TT-VidT: Decoupling the Temporal Axis for Efficient Motion-Centric Video Pretraining

Shih-Ying Yeh[♠♡†] **Daniel Z. Kaplan**[♢] **Xuehai Wang**[♣△]
Fu-En Yang[★] **Min-Hung Chen**[★] **Shang-Hong Lai**[♠]

[♠]National Tsing Hua University [♡]Comfy Org Research [♢]realiz.ai
[♣]Karolinska Institutet [△]Stockholm University [★]NVIDIA

[†]Corresponding author: kohaku@kblueleaf.net

Project page: https://kohakublueleaf.github.io/TTVidT/

Abstract

Comparisons in video self-supervised learning often evaluate complete training recipes rather than isolating the method itself: architecture, objective, data exposure, schedule, scale, and decoder capacity can all vary at once. This makes it hard to identify which choices yield motion-prioritized representations, whose gains concentrate on frame-to-frame change while retaining useful appearance. We address this with a matched $4 \times 6 = 24$ architecture-objective study at roughly 170M$\sim$190M encoder scale on $\sim$1.7M OpenVid and Moments-in-Time v2 clips for 8 epochs, and propose TT-VidT. TT-VidT combines a DINOv3-initialized ViT-B/16 per-frame spatial path with a compact Temporal Transfer Layer, trained by Diff Compression to reconstruct target frames from a first-frame appearance anchor and frame-specific motion tokens. The sweep shows that TT3D with Diff Compression, not either component alone, enters the strongest motion-sensitive regime, and decoder ablations favor a compact video-pretrained decoder. In final comparison, TT-VidT leads Jester, Something-Something V2, ARID, and Diving48 fine-tuning simultaneously, improving over the strongest non-TT row by 54%$\sim$121%, while using 48% fewer encoder FLOPs than DisMo and 55% fewer than VideoMAE or V-JEPA 2. HMDB51, IARD, and EPIC-Kitchens bound the claim.

1 Introduction

To be meaningfully different from frame understanding, video understanding must use information that no single frame contains, and we call the information recoverable from a single frame *appearance*. In practice, however, many action-recognition benchmarks contain strong appearance cues: objects, scenes, actors, clothing, and camera context can often predict the label [Liu et al., 2021, Li et al., 2018, Kowal et al., 2022, Fioresi et al., 2025]. A video self-supervised learning (SSL) method can therefore obtain competitive recognition accuracy while relying heavily on appearance. This raises a more specific question: under a controlled comparison, which architecture-objective combinations produce representations whose gains concentrate on motion-sensitive tasks?

Existing video SSL methods approach this problem through three families: masked reconstruction (e.g., VideoMAE [Tong et al., 2022]), latent prediction (e.g., V-JEPA, V-JEPA 2 [Bardes et al., 2024, Assran et al., 2025]), and motion-aware training (e.g., DisMo [Ressler-Antal et al., 2025]). These approaches are effective, but prior comparisons often entangle architecture, objective, data exposure, schedule, and parameter scale. As a result, it is difficult to attribute motion-oriented behavior to a specific architecture-objective choice rather than to incidental recipe differences.

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

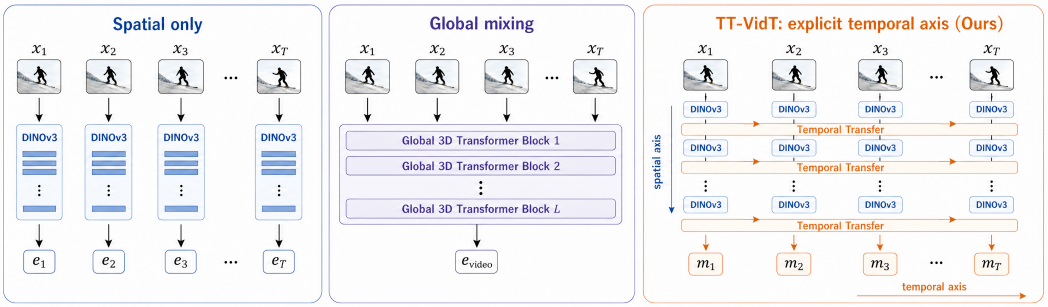

Figure 1: Overview of TT-VidT's decoupled pretraining design. A wide first-frame spatial feature supplies the appearance anchor, while a compact temporal transfer path emits frame-specific motion tokens for Diff Compression.

These observations suggest that temporal representation learning should be evaluated not only by average downstream accuracy, but also by whether a model improves in regimes where static appearance is insufficient. This distinction is difficult to isolate with conventional video SSL encoders, since spatial appearance and temporal evidence are typically mixed throughout the network [Arnab et al., 2021, Bertasius et al., 2021]. A model may therefore perform well on a video benchmark while relying heavily on identity, scene, or object cues that are visible in a single frame [Li et al., 2018, Kowal et al., 2022]. We use the term *motion-prioritized* to describe the empirical pattern in which a video representation improves most clearly on tasks whose labels depend on frame-to-frame change, while remaining competitive on appearance-dominated tasks.

TT-VidT is designed to make this pattern measurable under a controlled recipe. The model keeps a wide per-frame spatial stream for appearance information, while routing temporal information through a compact transfer path (Figure 1). During pretraining, Diff Compression reconstructs target frames from a first-frame spatial anchor and a small set of frame-specific motion tokens. This formulation limits the capacity of the frame-specific path and encourages it to carry information that cannot be recovered from the anchor alone. In the temporal transfer layers, motion tokens are conditioned on compressed spatial context, while the high-capacity spatial anchor is reserved for reconstruction. This design does not remove appearance information from the model; rather, it makes the frame-specific bottleneck narrow enough that improvements on motion-sensitive tasks can be interpreted as evidence for useful temporal transfer under the matched recipe.

Our protocol pretrains all entries at a roughly 170M$\sim$190M encoder scale on a $\sim$1.7M clip mixture from OpenVid [Nan et al., 2025] and Moments-in-Time v2 [Monfort et al., 2020] for 8 epochs under a matched recipe, then evaluates $4 \times 6 = 24$ architecture-objective combinations against three canonical baselines. TT-VidT's profile concentrates on motion-heavy evaluations: 37.47 on ARID, 73.25 on Jester, 25.92 on Something-Something V2, and 18.63 on Diving48 fine-tuning. The profile reaches the representation itself: when the input motion is flipped or time-reversed, TT-VidT abandons its original answer on all but 1% of clips while every baseline keeps it on 9$\sim$43%, and the lead survives end-to-end finetuning, seeds, and a near-doubled budget (§4.5). HMDB51, IARD, and EPIC-Kitchens bound the claim where appearance favors broader baselines.

Contributions. **(1)** We introduce a controlled video SSL protocol that compares 24 architecture-objective combinations under a shared recipe, supplemented by a single-frame appearance diagnostic to contextualize boundary cases. **(2)** We propose TT-VidT, an instance combining TT3D and Diff Compression. TT3D adds a compact Temporal Transfer Layer on top of a DINOv3-initialized ViT-B/16 spatial path, jointly trained from these initial weights under the matched recipe; Diff Compression reconstructs target frames from a first-frame spatial anchor and a small set of motion tokens. **(3)** We show that TT-VidT has an empirically motion-prioritized profile: it is the only method in our comparison group to lead Jester, Something-Something V2, ARID, and Diving48 fine-tuning simultaneously, with the gains confirmed per clip by a motion-inversion probe, while HMDB51, IARD, and EPIC-Kitchens expose the boundary of the claim.

2 Related Work

Augmentation-based motion-appearance disentanglement. DisMo is the closest comparison both in scale and conceptual lineage. Our matched recipe trains all entries at a roughly 170M$\sim$190M encoder scale on $\sim$1.7M video clips for 8 epochs ($\sim$13.6M samples seen), close to DisMo's reported 172M / $\sim$17M setting [Ressler-Antal et al., 2025]. DisMo learns a motion extractor and a motion-conditioned generator, using appearance augmentation to reduce identity leakage while preserving motion conditioning. Native DisMo uses a DINOv2-B frame encoder and a 3D ViT-B sequence embedder that produces motion tokens [Oquab et al., 2023, Ressler-Antal et al., 2025]. In our matched DisMo-style row, we use the same DINOv3-initialized 2D path plus 3D transformer-block setup as TT-VidT where applicable, but with DisMo's motion-control objective rather than Diff Compression [Siméoni et al., 2025]. The resulting comparison isolates a narrow delta: DisMo tests augmentation-based motion control, while TT-VidT combines a compact temporal path with Diff Compression, with capacity pressure motivated by information-bottleneck views [Tishby et al., 2000, Alemi et al., 2017, Achille and Soatto, 2018].

Masked and autoregressive video pretraining at comparable scale. VideoMAE and ARVideo are near peers by ViT-B-scale architecture, but their data exposure is much larger than the matched recipe: VideoMAE accounts for 343M total parameters and about 410M seen clips on Kinetics-710 in our comparison accounting, while ARVideo accounts for 304M total parameters and roughly 406M seen clips under its Something-Something V2 schedule [Tong et al., 2022, OpenMMLab, 2024, Ren et al., 2024]. Unlike DisMo's augmentation-based motion control, these methods train full video encoders with masked reconstruction or autoregressive token prediction, so architecture and objective comparisons remain informative although the exposure gap prevents a one-to-one controlled comparison. Motion-aware MAE variants sharpen this neighborhood: AdaMAE, MAM2, MotionMAE, MME, SMILE, TrackMAE, and No More Shortcuts alter mask selection, split appearance and motion decoders, reconstruct temporal differences or trajectories, inject synthetic motion, or remove local appearance shortcuts [Bandara et al., 2023, Song et al., 2022, Yang et al., 2022, Sun et al., 2023, Thoker et al., 2025b, Vandeghen et al., 2026, Dave et al., 2024]. They validate the need for temporal targets, while DisMo and TT-VidT help contextualize appearance leakage in evaluation.

