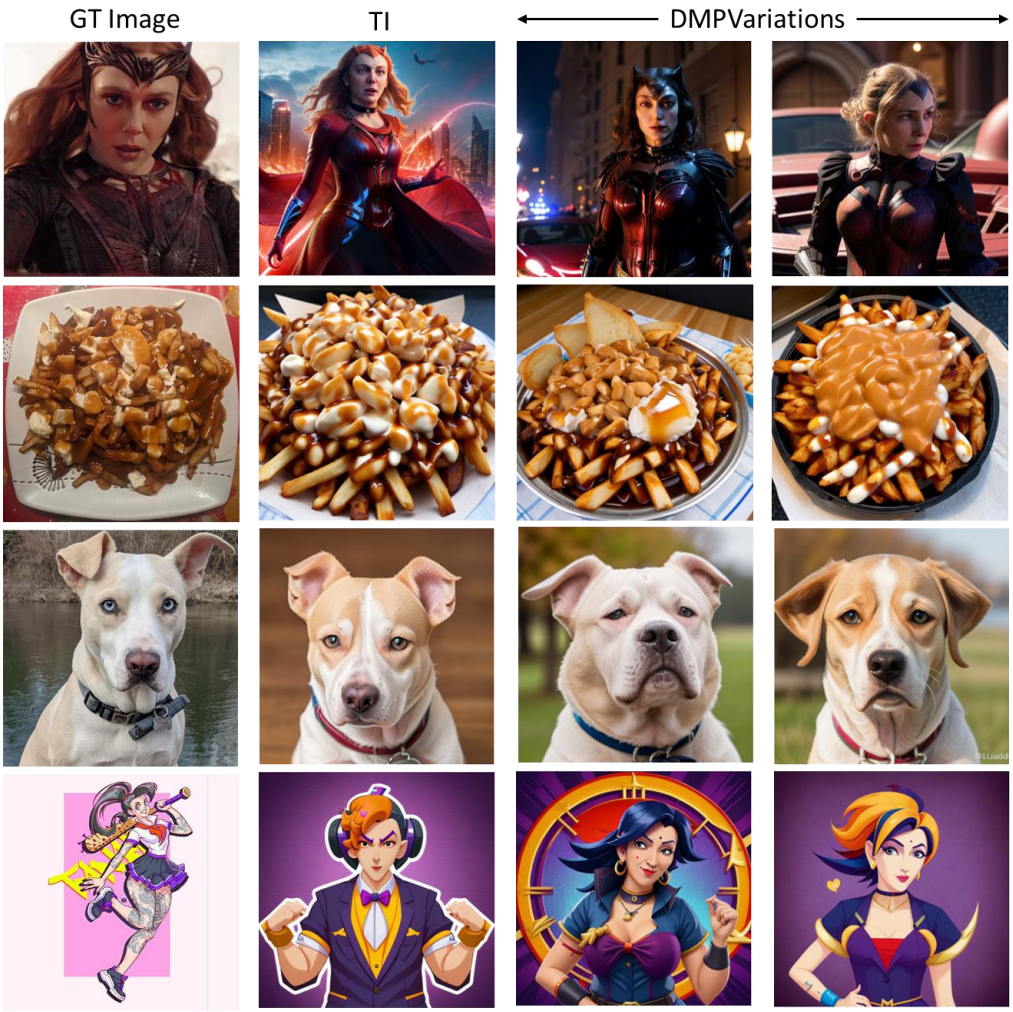

Figure 13: **Additional Qualitative results of generalization of DMPVariation model**. Left: groundtruth images. Second: images generated by SD for the original TI prompts and the last two columns are the images for variation prompts sampled by DMP.

small amount of noise can also help with generalization. This is similar to image diffusion models, which are also known to be robust to noise added during training.

A.7 Ablation on DMPMulti

We trained separate DMP models for identity synthesis and slider synthesis to compare their performance with the DMPMulti model, which was trained to generate both prompt types simultaneously. Table 16 presents the results of the DMPSlider model, trained solely for slider prompt generation. Table 17 presents the results of the DMPIdentity model, trained solely for identity prompt generation. The results indicate that its performance is comparable to that of the DMPMulti model, demonstrating that multi-task training in DMPMulti does not compromise its effectiveness.

A.7.1 Ablation on Identity Composition with Textual Inversion

For identity composition, we perform an ablation study using Stable Diffusion with Equation 5, generating new identities by combining prompts such as "a photo of id-1" and "a photo of id-2." Figure 14 illustrates the results of this process using Textual Inversion prompts with the Stable Diffusion v1.5 model. The generated images are often noisy, distorted, and tend to replicate the input identities rather than effectively merging their attributes. Additionally, running a full forward diffusion process with multiple identities doubles the inference time from 4 to 8 seconds per image.

