EVLF: Early Vision-Language Fusion for Generative Dataset Distillation Supplementary Material

Wenqi Cai[1] Yawen Zou[1] Guang Li[2] Chunzhi Gu[3] Chao Zhang[1]
[1]University of Toyama [2]Hokkaido University [3]University of Fukui

1. Training Details for Early Fusion

For the Early Fusion configuration, we used the VAE encoder pretrained with Stable Diffusion v1.5 (runwayml/stable-diffusion-v1-5) as a frozen feature extractor to obtain image embeddings. To reduce computational overhead, all latent representations were processed with automatic mixed precision in `fp16` during training.

Training jointly optimized two learnable components: a fusion module that integrates visual and textual representations, and a latent projector that aligns the fused latent with the text embedding space. Both components were trained with the AdamW optimizer. We used a total batch size of 64 with gradient accumulation over two steps to realize an effectively larger batch size. All experiments were conducted on two NVIDIA RTX A5000 GPUs in a distributed setting using the Hugging Face `Accelerate` library and PyTorch 2.3.

At each iteration, the frozen VAE encoder produced the latent mean of the input image, while the text description was tokenized and embedded by a pretrained CLIP text encoder. The embedding corresponding to the end-of-text token was taken as the pooled text representation. To improve robustness, 20% of the samples were randomly replaced with empty text embeddings, and token dropout with a rate of 10% was applied to the remaining samples, followed by Gaussian noise injection.

The fusion module combined the image and text embeddings to produce a fused latent representation, which was then projected into the text feature space by the latent projector for contrastive alignment. The total loss consisted of two components: a symmetric $\mathcal{L}_{\text{InfoNCE}}$ term enforcing bidirectional alignment between image and text features, and an $\mathcal{L}_{\text{MSE}}$ reconstruction term preserving the visual structure of the latent space. The overall objective was defined as a weighted sum of these losses, where the reconstruction loss weight was linearly increased during training. Gradient clipping with a maximum norm of one was applied before each optimization step to ensure stable convergence.

The training hyperparameters are summarized in Table 1.

Table 1. Training hyperparameters for the Early Fusion configuration.

Parameter	Value
Training Epochs	4
Batch size	64
Gradient accumulation steps	2
Optimizer	AdamW
Learning rate (Fusion module)	3e-4
Learning rate (Latent projector)	1e-4
Weight decay	1e-2

Table 2. Training hyperparameters for Denoiser fine-tuning.

Parameter	Value
Training Epochs	8
Validation Epochs	2
Batch size	8
Optimizer	AdamW
Learning rate	1e-5
Weight decay	1e-2
Augmentation	Center crop, Random flip

2. Denoiser Fine-Tuning

In this study, we fine-tune the pretrained Stable Diffusion model (runwayml/stable-diffusion-v1-5) using the fused embeddings produced by our Early Fusion module together with the corresponding label text information. The fine-tuning procedure follows the standard Stable Diffusion training pipeline, and all datasets are trained according to the hyperparameters summarized in Table 2.

The input resolution depends on the dataset: 512 for ImageNet-1K and its subsets (ImageWoof, ImageNette, ImageIDC, ImageNet-100), 32 for CIFAR-10/100, and 64 for Tiny-ImageNet. All training is conducted on two NVIDIA RTX A5000 GPUs in a distributed setup using the Hugging Face `Accelerate` library with PyTorch 2.4.1+cu121.

