Prototype-Guided Diffusion: Efficient Visual Conditioning without External Memory

1st Bilal Faye
LIPN, UMR CNRS 7030
Sorbonne Paris Nord University
faye@lipn.univ-paris13.fr

2nd Hanane Azzag
LIPN, UMR CNRS 7030
Sorbonne Paris Nord University
hanene.azzag@lipn.univ-paris13.fr

3rd Mustapha Lebbah
DAVID laboratory
Paris Saclay (UVSQ) University
mustapha.lebbah@uvsq.fr

***Abstract*—Diffusion models achieve state-of-the-art image generation but remain computationally costly due to iterative denoising. Latent-space models like Stable Diffusion reduce overhead yet lose fine detail, while retrieval-augmented methods improve efficiency but rely on large memory banks, static similarity models, and rigid infrastructures. We introduce the *Prototype Diffusion Model (PDM)*, which embeds prototype learning into the diffusion process to provide adaptive, memory-free conditioning. Instead of retrieving references, PDM learns compact visual prototypes from clean features via contrastive learning, then aligns noisy representations with semantically relevant patterns during denoising. Experiments demonstrate that PDM sustains high generation quality while lowering computational and storage costs, offering a scalable alternative to retrieval-based conditioning.**

I. Introduction

Diffusion models have rapidly become a dominant generative framework, known for high sample quality, training stability, and flexibility across tasks. Since the introduction of Denoising Diffusion Probabilistic Models (DDPMs) [1], they have surpassed GAN-based approaches [2], [3] and been extended to unconditional generation [4], super-resolution [5], inpainting [6], and text-to-image generation [7], [8].

Despite these advances, diffusion models remain slow due to iterative denoising. Latent-space methods such as Stable Diffusion [8] mitigate this cost but lose spatial detail and require powerful encoders/decoders. Retrieval-augmented approaches (RDM) [9] improve efficiency by conditioning on examples retrieved via pretrained vision-language models (e.g., CLIP [10]), but incur large memory/storage costs and rely on fixed features, limiting adaptability. An orthogonal direction, ProtoDiffusion [11], introduces prototype-based conditioning but in a two-stage pipeline: class-specific prototypes are precomputed and remain static, hindering adaptability and scalability to fine-grained or multimodal settings.

We address these limitations with the *Prototype Diffusion Model (PDM)*, which integrates online prototype learning directly into the diffusion process. Unlike ProtoDiffusion, PDM requires no labels and is fully unsupervised: prototypes are dynamically constructed from clean intermediate features using contrastive objectives, then reused as semantic anchors during noisy denoising steps. This joint, continuous learning allows prototypes to adapt alongside evolving representations, improving semantic grounding and efficiency without external memory. Furthermore, when labels are available, we extend PDM to a supervised variant, *s-PDM*, where class-specific prototypes are fixed by labels, eliminating prototype selection and compactness loss. Our contributions are:

- We propose *PDM*, the first diffusion model that integrates online, unsupervised prototype learning into the denoising process, enabling efficient and adaptive conditioning without external memory or annotations.
- We extend PDM to a supervised setting with *s-PDM*, where label-driven prototypes further simplify training and remove the need for compactness regularization.
- Through extensive experiments, we show that both PDM and s-PDM outperform standard DDPMs (no prototypes) and ProtoDiffusion (two-stage static prototypes), validating their scalability and effectiveness.

II. Related Work

A. Diffusion Models

Diffusion models have become a dominant generative framework, offering strong performance in image synthesis, inpainting, and super-resolution since DDPMs [1]. Latent diffusion [8] improves efficiency by operating in compressed spaces, while recent advances such as consistency models [12] and SDXL [13] demonstrate scalability. However, the denoising process remains slow due to many sequential steps, and latent-space acceleration often sacrifices spatial fidelity. These limitations motivate methods that incorporate additional context to guide denoising.

B. Retrieval-Augmented Diffusion Models

Retrieval-based approaches improve quality and controllability by conditioning diffusion on externally retrieved examples. RDM [9] aligns noisy features with CLIP-based retrieved patches, and IRDiff [14] applies similar ideas to molecular generation. While effective, these methods rely on large static memory banks, frozen encoders [10], and incur retrieval latency, reducing adaptability to evolving representations. Such constraints motivate lighter, internal conditioning mechanisms.