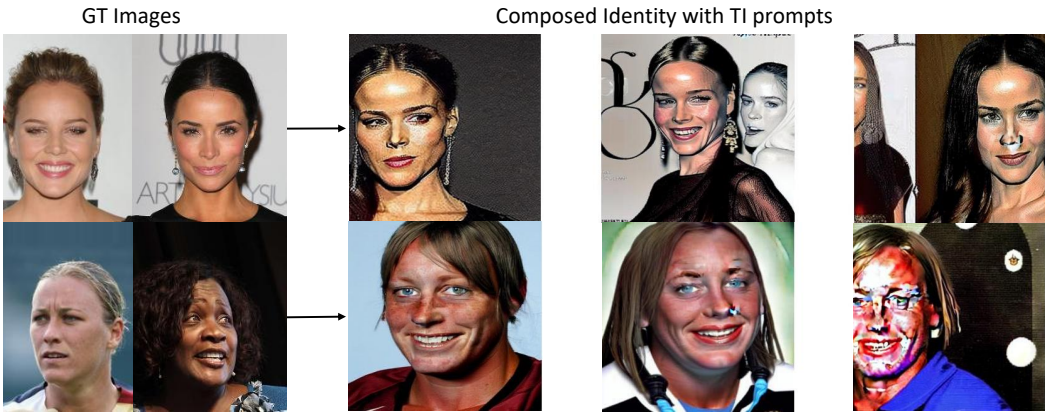

Figure 14: **Ablation study on Identity composition with Textual Inversion:** images generated with TI prompts by composing the Stable Diffusion outputs for the composition of the two identities shown on the left for each row. TI composition produces distorted identities or repeats the same identities.



Figure 15: **DMP Variations of Textual Inversion (TI) Prompts**: The leftmost image is the real groundtruth, the second is generated by TI, and the rest are DMP variations (each image represents a new identity) conditioned on the TI embedding. All images are generated with a fixed seed to the stable diffusion model. The FaceID similarity to the groundtruth image is listed on top of each image. Lower is better for diverse variations.

In contrast, our DMPMulti achieves high-quality identity compositions in approximately 5 seconds, introducing only a 1-second overhead compared to standard Stable Diffusion.

A.8 DMP for Personalization

The DMPVariation model enables the generation of prompt variations conditioned on a given Textual Inversion prompt, making it possible to train a text-to-prompt meta-diffusion model as a replacement for personalization prompt repositories. Users only require access to existing prompt repositories, as the DMPVariation model can directly generate the necessary intermediate embeddings. We choose the top-k ($k = 40$) embeddings generated from the DMPVariation model based on the cosine similarity with the available TI prompts. These embeddings can serve as training data for generating personalized prompts based on textual input. Once trained, this approach eliminates the need to search and retrieve prompts from a database, allowing for on-the-fly prompt generation.

Figure 8 presents a comparison of the images generated by DMPMulti model for the identity labeled "id-38" and "id-96" respectively. The figure presents three classes of images: groundtruth on the left, synthesized by SD prompted by the original TI prompts in the first and third row of the right side, and synthesized by SD prompted by the DMPMulti model, itself prompted for"identity-c." The images synthesized using DMPMulti model have quality comparable to those synthesized with TI prompts.

Negative Text Guidance. Figure 9 illustrates the effect of negative prompting by displaying images synthesized when DMPMulti model is prompted with identity-21 (leftmost) as positive and identity-24 (second from left) as negative prompt. The generated identity exhibits contrasting characteristics, such as fuller cheeks, smaller eyes, and a broader nose—features to those of the negative identity (id-24).

Novel Identities. Figure 11 shows qualitative results of novel identities sampled by DMP and their closest training images. Figure 24 shows additional qualitative results of novel identities sampled by DMPMulti.

Identity Prompt Diffusion. Figure 20 presents the qualitative results of various identities sampled by DMPMulti model. The figure contains the groundtruth on the left, and synthesized by SD prompted