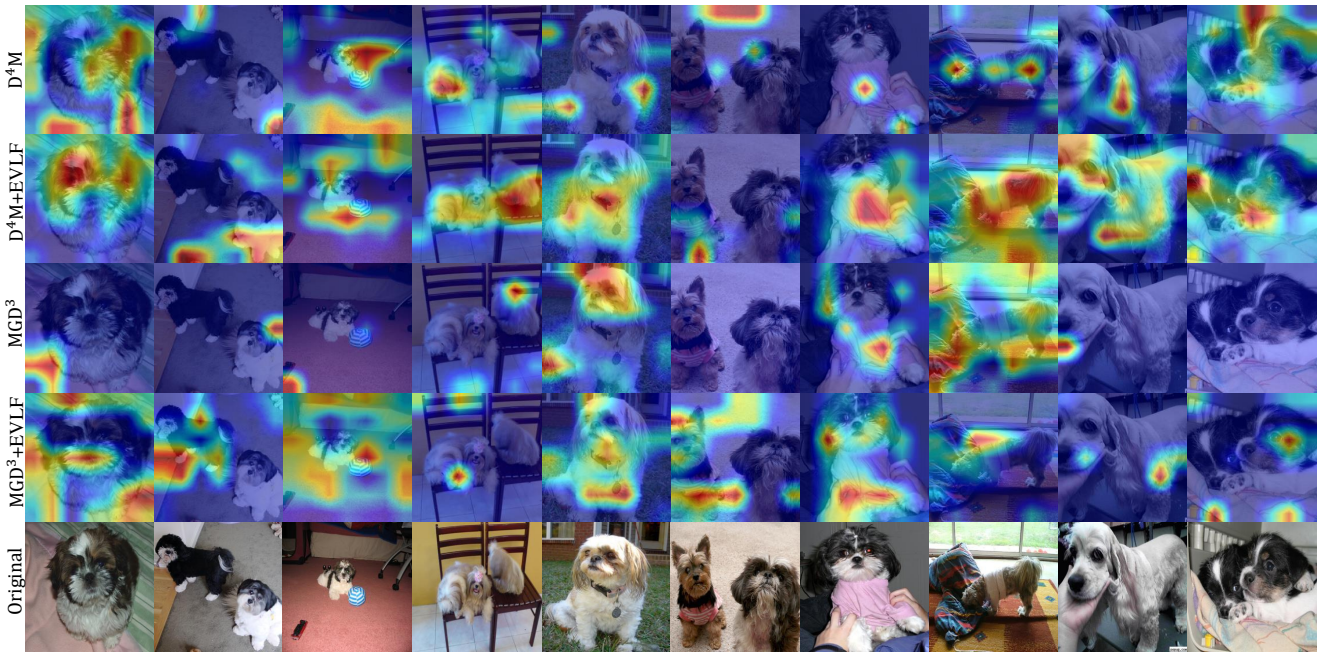


Figure 1. Grad-CAM visualizations for the Shih-Tzu class from the ImageWoof validation set. Each row shows attention maps of models trained on datasets generated by different methods, with the last row displaying the original images. When integrated with our method, D^4M exhibits more concentrated attention on the target object, whereas MGD^3 captures a broader and more complete region of the object.

3. Details of the Generation Pipeline

In this section, we provide a comprehensive description of the entire generation pipeline, including both architectural and implementation details. Specifically, we outline (1) the pretrained models employed in our framework and the sources from which they are obtained, (2) the design and configuration of our early fusion modules, and (3) the step-by-step procedures through which different methods generate their distilled datasets after Early Fusion is applied.

Pretrained Models. All pretrained components used in our generation pipeline are derived from Stable Diffusion v1.5 (runwayml/stable-diffusion-v1-5). Specifically, we adopt its Variational Autoencoder (VAE), consisting of both an encoder and a decoder, as well as its pretrained text encoder. For the denoiser, however, we follow the architectures used in the respective baseline methods that our framework integrates with: D^4M employs a U-Net–based denoiser, whereas MGD^3 uses a DiT-based denoiser.

Early Fusion Module. To integrate semantic information into the latent space in a structured and controllable manner, we introduce the Early Vision–Language Fusion (EVLF) module, which consists of two components: a cross-attention fusion block and a latent projector. The early fusion module first maps the VAE latent features and text embeddings into a shared embedding space and augments the latent tokens with learnable positional encodings. A stack of multi-head cross-attention layers is then applied, where the latent tokens serve as queries and both latent and textual tokens serve as keys and values. This design enables the model to inject text-conditioned semantics directly into the spatial latent representation while preserving its structural coherence. Following attention refinement, the fused embedding is projected back into the original latent dimensionality.

Complementing this mechanism, the *LatentProjector* provides a compact global representation of the latent features. It applies Gaussian smoothing to reduce high-frequency artifacts, followed by a strided convolution and adaptive pooling to derive a spatially compressed latent map. The resulting features are processed by a linear projection and normalization layer, producing a 768-dimensional vector aligned with the text-encoder embedding space. This global descriptor serves as an additional semantic signal that stabilizes the fusion process and improves conditioning quality. Together, these two components form the EVLF module, which effectively embeds text-conditioned semantic cues into the latent space before downstream dataset-distillation procedures are applied.

Downstream Dataset-Distillation Procedures. To clarify how different baseline methods produce the final distilled datasets within our pipeline, we describe the down-

stream generation procedures for D^4M and MGD^3 when operated after EVLF. In both cases, EVLF outputs a fused latent tensor that carries spatially preserved visual features together with text-conditioned semantic cues. This fused latent serves as the primary input to the following generative routines.