Large-scale latent-prediction and masked-video systems. V-JEPA, V-JEPA 2, and VideoMAE v2 define the latent-prediction and masked-video scale context, with V-JEPA 2 moving into VM22M-scale data and ViT-L to ViT-g encoders [Bardes et al., 2024, Assran et al., 2025, Wang et al., 2023a]. Toto, VideoMAP, NExT-Vid, and SALT extend the frontier through autoregressive video pretraining, Mamba-Transformer hybrids, next-frame objectives, or static-teacher latent training [Rajasegaran et al., 2025, Liu et al., 2025, Li et al., 2025a,b]. These systems set useful upper-bound context and baseline families, but their native recipes differ in model size, data volume, schedules, and often teacher compute. Even at our matched-recipe scale, close to DisMo's, TT-VidT is framed as a small-recipe design study; full-scale V-JEPA 2, VideoMAE v2, Toto, VideoMAP, NExT-Vid, and SALT operate in different regimes.

Image-pretrained substrates with temporal modules. A second lineage reuses strong image-pretrained features and adds video-specific temporal computation. AdViSe trains a lightweight R3D temporal module on top of an image foundation model with a playback-rate perception objective, while FRAME, SALT, and MVD reuse DINO, CLIP, image, or video teacher features for temporally consistent representation learning or distillation [Wu et al., 2025, TV et al., 2025, Li et al., 2025b, Wang et al., 2023b]. ST-Adapter, AIM, DiST, and ZeroI2V adapt image transformers for supervised image-to-video transfer through temporal adapters, joint adapters, distillation, or efficient transfer modules [Pan et al., 2022, Yang et al., 2023, Qing et al., 2023, Li et al., 2024]. We cite these methods as architectural precedent for a DINOv3-initialized 2D path plus a narrow temporal pathway, with all parameters trained jointly, rather than as SSL pretraining baselines under the matched recipe.

Benchmarks, appearance bias, and diagnostic evaluation. Action-recognition benchmarks differ in motion demand, so the evaluation is written as a diagnostic ladder rather than a single leaderboard. HMDB51 is classic but biased toward scene, background, and subject appearance [Kuehne et al., 2011, Liu et al., 2021, Fioresi et al., 2025]. Jester, Something-Something V2, and Diving48 stress hand motion, temporal order, human-object interaction, or fine-grained dynamics, yet static-dynamic analyses show that appearance cues can remain predictive [Materzynska et al., 2019, Goyal et al.,

2017, Li et al., 2018, Kowal et al., 2022]. ARID reduces appearance reliability through low-light capture, IARD tests identity-controlled invariance, and EPIC-Kitchens-100 mixes motion-heavy verbs with noun and scene signals [Xu et al., 2020, Tacchetti et al., 2017, Isik et al., 2018, Damen et al., 2022]. SEVERE, SEVERE++, and VSSL benchmarking argue that single-score summaries hide domain, scale, granularity, and probe-capacity variation [Thoker et al., 2022, 2025a, Kumar et al., 2024]. Each entry in our sweep is trained under the shared video pretraining recipe; image-path entries (TT3D, TT1D, and DisMo) initialize the spatial encoder from DINOv3 ViT-B/16, while VideoMAE-style 3D ViT entries are trained from scratch. All entries are paired with a single-frame DINOv3 appearance control [Siméoni et al., 2025]; the probe ladder combines kNN, linear and MLP probes over mean-pooled or learned-weight features, attentive aggregation, and DisMo-style identity checks [Caron et al., 2021, Chen et al., 2020, He et al., 2022, Bardes et al., 2024, Ressler-Antal et al., 2025].

3 Method

We present TT-VidT, a video self-supervised method built from two parts. The encoder pairs a DINOv3-initialized 2D spatial path with a compact *Temporal Transfer* pathway that produces per-frame motion embeddings; we instantiate it in two forms, *TT1D* (Section 3.1) and the joint space-time variant *TT3D* (Section 3.2). The pretraining objective is *Diff Compression* (Section 3.3): the decoder reconstructs each later frame from a single first-frame appearance anchor paired with a per-target-frame motion token, so the encoder must place into each motion token whatever frame-specific information the decoder needs beyond the anchor.

We use 1-indexed frames throughout: $X = (x_1, \ldots, x_T)$ with $T = 8$, $z_t = E(x_t) \in \mathbb{R}^{N \times d}$ from a DINOv3 ViT-B/16 encoder E [Siméoni et al., 2025] with $N = 256$, $d = 768$, $m_t \in \mathbb{R}^{K \times d}$ with $K = 8$ motion tokens per frame, decoder D, and per-frame loss ℓ. All parameters are trained jointly under the matched recipe.

3.1 Temporal Transfer

The temporal channel of TT-VidT is built around a small set of motion tokens that summarize per-frame spatial context and exchange information across frames. The spatial path is the standard DINOv3-initialized ViT applied independently per frame, with no cross-frame mixing.

We attach $K = 8$ learnable motion-token embeddings to each frame. The Temporal Transfer Layer $\mathcal{T}$ has $L_{\mathcal{T}} = 12$ layers with hidden dimension $d = 768$. In each layer, the motion tokens of frame t are concatenated with that frame's spatial tokens and processed by the self-attention of the ViT layer, so each motion group summarizes its own frame. The motion tokens then attend to each other under a block-causal mask: the $T \cdot K$ motion tokens form a single sequence in which frame t may attend to frames $1, \ldots, t$. After the final layer, the T groups $\{m_1, \ldots, m_T\}$ are read out for the pretraining objective (Section 3.3).

The cross-frame attention has length $T \cdot K = 64$, much smaller than the $T \cdot N = 2048$ that running attention over full spatial features would require. This keeps the matched comparison setting feasible, where data, schedule, and parameter scale are held constant across the sweep so that the properties of objective and encoder can be isolated. We refer to this configuration as *TT1D*, since the cross-frame mixing runs purely along the time dimension over motion tokens.

3.2 TT3D: Joint Spatial-Temporal Mixing

TT3D is a variant of Temporal Transfer in which the cross-frame step sees more than the motion tokens. Instead of letting only the motion tokens exchange information across frames, we admit a coarse view of the spatial features directly into the cross-frame attention.

In each layer of $\mathcal{T}$, spatial tokens are reduced from $N = 256$ to $N' = 16$ per frame by a $4\times$ per-axis downsample, then concatenated with the $K = 8$ motion tokens of that frame. A single block-causal 3D self-attention runs over the joint per-frame sequence of $K + N' = 24$ tokens, with frame t attending to frames $1, \ldots, t$. Motion tokens and downsampled spatial tokens therefore share one temporal attention, so the motion tokens read spatial context from every earlier frame. The motion outputs are read out as in TT1D and consumed by the pretraining objective (Section 3.3).

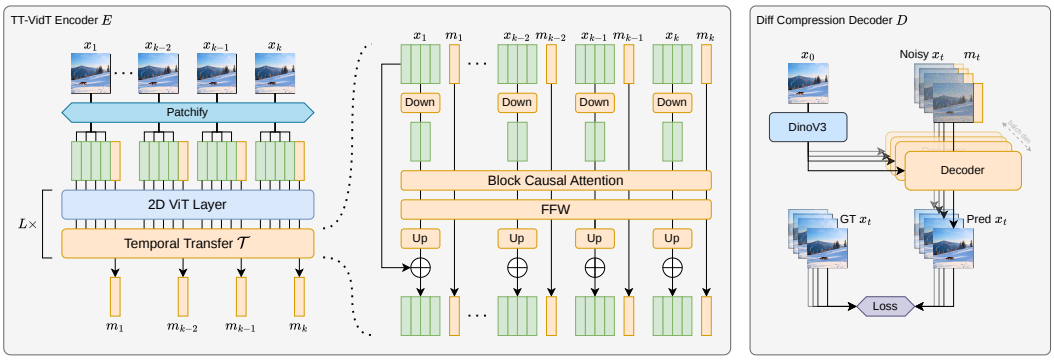

Figure 2: TT-VidT pretraining overview. The encoder interleaves DINOv3 2D ViT processing with Temporal Transfer. The Diff Compression decoder performs diffusion-style denoising conditioned on the first-frame anchor and motion tokens, with loss computed against the target frame.

The joint sequence has length $T \cdot (K + N') = 192$, still well below $T \cdot N = 2048$. The decoder continues to cross-attend to the full z_1; the downsample is local to $\mathcal{T}$ and does not propagate to the appearance pathway used by Diff Compression.

3.3 Diff Compression

Diff Compression takes the first frame's spatial features as appearance anchor and the t-th frame's motion embedding as the carrier of frame-specific information:

$$\hat{x}_t = D(z_1,\, m_t), \qquad t = 2, \ldots, T. \tag{1}$$

The per-frame loss is $\mathcal{L}_t = \ell(\hat{x}_t, x_t)$ with ℓ a diffusion or flow-matching reconstruction loss [Peebles and Xie, 2023, Lipman et al., 2023] by default and latent regression as an ablation. The pretraining loss averages over non-anchor frames, $\mathcal{L}(X) = \frac{1}{T-1} \sum_{t=2}^{T} \mathcal{L}_t$, and the encoder E, Temporal Transfer Layer $\mathcal{T}$, decoder D, and motion-token embeddings are all trained jointly; recipe details are deferred to Section 4.1.

The factorization inspired by VTok [Wang et al., 2026]: a single key frame z_1 is paired with per-target-frame motion tokens m_t for $t = 2, \ldots, T$, rather than a frame-by-frame autoregressive chain. We differ from VTok in how each m_t is produced. VTok computes it explicitly, by a feature subtraction between the target frame and the key frame followed by a projection. In our setup m_t is the direct output of the Temporal Transfer pathway, learned end-to-end under the reconstruction objective; the architecture contains no built-in feature diff, so the encoder is free to place into m_t whatever information lets $D(z_1, m_t)$ reproduce x_t. The decoder cross-attends to the full $z_1 \in \mathbb{R}^{N \times d}$, not the N'-downsampled features used inside $\mathcal{T}$ (Section 3.2); the appearance pathway stays wide while the motion pathway remains compact. We hold the decoder at a compact S configuration; ablations in Section 4.3 show that scaling decoder capacity degrades motion-focused performance.