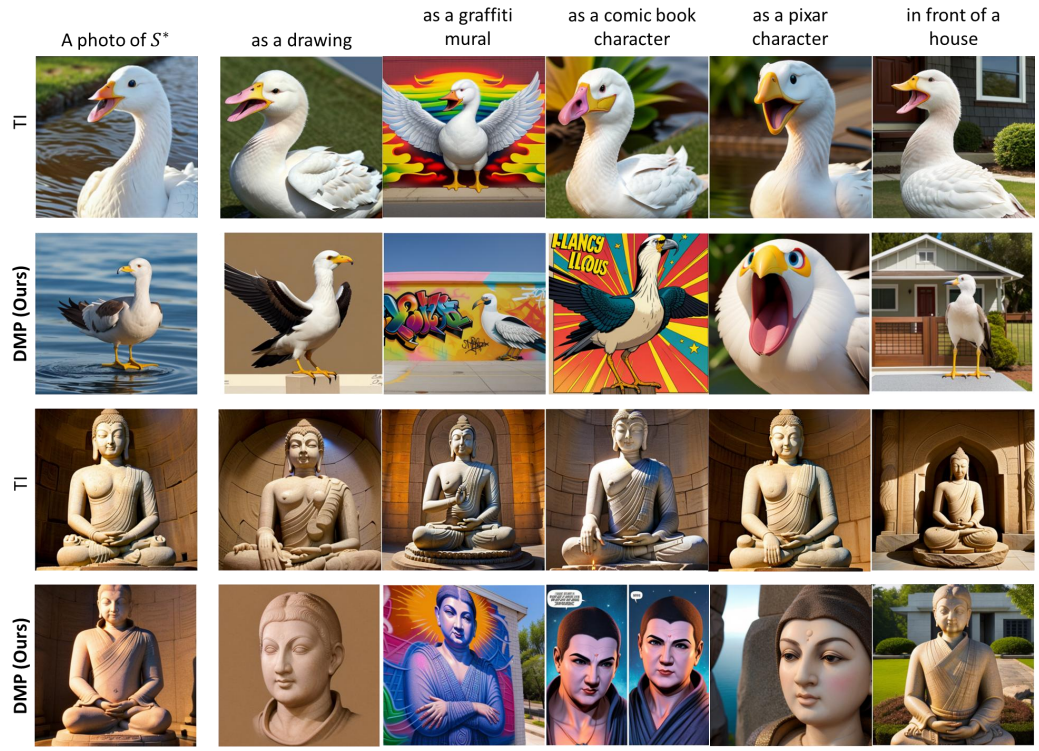

Figure 16: **Qualitative results** of images generated from prompts synthesized by DMPVariation. Our Diffusion Meta-Prompts are more robust and less overfitted than the baseline textual inversion which fails to generalize.

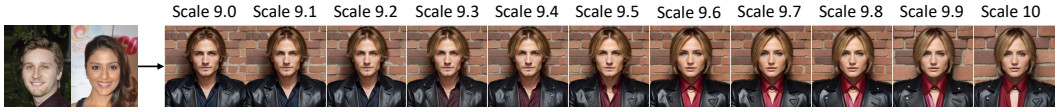

Figure 17: **Identity composition:** images generated with DMPVariation model prompts (right) for increasing guidance scales from 9 to 10 for the composition of the two identities shown on the left.

by the DMPMulti, itself prompted for“identity-c" on the right. The images synthesized using both models reflect the original identities in the groundtruth images.

Identity Composition. Figure 21 demonstrates additional results of combining two identities, displayed on the left, using DMPMulti to synthesize identity prompts with (5). The synthesized identity clearly incorporates prominent features from both original faces, such as the nose and chin, resulting in a cohesive blend of attributes.

Figure 22 shows the results of identity composition using SDv1.5 checkpoint that uses the same CLIP text encoder as SD-Realistic Vision checkpoint. It shows that DMP performs effectively without requiring retraining for this version. Since DMP was trained in CLIP text space, it generalizes to all models sharing the CLIP text encoder, eliminating the need for retraining on specific model versions.

Interpolation. Figure 23 shows the qualitative results of interpolating between two faces using the DMPMulti model with classifier-free guidance scale between 0 to 5. The results show that DMPMulti enables fine-grained interpolation by simply manipulating the guidance scale.

A.9 Limitations and Future Work

While the DMP framework unifies and improves prompt generation and generalization, simplifying deployment, the effectiveness of DMP is fundamentally constrained by the quality and expressiveness of the underlying prompt learning method used to construct the training repository. Second, DMP inherits the limitations of text prompts such as lack of fine-grained control and is sensitive to noise seeds similar to image diffusion models with a small variance between the sampled prompts. In our experiments, DMP maintained high accuracy even at different noise seeds with only a standard deviation of 0.7 % from the mean accuracy for classification tasks.



Figure 18: **Negative Prompt Sliders** images synthesized with sliders sampled by DMPSlider when negatively prompted for different concepts. The prompt used for the SD-XL model is *"A photo of a girl"*.