C. Prototype Learning in Diffusion Models

Prototype-based conditioning offers compact alternatives to external memory. ProtoDiffusion [11] employs class-specific

prototypes but learns them in a separate, static stage, limiting adaptability. Other works use prototypes for few-shot or continual learning [15], [16], yet typically rely on coarse class-level structures or focus on discriminative tasks. Existing methods thus lack fully joint, fine-grained prototype learning integrated into diffusion.

Our Prototype Diffusion Model (PDM) addresses this gap by learning patch-level prototypes online from clean features and aligning them with noisy representations during denoising, enabling adaptive conditioning without external memory.

III. Background

A. Diffusion Models

Diffusion models are a class of generative models that learn to synthesize data by inverting a stochastic noising process. A prominent and widely used variant is the *Denoising Diffusion Probabilistic Model* (DDPM) [1], [17], which formulates generation as the reversal of a predefined forward diffusion process.

The forward process gradually corrupts a clean data sample $x_0 \in \mathbb{R}^d$, drawn from an unknown data distribution $q(x_0)$, by adding Gaussian noise over T discrete timesteps. This results in a sequence of latent variables $x_1, \ldots, x_T$, defined recursively by:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t\, \mathbf{I}), \tag{1}$$

where $\beta_t \in [0,1]$ is a variance schedule controlling the amount of noise injected at each timestep, and $\mathbf{I}$ is the identity matrix. By defining $\alpha_t = 1 - \beta_t$ and the cumulative product $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, the marginal distribution of x_t given x_0 admits a closed-form expression:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

The goal of the model is to learn the reverse process $p_\theta(x_{t-1} \mid x_t)$, which is intractable in general. DDPM approximates this reverse process by a parameterized Gaussian distribution:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_t), \tag{3}$$

where the mean $\mu_\theta(x_t, t)$ is predicted by a neural network and the variance Σ_t is typically fixed or learned.

A common parameterization predicts the noise ϵ added during the forward process. The mean of the reverse distribution is then recovered as:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon_\theta(x_t, t) \right), \tag{4}$$

where $\epsilon_\theta(x_t, t)$ is the model's estimate of the noise component ϵ at timestep t.

The model is trained to minimize the expected mean squared error between the true noise ϵ and the predicted noise ϵ_θ, resulting in the simplified training objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t, x_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta \left(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, t \right) \right\|^2 \right]. \tag{5}$$

This objective encourages the model to accurately denoise corrupted inputs at arbitrary timesteps, ultimately enabling sampling by iteratively denoising from pure Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$ back to a coherent data sample x_0.

B. Retrieval-Augmented Diffusion Models

Retrieval-Augmented Diffusion Models (RDMs) enhance classical diffusion models by incorporating a retrieval mechanism that conditions generation on external, real examples. This semi-parametric approach supplements the learned generative model with non-parametric memory, enabling improved sample diversity, controllability, and generalization [8], [9], [18].

Given an input image x, its latent representation $z = E(x)$ is computed using a pretrained encoder E, typically from an autoencoder such as VQ-GAN [19]. A forward diffusion process is then applied in the latent space to produce noisy samples $z_1, \ldots, z_T$, analogous to the DDPM framework.

To enrich the generative process with semantic context, a retrieval function $\xi_k(x, D)$ selects the k nearest neighbors of x from a dataset D, based on similarity in a learned embedding space. This embedding space is induced by a pre-trained model ϕ, such as CLIP [10], and the retrieval is defined as:

$$\xi_k(x, D) = \text{Top-}k \left(\text{sim}(\phi(x), \phi(y)) \mid y \in D \right), \tag{6}$$

where $\text{sim}(\cdot, \cdot)$ denotes a similarity metric, such as cosine similarity, in the embedding space $\phi(\cdot)$.

The denoising model is then conditioned not only on the noisy latent z_t and timestep t, but also on the retrieved support set $\{\phi(y) \mid y \in \xi_k(x, D)\}$. The denoising function becomes: $\epsilon_\theta(z_t, t, \{\phi(y)\})$, where conditioning is typically implemented using cross-attention layers that attend to the retrieved features during noise prediction [8].