On the same encoder, MAE-Diffusion underperforms Diff Compression in our sweep, identifying the first-frame appearance anchor as the operative difference (Table 1). The pairing also exhibits a coupling property (Section 4.2): on motion-heavy benchmarks, neither Diff Compression with a strong full-3D encoder nor TT3D with a mask-and-reconstruct objective unlocks the regime that the pair does; on appearance-discriminable benchmarks the non-leading cases bound the motion-priority interpretation (Appendix A).

4 Experiments

4.1 Setup

All video SSL entries are evaluated under a shared small-scale pretraining and probing recipe. Entries that use an image-pretrained 2D path (TT3D, TT1D, and DisMo) initialize the spatial encoder from DINOv3 ViT-B/16 [Siméoni et al., 2025]; all parameters are trained jointly under the matched recipe,

and partial-unfreezing variants are out of scope. This protocol is designed to compare architecture-objective choices while controlling data, schedule, parameter scale, and downstream evaluation. The claim we test is empirical: under the same shared video pretraining recipe, which combination yields a motion-prioritized representation? We use motion- and appearance-oriented dataset descriptions as empirical shorthand; Appendix A gives the quick diagnostic behind this interpretation.

Pretraining uses OpenVid [Nan et al., 2025] at approximately 1M clips and Moments-in-Time v2 [Monfort et al., 2020] at approximately 700k clips. We sample 8 frames at 6 fps. The default recipe is 8 epochs, effective global batch size 32, AdamW with peak learning rate 5e-4, betas $(0.9, 0.98)$, weight decay 0.01, gradient clipping at 0.1, 10k linear warmup, cosine decay to 1% of the peak learning rate, and fp16 mixed precision. We use μP [Yang et al., 2021] with base dimension 256. The sweep covers four encoders: the VideoMAE-style joint space-time 3D ViT [Tong et al., 2022] (henceforth ViT3D), DisMo-style 2D+3D, TT1D, and TT3D, at a roughly 170M$\sim$190M encoder scale. It also covers six objectives: MAE, Adaptive AR, naive AR, two-jump AR, MAE-Diff, and Diff Compression. MAE and Adaptive AR are established prior-work objectives; Diff Compression is the proposed objective, and naive AR, two-jump AR, and MAE-Diff are ablation rows that vary autoregressive horizon, multi-step prediction, and the addition of a diffusion head, respectively. In total, each entry sees approximately 13.6M video samples over approximately 425k optimizer steps.

The decoder has three initialization regimes. No pretraining uses a random decoder with regression or diffusion auxiliary loss. ImageNet pretraining uses ImageNet-1k [Deng et al., 2009] and the DINOv3 ViT-B class token as diffusion guidance to reconstruct the full image; cross-attention is not trained in this stage [Siméoni et al., 2025]. Video pretraining samples two frames from a training video, gives the decoder the later frame's class token and the earlier frame's spatial features as cross-attention guidance, and reconstructs the later frame. All pretrained decoders use effective global batch size 256 for 3 epochs.

The V-JEPA 2 row is trained at the same scale/recipe using its native 2+8 schedule. Since it has no per-frame cls-token, we mean-pool per-frame tokens as the motion embedding. The sweep uses no augmentation as a uniform constraint. This is a controlled comparison, although not a neutral one in every respect: DisMo's native training scheme uses motion-preserved augmentations, so no augmentation can handicap non-TT3D objectives. We therefore report DisMo's dual-augmentation ablation in §4.2.

Benchmarks and scope. Jester and Something-Something V2 define their classes by the movement itself [Materzynska et al., 2019, Goyal et al., 2017], Diving48 separates dives only by body dynamics [Li et al., 2018], and ARID's low light makes appearance unreliable [Xu et al., 2020]. HMDB51, IARD, and EPIC-Kitchens sit on the appearance side. §4.5 then measures motion sensitivity per clip rather than per dataset. The 8-epoch budget is itself a design choice matched to DisMo's published scale, so learning speed under the shared recipe is a measured property of each pair rather than a confounder, and all conclusions are stated for this matched budget.

4.2 Synergy: Architecture-Objective Interaction

We sweep all four architectures and six objectives under the shared video pretraining recipe, evaluated with attentive probing on Jester, Something-Something V2 (SSv2), and ARID. Table 1 reports the no-augmentation sweep with a common `decS_imgnet` decoder and a single run per cell. Three cells in this grid coincide with canonical configurations from prior or proposed work: ViT3D+MAE recovers standard VideoMAE [Tong et al., 2022], DisMo+AdaAR recovers standard DisMo [Ressler-Antal et al., 2025], and TT1D/TT3D+DiffComp are the configurations we propose. We shade these cells in Table 1 so they can be located at a glance.

The successful region is sparse. Of the 24 architecture-objective cells, only ViT3D+MAE and TT3D+Diff Compression reach a high tier on the motion-focused Jester and SSv2 benchmarks in the no-augmentation condition. ViT3D+MAE gives 39.47 on Jester and 14.95 on SSv2. TT3D+Diff Compression gives 53.89 and 18.28, the highest cells for both datasets. With TT3D fixed, replacing Diff Compression by MAE drops Jester from 53.89 to 11.07 and SSv2 from 18.28 to 4.98. With Diff Compression fixed, replacing TT3D by ViT3D or DisMo gives 19.82 or 12.98 on Jester and 6.57 or 5.53 on SSv2. This paired drop indicates an interaction effect between TT3D and Diff Compression, not a generic effect of either component alone.

Table 1: Pretraining sweep: top-1 classification accuracy (%) via frozen attentive probing on Jester, Something-Something V2, and ARID. Each cell is one architecture-objective configuration under the matched recipe (§4.1) with a common `decS_imgnet` decoder, single run. **Bold** = best in block; underline = second-best. Blue = VideoMAE, green = DisMo (second number with dual augmentation), orange = ours.

		Objective					
Dataset	Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
Jester	ViT3D	39.47	12.85	13.01	12.85	31.68	19.82
	DisMo	29.71	16.46 / 46.95	15.94	13.54	21.66	12.98
	TT1D	26.65	11.70	10.45	11.86	15.68	24.32
	TT3D	11.07	12.85	10.42	11.83	10.32	**53.89**
SSv2	ViT3D	14.95	3.99	4.86	3.93	11.73	6.57
	DisMo	11.20	5.29 / 13.33	4.35	3.54	7.89	5.53
	TT1D	10.05	4.71	2.53	4.86	6.00	7.91
	TT3D	4.98	2.26	4.16	4.39	4.82	**18.28**
ARID	ViT3D	**24.37**	13.58	15.66	14.12	18.75	17.70
	DisMo	18.15	15.46 / 22.45	14.87	14.02	15.71	16.51
	TT1D	17.65	13.03	12.73	13.38	14.52	16.41
	TT3D	11.49	17.65	13.92	15.76	7.71	21.98

ARID is more mixed in the base sweep. ViT3D+MAE leads the ARID block at 24.37, while TT3D+Diff Compression reaches 21.98. After the decoder ablation selects the decoder S, video-pretrained setting, TT-VidT reaches 37.47 on ARID in the final comparison. The final method leads the selected final-comparison row by +13.1 on ARID, compared with VideoMAE at 24.37; +26.3 on Jester, compared with DisMo with dual augmentation at 46.95; and +11.0 on SSv2, compared with VideoMAE at 14.95. These gaps summarize the selected final-comparison rows reported in Table 5. They describe the operating profile of TT-VidT under the matched recipe, not a universal ranking of architectures or objectives.

The non-leading cases bound the claim. On IARD, TT3D's lead does not hold: a ViT3D+DiffComp configuration reaches 77.03, compared with 67.03 for TT3D. IARD rewards retention of per-frame appearance features such as identity, clothing, and viewpoint context. On HMDB51, VideoMAE leads at 27.73 in the final table. Appendix A provides a quick appearance-vs-motion diagnostic that helps interpret these boundary cases.

DisMo needs its native encoder/decoder dual augmentation [Ressler-Antal et al., 2025]. Enabling this scheme in our controlled recipe improves DisMo substantially: Jester $16.46 \rightarrow 46.95$, SSv2 $5.29 \rightarrow 13.33$, and ARID $15.46 \rightarrow 22.45$ with attentive probing. We call this entry DisMo with dual augmentation in the final comparison. Even with dual augmentation, it trails TT-VidT by 26.3 on Jester and 12.6 on SSv2.

4.3 Decoder Ablation

We ablate the DiT decoder [Peebles and Xie, 2023] with the encoder fixed to TT3D+Diff Compression. Table 2 varies decoder size at three settings (S, B, L) and initialization source across random, ImageNet-pretrained, and video-pretrained forms.

Pretraining matters even after 3 epochs. At size S, where all initialization sources are measured, video pretraining dominates ImageNet pretraining, and ImageNet pretraining dominates random initialization on motion-focused columns. On Jester, decoder S, video-pretrained reaches 73.25, compared with decoder S, ImageNet-pretrained at 53.89, decoder S, random init with diffusion auxiliary loss at 28.46, and decoder S, random init with regression auxiliary loss at 20.21. On SSv2, the same ordering is 25.92, 18.28, 11.71, and 7.53. Thus the decoder is not merely an output head; its pretraining changes the pressure placed on the encoder.

Larger is not always better. With video pretraining, decoder S reaches 73.25 on Jester and 25.92 on SSv2, while decoder L drops to 34.37 and 12.15. L therefore underperforms S by approximately 50%$\sim$53% on these motion-focused benchmarks. One plausible interpretation is that a larger decoder

Table 2: Decoder ablation on TT3D + Diff Compression (no augmentation). Rows: decoder size $\times$ init source (`rand` = random init with regression/diffusion auxiliary loss; `imgnet`/`video` = DiT decoder pretrained on ImageNet-1k or our video task). Cells: top-1 classification accuracy (%) via frozen attentive probing (D48 = Diving48; EK-V = EPIC-Kitchens-100 verb, trimmed). Best per column **bold**, second underlined.