Figure 19: **Subject composition:** images generated with DMPVariation model prompts (right) for the composition of the two identities shown on the left with the guidance scale denoted at the top. See Fig. 17 for additional results at finer scale.

Table 27: Ablation study with different number of prompts per concept. Base2new generalization per dataset: performance for All, Base, and New classes, and HM (Harmonic Mean). The results reported are the average over three seed runs.

	(a) Average				**(b) ImageNet**				**(c) Caltech101**				**(d) OxfordPets**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
CoOp [Zhou et al., 2022b]	67.1	80.9	68.6	74.2	68.0	76.2	66.8	71.2	93.8	98.1	93.0	95.5	90.6	95.2	96.5	95.8
DMPCoOp	**70.1**	80.3	**73.4**	76.5	68.7	75.4	**68.8**	72.0	**94.8**	**98.3**	**95.3**	**96.8**	91.7	95.4	97.3	96.3
DMPCoOp (5 prompts)	69.1	81.4	71.6	75.9	68.9	76.6	68.0	72.0	**94.8**	97.9	95.2	96.5	91.1	95.4	97.1	96.2
DMPCoOp (10 prompts)	69.3	**81.8**	72.0	76.3	69.0	**76.6**	68.5	72.3	94.5	98.2	95.2	96.7	92.9	**96.0**	**97.5**	**96.7**
DMPCoOp (20 prompts)	69.1	81.2	72.9	**76.6**	**69.3**	76.4	**68.8**	**72.4**	94.3	98.2	95.1	96.6	**93.0**	**96.0**	**97.5**	**96.7**

	(e) StanfordCars				**(f) Flowers102**				**(g) Food101**				**(h) FGVC Aircraft**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
CoOp [Zhou et al., 2022b]	68.3	**75.8**	68.2	71.8	72.2	96.2	66.7	78.8	83.8	89.8	88.0	88.9	25.9	37.0	30.4	33.4
DMPCoOp	**69.3**	74.5	**72.0**	**73.2**	**75.6**	93.4	**72.2**	81.4	**86.4**	89.6	89.9	89.7	26.3	35.7	31.6	33.5
DMPCoOp (5 prompts)	68.8	75.2	70.8	72.9	74.5	96.0	71.1	81.7	84.5	89.5	89.1	89.3	**27.9**	**37.5**	**33.3**	**35.3**
DMPCoOp (10 prompts)	68.5	**75.8**	69.9	72.7	73.7	95.7	69.8	80.7	85.5	89.8	**91.2**	**90.5**	27.8	37.0	32.0	34.3
DMPCoOp (20 prompts)	68.7	75.0	70.8	72.8	75.9	**96.4**	72.1	**82.5**	85.4	**90.0**	91.1	**90.5**	**27.9**	37.3	31.6	34.2

	(i) SUN397				**(j) DTD**				**(k) EuroSAT**				**(l) UCF101**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
CoOp [Zhou et al., 2022b]	67.2	**80.7**	71.8	76.0	51.0	**78.5**	50.5	61.5	49.7	78.3	58.3	66.8	68.1	84.0	64.2	72.8
DMPCoOp	67.3	80.4	72.9	76.5	**53.7**	75.1	**55.8**	**64.0**	**65.9**	85.6	**80.7**	**83.1**	71.7	79.6	70.7	74.9
DMPCoOp (5 prompts)	66.8	79.8	71.7	75.5	52.1	76.4	54.3	63.5	59.1	85.7	67.2	75.3	71.1	**85.6**	69.6	76.8
DMPCoOp (10 prompts)	**67.6**	80.3	**73.4**	**76.7**	52.1	75.6	53.4	62.6	60.1	**89.0**	71.7	79.4	70.7	85.3	69.2	76.4
DMPCoOp (20 prompts)	**67.6**	80.4	73.3	**76.7**	51.4	75.6	54.1	63.1	54.6	84.8	74.4	79.3	**71.8**	82.9	**72.6**	**77.4**

Future research directions can address these limitations by exploring joint training of the meta-model and the downstream foundation models, potentially overcoming the performance ceiling imposed by existing prompt learning techniques. Other directions for future work can explore ways for distilling LoRA [Hu et al., 2022] adapters into prompts and the design of a Meta-LoRA model, which synthesizes weight matrices instead of prompts (a more complex problem due to the large parameter cardinality of LoRA weights).

A.10 Implementation details

In this section, we describe the evaluation setup followed in our experiments and the rationale behind choosing the setup. We then describe the hyperparameter settings used to train all DMP models and finally the prompt format used in DMPMulti model.