For D^4M, we first apply class-wise k-means clustering to the fused latent representations. For each class, the number of cluster centers is set to match the target IPC (images per class). These cluster centers serve as class-specific latent *prototypes*, which act as the starting latent codes from which the final synthetic images are generated. Each center is then processed using D^4M's original U-Net denoising and sampling procedure without modification to its noise schedule or sampling hyperparameters. The refined latents are subsequently decoded by the pretrained VAE decoder to produce RGB samples, which are directly collected as the distilled dataset.

For MGD^3, we apply an analogous class-wise k-means step on the fused latents. The obtained cluster centers also match the target IPC, but here they are interpreted as distinct *modes* within each class. Instead of using the prototypes as initial latents, MGD^3 generates images by sampling pure Gaussian noise and denoising it using its DiT-based denoiser. During the sampling trajectory, the model is guided toward the corresponding class modes, effectively pulling the generated samples toward these mode centers. After denoising convergence, the resulting latents are decoded via the shared VAE decoder to yield the final distilled dataset.

4. Overhead Analysis

To verify that the proposed EVLF module introduces only minimal computational burden, we assess its overhead in terms of parameter count and the cost of its training stage. The EVLF module contains 1.52M parameters, which is negligible compared with the pretrained components inherited from Stable Diffusion v1.5, namely the VAE encoder (83.64M) and the text encoder (123.06M). In total, EVLF accounts for only 0.73% of the combined parameter count of these pretrained backbones, indicating that the additional model capacity introduced by our method is minimal.

The training cost of EVLF is also modest. The CrossAttentionFusion module requires approximately 9 minutes per epoch under a two-GPU distributed data parallel setting, and it converges within 4 epochs, yielding a total training time of roughly 36 minutes. Since this one-time training is performed only once and the resulting module is reused across all downstream dataset distillation pipelines, the added overhead is marginal relative to the overall computational demands of diffusion-based distillation frameworks.

5. Visualization of Attention Maps

To verify that models trained on our synthesized datasets capture more discriminative and semantically relevant features, we use Grad-CAM to visualize attention maps for validation samples from the Shih-Tzu class in the ImageWoof dataset during inference. Specifically, we compare the attention responses of the ResNetAP-10 model trained on datasets distilled by prior methods (D^4M and MGD^3) with those of the same model trained on datasets distilled by these methods when enhanced with our proposed approach.

As shown in Fig. 1, the model trained on the dataset synthesized by D^4M not only attends to the target object but also exhibits noticeable activation in background regions, indicating a less discriminative focus. In contrast, our method effectively suppresses these redundant responses and concentrates attention on the target area. For the model trained on the dataset generated by MGD^3, the attention tends to cover only a small portion of the target object. When combined with our method, however, the model develops a more complete and semantically consistent focus on the entire object, leading to improved classification confidence.