Size	Init	HMDB	ARID	IARD	Jester	SSv2	D48	EK-V
S	rand+reg	4.57	13.28	71.76	20.21	7.53	5.53	24.89
	rand+diff	7.61	13.18	79.01	28.46	11.71	6.70	23.88
	imgnet	12.64	21.98	67.03	53.89	18.28	5.63	28.23
	video	**25.15**	37.47	74.84	**73.25**	**25.92**	8.17	32.54
B	imgnet	9.20	19.99	**83.19**	61.00	17.00	6.09	26.56
	video	23.16	**37.84**	77.36	69.99	23.99	**9.64**	**32.82**
L	imgnet	6.15	14.52	76.37	26.95	8.84	5.69	24.99
	video	11.98	17.90	74.73	34.37	12.15	7.16	27.52

reduces pressure on the compact motion tokens, allowing more of the reconstruction burden to shift into decoder capacity; the motion embedding then carries less information overall, including less motion. We treat this as an empirical pattern observed in Table 2, consistent with non-monotonic decoder behavior reported for masked autoencoding in VideoMAE v2 [Wang et al., 2023a].

The S/B comparison is scope-dependent. On video-pretrained decoders, decoder S and decoder B are comparable on the main motion-heavy columns: Jester is 73.25 versus 69.99 and SSv2 is 25.92 versus 23.99. On ImageNet-pretrained decoders, the gap can reach 27% on HMDB. We adopt decoder S, video-pretrained as the default decoder for TT-VidT, which reaches 25.15 on HMDB, 37.47 on ARID, 74.84 on IARD, 73.25 on Jester, and 25.92 on SSv2. The same pattern is consistent in extended evaluation: video-pretrained decoders dominate ImageNet-pretrained decoders on EK-V verb classification, EK-V anticipation, and Diving48 fine-tuning, with per-size breakdowns reported in the supplementary material.

4.4 Efficiency: Encoder FLOPs

TT3D's Temporal Transfer Layer operates on a compressed per-frame token budget. A $4\times$ per-axis spatial downsample reduces spatial tokens from $N = 256$ to $N' = 16$ per frame; with $K = 8$ motion tokens, each frame contributes 24 tokens and the temporal-attention sequence has 192 tokens instead of 2048 for full spatial-temporal attention. Table 3 reports the corresponding encoder-only analytical FLOPs: TT3D costs 456.1 GF, about half of DisMo-2D-3D and less than half of VideoMAE-3D under the same T=8, 256^2 setting. V-JEPA 2 is listed for context with its native tube setup, T=16 raw frames to 8 latent frames; all other entries use T=8.

Table 3: FLOPs at 256^2 For each model

Model	FLOPs (GF)
TT-VidT TT1D	400.5
TT-VidT TT3D	456.1
DisMo	874.7
V-JEPA 2	1009.9
VideoMAE	1012.6

4.5 Motion Probe and Controls

Table 4: Left: motion-inversion probe on SSv2, flip / stay (%) per input transform, plus relative accuracy drop with shuffled frames. Right: attribution controls, frozen attentive probing.

Model	H-flip	V-flip	Reverse	Shuf. drop
TT-VidT (Ours)	**81.2 / 0.6**	**59.5 / 0.9**	**71.4 / 1.0**	50.9%
VideoMAE-3D	66.9 / 9.7	55.1 / 9.1	38.6 / 20.9	50.0%
DisMo	34.5 / 35.0	8.4 / 43.2	34.5 / 16.4	31.9%
DINOv3-1f	37.1 / 24.5	11.8 / 22.7	0.0 / 100.0	3.1%

Model	Jester	SSv2	ARID
ViT3D+MAE	39.47	14.95	24.37
+ DINOv3 init	39.53	14.00	24.86
TT3D expl. diff	64.7	24.8	26.5
TT-VidT (Ours)	**73.25**	**25.92**	**37.47**

Table 4 asks whether the motion-prioritized profile of §4.2 lives in the representation itself, and which ingredient of TT-VidT produces it. The left half inverts the motion of SSv2 clips whose classes are exact mirrors under horizontal flip, vertical flip, or time reversal, re-encodes them, and asks an attentive probe trained on clean clips whether its decision follows (Appendix I). A prediction

that moves to the mirror class shows that the encoder re-read the motion, and one that stays shows that appearance decided it, since the content is otherwise unchanged. TT-VidT follows the inverted motion on nearly every clip, whereas every baseline keeps a share of its original answers, most of all under time reversal, the one transform that leaves every pixel untouched. The single-frame DINOv3 control cannot see frame order and never follows a reversal, which validates the probe and shows that Temporal Transfer and Diff Compression turn an order-blind substrate into the most direction-faithful encoder of the comparison. The shuffle column separates using frame order from reading it: VideoMAE depends on order as much as TT-VidT, yet reads its direction far less faithfully. The benchmark profile therefore reflects what the representation encodes rather than which datasets happen to reward it.

The right half isolates the two ingredients TT-VidT shares with other work. A ViT3D initialized from DINOv3 in the 12 of its 24 layers that the teacher can fill lands on its scratch counterpart on every motion-heavy benchmark and moves only the appearance-heavy IARD, so the image substrate shapes appearance rather than the motion regime. A VTok-style motion token computed as an explicit feature difference [Wang et al., 2026] already surpasses every non-TT configuration on Jester, which credits the keyframe factorization, and the learned token of TT-VidT improves on it on all five datasets, most where subtraction discards appearance that the task still needs. Neither the initialization nor the factorization alone explains the gain, which again locates it in the pairing of Temporal Transfer and Diff Compression. The gain also exceeds seed variance, since the weakest of nine TT-VidT measurements stays above the strongest ViT3D+MAE measurement on every motion-heavy benchmark, and it persists at a near-doubled budget, where TT-VidT has already saturated while the gap to the baseline stays wide (Appendix I). Under the matched recipe, the motion advantage is thus a property of the method rather than of training speed.

4.6 Final Comparison

We compare TT-VidT against the entries identified by the sweep and ablations: VideoMAE, DisMo with dual augmentation, and an internally retrained V-JEPA 2 under the shared video pretraining recipe. Table 5 reports frozen attentive probing on HMDB, ARID, IARD, Jester, SSv2, and EK-V verb classification, plus Diving48 full fine-tuning and EK-V anticipation.

Table 5: Final comparison: V-JEPA2 trained at the same scale/recipe plus selected entries from the sweep and ablations. Frozen attentive probe on the five sweep datasets and EK-V (verb classification, trimmed clips); D48 FT is full finetuning (50 ep); EK-V Antic. is verb anticipation (frozen attentive on untrimmed observation windows). Best per column **bold**, second underlined.

Method	HMDB	ARID	IARD	Jester	SSv2	EK-V	D48 FT	EK-V Antic.
V-JEPA2 (24L×768d, 16f@12fps, tube=2)	22.70	16.01	79.34	27.41	10.59	32.61	8.02	**23.07**
VideoMAE (768d, 24L)	**27.73**	24.37	80.44	39.47	14.95	**35.62**	8.43	22.81
DisMo (768d, 12s+12t)	22.57	22.45	**89.89**	46.95	13.33	31.97	8.12	22.72
TT-VidT TT3D(Ours)	25.15	**37.47**	74.84	**73.25**	**25.92**	32.54	**18.63**	21.33

The four final-comparison rows expose different operating regimes.

TT-VidT is the only method in our comparison group to lead Jester, Something-Something V2, ARID, and Diving48 fine-tuning simultaneously, with the largest gains concentrated on these motion-heavy evaluations: 37.47 on ARID, 73.25 on Jester, 25.92 on Something-Something V2, and 18.63 on Diving48 fine-tuning. These four datasets are where motion features add the most value relative to a single-frame appearance baseline.

Table 6: Frozen probe vs. 30-epoch end-to-end finetuning.

	Frozen	FT
Jester		
ViT3D+MAE	39.47	51.50
DisMo dual-aug	46.95	51.76
TT-VidT	**73.25**	**72.79**
SSv2		
ViT3D+MAE	14.95	13.40
DisMo dual-aug	13.33	14.85
TT-VidT	**25.92**	**27.72**

TT-VidT's lead survives end-to-end finetuning. Table 6 finetunes the full encoder on Jester and SSv2 for 30 epochs under one shared recipe with identical learning rates for all rows. TT-VidT stays 21.0 and 12.9 points ahead of the strongest baseline. The baselines need 12 to 35 points of supervised correction yet converge near 51.5 on Jester, below TT-VidT's frozen 73.25, while TT-VidT barely moves from its frozen probe, so supervised adaptation saturates the baselines rather than closing the gap.

VideoMAE is the broadest baseline under this recipe. It leads HMDB51 (27.73) and EPIC-Kitchens verb classification (35.62), and remains competitive on several motion-heavy datasets. This breadth, combined with its strong appearance retention, makes it the most reasonable single fallback when the downstream task mixes motion and static cues.

DisMo improves substantially when its native dual-augmentation training is restored. Without dual augmentation, DisMo+AdaAR is not competitive in the no-augmentation sweep; with dual augmentation, the same architecture leads IARD (89.89) and reaches competitive scores on Jester and SSv2. This indicates that DisMo's augmentation design is integral to its architecture rather than incidental.

V-JEPA 2 trails most evaluations under the matched-recipe small-scale setting, indicating that latent prediction at this small scale does not match its native large-scale operating point on motion-heavy or appearance-discriminable benchmarks. Its native large-scale recipe is outside our protocol. It does, however, lead EPIC-Kitchens verb anticipation (23.07), indicating that latent prediction retains an advantage on long-horizon prediction even under the matched recipe.

The mixed tasks expose the boundary. On EK-V verb classification, VideoMAE leads at 35.62 because verb recognition still benefits from object and scene context; TT-VidT reaches 32.54, DisMo 31.97, and V-JEPA 2 32.34. On EK-V anticipation, V-JEPA 2 reaches 23.07, compared with VideoMAE at 22.81, DisMo at 22.72, and TT-VidT at 21.33. Anticipation under untrimmed observation windows is a setting where V-JEPA 2's temporal prediction can show, even though the same row is weak on Jester and SSv2. HMDB51 and IARD follow the appearance-sensitive diagnostic in Appendix A: VideoMAE leads HMDB at 27.73, and DisMo leads IARD at 89.89. On Jester and SSv2, TT-VidT's lead holds across all six probe configurations: kNN, linear and MLP over mean-pooled or learned-weight features, and attentive probing. The supplementary material provides per-probe breakdowns for EK-V verb classification, EK-V anticipation, and Diving48 fine-tuning.

