

The Diashow Paradox: Stronger 3D-Aware Representations Emerge from Image Sets, Not Videos

– Supplementary Material –

A. Adapting Video ViTs for Image Tasks

In contrast to image-based VFMs, video-based VFMs have spatio-temporal positional embeddings. To establish an effective inference mode of video-based VFMs on dense image tasks, we evaluate and compare multiple inference modes of pre-trained V-JEPA and VideoMAE on dense geometric tasks: depth and surface normal estimation.

A straightforward strategy to adapt video-based VFMs for static image tasks is to duplicate the input image N times, where N is the number of frames used during video pre-training. We obtain the image features by concatenating the output tokens at corresponding spatial positions across all replicated frames. These temporally aggregated features are then passed through a probing head for downstream evaluation. Fig. 4 illustrates the aggregation process used for probing. One downside of this approach is the computational cost. An inference pass on a single image now scales with the number of frames divided by the tubelet. In practice, this overhead amounts to a multi-fold increase in FLOPs compared to image-based counterparts, such as VideoMAE/MAE or V-JEPA/I-JEPA. To avoid the computational overhead, we adopt the Conv2D head from Eq. (1) and apply three different strategies of handling the temporal dimension of the positional embeddings as shown in Fig. 5. In total, we evaluate four adaptation strategies for video ViTs on dense image tasks:

- **Temporal Duplication and Aggregation:** Replicate the image N times and concatenate tokens across the temporal axis before probing.
- **Temporal Interpolation:** In this strategy, we utilize the adapted Conv2D as in Eq. (1) and interpolate the temporal dimension of the pre-trained positional embeddings to single-image input.
- **Temporal Frame Alignment:** Using the Conv2D head from Eq. (1), we realign the temporal positional embeddings to a single input image at inference/probing, matching the model’s temporal prior to a static input.
- **Representation Replication:** To disentangle gains from representation ability versus parameter count, we replicate the *Temporal Frame Alignment* features to match the *Temporal Duplication and Aggregation* tensor shape before decoding.

We evaluate four adaptation strategies of the temporal positional embeddings with two video-based VFMs. We use NYUv2 depth estimation as the evaluation benchmark. Tab. 4 presents the quantitative results.

While *Temporal Duplication and Aggregation* achieves

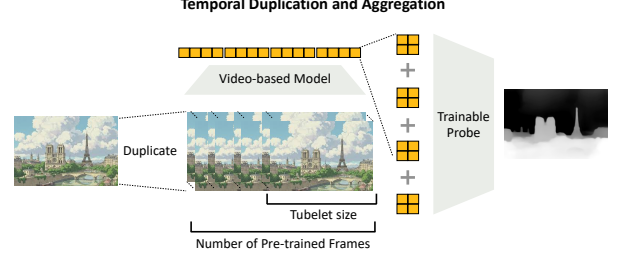


Figure 4. **Input duplication and temporal aggregation.** Input image is replicated N times; tokens at corresponding spatial positions across frames are concatenated along the temporal axis before probing.

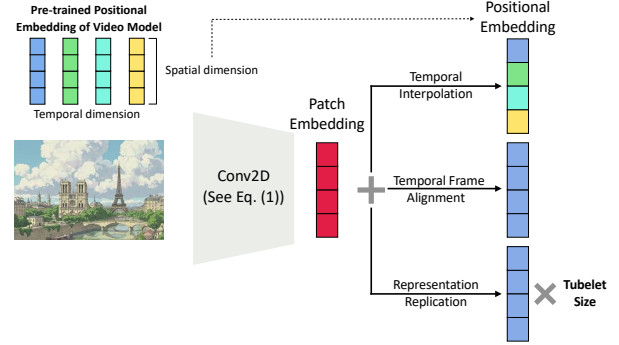


Figure 5. **Temporal adaptation for a single-image input.** Patch embedding from a single image is encoded via adapted Conv2D, the pre-trained positional embeddings are adapted with three strategies: *Temporal Interpolation*, *Temporal Frame Alignment* and *Representation Replication*.

Model	Temporal Pos Emb Strategy	RMSE ↓
VideoMAE-b16	Temporal Interpolation	0.4184
	Temporal Frame Alignment	0.3906
	Temporal Duplication and Aggregation	0.3805
	Representation Replication	0.3969
V-JEPA-h16	Temporal Interpolation	0.2866
	Temporal Frame Alignment	0.2685
	Temporal Duplication and Aggregation	0.2649
	Representation Replication	0.2677

Table 4. **Adaptation of temporal positional embeddings in video ViTs.** We evaluate different strategies for adapting video ViTs for image tasks on NYUv2 depth estimation with VideoMAE and V-JEPA.

the lowest RMSE across both architectures (0.3805 for VideoMAE-b16 and 0.2649 for V-JEPA-h16), the performance gains over *Temporal Frame Alignment* are modest,

2.6% and 1.3% relative improvement, respectively, while incurring a substantial computational overhead.