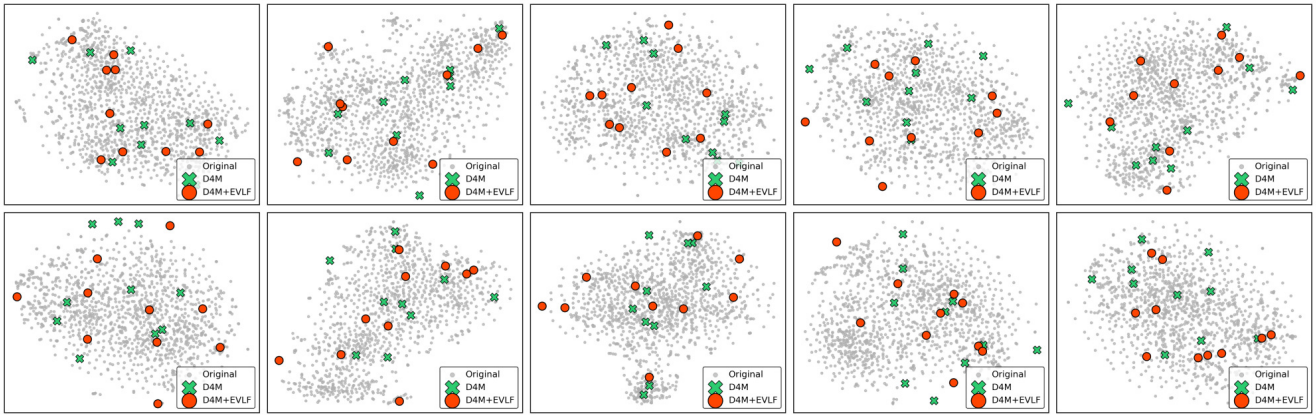
6. t-SNE Visualization

To further analyze the distributional coverage and diversity of the generated samples, we use t-SNE visualizations to examine how integrating our method affects the synthetic data produced by D^4M and MGD^3 across all ten classes of the ImageIDC dataset. As shown in Fig. 2, both MGD^3 and D^4M tend to produce tightly clustered samples, indicating limited diversity in the generated data. Such concentration hinders the model's ability to learn representative features from the distilled datasets, especially under low-IPC settings. In contrast, after integrating EVLF, both the coverage and diversity of the generated samples improve noticeably. This demonstrates that our method effectively alleviates the over-correction issue, enabling the distilled dataset to capture richer visual information from the original data rather than merely mirroring the prompt pattern.

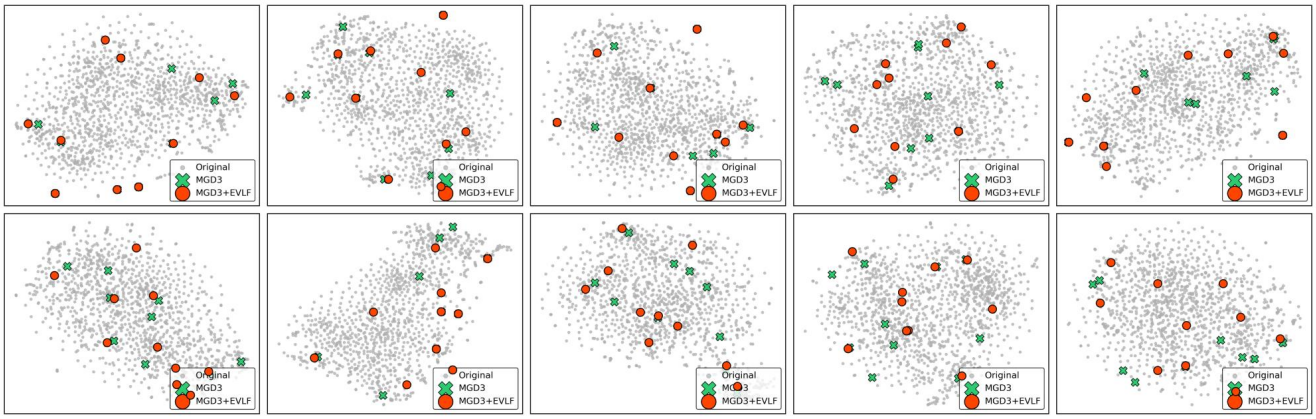
7. Experiments on ImageNet-100

We conduct experiments on ImageNet-100 to compare all methods under different IPC settings and model architectures. ImageNet-100 is constructed by selecting 100 representative categories from ImageNet-1K, providing a balanced and diverse benchmark for evaluating image classification models. As shown in Tab. 3, incorporating our method into MGD^3 consistently improves performance across all IPC settings and architectures. Notably, at IPC = 20, our approach surpasses all previous methods.

These results demonstrate that our method remains effective on ImageNet-100, which contains fine-grained and



(a) Comparison of D^4M and D^4M+EVLF



(b) Comparison of MGD^3 and MGD^3+EVLF

Figure 2. t-SNE comparison on ImageIDC. (a) Samples generated by D^4M and D^4M+EVLF. (b) Samples generated by MGD^3 and MGD^3+EVLF. Both D^4M and MGD^3 tend to produce tightly clustered samples, resulting in limited coverage and diversity. In contrast, incorporating EVLF yields more dispersed and well-separated samples, indicating improved coverage of the original feature space.

Table 3. Dataset distillation results on ImageNet-100.

IPC (Ratio)	Test Model	Random	Herding	IDC-1	Minimax	MGD^3	**MGD^3+EVLF**	Full
10 (0.8%)	ConvNet-6	17.0±0.3	17.2±0.3	24.3±0.5	22.3±0.5	23.4±0.9	**24.4±0.4**	79.9±0.4
	ResNetAP-10	19.1±0.4	19.8±0.3	25.7±0.1	24.8±0.2	25.8±0.5	**26.5±1.0**	80.3±0.2
	ResNet-18	17.5±0.2	16.1±0.2	**25.1±0.2**	22.5±0.3	23.6±0.4	24.6±0.4	81.8±0.7
20 (1.6%)	ConvNet-6	24.8±0.2	24.3±0.4	28.8±0.3	29.3±0.4	30.6±0.4	**31.7±0.6**	79.9±0.4
	ResNetAP-10	26.7±0.5	27.6±0.1	29.9±0.2	32.3±0.1	33.9±1.1	**34.2±0.5**	80.3±0.2
	ResNet-18	25.5±0.3	24.7±0.1	30.2±0.2	31.2±0.1	32.6±0.4	**33.4±0.1**	81.8±0.7

visually similar classes that pose significant challenges for dataset distillation. By aligning textual and visual representations through the fused latent design, our approach preserves class-discriminative semantics and suppresses overcorrection during synthesis. Consequently, it produces high-quality, semantically consistent samples that generalize well across different architectures and data regimes.