5 Conclusion

We presented TT-VidT, a video self-supervised learning design that decouples appearance and temporal change during pretraining. TT-VidT keeps static appearance in a wide first-frame spatial anchor and routes frame-specific change through a compact Temporal Transfer pathway. Diff Compression makes this bottleneck explicit: target frames are reconstructed from the anchor and motion tokens, so the useful information in the motion tokens is precisely what cannot be recovered from the anchor alone.

We tested this design with a controlled 4×6 architecture-objective sweep under a shared data, schedule, and encoder-scale recipe. The main empirical finding is that the effect comes from the pairing, not from either component independently: TT3D with MAE does not enter the strong-motion regime, and Diff Compression with ViT3D or DisMo encoders remains far below TT-VidT on Jester and Something-Something V2. After the decoder ablation selects the video-pretrained decS setting, TT-VidT is the only final-comparison method to lead Jester, Something-Something V2, ARID, and Diving48 fine-tuning simultaneously.

This explains why TT-VidT is strong on motion-sensitive evaluations: the narrow temporal path makes frame-to-frame change cheap and salient while the wide anchor still supplies appearance for reconstruction. The boundary cases are consistent with the same profile: HMDB51, IARD, and EPIC-Kitchens can reward identity, object and scene context, or anticipation, where broader appearance-sensitive or predictive baselines remain preferable. The same decoupling is also efficient: TT3D uses a 192-token temporal-attention sequence and 456.1 GF encoder FLOPs, roughly half of the parameter-comparable full video encoders in our accounting.

Limitations. This study is intentionally scoped to a small matched recipe: one pretraining mixture (approximately 1.7M OpenVid and Moments-in-Time v2 clips), single-run sweep cells with multi-seed checks on the canonical ones, and one roughly 170M$\sim$190M encoder scale. The resulting profile should therefore be read as evidence about this controlled regime rather than as a claim about all data scales or schedules. We also did not test combinations such as TT3D with DisMo-style dual augmentation, alternative appearance probes, or partial unfreezing of the spatial encoder.

Acknowledgments and Disclosure of Funding

This work is supported by the NVIDIA Taiwan AI Research & Development Center (TRDC). The authors declare no other competing interests.

References

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. In *International Conference on Learning Representations*, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning, 2025.

Wele Gedara Chaminda Bandara, Naman Patel, Ali Gholami, Mehdi Nikkhah, Motilal Agrawal, and Vishal M. Patel. Adamae: Adaptive masking for efficient spatiotemporal learning with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. In *International Conference on Learning Representations*, 2024.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 2022.

Ishan Rajendrakumar Dave, Simon Jenni, and Mubarak Shah. No more shortcuts: Realizing the potential of temporal self-supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

Joseph Fioresi, Ishan Rajendrakumar Dave, and Mubarak Shah. Albar: Adversarial learning approach to mitigate biases in action recognition. In *International Conference on Learning Representations*, 2025.

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The something something video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Leyla Isik, Andrea Tacchetti, and Tomaso Poggio. Fast, invariant representation for human action in the visual system. *Journal of Neurophysiology*, 2018.

Matthew Kowal, Mennatullah Siam, Md Amirul Islam, Neil D. B. Bruce, Richard P. Wildes, and Konstantinos G. Derpanis. A deeper dive into what deep spatiotemporal networks encode: Quantifying static vs. dynamic information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

Akash Kumar, Ashlesha Kumar, Vibhav Vineet, and Yogesh S. Rawat. Benchmarking self-supervised video representation learning, 2024. OpenReview, NeurIPS 2024 Datasets and Benchmarks Track submission.

Jinghan Li, Yang Jin, Hao Jiang, Yadong Mu, Yang Song, and Kun Xu. Learning from next-frame prediction: Autoregressive video modeling encodes effective representations, 2025a.

Xianhang Li, Chen Huang, Chun-Liang Li, Eran Malach, Josh Susskind, Vimal Thilak, and Etai Littwin. Rethinking jepa: Compute-efficient video ssl with frozen teachers, 2025b.

Xinhao Li, Yuhan Zhu, and Limin Wang. Zeroi2v: Zero-cost adaptation of pre-trained transformers from image to video, 2024.

Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision*, pages 513–528, 2018.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/forum?id=PqvMRDCJT9t`.

Xin Liu, Silvia L. Pintea, Fatemeh Karimi Nejadasl, Olaf Booij, and Jan C. van Gemert. No frame left behind: Full video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Yunze Liu, Peiran Wu, Cheng Liang, Junxiao Shen, Limin Wang, and Li Yi. Videomap: Toward scalable mamba-based video autoregressive pretraining, 2025.

Joanna Materzynska, Guillaume Berger, Ingo Bax, and Roland Memisevic. The jester dataset: A large-scale video dataset of human gestures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

Mathew Monfort, Carl Vondrick, Aude Oliva, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa M. Brown, Quanfu Fan, and Dan Gutfreund. Moments in time dataset: One million videos for event understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42 (2):502–508, 2020. doi: 10.1109/TPAMI.2019.2901464. URL `https://doi.org/10.1109/TPAMI.2019.2901464`.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=j7kdXSrISM`.

OpenMMLab. Preparing Kinetics-710. MMAction2 dataset documentation, 2024. https://mmaction2.readthedocs.io/en/latest/dataset_zoo/kinetics710.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. In *Advances in Neural Information Processing Systems*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Zhiwu Qing, Shiwei Zhang, Ziyuan Huang, Yingya Zhang, Changxin Gao, Deli Zhao, and Nong Sang. Disentangling spatial and temporal learning for efficient image-to-video transfer learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pre-training from videos, 2025.

Sucheng Ren, Hongru Zhu, Chen Wei, Yijiang Li, Alan Yuille, and Cihang Xie. Arvideo: Autoregressive pretraining for self-supervised video representation learning, 2024.

Thomas Ressler-Antal, Frank Fundel, Malek Ben Alaya, Stefan Andreas Baumann, Felix Krause, Ming Gui, and Björn Ommer. Dismo: Disentangled motion representations for open-world motion transfer. In *Advances in Neural Information Processing Systems*, 2025.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025.

Yuxin Song, Min Yang, Wenhao Wu, Dongliang He, Fu Li, and Jingdong Wang. It takes two: Masked appearance-motion modeling for self-supervised video transformer pre-training, 2022.

Xinyu Sun, Peihao Chen, Liangwei Chen, Changhao Li, Thomas H. Li, Mingkui Tan, and Chuang Gan. Masked motion encoding for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Andrea Tacchetti, Leyla Isik, and Tomaso Poggio. Invariant action recognition dataset, 2017. Center for Brains, Minds and Machines dataset.

Fida Mohammad Thoker, Hazel Doughty, Piyush Bagad, and Cees G. M. Snoek. How severe is benchmark-sensitivity in video self-supervised learning? In *European Conference on Computer Vision*, 2022.

Fida Mohammad Thoker, Letian Jiang, Chen Zhao, Piyush Bagad, Hazel Doughty, Bernard Ghanem, and Cees G. M. Snoek. Severe++: Evaluating benchmark sensitivity in generalization of video representation learning, 2025a.

Fida Mohammad Thoker, Letian Jiang, Chen Zhao, and Bernard Ghanem. Smile: Infusing spatial and motion semantics in masked video learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025b.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method, 2000.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

Sethuraman TV, Savya Khosla, Vignesh Srinivasakumar, Jiahui Huang, Seoung Wug Oh, Simon Jenni, Derek Hoiem, and Joon-Young Lee. Frame: Pre-training video feature representations via anticipation and memory, 2025.

Renaud Vandeghen, Fida Mohammad Thoker, Marc Van Droogenbroeck, and Bernard Ghanem. Trackmae: Video representation learning via track mask and predict, 2026.

Feng Wang, Yichun Shi, Ceyuan Yang, Qiushan Guo, Jingxiang Sun, Alan Yuille, and Peng Wang. Vtok: A unified video tokenizer with decoupled spatial-temporal latents, 2026. URL https://arxiv.org/abs/2602.04202.

Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023a.

Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Lu Yuan, and Yu-Gang Jiang. Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023b.

Jingwei Wu, Zhewei Huang, and Chang Liu. Advancing video self-supervised learning via image foundation models. *Pattern Recognition Letters*, 192:22–28, 2025. doi: 10.1016/j.patrec.2025.03.015.

Yuecong Xu, Jianfei Yang, Haozhi Cao, Kezhi Mao, Jianxiong Yin, and Simon See. Arid: A new dataset for recognizing action in the dark, 2020.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=Bx6qKuBM2AD.

Haosen Yang, Deng Huang, Bin Wen, Jiannan Wu, Hongxun Yao, Yi Jiang, Xiatian Zhu, and Zehuan Yuan. Self-supervised video representation learning with motion-aware masked autoencoders, 2022.

Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. In *International Conference on Learning Representations*, 2023.

Appendix

Table of Contents

A Appearance-vs-Motion Diagnostic 16

B Architecture Sizing 17

C Encoder Pretraining 17

D Decoder Pretraining 17

E Frozen-Probe Evaluation 17

F End-to-End Finetuning 18

G Total Compute Footprint 18

H Complete Results 18

- H.1 Architecture-Objective Sweep under Every Probe . . . 19
- H.2 Decoder Ablation under Every Probe . . . 19
- H.3 Training-Recipe Variants . . . 19
- H.4 Final Comparison and Every Run . . . 19

I Motion Probe, Attribution Controls and Robustness 20

- I.1 Motion-Inversion Probe . . . 21
- I.2 Attribution Controls . . . 24
- I.3 End-to-End Finetuning. . . . 25
- I.4 Seeds and Training Budget . . . 26

J Broader Impact 26

K Limitations and Future Work Discussion 27

L Data, Licenses, and Release 27

A Appearance-vs-Motion Diagnostic

This section provides a lightweight diagnostic for interpreting where TT-VidT helps and where broader appearance-sensitive representations remain stronger. The experiment is intended as a quick observation and source of insight, not as a formal benchmark or proof of motion understanding.

