A VAE

A.1 2D-VAE QUANTIZATION

Variational Autoencoders (VAEs) enable significant data compression by encoding each image as a probability distribution in a learned latent space, having the architecture like in 2. The 2D-VAE used in this paper optimizes the following loss function:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - D_{\text{KL}}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p_\theta(\mathbf{z})\right) \tag{5}$$

The first term minimizes the reconstruction loss when decoding the latent representation of an image, while the second term, the KL divergence, ensures each encoded distribution aligns with a normal prior distribution. Combined, the objective balances the quality of decoded images and the smoothness of the latent distribution.

In order to ensure a fair comparison with previous work, the weights of the VAE are quantized through post-training static quantization, reducing the bid-width from 32 to 8 bits:

$$x_q = \text{round}\left(\frac{x}{s}\right) + z \tag{6}$$

Where s is the scaling factor, and z is the zero point.

By applying linear quantization, the size of the pre-trained model is reduced to one-fourth of its original size. Empirically, the quantized VAE continues to yield high accuracy during experimentation. Compared to other methods such as quantization-aware training, static quantization has the advantage of retaining a high level of accuracy while offering lower computational complexity during the quantization phase.

B IMPLEMENTATION DETAILS

Our code can be accessed by:`https://anonymous.4open.science/r/Latent_Video_Dataset_Distillation-AFF3` In this section, we provide implementation details of our experiments, including the computing infrastructure, the selection of VAEs, the preprocessing steps applied to video datasets, and the measures taken to ensure a fair comparison.

B.1 COMPUTING INFRASTRUCTURE

All experiments were conducted on an NVIDIA H100 SXM GPU with 94 GB of memory, running on a Linux operating system. The versions of all relevant libraries and frameworks are documented in the anonymous GitHub repository. Please note that the reported GPU model and memory reflect the hardware used in our experiments, not the minimum required to reproduce the results.

B.2 ADDITIONAL VAE SELECTION

We have adopted and quantized SD-VAE-FT-MSE(StabilityAI, n.d.) and CV-VAE(Zhao et al., 2025) in our experiments. The variational autoencoders are used to encode video sequences into a compact latent space, enabling efficient dataset distillation. When dealing with IPC 1, where storage constraints are particularly strict, we employ SD-VAE-FT-MSE, a 2D-VAE, which compresses videos as independent frames, allowing for highly compact storage. In contrast, for IPC 5, we utilize CV-VAE, a 3D-VAE, which explicitly models temporal dependencies in video sequences. Unlike 2D-VAEs, which treat frames as separate entities, 3D-VAEs capture motion continuity and temporal redundancy, effectively reducing redundant information across consecutive frames. This results in a more structured latent representation, ensuring that only the most informative motion features are retained, leading to improved efficiency in video dataset distillation. This selective choice of VAE architectures ensures that our distilled datasets achieve the optimal balance between compression efficiency and information retention across different IPC levels.

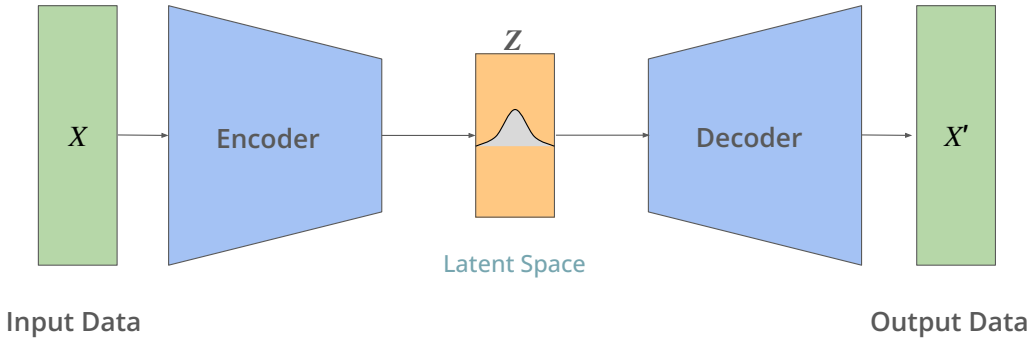


Figure 2: Architecture of Variational Autoencoder(VAE).

B.3 Quantized VAE Model Size

We apply post-training static quantization on SD-VAE-FT-MSE, compressing the model from original 335MB to 80MB, achieving around 76% compression rate.