The training objective follows the denoising loss paradigm, extended to include retrieval conditioning:

$$\mathcal{L}_{\text{RDM}} = \mathbb{E}_{x, \epsilon, t} \left[\left\| \epsilon - \epsilon_\theta \left(z_t, t, \{\phi(y) \mid y \in \xi_k(x, D)\} \right) \right\|^2 \right]. \tag{7}$$

This formulation enables the model to ground its generation in real, semantically similar samples, which is particularly beneficial in data-scarce regimes or when fine control over the output is desired. Unlike purely parametric approaches, RDMs flexibly adapt to new domains by querying from an updated external memory without retraining the entire model.

C. Prototype Learning in Diffusion Models

Prototype-based learning has become a cornerstone of few-shot learning, where the objective is to generalize to novel classes from only a few labeled examples. One of the most influential approaches in this domain is the Prototypical Network (ProtoNet) [20], which embeds both support and query samples into a common metric space and performs classification based on distances to class prototypes.

Given a set of K support examples $\{x_{c,k}\}_{k=1}^{K}$ for a class c, a class prototype $z_c \in \mathbb{R}^d$ is computed as the mean of their embeddings under a shared encoder f_ϕ: $z_c = \frac{1}{K} \sum_{k=1}^{K} f_\phi(x_{c,k})$, where f_ϕ is typically a neural network trained to produce discriminative embeddings. A query sample x_q is classified by computing the softmax over negative distances to each prototype: $p(y = c \mid x_q) = \frac{\exp(-d(f_\phi(x_q), z_c))}{\sum_{c'} \exp(-d(f_\phi(x_q), z_{c'}))}$, where $d(\cdot, \cdot)$ is usually the squared Euclidean distance in the embedding

space. In the context of diffusion models, class prototypes can be leveraged to guide the generative process, especially in scenarios where class-conditioning is required with limited labeled data. Instead of conditioning the diffusion process on textual prompts or class labels alone, prototype-guided diffusion injects semantic structure directly from few-shot support examples.

Let z_c denote the prototype for the target class c, and let x_t be a noisy sample at timestep t. The denoising model can be conditioned on z_c to produce a prototype-guided noise estimate $\epsilon_\theta(x_t, t, z_c)$. Following the classifier-free guidance strategy [21], the final guided noise estimate is computed as:

$$\epsilon_{\text{guided}}(x_t, t) = (1 + s) \cdot \epsilon_\theta(x_t, t, z_c) - s \cdot \epsilon_\theta(x_t, t), \tag{8}$$

where $\epsilon_\theta(x_t, t)$ is the unconditional noise prediction, and $s \geq 0$ is a guidance scale parameter controlling the strength of conditioning. When $s = 0$, the model generates unconditionally; larger values of s encourage outputs more aligned with the class prototype.

This formulation enables the diffusion model to generate samples that are consistent with the semantic identity of a class, even in few-shot or low-resource regimes. Moreover, prototype guidance introduces minimal overhead while providing strong inductive bias via non-parametric, data-driven class priors.

IV. Method

The Prototype Diffusion Model (PDM) is a generative model that leverages prototype learning to condition the denoising process in diffusion models. Unlike previous methods such as ProtoDiffusion, which rely on labeled datasets for supervised prototype assignment, PDM is trained in a fully unsupervised manner. This allows the model to discover and utilize semantic prototypes without the need for annotations, making it particularly well-suited for unlabeled data.

The model is composed of two jointly trained neural networks. The first, denoted f_ϕ, is a convolutional neural network responsible for feature extraction. Given an input image $x \in \mathbb{R}^{H \times W \times C}$, it produces a latent representation:

$$\hat{x} = f_\phi(x), \quad \hat{x} \in \mathbb{R}^D \tag{9}$$

A set of K prototypes $\{e_1, \ldots, e_K\} \subset \mathbb{R}^D$ is maintained as learnable parameters, randomly initialized at the start of training. For each image x, the closest prototype is selected based on Euclidean distance in the latent space:

$$e_x = \arg\min_{e_i} \|f_\phi(x) - e_i\|_2 \tag{10}$$

The second component of PDM is the denoising network f_θ, implemented as a U-Net [22], which takes as input a noisy image and is conditioned on the selected prototype. *To incorporate the prototype into the U-Net, cross-attention layers are added such that the prototype e_x acts as both key and value, while the query originates from the output of the previous block processing the noisy image.* This enables the network to guide the denoising process using the most relevant semantic concept.