We compare each dataset's best motion-aware score from our trained video entries against a single-frame DINOv3 ViT-B attentive baseline, used as an appearance-only reference [Siméoni et al., 2025]. Figure 3 plots DINOv3 single-frame accuracy on the x-axis and the best motion-model attentive score on the y-axis. Points above the diagonal indicate datasets where temporal information appears to add value over single-frame recognition; points at or below the diagonal suggest appearance-dominated regimes. Since a single frame can still contain implicit motion cues, this comparison should be read as suggestive evidence rather than a strict separation of appearance and motion.

The diagnostic helps explain the profile observed in the main experiments. Jester and ARID lie above the diagonal: their best frozen attentive probe over trained video entries exceeds the single-frame DINOv3 baseline by a clear margin, consistent with TT-VidT's gains concentrating on motion-sensitive recognition. Something-Something V2 is not shown in the diagnostic plot but exhibits the same pattern in the architecture-objective sweep (Table 1). Diving48 lies below the diagonal in the frozen-probe view, but full fine-tuning recovers a strong gain (18.63 vs. DisMo with dual augmentation 8.12); we therefore treat Diving48 as complementary fine-tuning evidence rather than a frozen-probe diagnostic point.

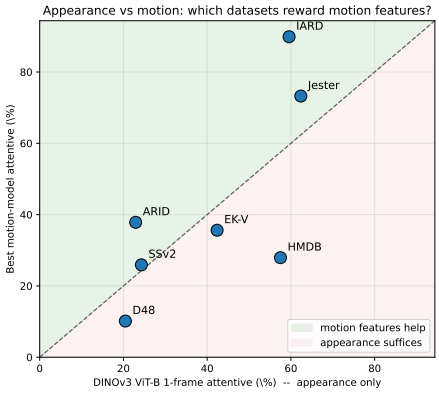

Figure 3: Each point is a dataset. X-axis: DINOv3 ViT-B single-frame attentive accuracy (appearance baseline). Y-axis: best motion-model attentive across our entries. Region above the diagonal: motion features add value over appearance.

HMDB51 and IARD illustrate the opposite side of the profile. HMDB51 is appearance-discriminable: a single-frame DINOv3 probe already explains much of the recognition signal, leaving less room for motion features. IARD requires a separate identity-control check because its appearance-vs-motion position is dominated by actor identity rather than by motion modeling. Table 7 reports frozen kNN@20 identity classification over the five actors in IARD, where lower identity accuracy indicates less retained per-frame identity.

Table 7: IARD identity classification (5 actors, random split). kNN@20 over frozen attentive features. **Lower is better:** a motion-prioritized representation should retain less per-frame identity. Random baseline is 20.00%.

Method	kNN@20
DINOv3 1f (appearance reference)	100.00
V-JEPA 2, 16f	98.79
DisMo with dual augmentation	99.56
VideoMAE	98.90
TT-VidT	**70.99**
random baseline	20.00

Across the four pretrained video baselines, only TT-VidT's identity accuracy (70.99) drops substantially below the appearance reference (100.00); V-JEPA 2 (98.79), DisMo with dual augmentation (99.56), and VideoMAE (98.90) all retain near-complete identity information. This supports the interpretation that TT-VidT's compact motion channel discards more per-frame identity than the broader baselines. EPIC-Kitchens similarly mixes verb, noun, and scene signals, so broader appearance-sensitive baselines retain an advantage on its mixed evaluations. These observations bound the motion-priority claim: TT-VidT targets motion-sensitive recognition, while appearance-heavy or anticipation-heavy settings can favor other operating points.

B Architecture Sizing

We report analytical parameter counts and forward-pass FLOPs for every encoder and decoder configuration used in the paper. Encoder FLOPs in Table 8 cover one full forward pass over a video of $T{=}8$ frames at $256{\times}256$, including the per-frame spatial path and any temporal mixing. The DINOv3 ViT-B row is reported both per-frame and at the $T{=}8$ scale, so it can be read against the temporal architectures on equal footing. V-JEPA2 ingests $T{=}16$ raw frames and produces 8 latent frames through tube embedding; we list both numbers for transparency. TT-VidT in either TT1D or TT3D form remains within roughly $1.2\times$ of the $T{=}8$ per-frame DINOv3 cost, which is the basis for the matched comparison setting in the main paper.

Table 9 lists the three decoder sizes (S, B, L) used in the decoder ablation, with and without first-frame cross-attention. The xattn rows correspond to the appearance-anchored configuration used by Diff Compression or AR-series setup, where the decoder cross-attends to the full z_k ($k = 1$ for diff compression, $k = t - dt$ for AR series). The default decoder used throughout the main experiments is decS with cross-attention enabled.

Table 8: Encoder analytical parameters and forward-pass FLOPs at $T{=}8$, $256{\times}256$.

Encoder	Params	FLOPs (GF)	Scope
DINOv3 ViT-B (1-frame)	85.5M	47.1	1 frame, 256^2
DINOv3 ViT-B ($T{=}8$ per-frame)	85.5M	377.2	$T{=}8$, 256^2
VideoMAE-3D (24L, 768d)	170.5M	1012.6	$T{=}8$, 256^2
DisMo-2D-3D (12s+12t, 768d)	170.5M	874.7	$T{=}8$, 256^2
TT-VidT TT1D (ours)	175.2M	400.5	$T{=}8$, 256^2
TT-VidT TT3D (ours)	196.4M	456.1	$T{=}8$, 256^2
V-JEPA2 (24L $\times$ 768d, tube=2)	171.0M	1009.9	$T{=}16$ (8 latent), 256^2

Table 9: Decoder analytical parameters and forward-pass FLOPs at $T{=}8$. L, D, and I denote depth, hidden dim, and FFN inner dim. xattn marks decoders with first-frame cross-attention.

Variant	L	D	I	xattn	Params	FLOPs (GF)
decS	12	768	3072	–	129.8M	133.8
decS	12	768	3072	✓	165.2M	157.8
decB	16	1024	4096	–	306.2M	316.3
decB	16	1024	4096	✓	390.1M	373.2
decL	28	1152	3456	–	562.7M	584.1
decL	28	1152	3456	✓	748.5M	710.1

C Encoder Pretraining

Encoder pretraining uses OpenVid-1M [Nan et al., 2025] at 384 px together with Moments-in-Time [Monfort et al., 2020], sampled at 6 FPS to $T{=}8$ frames at $256{\times}256$. Frames are mapped to a $16{\times}16{\times}4$ latent through a frame VAE before reconstruction. The same recipe applies to every encoder in the architecture-objective sweep, so differences across rows of the main sweep table reflect architecture and objective rather than schedule. All encoder pretraining ran on B200 GPUs. Hyperparameters are listed in Table 10.

D Decoder Pretraining

The decoder is pretrained separately so that the same checkpoint can be reused across encoder runs in the sweep. We use either ImageNet-1k [Deng et al., 2009] (single-frame, repeated to fill the $T{=}33$ frame budget) or the same video corpus as the encoder, and evaluate both initializations in the main paper. For decoder-only pretraining, we use latent regression as a lightweight surrogate for the Diff Compression reconstruction target. Decoder pretraining ran on H100 GPUs, separate from the B200 hardware used for encoder pretraining and finetuning. Hyperparameters are in Table 11.

E Frozen-Probe Evaluation

Frozen-probe evaluation uses the protocol summarized in Table 12. We report six probes (kNN, linear, layer-weighted linear, MLP, layer-weighted MLP, attentive) over mean-pooled features. Small datasets (HMDB51, ARID, IARD, Diving48) use batch 64 for 100 epochs; large datasets (Jester, SSv2, EK-100) use batch 256 for 20 epochs. Features are extracted at $T{=}8$ by default, with $T{=}16$ used for Diving48 where longer temporal context matters. We use ImageNet normalization for DINOv3 and V-JEPA2 baselines, and $[0.5, 0.5, 0.5]$ normalization for our models, matching the statistics each backbone was trained on.

Table 10: Encoder pretraining hyperparameters. Hardware: B200.

Setting	Value
Pretraining datasets	OpenVid-1M (384 px) + Moments-in-Time v2
Frame count T	8
Sample FPS	6
Resolution	256×256
Latent space (frame VAE)	$16 \times 16 \times 4$
Spatial backbone init	DINOv3 ViT-B/16
Decoder	DiT, S (default) / B / L
Decoder pretrain init	ImageNet / video
Motion tokens K	8
TT3D spatial downsample f	4 ($16\times16 \rightarrow 4\times4$)
Optimizer	AdamW
Betas	$(0.9, 0.98)$
Weight decay	0.01
Peak learning rate	5e-4
LR schedule	cosine, min ratio 0.01
Warmup steps	10 000
Gradient clip	0.1
Epochs	8 (main) / 20 (long-training)
Per-GPU batch size	16
GPUs	2 B200
Effective global batch	32 (2 GPU $\times$ 16)
Gradient accumulation	1

Table 11: Decoder pretraining hyperparameters (DiT, S/B/L). Effective global batch 256 across all three sizes. Hardware: H100.

Setting	Value
Epochs	3
Effective global batch	256
Per-GPU batch	8
Gradient accumulation	1 (S) / 2 (B) / 4 (L)
Optimizer	AdamW
Betas	$(0.9, 0.98)$
Weight decay	0.01
Peak learning rate	5e-4
LR schedule	cosine, min ratio 0.01
Warmup steps	1 000
Frame count T	33 (multi-jump token counts $1/2/4/8$)
Resolution	256×256 pixel (16×16 latent)
Latent space	$16 \times 16 \times 4$ (frame VAE)
Decoder objective	latent regression (DiffComp-style)
Pretrain data, ImageNet init	ImageNet-1k (single frame, repeated)
Pretrain data, Video init	OpenVid-1M (384 px) + Moments-in-Time v2

F End-to-End Finetuning

We finetune end-to-end on Diving48 for 50 epochs. The encoder uses a much smaller learning rate than the head ($1e{-5}$ vs $1e{-3}$) to preserve the pretrained features while still allowing adaptation. The classification head is an attentive pool followed by a 2-layer MLP. We keep augmentation deliberately light (resize and horizontal flip on train; resize only on eval) so that the comparison reflects what the encoder offers rather than augmentation engineering. Finetuning ran on B200 GPUs. Hyperparameters are in Table 13.