Evaluation Setup. For downstream model, we use stable diffusion [Rombach et al., 2022] `Realistic-Vision-v4` checkpoint using classifier-free guidance with a scale of 4.5 and 30 DDIM steps for the image synthesis. For SD-XL [HuggingFace, 2023] model, we use a scale of 7.5 with 20 DDIM steps. Note that, because DMP prompts are introduced in the CLIP text encoder, they can be interchangeably used with any diffusion model using this encoder. Our choice of downstream diffusion model follows the original prompting methods.

For models other than CoOp, additional weights or head layers are optimized to prompt the deeper layers of the CLIP encoders. Due to the large number of parameters associated with these weights, it is infeasible to train a diffusion model to synthesize these parameters. For example, the projection matrices of MaPLe have 3.55 million parameters while CoPrompt and TAC have 4.65 million parameters each. This is much larger than even the images produced by Stable diffusion (65536 parameter latent). In contrast, all text prompts have only 2048 or fewer parameters (only 256 parameters in the latent space). So, we only synthesize the text prompts attached to the input of CLIP model with DMP. During inference, we replace the learned textual prompt from the baseline method with the prompt sampled with DMP while keeping the other weights of the baseline method fixed. As followed in prompt learning literature, we pick three random seeds and report the average results.

Table 28: Base2new generalization per dataset: performance for All, Base, and New classes, and HM (Harmonic Mean). The results reported are the average over three seed runs.

	(a) Average				**(b) ImageNet**				**(c) Caltech101**				**(d) OxfordPets**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
CoOp [Zhou et al., 2022b]	67.1	**80.9**	68.6	74.2	68.0	76.2	66.8	71.2	93.8	98.1	93.0	95.5	90.6	95.2	96.5	95.8
DMPCoOp	**70.1**	80.3	**73.4**	76.5	68.7	75.4	68.8	72.0	**94.8**	**98.3**	**95.3**	**96.8**	91.7	95.4	97.3	96.3
DMPCoOp (10% noise)	69.6	81.7	72.3	**76.7**	**69.2**	76.5	**69.0**	**72.6**	94.0	98.1	94.2	96.1	91.5	95.0	96.9	95.9
DMPCoOp (40% noise)	68.9	81.2	70.3	75.0	68.5	**76.6**	67.3	71.6	94.1	97.9	95.2	96.5	**92.8**	**95.7**	**97.9**	**96.8**
	(e) StanfordCars				**(f) Flowers102**				**(g) Food101**				**(h) FGVC Aircraft**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
CoOp [Zhou et al., 2022b]	68.3	75.8	68.2	71.8	72.2	96.2	66.7	78.8	83.8	89.8	88.0	88.9	25.9	37.0	30.4	33.4
DMPCoOp	**69.3**	74.5	**72.0**	**73.2**	75.6	93.4	72.2	81.4	**86.4**	89.6	89.9	89.7	26.3	35.7	31.6	33.5
DMPCoOp (10% noise)	69.1	**76.7**	69.4	72.9	77.1	**96.4**	72.8	83.0	85.6	**89.9**	**91.1**	**90.5**	**27.7**	**37.6**	**33.8**	**35.6**
DMPCoOp (40% noise)	68.3	76.6	69.1	72.7	**79.4**	95.9	**75.0**	**84.2**	84.6	89.4	90.3	89.8	26.4	36.6	32.8	34.6
	(i) SUN397				**(j) DTD**				**(k) EuroSAT**				**(l) UCF101**			
Method	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM	All	Base	New	HM
CoOp [Zhou et al., 2022b]	67.2	**80.7**	71.8	76.0	51.0	78.5	50.5	61.5	49.7	78.3	58.3	66.8	68.1	**84.0**	64.2	72.8
DMPCoOp	**67.3**	80.4	**72.9**	**76.5**	**53.7**	75.1	**55.8**	**64.0**	**65.9**	**85.6**	**80.7**	**83.1**	71.7	79.6	70.7	74.9
DMPCoOp (10% noise)	66.2	**80.7**	70.5	75.3	53.5	**79.9**	51.9	62.9	59.3	84.5	71.6	77.5	**72.5**	83.0	**74.0**	**78.2**
DMPCoOp (40% noise)	65.9	79.7	69.8	74.4	49.8	78.4	48.6	60.0	59.6	84.0	62.8	71.9	68.4	82.3	64.1	72.1

We note that different initialization seeds result in minor performance differences which explains the performance difference from the original paper reported results.

A.10.1 Hyperparameter Settings

Table 30 summarizes the detailed hyperparameter settings of the DMP models trained from scratch reported in the main paper.