Following the standard diffusion framework, Gaussian noise is added to the input image over T steps using the forward process:

$$\tilde{x}_t = \sqrt{\bar{\alpha}_t} x + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{11}$$

At each time step t, the denoising network receives $\tilde{x}_t$, the selected prototype e_x, and a time embedding $\gamma(t)$. The prototype is enriched with this temporal information before being passed to the attention mechanism:

$$\hat{\epsilon} = f_\theta(\tilde{x}_t, e_x + \gamma(t)) \tag{12}$$

The training objective of PDM includes several loss components. First, a contrastive loss encourages the feature extractor f_ϕ to bring the latent representation $f_\phi(x)$ close to its associated prototype while pushing it away from others:

$$\mathcal{L}_{\text{contrastive}} = -\log\left(\frac{\exp(-\tau\|f_\phi(x) - e_x\|_2^2)}{\sum_{k=1}^{K}\exp(-\tau\|f_\phi(x) - e_k\|_2^2)}\right) \tag{13}$$

In addition, an alignment loss directly penalizes the squared Euclidean distance between the image representation and the selected prototype:

$$\mathcal{L}_{\text{align}} = \|f_\phi(x) - e_x\|_2^2 \tag{14}$$

To ensure that the learned prototypes are diverse and semantically meaningful, a compactness regularization term is introduced. This term penalizes high similarity between different prototypes using cosine similarity:

$$\mathcal{L}_{\text{compact}} = \sum_{k \neq k'} \text{sim}(e_k, e_{k'}) \tag{15}$$

where $\text{sim}(e_k, e_{k'})$ denotes the cosine similarity between prototypes e_k and $e_{k'}$. This encourages the prototypes to capture distinct concepts rather than collapsing into similar representations.

The diffusion objective follows the standard DDPM loss, measuring the mean squared error between the predicted noise and the true noise added at each time step:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{x,t,\epsilon}\left[\|\epsilon - f_\theta(\tilde{x}_t, e_x + \gamma(t))\|_2^2\right] \tag{16}$$

The overall training loss for PDM combines all components:

$$\mathcal{L}_{\text{PDM}} = \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{compact}} \tag{17}$$

All loss components are combined with equal weighting, providing effective prototype guidance while maintaining simplicity (see Algorithm 1 for details).

During inference, PDM supports both conditional and unconditional generation. If an input image is provided, it is passed through f_ϕ to select the most appropriate prototype e_x, which is then used to condition the denoising process. If no image is provided, a prototype is selected at random. The complete inference process is summarized in Algorithm 2.

PDM can be readily adapted to a supervised context in which labeled data is available. We create and refer to this variant as *supervised Diffusion Prototype Model (s-PDM)*. In this

Algorithm 1: Training Prototype Diffusion Model (PDM)

Input: Dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, number of prototypes K, noise schedule $\{\alpha_t\}_{t=1}^T$, weight τ
Output: Trained networks f_ϕ, f_θ, and prototypes $\{e_1, \ldots, e_K\}$