B.4 Fair Comparison

Throughout our experiments across four video datasets under two IPC settings (1 and 5), we rigorously ensure that the storage used by our method does not exceed predefined storage constraints. For example, in MiniUCF IPC 1, previous methods allocate a storage limit of 115MB. Under the same setting, we sample 24 instances per class and apply HOSVD with a compression rate of 0.75, saving the core tensor and factor matrices. The resulting distilled dataset occupies 27MB, while the quantized 2D-VAE requires 80MB, leading to a total memory consumption of 107MB, which remains within the 115MB storage budget. The detailed storage consumption can be found in Tab. 7.

Dataset	MiniUCF	HMDB51	Kinetics-400	SSv2
IPC 1	107 MB	107 MB	148 MB	223 MB
IPC 5	475 MB	475 MB	455 MB	458 MB

Table 7: Storage consumed by our method for each dataset. Storage represents the total size of the distilled tensors and the associated VAE model.

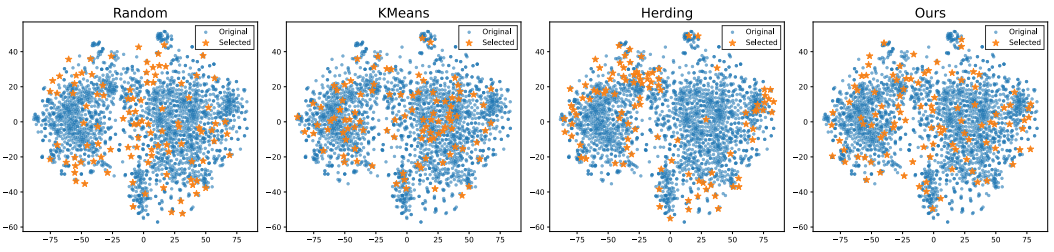


Figure 3: Distributions of sampled videos in the latent space.

B.5 Sampling Methods

We evaluate the impact of different sampling strategies on dataset distillation, comparing our Diversity-Aware Data Selection using Determinantal Point Processes (DPPs) against various selection methods: Random, Herding, K-Center(Sener & Savarese, 2018), Kmeans, Kmeans + Kernel

Density Estimation(KDE), and Mini-Batch KMeans. As shown in Tab. 8, our method achieves the highest performance, demonstrating the effectiveness of DPP-based selection in video dataset distillation.

These results highlight the importance of an effective data selection strategy in video dataset distillation. Our approach leverages DPPs to maximize diversity while retaining representative samples, leading to superior generalization in downstream tasks.

Dataset	MiniUCF		Kinetics-400	
IPC	1	5	1	5
Random	33.4	37.2	7.2	11.3
Herding	<u>34.6</u>	37.2	<u>7.3</u>	<u>12.4</u>
K-Center	33.5	<u>38.8</u>	6.8	11.5
KMeans	31.8	35.1	7.0	11.0
KMeans+KDE	29.9	38.0	7.2	12.1
Mini-Batch KMeans	34.3	36.4	6.7	11.2
Ours	**34.8**	**41.1**	**9.0**	**13.8**

Table 8: Results of different sampling methods in latent space.

B.6 Diversity Analysis

We visualize our selected data together with three other sampling methods (Random, KMeans, Herding) using t-SNE in Fig. 3. KMeans relies on distance metrics, which usually struggles in high-dimensional space, while Herding has limited diversity due to overlooking underrepresented regions of the data distribution. As shown in Fig. 3, our method does not sacrifice diversity for performance. In fact, it achieves a more balanced coverage of encoded videos in the latent space, highlighting its ability to retain representative and diverse samples that are crucial for generalization.

B.7 Effect of Latent Compression

In Fig. 9, we have provided a detailed comparison between our full method, DPPs-only approach, and prior methods evaluating on the dataset SSv2 when IPC is 5.

DM + VDSD	MTT + VDSD	IDTD	DPPs only	Ours
4.0 ± 0.1	8.3 ± 0.1	9.5 ± 0.3	9.3 ± 0.1	$\mathbf{10.5 \pm 0.2}$

Table 9: Performance of different dataset distilation and data sampling methods on the SSv2 dataset under IPC 1.

C Peak Memory Analysis

To assess the efficiency of our method in terms of memory consumption, we compare the peak GPU memory usage during dataset distillation with other methods: DM and VDSD. As shown in Tab. 10, our method achieves the lowest peak memory consumption at 11,085 MiB, significantly reducing memory usage compared to DM (20,457 MiB) and VDSD (12,545 MiB).

Method	DM	VDSD	Ours
GPU Memory	$20,457$ MiB	$12,545$ MiB	$11,085$ MiB

Table 10: Peak memory comparsion between different dataset distillation methods on MiniUCF when IPC is 5.

Our method minimizes peak memory usage by operating in the latent space and leveraging training-free compression via HOSVD, significantly reducing redundant memory allocation during dataset distillation. This lower memory footprint allows our approach to scale to larger datasets and higher IPC settings while maintaining efficiency.