A.10.2 Evaluation Prompts for DMPMulti

The example format of the prompts used for evaluation is shown below for the concept "Age",

- A portrait of a woman with a warm smile, {}
- A person's face, {}
- A man sitting on a park bench, reminiscing about his youth, {}
- A couple of friends enjoying a picnic together, {}
- A photo of a person, {}

A.11 LLM usage

LLM was used to polish the writing (e.g., clarity, grammar). It was not used in ideation or in any other stage.

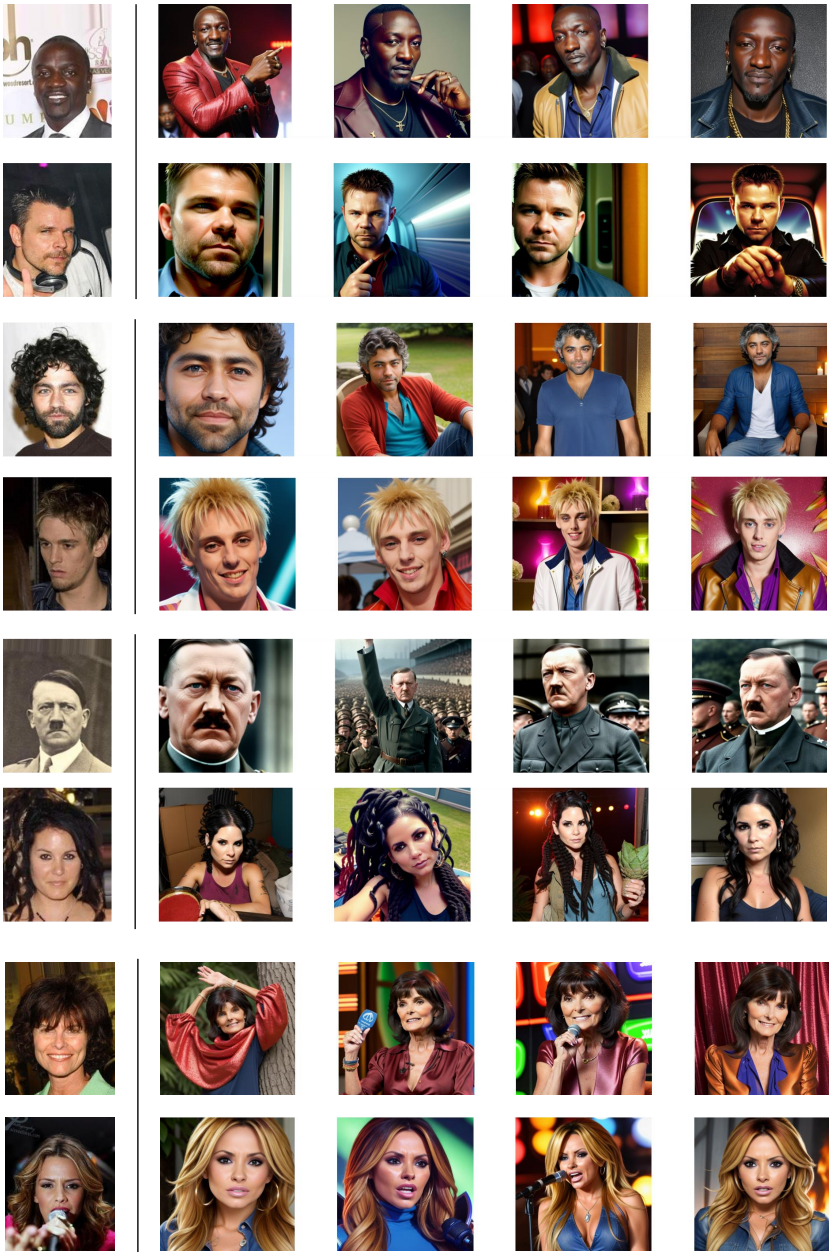

Figure 20: **Qualitative results** of image synthesis with DMPMulti prompts. Left: groundtruth images. Right: images synthesized by SD with prompts sampled from the DMPMulti model.