To investigate whether these gains stem from increased model capacity rather than the representation capability, we evaluate *Representation Replication*, which matches the spatial dimensionality of the duplication setup. The results (0.3969 for VideoMAE-b16 and 0.2677 for V-JEPA-h16) demonstrate that the benefits of *Temporal Duplication and Aggregation* cannot be attributed solely to increased parameter count. Rather, the performance improvements arise from the aggregation of multiple temporal representations, indicating that ensemble effects contribute significantly to the observed gains.

Temporal Interpolation consistently underperforms across both models, suggesting that interpolation of the temporal positional embeddings is not the best strategy to preserve the learned temporal relationships critical for effective feature extraction from a single image.

Conclusion. *Temporal Frame Alignment* emerges as a strong baseline for adapting video-based VFMs to dense image tasks, achieving competitive performance while maintaining computational efficiency. This approach preserves the learned temporal structure of the model without the computational overhead of frame duplication. These properties make the approach practical for real-world deployment, where computational resources are constrained.

B. Adapting Video ViTs for Multi-View Tasks

We adapt video-based models for inference on multi-view tasks. Here, we focus on the multi-view correspondence task as the main benchmark [1, 5, 27], which has previously been studied in the context of *image* encoders. We explore four different strategies for adapting *video* models:

- **Temporal Duplication and Aggregation:** As in Appendix A, each input image is temporally replicated to fill the expected number of frames for the pre-trained backbone. The resulting features are then concatenated as in Fig. 4.
- **Temporal Frame Alignment:** Following Appendix A, we also align temporal positional embeddings to a single-frame input and extract correspondence features.
- **Joint Temporal Attention:** To capture cross-frame attention from video-based models, we present both input images as temporally adjacent frames. We extract features from the correspondence tokens of each frame, utilizing cross-frame attention in the video backbone.
- **Asymmetric Frame Composition:** Temporal positional embeddings in video models can introduce representational asymmetries between otherwise identical frames, leading to instability in direct feature comparisons. To address this, we adopt an asymmetric composition strategy designed to neutralize position-induced drift: (1) image

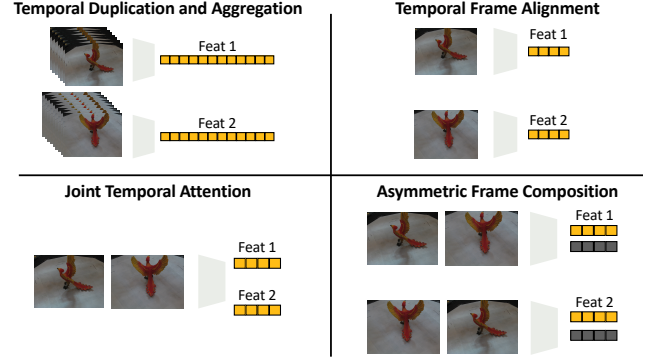


Figure 6. **Adapting video ViTs for multi-view correspondence.** We compare four feature-extraction modes: (i) Temporal Duplication and Aggregation, (ii) Temporal Frame Alignment, (iii) Joint Temporal Attention (two views as adjacent frames), and (iv) Asymmetric Frame Composition to cancel position-induced drift.

Model	Strategies	θ_{30}^0	θ_{60}^{30}	θ_{90}^{60}	θ_{120}^{90}
VideoMAE-b16	Temporal Frame Alignment	88.32	48.32	20.38	11.58
	Asymmetric Frame Composition	68.90	35.04	16.60	11.78
	Joint Temporal Attention	12.84	10.93	7.52	7.35
	Temporal Duplication and Aggregation	89.63	48.24	18.87	10.20
V-JEPA-h16	Temporal Frame Alignment	86.39	52.56	23.03	13.79
	Asymmetric Frame Composition	51.32	27.27	15.53	10.52
	Joint Temporal Attention	11.19	9.23	6.67	6.00

Table 5. **Performance metrics summary for feature extraction strategies.** Comparison of four feature-extraction modes for video pre-trained backbones on NAVI with 1000 keypoints: Temporal Duplication and Aggregation, Temporal Frame Alignment, Joint Temporal Attention, and Asymmetric Frame Composition. Bold indicates the best per column within each model.

A followed by image B, from which we extract features corresponding to image A; and (2) image B followed by image A, from which we derive the features corresponding to image B.

Fig. 6 demonstrates the setup and the structural differences among the approaches.

B.1. Results and Analysis

The results reveal key insights into probing video-based ViTs’ representations for multi-view correspondence.

Temporal Frame Alignment preserves cross-view consistency in both VideoMAE and V-JEPA and nearly dominates across view changes, where it matches *Temporal Duplication and Aggregation*, while employing the same compute as the image-based ViTs.