D RUNTIME ANALYSIS

To assess the computational efficiency of our method, we compare its distillation runtime with VDSD across different datasets. All experiments are conducted on an NVIDIA H100 SXM GPU. Our training-free method demonstrates a significant speed advantage, particularly on large-scale datasets, due to its latent-space processing and training-free compression strategy.

On small-scale datasets, such as HMDB51 and MiniUCF, our method completes the dataset distillation process in under 10 minutes, whereas VDSD requires 2.5 hours. The efficiency gain is even more pronounced on large-scale datasets, where our method finishes in approximately 1 hour on Kinetics-400 and SSv2, while VDSD exceeds 5 hours.

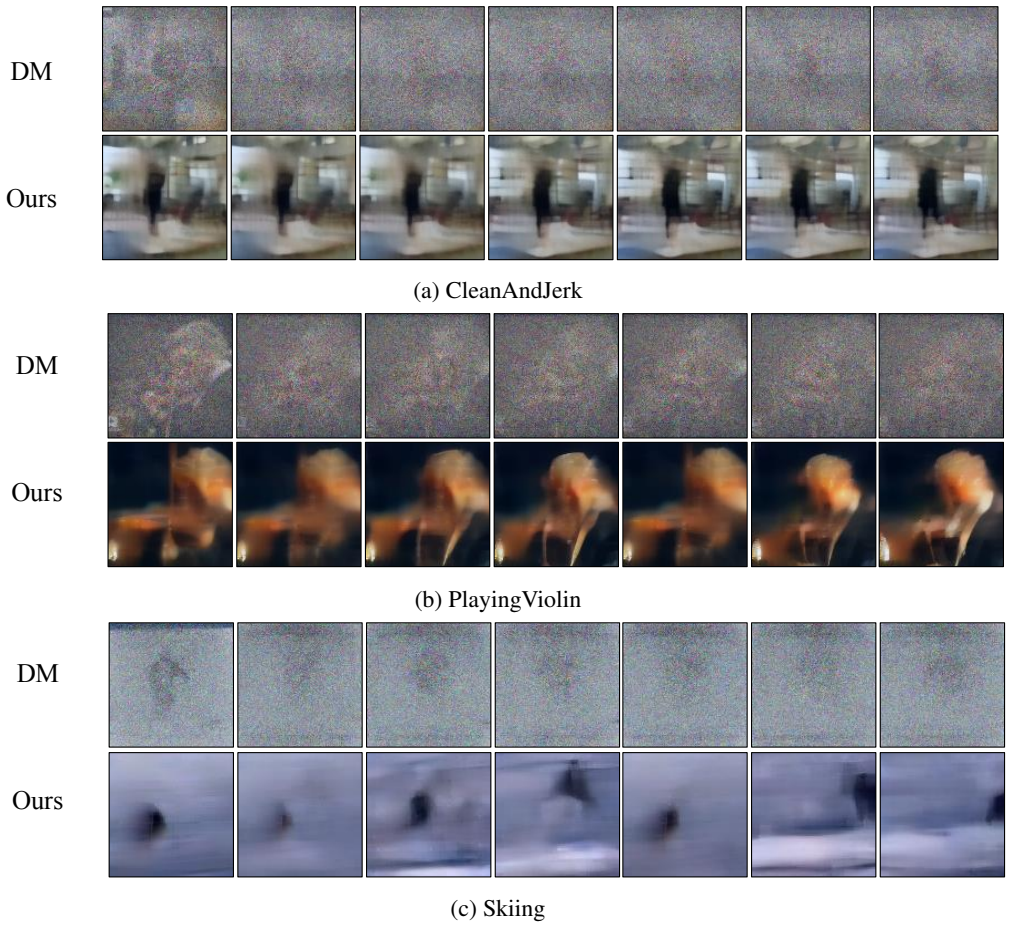
These results confirm that our latent-space approach significantly reduces computational overhead compared to pixel-space distillation methods like VDSD. By leveraging structured compression techniques such as HOSVD and eliminating costly iterative optimization steps, our method achieves faster dataset distillation without compromising performance. This makes our approach highly scalable and practical for real-world applications, especially in large-scale video analysis scenarios.

E LLM USAGE DISCLOSURE

In accordance with the ICLR 2026 policies on large language model (LLM) usage, we disclose that an LLM was used during the preparation of this paper. The assistance was limited to language polishing. All technical content, research design, experiments, analysis, and conclusions were conceived, implemented, and verified by the authors.