Initialize feature extractor f_ϕ, denoiser f_θ, and prototypes $\{e_1, \ldots, e_K\} \subset \mathbb{R}^D$;
foreach *minibatch* $\{x_1, \ldots, x_B\} \subset \mathcal{D}$ **do**
  foreach $x_i \in \{x_1, \ldots, x_B\}$ **do** `// Feature encoding and prototype assignment`
    Compute $\hat{x}_i = f_\phi(x_i)$;
    Select prototype: $e_{x_i} = \arg\min_{e_k} \|\hat{x}_i - e_k\|_2$;
    Sample $t \sim \mathcal{U}\{1, T\}$, $\epsilon \sim \mathcal{N}(0, I)$;
    Compute $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$;
    Compute $\tilde{x}_i = \sqrt{\bar{\alpha}_t} x_i + \sqrt{1 - \bar{\alpha}_t}\epsilon$;
    Predict $\hat{\epsilon}_i = f_\theta(\tilde{x}_i, e_{x_i} + \gamma(t))$;
  Compute loss terms:
  $\mathcal{L}_{\text{contrastive}} = -\frac{1}{B}\sum_{i=1}^B \log\left(\frac{\exp(-\tau\|\hat{x}_i - e_{x_i}\|_2^2)}{\sum_{k=1}^K \exp(-\tau\|\hat{x}_i - e_k\|_2^2)}\right)$;
  $\mathcal{L}_{\text{align}} = \frac{1}{B}\sum_{i=1}^B \|\hat{x}_i - e_{x_i}\|_2^2$;
  $\mathcal{L}_{\text{compact}} = \beta \sum_{k \neq k'} \text{sim}(e_k, e_{k'})$;
  $\mathcal{L}_{\text{diff}} = \frac{1}{B}\sum_{i=1}^B \|\epsilon_i - \hat{\epsilon}_i\|_2^2$;
  Combine losses:
  $\mathcal{L}_{\text{PDM}} = \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{align}} + \mathcal{L}_{\text{compact}}$;
  Update $f_\phi, f_\theta, \{e_k\}$ via gradient descent on $\mathcal{L}_{\text{PDM}}$;
return $f_\phi, f_\theta, \{e_k\}_{k=1}^K$;

Algorithm 2: Inference in Prototype Diffusion Model (PDM)

Input: Prototype set $\{e_1, \ldots, e_K\}$, trained feature extractor f_ϕ, denoiser f_θ, optional image x, noise schedule $\{\alpha_t\}_{t=1}^T$
Output: Generated image $\hat{x}_0$

if *image* x *is provided* **then**
  Compute $\hat{x} = f_\phi(x)$;
  Select prototype: $e_x = \arg\min_{e_i} \|\hat{x} - e_i\|_2$;
else
  Sample prototype $e_x \sim \{e_1, \ldots, e_K\}$;
Initialize $\tilde{x}_T \sim \mathcal{N}(0, I)$;
for $t \leftarrow T$ **to** 1 **do** `// Iterative denoising`
  Compute $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$;
  Compute $\hat{\epsilon}_t = f_\theta(\tilde{x}_t, e_x + \gamma(t))$;
  Compute $\sigma_t = \sqrt{1 - \alpha_t}$, sample $z \sim \mathcal{N}(0, I)$;
  Update:
$$\tilde{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\tilde{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\hat{\epsilon}_t\right) + \sigma_t z$$
return $\hat{x}_0 = \tilde{x}_0$;

case, each class is associated with a fixed prototype. Therefore, during training, prototype selection is unnecessary since the correct prototype is known. Moreover, the compactness loss $\mathcal{L}_{\text{compact}}$ is no longer needed, as inter-class semantic separation is enforced by the labels themselves. The resulting supervised loss simplifies to:

$$\mathcal{L}_{\text{PDM-supervised}} = \mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{align}} \tag{18}$$

The inference procedure remains unchanged, conditioned either on a labeled image or a sampled prototype for unconditional generation. This flexibility allows PDM to operate seamlessly in both supervised and unsupervised settings.

V. Experiments

We evaluate PDM and its supervised variant s-PDM on CIFAR-10 [23], STL-10 [24], EuroSAT [25], and Tiny ImageNet [26] using two NVIDIA A100 GPUs.

A. Experimental Setup

All experiments use a compact U-Net with channel configuration `[128, 256, 256, 256]`. Prototypes are injected at the bottleneck through a single cross-attention block, enabling global semantic conditioning with minimal computational cost. The feature extractor f_ϕ is a lightweight 4-layer CNN with adaptive average pooling.

Following common practice in diffusion studies, we intentionally focus on low-resolution benchmarks to isolate and analyze the effect of prototype-guided conditioning independently of scale and architectural complexity.

We compare our models to two baselines: (i) a standard DDPM (no prototypes) and (ii) ProtoDiffusion, where prototypes are learned in a separate stage and kept frozen. These comparisons allow us to isolate the effect of *joint prototype learning during diffusion*, which is the primary novelty of PDM. Importantly, our goal is not to surpass state-of-the-art diffusion performance, but to demonstrate that **learning prototypes jointly with the denoiser leads to systematically improved generative quality**.