Concept	DMP	TI	Delta
ettblackteapot	0.2905	0.2778	+0.0127
dicoo	0.2837	0.2700	+0.0137
goku	0.2830	0.2800	+0.0030
degodsheavy	0.2827	0.2754	+0.0073
spider-gwen	0.2825	0.2810	+0.0015
johnny-silverhand	0.2812	0.2760	+0.0052
blue-haired-boy	0.2812	0.2800	+0.0012
bullvbear	0.2798	0.2683	+0.0115
black-waifu	0.2790	0.2715	+0.0075
a-female-hero-from-the-legend-of-mir	0.2788	0.2710	+0.0078
freddy-fazbear	0.2786	0.2750	+0.0036
ldrs	0.2783	0.2642	+0.0141
chonkfrog	0.2778	0.2756	+0.0022
concept-art	0.2770	0.2715	+0.0055
degods	0.2769	0.2686	+0.0083
hanfu-anime-style	0.2766	0.2703	+0.0063
joemad	0.2764	0.2673	+0.0091
dog	0.2761	0.2780	-0.0019
stuffed-penguin-toy	0.2760	0.2756	+0.0004
fox-purple	0.2760	0.2734	+0.0026
colossus	0.2756	0.2634	+0.0122
chungus-poodl-pet	0.2754	0.2637	+0.0117
anya-forger	0.2754	0.2737	+0.0017
furrpopasthetic	0.2751	0.2737	+0.0014
bob-dobbs	0.2751	0.2664	+0.0087
eddie	0.2751	0.2607	+0.0144
arthur1	0.2750	0.2666	+0.0084
dragonborn	0.2750	0.2556	+0.0194
kay	0.2747	0.2605	+0.0142
tesla-bot	0.2747	0.2634	+0.0113
borderlands	0.2747	0.2637	+0.0110
lavko	0.2744	0.2588	+0.0156
gim	0.2740	0.2659	+0.0081
hubris-oshri	0.2740	0.2659	+0.0081
tubby	0.2737	0.2673	+0.0064
finn-token	0.2737	0.2734	+0.0003
moxxi	0.2730	0.2722	+0.0008
altvent	0.2730	0.2676	+0.0054
omlettehaai	0.2730	0.2600	+0.0130
jos-de-kat	0.2730	0.2651	+0.0079
loab-style	0.2730	0.2573	+0.0157
crinos-form-garou	0.2727	0.2693	+0.0034
blue-zombie	0.2727	0.2693	+0.0034
cgdonny1	0.2725	0.2610	+0.0115
amogus	0.2725	0.2551	+0.0174
lucky-luke	0.2725	0.2751	-0.0026
captain-haddock	0.2725	0.2725	+0.0000
manga-nov-23	0.2725	0.2551	+0.0174
button-eyes	0.2722	0.2683	+0.0039
bruma	0.2722	0.2632	+0.0090
ouroboros	0.2720	0.2734	-0.0014
fursona	0.2720	0.2646	+0.0074
kanovt	0.2720	0.2522	+0.0198
doc	0.2717	0.2693	+0.0024
warhammer-40k-drawing-style	0.2715	0.2727	-0.0012
nard-style	0.2715	0.2698	+0.0017
baluchitherian	0.2715	0.2617	+0.0098
insidewhale	0.2715	0.2630	+0.0085
irasutoya	0.2715	0.2598	+0.0117
devonm	0.2712	0.2617	+0.0095
edgerunners-style-v2	0.2712	0.2660	+0.0052
nixeu	0.2710	0.2660	+0.0050
shek-9-12-opening	0.2710	0.2708	+0.0002
loab-character	0.2710	0.2656	+0.0054
ldr	0.2710	0.2666	+0.0044
malika-favre-art-style	0.2710	0.2632	+0.0078
drive-scorpion-jacket	0.2710	0.2686	+0.0024
apulian-rooster-v0-1	0.2710	0.2351	+0.0359
fftstyle	0.2708	0.2593	+0.0115
alf	0.2708	0.2664	+0.0044
wheelchair	0.2708	0.2730	-0.0022
obama-self-2	0.2705	0.2705	+0.0000
dog-chip	0.2703	0.2676	+0.0027
cheburashka	0.2700	0.2705	-0.0005
ihylc	0.2700	0.2580	+0.0120
cat-toy	0.2695	0.2600	+0.0095
Average	0.2731	0.2663	+0.0068

Table 29: HPSv2 comparison for 75 random concepts where each concept is evaluated for 6 different prompts.

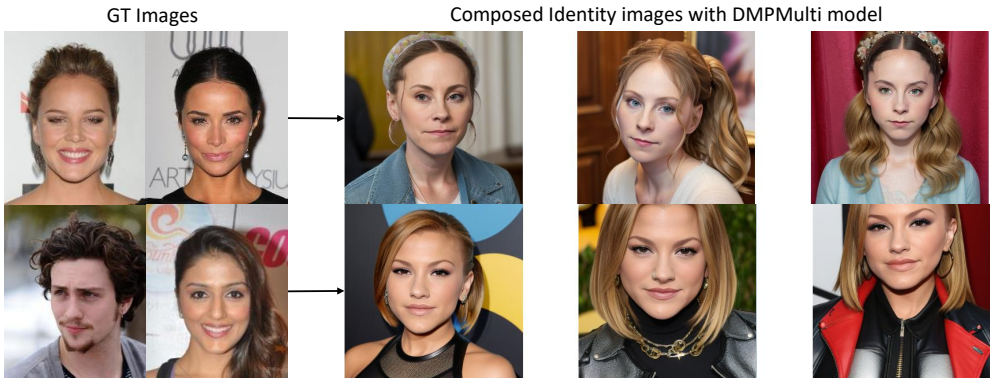

Figure 21: **DMPMulti composition:** Images generated with DMPMulti model prompts (right) for the composition of the two identities shown on the left.