Cross-frame attention variants perform poorly compared to other variants. *Joint Temporal Attention* collapses below 13% for all viewpoint changes, while *Asymmetric Frame Composition*, which extracts features from A in context (A, B) and from B in context (B, A), exposes positional asymmetry in otherwise identical frames, yielding large gains over *Joint Temporal Attention* across settings (e.g., 68.90%

vs. 12.84% for VideoMAE at θ_{30}^0). Yet, there remains a substantial gap to *Temporal Frame Alignment*, indicating that generic cross-frame attention does not enforce view-consistent geometry representation.

Conclusion. The results in Tab. 5 indicate that for video-based VFMs, such as VideoMAE and V-JEPA, temporal frame alignment offers a more stable and consistent representation for multi-view correspondence and preserves the spatial semantics of the input.

C. Adapting ViTs for Different Resolutions

Downstream tasks may require inference at a different input resolution than the one used for model training. When the test-time input resolution deviates from the pre-training resolution, we can either *recompute* the positional embeddings on the new grid or *interpolate* them from the pre-trained grid [5, 8]. In our study, we consider three strategies:

- **Recompute Positional Embedding:** We recompute the positional sinusoidal embeddings on the new grid.
- **Interpolate Positional Embedding:** We interpolate pre-trained embeddings to the new grid.
- **Resize Input Images:** We resize the image to the pre-training resolution, which requires no changes to the pre-trained backbone. We then upsample the features to the target size for the probe.

Motivation. With frozen encoders, changing the positional code or sequence length induces a distribution shift. Interpolation is common at inference, but with relative positions, this change to the spatial/temporal grid can impair performance [5]. If this is not carefully controlled, comparisons become biased by the adaptation strategy, not the underlying representation quality.

In each approach, the input image is divided into patches consistent with the pre-trained configuration. However, the first two methods (*Recomputing Positional Embeddings*, *Interpolating Pre-trained Positional Embeddings*) alter the input size, thereby changing the number of tokens and the sequence length processed by the transformer. This variation can lead to increased computational demands, particularly with high-resolution images.

In probing scenarios, models are typically frozen, making the introduction of new positional embeddings through recomputation potentially inconsistent. The models pre-trained on specific image sizes may not generalize well to different resolutions if we alter their positional embeddings. Conversely, interpolating existing positional embeddings is a widely adopted practice during inference, enabling models to handle varying image sizes effectively without re-training. However, as discussed by Banani et al. [5], both recomputing and interpolating positional embeddings may

Model	Strategy	δ_1	δ_2	δ_3	Depth-RMSE
MAE-b16	Recompute Pos Emb	0.818	0.961	0.991	0.4779
	Interpolate Pos Emb	0.883	0.980	0.995	0.3875
	Resize Input Images	0.932	0.988	0.997	0.3187
SAM-b16	Interpolate Pos Emb	0.821	0.965	0.991	0.4865
	Resize Input Images	0.881	0.978	0.996	0.3998
DeiT-3-b16	Recompute Pos Emb	0.940	0.991	0.998	0.3063
	Resize Input Images	0.942	0.991	0.998	0.3028
CLIP-b16	Recompute Pos Emb	0.721	0.932	0.982	0.6082
	Resize Input Images	0.763	0.944	0.984	0.5589
MiDaS-I16	Recompute Pos Emb	0.915	0.984	0.996	0.3491
	Resize Input Images	0.914	0.984	0.995	0.3490
DoRA-b16	Recompute Pos Emb	0.793	0.953	0.988	0.5227
	Resize Input Images	0.850	0.969	0.991	0.4373
DINOv2-b14	Interpolate Pos Emb	0.975	0.997	0.999	0.2223
	Resize Input Images	0.975	0.997	0.999	0.2223
DINO-s16	Interpolate Pos Emb	0.895	0.980	0.995	0.3790
	Resize Input Images	0.913	0.983	0.996	0.3528
DINO-b16	Interpolate Pos Emb	0.909	0.983	0.996	0.3571
	Resize Input Images	0.920	0.986	0.996	0.3340

Table 6. **Adapting ViTs to input resolution.** We evaluate depth estimation as positional embeddings are adapted across resolutions via interpolation, resizing, or recomputation.

not be effective with models that use relative positional embeddings and can significantly degrade model performance. In contrast, resizing input images to the pre-trained dimensions allows the model to operate under familiar conditions, providing a more stable foundation for parametric probes and accommodating a broader variety of models.

Tab. 6 (NYUv2, 480×480) shows that *Resizing Input Images* consistently matches or outperforms recomputation/interpolation across MAE, SAM, DeiT-3, CLIP, MiDaS, DoRA, DINO/DINOv2 in terms of depth accuracy (higher δ ; lower RMSE). When pre-training includes higher-resolution exposure (e.g., DINOv2), the gap to interpolation vanishes.

Conclusion. For trainable probes on dense image and video tasks, resizing the inputs to the backbone’s pre-training resolution appears to be a superior configuration compared to the alternatives. This approach preserves the computational cost and attention statistics, avoids positional drift, and consistently yields the strongest, most stable results.