Performance is reported using Inception Score (IS) [27], Fréchet Inception Distance (FID), and Kernel Inception Distance (KID) [28].

B. Quantitative Evaluation

Table I shows that both PDM and s-PDM outperform the baselines across all datasets and metrics. The gains are especially clear for FID and KID, which require matching the real data distribution.

Key finding: jointly learning prototypes within the diffusion process (PDM) consistently improves generative fidelity over DDPM and ProtoDiffusion. This confirms that prototypes must remain trainable and aligned with the evolving denoiser features; otherwise, as in ProtoDiffusion, they gradually lose relevance.

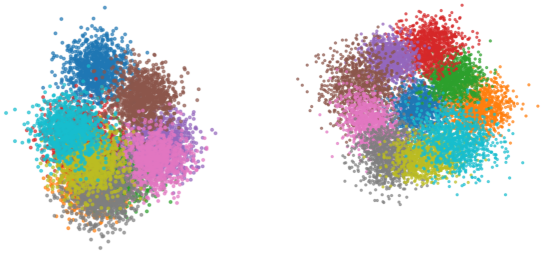
s-PDM obtains the strongest results overall due to label-informed prototype assignment. Its advantage is most visible

Method	CIFAR-10			STL-10			EuroSAT			Tiny ImageNet		
	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓	IS ↑	FID ↓	KID ↓
DDPM	7.12	18.45	0.021	7.85	34.20	0.038	3.75	28.47	0.018	9.05	48.22	0.08
ProtoDiffusion	**8.50**	11.70	0.009	9.40	22.70	0.015	4.45	15.20	0.010	10.55	31.00	0.04
PDM (unsup.)	8.35	8.10	0.007	10.28	21.30	0.011	4.40	13.50	0.008	11.20	25.40	0.03
s-PDM (sup.)	8.20	**6.58**	**0.004**	**11.77**	**20.32**	**0.007**	**4.52**	**11.29**	**0.005**	**11.65**	**23.00**	**0.02**

TABLE I: Comparison of DDPM, ProtoDiffusion, PDM, and s-PDM across multiple datasets. Best values are in **bold**.

on FID/KID, where class-consistent conditioning directly improves distributional accuracy. In contrast, IS does not rely on real samples and favors diversity, which slightly benefits the unsupervised PDM.

Figure 1 illustrates this difference: PCA of f_ϕ features reveals sharper class separation in s-PDM, while PDM forms compact but unlabeled semantic groups. This confirms that supervised prototypes enforce stronger structure in latent space.



(a) PDM (b) s-PDM

Fig. 1: PCA of f_ϕ features on CIFAR-10. s-PDM shows clearer class separation, while PDM forms semantic clusters without labels.

C. Effect of the Number of Prototypes

To assess the importance of prototype granularity, we vary the number of prototypes in PDM while keeping all other settings fixed.

Prototypes	IS ↑	FID ↓	KID ↓
3	7.50	11.90	0.010
5	7.85	10.20	0.009
7	8.10	9.10	0.008
10	8.35	8.10	0.007
13	8.42	7.95	0.006
15	8.48	7.85	0.006
20	8.20	7.70	0.006

TABLE II: Ablation on the number of prototypes (CIFAR-10).

Results (Table II) show that using fewer prototypes than the dataset's intrinsic semantic modes reduces performance, as multiple classes collapse into a single centroid. Increasing prototypes beyond the number of classes yields marginal gains but eventually causes over-fragmentation. These findings confirm that prototypes should match the dataset's semantic granularity—a desirable property for adaptive or continual generative modeling.

D. Visualization of Generated Samples

Figure 2 presents randomly generated 2×2 image grids from four models trained on STL-10: DDPM, ProtoDiffusion, PDM, and s-PDM. All four models are shown side by side to emphasize differences in semantic coherence and prototype-guided generation.

These visualizations confirm that jointly learned prototypes substantially improve generation quality. While DDPM and ProtoDiffusion produce plausible images, they lack semantic consistency and class-specific alignment. PDM, even without labels, captures coherent structures and realistic textures, demonstrating that prototypes learned online can effectively guide the denoiser. s-PDM further enhances sharpness and class alignment by leveraging label-informed prototypes. Together, these results illustrate the central contribution of our work: *learning prototypes jointly with the diffusion model provides strong semantic conditioning that improves both visual fidelity and structural coherence*.