F VISUALIZATION

Following previous works, we provide an inter-frame contrast between DM and our method to illustrate the differences in temporal consistency in Fig. 4. Specifically, we sample three representative classes (CleanAndJerk, Playing Violin, and Skiing) from the MiniUCF dataset and visualize the temporal evolution of distilled instances. The results clearly demonstrate that our method retains more temporal information, preserving smooth motion transitions across frames. We also provide the reconstructed and decoded frames of our method for MiniUCF across 20 classes in Fig. 5. These visualizations further validate the effectiveness of our latent-space video distillation framework in preserving critical spatiotemporal dynamics.



(a) CleanAndJerk

(b) PlayingViolin

(c) Skiing

Figure 4: **Inter-frame comparison between DM and our method.** Frames from our method are reconstructed from saved tensors and decoded using a 3D-VAE. We adopt the same resolution setting (112×112) as used in prior works for fair comparison. The visible blur in the reconstructed frames reflects the trade-off between compression ratio and downstream performance, as discussed in our ablation study.

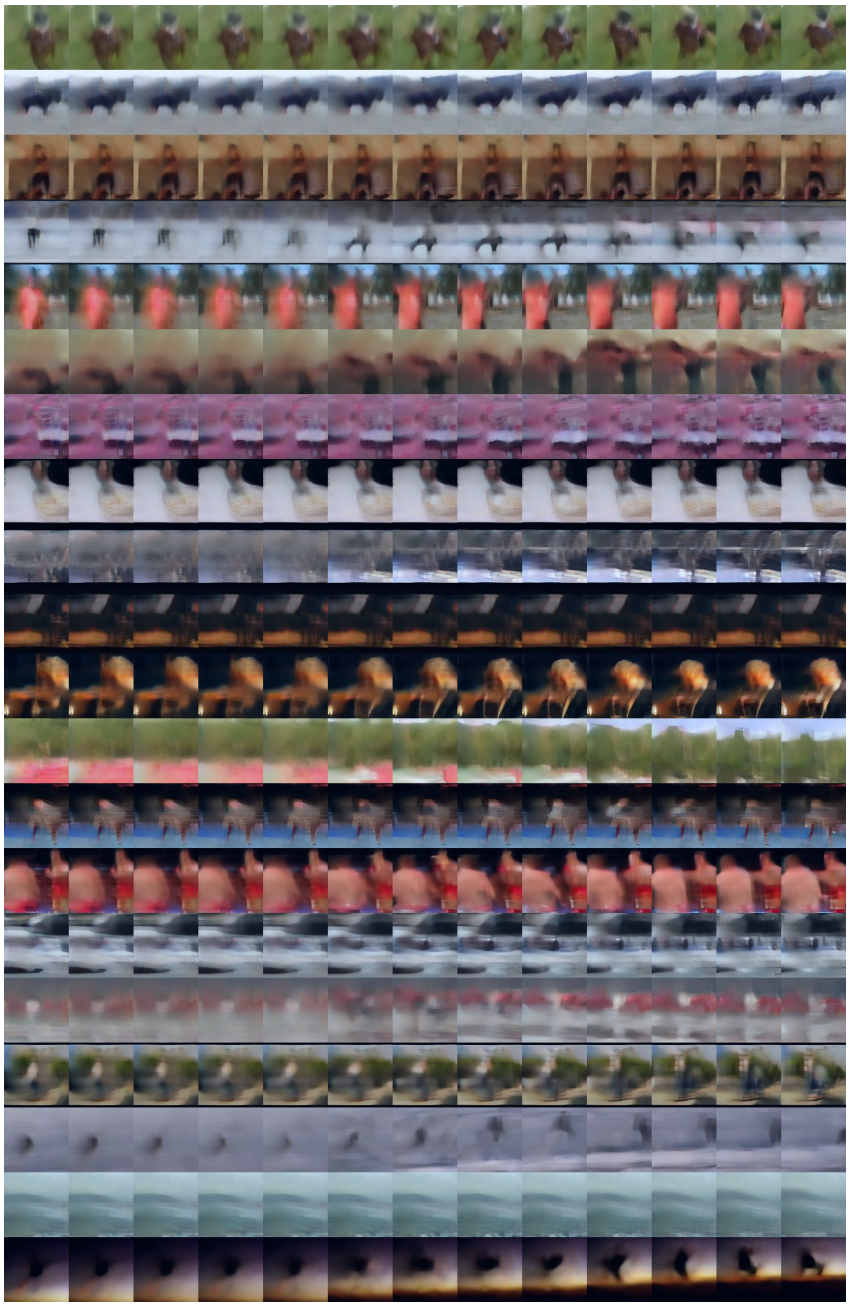


Figure 5: Reconstructed and decoded frames of our method for MiniUCF with a 3D-VAE.