Table 12: Frozen-probe evaluation hyperparameters.

Setting	Value
Frozen probes	kNN, linear, linear-lw, MLP, MLP-lw, attentive (6)
kNN k	20 (cosine NN on mean-pooled features)
Probe MLP hidden	512
Probe MLP dropout	0.1
Probe optimizer	AdamW
Probe learning rate	1e-3
Probe weight decay	1e-4
Probe LR schedule	constant
Probe batch (small datasets)	64 (HMDB51, ARID, IARD, Diving48)
Probe batch (large datasets)	256 (Jester, SSv2, EK-100)
Probe epochs (small)	100
Probe epochs (large)	20
Feature extraction frames T	8 (default) / 16 (Diving48)
Sampling mode	center (eval), uniform (train)
Resolution	256×256
Normalization	ImageNet (DINOv3, V-JEPA2) or $[0.5, 0.5, 0.5]$ (ours)

Table 13: End-to-end finetuning hyperparameters. Hardware: B200.

Setting	Value
Finetune dataset	Diving48 (50 epochs)
Encoder LR	1e-5
Head LR	1e-3
Weight decay	0.05
Warmup epochs	5
Per-GPU batch	2
Gradient accumulation	16 (effective batch 32)
Head	Attentive pool + 2-layer MLP (hidden 512, dropout 0.1)
Augmentation (train)	Resize 256, RandomHorizontalFlip, ImageNet norm
Augmentation (eval)	Resize 256, ImageNet norm
LR schedule	cosine
Optimizer	AdamW

G Total Compute Footprint

We report the wall-clock cost of every component of the work. Encoder pretraining and downstream finetuning ran on B200 GPUs; decoder pretraining ran on H100 GPUs. GPU-hours are reported on the hardware actually used for each component, and we do not attempt a cross-GPU normalization. The encoder pretrain row in Table 14 reports the median run cost per architecture using decS, taken over the runs in the architecture-objective sweep. Table 15 gives the per-checkpoint cost for the six decoder configurations (S/B/L $\times$ ImageNet/Video init), all at effective global batch 256. Table 16 aggregates across all components for the full study.

H Complete Results

This section reports every pretrained run of this work on every evaluation. All runs follow the matched recipe of Section 4.1 unless a table names a change. Frozen evaluation uses the six probes of Appendix E: kNN@20, linear and MLP heads over mean-pooled or learned-weight pooled tokens, and the attentive probe used in the main text. Finetuning follows Appendix F.

Table 14: Encoder pretraining cost on B200, median across runs in the sweep, decS only.

Architecture	GPU-hours (B200)
TT1D (ours)	78.5
TT3D (ours)	87.7
DisMo-2D-3D	91.1
VideoMAE-3D	92.0

Table 15: Decoder pretraining cost on H100, per checkpoint at effective global batch 256.

Variant	Init	GPUs	Wall time	GPU-hours (H100)
decS	ImageNet	2	9 h 00	18.0
decS	Video	2	12 h 20	24.7
decB	ImageNet	2	11 h 15	22.5
decB	Video	2	15 h 20	30.7
decL	ImageNet	4	8 h 20	33.3
decL	Video	4	11 h 50	47.3

Table 16: Total compute footprint of this work. Decoder pretraining is on H100; all other components are on B200. We do not normalize across GPU types.

Component	Hardware	GPU-hours
Decoder pretrain (6 ckpts)	H100	$\approx$ 177
Encoder pretrain (decS, n=35)	B200	3,663
Diving48 finetune ($\approx$ 30 runs)	B200	$\approx$ 700
Feature extraction	B200	24
Frozen probing	B200	19
Total		$\approx$ 4,580

H.1 Architecture-Objective Sweep under Every Probe

Figure 4 follows the canonical cells across the six probes, and Figure 5 shows the attentive probe of all 24 cells on every task. TT3D with Diff Compression is the darkest cell of its row on Jester, SSv2, ARID and Diving48 finetuning, and the lead of TT-VidT on Jester and SSv2 holds under every probe. Tables 19 to 22 give every cell under every probe, and Tables 17 and 18 the end-to-end finetunes.

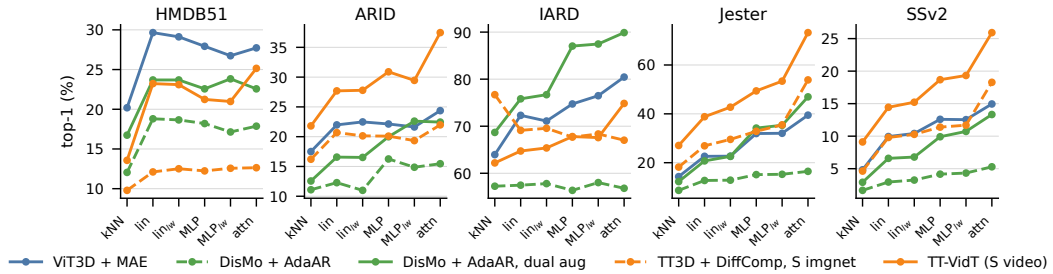

Figure 4: The canonical configurations under the six frozen probes, from the weakest (kNN) to the strongest (attentive) readout.

Table 17: End-to-end finetuning of every sweep cell on Diving48, 50 epochs.

Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
ViT3D	8.43	6.29	6.45	7.66	7.16	5.84
DisMo	5.28	6.29	6.45	6.70	5.43	6.65
TT1D	10.91	9.04	8.58	7.66	13.71	10.61
TT3D	15.38	9.04	11.42	10.30	<u>15.99</u>	**16.35**

Table 18: End-to-end finetuning of the sweep cells on EPIC-Kitchens-100 verb classification. ViT3D and DisMo were not finetuned.

Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
ViT3D	-	-	-	-	-	-
DisMo	-	-	-	-	-	-
TT1D	37.92	37.82	36.07	38.54	39.24	38.23
TT3D	<u>42.78</u>	36.61	37.29	36.19	**43.80**	42.49

H.2 Decoder Ablation under Every Probe

Figure 6 plots the decoder ablation of Section 4.3 on every task. The video-pretrained decoder is above the ImageNet-pretrained one at every size on the motion-heavy tasks, and size S or B gives the best result on every task. Table 23 repeats the ablation under every probe, Table 24 on the downstream tasks, and Table 25 separates decoder pretraining from the objective.

H.3 Training-Recipe Variants

Three variants change one element of the recipe. Table 26 applies DisMo-style dual augmentation to every architecture. It raises Jester for every architecture and DisMo the most, which is why the final comparison uses DisMo in this setting. Table 27 varies the DisMo decoder, and Table 28 adds Kinetics-710 [OpenMMLab, 2024] to the pretraining mixture.

H.4 Final Comparison and Every Run

Table 29 extends Table 5 to every probe and to V-JEPA 2 at two frame rates. Table 30 lists every pretrained run of this work on every task.

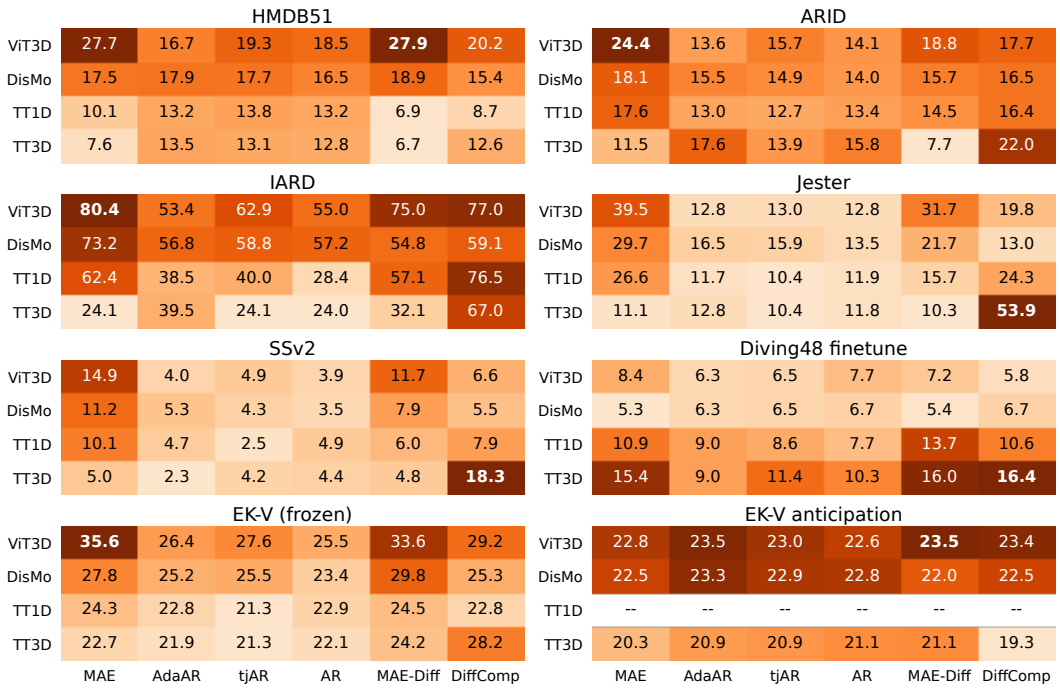

Figure 5: Attentive probe (finetune for Diving48) of the 24 architecture-objective cells on every task. Colour is scaled per panel, the best cell is bold.

Table 19: Architecture-objective sweep on HMDB51 (left) and ARID (right), top-1 accuracy (%) under all six frozen probes. **Bold** and underline mark the best and second-best cell within each probe and dataset.