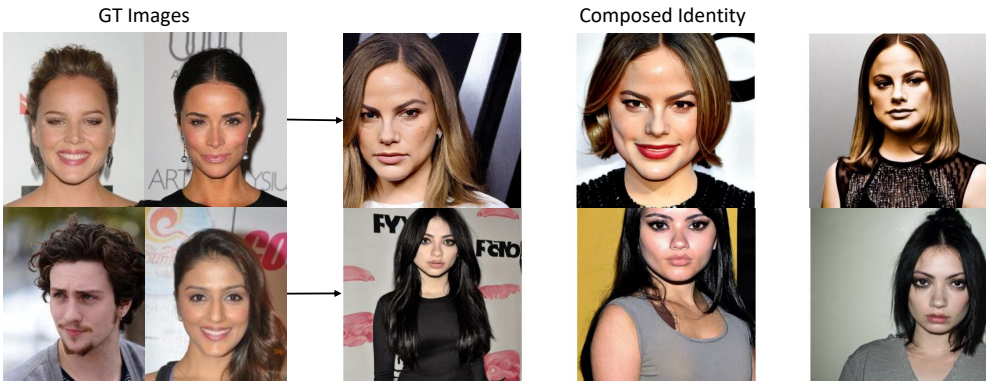

Figure 22: **DMPMulti composition with SDv1.5 model:** Prompts synthesized by **DMP** generalize well to different downstream models sharing the same CLIP text encoder without any re-training.

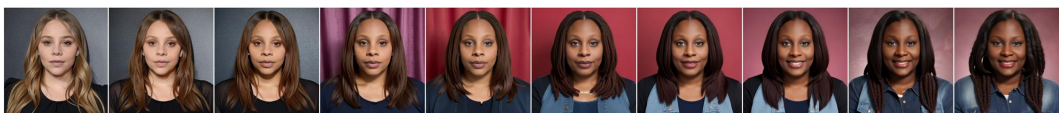

Figure 23: **Interpolating** between two faces using DMPMulti with concept composition.

Table 30: Hyperparameter Settings for the DMP models trained across three different tasks namely personalization, concepts and classification.

	DMPVariation	DMPMulti	DMPCoOp	Autoencoder
z-shape	-	-	16 x 8	16 x 8
x-shape	768 x 1	768 x 1	2048 x 1	2048 x 1
$\|Z\|$	-	-	128	128
$\|X\|$	768	768	2048	768
Diffusion steps	1000	1000	1000	1000
Optimizer	AdamW	AdamW	AdamW	AdamW
Noise Schedule	linear	linear	linear	linear
Nparams	33M	33M	33M	1M
Channels	128	128	128	128
Depth	1	1	1	1
Channel Multiplier	1,2,2,2	1,2,2,2	1,2,2,2	1,1,1,1,2,2,2,2
Attention resolutions	16, 8, 4	16, 8, 4	8, 4, 2	-
Head Channels	8	8	8	8
Batch Size	320	320	256	128
Iterations	100k	100k	100k	100k
Learning Rate	1e-6	1e-6	2.0e-7	4.5e-6



Figure 24: **Qualitative results of generating novel identities during inference** using random identity conditioning with DMPMulti model. The prompt used for the Stable Diffusion Realistic Vision model is *"A photo of a id-x"* where x is the id not present in the training set. The identities do not overfit (**mean face ID similarity of 0.0102 across training images**). Note that all the images use a fixed seed to the diffusion model.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We train a Diffusion model for meta-prompting from a learned repository and show that it generalizes better to novel tasks without any task-specific training than the repository prompts. Results are shown across classification, image generation and retrieval settings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Discussed in Appendix A.9

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Proof is included in Appendix

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details are included in appendix. Code will be released upon acceptance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Yes, code will be made public after acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes they are included in the Implementation details and Appendix

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The results follow the standard practice in literature reporting the average over three seed runs for the appropriate experiments

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: Yes in Implementation details

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Yes after conclusion

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Not applicable as it uses only open source models and data

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets included

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Not applicable

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs used only for editing or polishing text

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.