VI. Conclusion

We presented the *Prototype Diffusion Model* (PDM) and its supervised variant *s-PDM*, which integrate prototype learning directly into the diffusion process. By jointly updating prototypes with the denoiser, PDM provides label-free semantic guidance without external memory or retrieval, while s-PDM leverages class labels for improved FID/KID performance through stable, class-specific prototypes.
Experiments demonstrate that prototype-guided conditioning consistently enhances generative fidelity: s-PDM achieves the strongest quantitative results, and PDM offers a competitive, annotation-free alternative. Ablations confirm the importance of aligning the number of prototypes with dataset semantics.
Future work includes extending prototype-guided diffusion to **text-to-image** tasks, exploring semi- and weakly supervised variants, and developing hierarchical or adaptive prototypes for complex and evolving data distributions.

Acknowledgments

This work has received funding from the European Commission's Horizon Europe research and innovation programme under the Marie Sklodowska-Curie Actions (MSCA) grant agreement No. 101236749.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

(a) DDPM (b) ProtoDiffusion (c) PDM (unsupervised) (d) s-PDM (supervised)

Fig. 2: Random 2×2 sample generations from models trained on STL-10 after 500 training epochs. Baseline models (DDPM and ProtoDiffusion) produce plausible but less semantically coherent images. Prototype-guided methods (PDM and s-PDM) demonstrate stronger semantic structure, with s-PDM further improving class consistency and visual fidelity. This highlights the impact of jointly learning prototypes on guiding the diffusion process.

[2] F. Mazé and F. Ahmed, "Diffusion models beat gans on topology optimization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 8, 2023, pp. 9108–9116.

[3] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[4] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models." in *International Conference on Learning Representations (ICLR)*, 2021.

[5] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Advances in neural information processing systems*, vol. 35, pp. 36 479–36 494, 2022.

[6] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 461–11 471.

[7] Y. Zhou, B. Liu, Y. Zhu, X. Yang, C. Chen, and J. Xu, "Shifted diffusion for text-to-image generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 10 157–10 166.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[9] A. Blattmann, R. Rombach, K. Oktay, J. Müller, and B. Ommer, "Retrieval-augmented diffusion models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 309–15 324, 2022.

[10] A. Radford, J. W. 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*. PMLR, 2021, pp. 8748–8763.

[11] G. Baykal, H. F. Karagoz, T. Binhuraib, and G. Unal, "Protodiffusion: Classifier-free diffusion guidance with prototype learning," in *Asian Conference on Machine Learning*. PMLR, 2024, pp. 106–120.

[12] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, "Consistency models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 32 211–32 252.

[13] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "SDXL: improving latent diffusion models for high-resolution image synthesis," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[14] Z. Huang, L. Yang, X. Zhou, C. Qin, Y. Yu, X. Zheng, Z. Zhou, W. Zhang, Y. Wang, and W. Yang, "Interaction-based retrieval-augmented diffusion models for protein-specific 3d molecule generation," in *Forty-first International Conference on Machine Learning*, 2024.

[15] Y. Du, Z. Xiao, S. Liao, and C. Snoek, "Protodiff: Learning to learn prototypical networks by task-guided diffusion," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[16] K. Doan, Q. Tran, T. L. Tran, T. Nguyen, D. Phung, and T. Le, "Class-prototype conditional diffusion model with gradient projection for continual learning," *arXiv preprint arXiv:2312.06710*, 2023.

[17] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International conference on machine learning*. pmlr, 2015, pp. 2256–2265.

[18] Z. Hu, A. Iscen, C. Sun, Z. Wang, K.-W. Chang, Y. Sun, C. Schmid, D. A. Ross, and A. Fathi, "Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 23 369–23 379.

[19] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[21] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[22] O. Ronneberger, P. Fischer, and T. 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*. Springer, 2015, pp. 234–241.

[23] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[24] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[25] P. Helber, B. Bischke, A. Dengel, and D. Borth, "Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2217–2226, 2019.

[26] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[27] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in neural information processing systems*, vol. 29, 2016.

[28] M. Binkowski, D. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," in *International Conference on Learning Representations (ICLR)*, 2018.