8. Visualization of Generated Samples

Visualizations of the distilled datasets generated by each method are shown in the following figures: ImageWoof (Fig. 3), ImageIDC (Fig. 4), and ImageNette (Fig. 5). Unlike the attention maps and t-SNE visualizations, which highlight how the distilled datasets influence model training and reflect their coverage of the original data distribu-

Figure 3. Comparison of various methods on the ImageWoof dataset.

Figure 4. Comparison of various methods on the ImageIDC dataset.

tion, these figures focus purely on the visual appearance of the synthesized samples. By directly presenting the images generated by each method, they allow a clearer comparison of the perceptual quality and fidelity of the distilled datasets. Through comparison, we observe that D^{4}M tends to produce images that capture only partial texture cues and often fail to achieve high visual fidelity. MGD^3, in contrast, tends to generate samples that exhibit structural inconsistencies and artifact-like distortions. When incorporated with our method, the synthesized images achieve superior detail representation and structural alignment, yielding an overall improvement in visual fidelity.

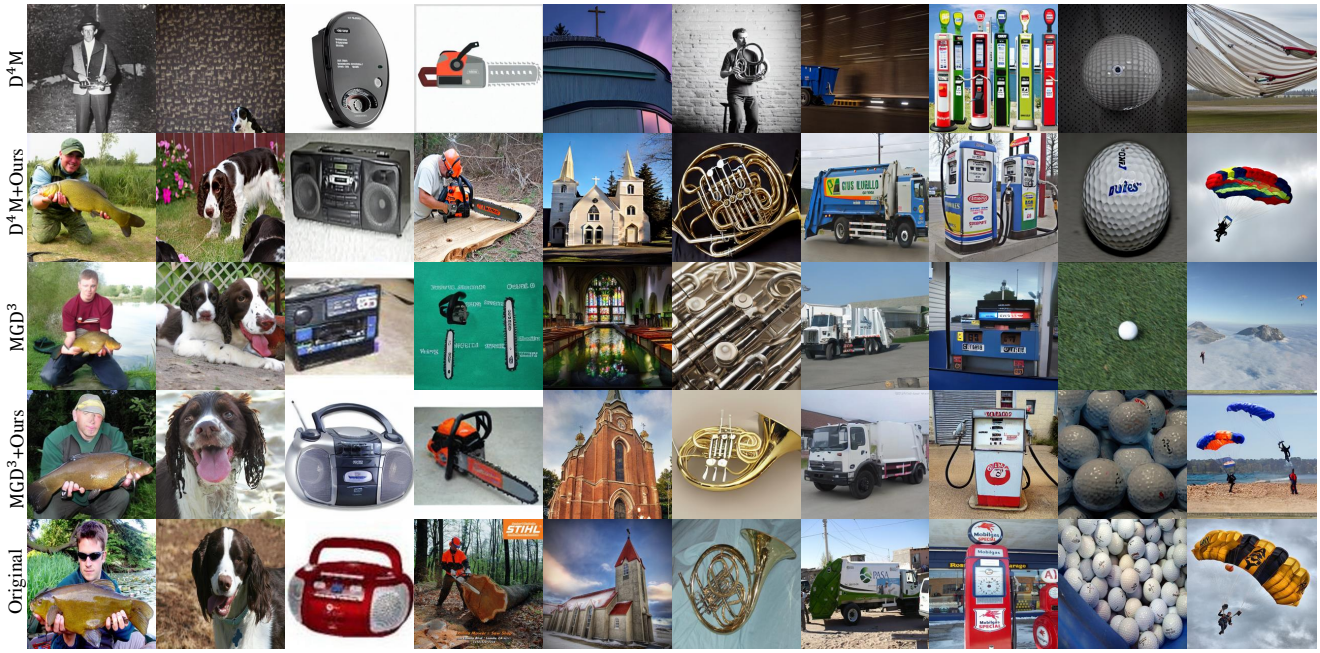


Figure 5. Comparison of various methods on the ImageNette dataset.