		HMDB51						ARID					
Probe	Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
kNN	ViT3D	**20.19**	10.46	13.24	12.51	18.20	13.50	**17.50**	9.30	9.95	10.00	17.40	12.63
	DisMo	12.77	12.05	12.11	11.98	11.71	11.58	12.78	11.09	13.58	10.64	13.13	12.18
	TT1D	5.89	9.53	8.93	10.46	4.90	4.90	12.38	5.72	8.50	9.85	11.24	12.18
	TT3D	5.10	9.00	9.07	9.13	5.03	9.79	10.64	6.32	7.41	10.49	12.78	16.21
linear	ViT3D	**29.65**	17.21	19.72	18.40	27.47	19.46	**21.98**	11.93	11.69	10.94	17.90	14.42
	DisMo	20.71	18.80	18.07	17.14	19.26	17.87	17.45	12.28	15.32	11.29	16.21	16.41
	TT1D	11.05	13.83	11.25	12.84	10.52	10.66	14.52	6.66	12.73	6.17	11.88	15.71
	TT3D	11.65	12.64	11.85	13.30	11.18	12.11	14.82	10.84	11.34	5.82	10.89	20.69
$\text{linear}_{\text{1w}}$	ViT3D	**29.12**	17.54	19.99	18.73	27.20	19.13	**22.48**	11.93	10.49	12.48	17.90	13.03
	DisMo	20.45	18.66	18.27	17.47	20.91	17.34	16.46	10.99	15.17	11.74	16.16	13.67
	TT1D	12.84	13.63	11.65	12.91	11.65	10.13	14.62	9.90	11.34	6.02	13.28	16.01
	TT3D	11.78	12.38	12.51	12.97	11.38	12.51	16.56	8.70	11.24	6.22	9.60	20.14
MLP	ViT3D	**27.93**	16.81	19.92	18.73	26.80	20.98	**22.13**	14.87	17.01	15.37	16.36	20.64
	DisMo	18.40	18.20	16.88	17.14	16.28	17.34	16.91	16.26	14.57	15.86	14.47	16.81
	TT1D	8.93	13.10	12.11	12.57	6.55	7.21	14.87	13.87	14.97	14.57	15.27	14.87
	TT3D	6.55	13.96	11.45	11.71	7.81	12.24	10.94	17.21	15.17	13.33	8.06	20.09
MLP_{1w}	ViT3D	**26.74**	17.87	20.05	18.13	26.41	22.44	**21.63**	14.52	17.75	14.12	18.25	20.59
	DisMo	19.39	17.14	16.68	15.35	18.40	16.81	16.16	14.87	15.61	13.97	15.71	15.56
	TT1D	9.33	11.98	13.50	12.57	5.96	7.48	13.87	13.72	13.13	17.75	13.92	15.42
	TT3D	6.88	13.10	10.92	12.31	7.15	12.57	12.03	14.72	14.77	13.58	9.85	19.34
attentive	ViT3D	27.73	16.68	19.32	18.46	**27.93**	20.19	**24.37**	13.58	15.66	14.12	18.75	17.70
	DisMo	17.54	17.87	17.74	16.48	18.86	15.42	18.15	15.46	14.87	14.02	15.71	16.51
	TT1D	10.06	13.24	13.83	13.24	6.88	8.74	17.65	13.03	12.73	13.38	14.52	16.41
	TT3D	7.61	13.50	13.10	12.84	6.68	12.64	11.49	17.65	13.92	15.76	7.71	21.98

I Motion Probe, Attribution Controls and Robustness

This section collects the experiments added after submission. Figure 7 summarizes them.

Table 20: Architecture-objective sweep on IARD (left) and Jester (right), top-1 accuracy (%) under all six frozen probes. **Bold** and underline mark the best and second-best cell within each probe and dataset.

		IARD						Jester					
Probe	Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
kNN	ViT3D	63.96	40.99	61.21	41.98	64.73	71.87	14.30	6.38	5.55	5.92	14.59	11.27
	DisMo	50.55	57.25	49.45	46.81	57.36	50.55	9.04	8.68	9.76	6.21	9.37	8.02
	TT1D	48.35	28.68	24.62	27.80	41.54	49.23	11.21	4.63	4.34	4.67	8.64	10.25
	TT3D	35.16	34.62	33.19	26.04	28.46	**76.70**	5.69	5.46	4.47	4.60	5.26	**18.21**
linear	ViT3D	**72.31**	44.51	55.27	54.73	69.12	59.12	22.61	12.45	11.63	12.36	22.71	13.34
	DisMo	61.76	57.47	54.29	52.31	62.64	52.42	15.18	12.72	13.21	12.36	15.54	12.68
	TT1D	49.12	34.29	33.19	42.20	57.36	50.44	14.36	10.42	10.42	10.42	12.22	13.93
	TT3D	52.75	32.53	49.23	36.26	28.68	69.12	12.19	10.42	10.42	10.42	11.76	**26.91**
$\text{linear}_{\text{lw}}$	ViT3D	**71.10**	42.75	51.43	54.29	70.22	60.11	22.67	12.45	11.63	12.22	23.73	13.31
	DisMo	62.09	57.80	55.93	52.31	65.60	56.26	15.64	12.85	13.08	12.39	17.75	12.65
	TT1D	49.12	33.08	37.14	42.53	52.20	50.33	15.74	10.42	10.42	10.42	14.53	15.15
	TT3D	52.75	31.32	46.59	42.09	24.40	69.56	13.01	10.42	10.42	10.42	13.21	**29.54**
MLP	ViT3D	**74.73**	57.47	69.67	61.21	70.88	66.48	31.94	13.08	14.20	12.65	30.00	18.80
	DisMo	57.91	56.37	57.47	51.10	58.57	61.98	20.08	15.12	15.51	13.54	19.36	12.72
	TT1D	56.15	53.63	40.33	60.55	58.90	60.44	18.24	10.45	10.42	10.45	13.70	16.37
	TT3D	33.85	44.84	46.04	53.41	28.68	67.69	11.40	10.42	10.42	10.45	9.83	**32.86**
MLP_{lw}	ViT3D	**76.48**	56.59	70.44	63.74	70.33	68.79	32.04	13.05	14.23	12.82	30.66	18.40
	DisMo	57.80	58.02	58.57	51.87	59.67	61.21	20.37	15.25	15.77	13.77	18.11	12.45
	TT1D	57.03	53.08	43.19	60.77	59.56	60.22	18.63	10.42	10.42	10.48	15.28	16.79
	TT3D	37.36	46.81	45.05	54.18	32.86	68.35	11.73	10.78	10.45	10.45	10.65	**35.49**
attentive	ViT3D	**80.44**	53.41	62.86	54.95	74.95	77.03	39.47	12.85	13.01	12.85	31.68	19.82
	DisMo	73.19	56.81	58.79	57.25	54.84	59.12	29.71	16.46	15.94	13.54	21.66	12.98
	TT1D	62.42	38.46	40.00	28.35	57.14	76.48	26.65	11.70	10.45	11.86	15.68	24.32
	TT3D	24.07	39.45	24.07	23.96	32.09	67.03	11.07	12.85	10.42	11.83	10.32	**53.89**

Table 21: Architecture-objective sweep on Something-Something V2 (left) and Diving48 (frozen) (right), top-1 accuracy (%) under all six frozen probes. **Bold** and underline mark the best and second-best cell within each probe and dataset.

		Something-Something V2						Diving48 (frozen)					
Probe	Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
kNN	ViT3D	**4.83**	1.44	1.61	1.51	4.24	1.99	8.58	7.41	6.95	8.32	**8.98**	6.95
	DisMo	1.90	1.67	1.69	1.42	1.89	1.67	6.60	7.66	8.07	7.21	6.45	7.92
	TT1D	2.12	1.12	1.10	1.07	1.67	2.30	6.09	6.35	6.40	6.04	5.08	5.23
	TT3D	1.26	1.08	1.10	1.14	1.33	4.63	4.92	6.65	6.75	6.45	5.28	6.14
linear	ViT3D	**9.94**	2.66	2.99	2.73	9.41	4.35	8.22	7.82	7.36	8.38	7.21	**9.09**
	DisMo	5.64	2.95	3.48	2.39	6.22	2.93	7.46	7.41	7.72	7.31	8.07	8.73
	TT1D	5.16	1.93	1.60	1.79	4.06	4.56	6.55	6.40	5.63	5.89	5.94	6.75
	TT3D	3.29	1.74	1.82	1.78	3.45	9.80	6.60	5.43	5.94	5.48	6.80	7.77
$\text{linear}_{\text{lw}}$	ViT3D	**10.41**	2.72	3.02	2.58	9.44	4.29	8.48	7.01	8.88	7.77	**9.29**	8.22
	DisMo	6.06	3.25	3.64	2.59	6.48	3.02	8.43	7.41	8.07	7.31	8.78	7.26
	TT1D	5.50	1.98	1.89	1.94	4.46	5.32	6.14	6.24	5.63	6.04	6.40	5.69
	TT3D	3.89	1.73	1.77	1.81	3.79	10.28	6.95	5.94	6.04	6.19	6.04	6.60
MLP	ViT3D	**12.59**	4.23	4.34	3.66	11.51	5.88	9.14	8.73	8.83	7.97	8.93	7.26
	DisMo	7.05	4.16	4.15	3.31	7.94	3.59	7.72	7.87	8.43	**9.19**	8.73	8.43
	TT1D	5.34	2.21	1.95	2.40	5.12	4.75	5.48	6.95	5.89	5.94	4.87	5.84
	TT3D	4.21	2.18	2.04	2.12	4.44	11.39	5.48	6.45	7.56	6.70	5.74	5.63
MLP_{lw}	ViT3D	**12.54**	4.14	4.55	3.73	11.85	5.92	**9.19**	7.56	7.92	8.48	8.43	8.48
	DisMo	7.29	4.33	4.97	3.25	8.27	3.80	8.93	7.66	8.17	8.32	8.53	7.41
	TT1D	5.17	2.14	2.28	2.41	5.78	5.17	5.53	6.55	6.75	6.35	4.72	6.40
	TT3D	4.62	2.40	2.04	2.23	4.63	11.69	5.84	6.14	7.72	6.45	6.65	6.35
attentive	ViT3D	14.95	3.99	4.86	3.93	11.73	6.57	8.68	7.11	7.87	8.83	8.93	7.16
	DisMo	11.20	5.29	4.35	3.54	7.89	5.53	8.17	**9.19**	8.17	8.58	8.32	6.70
	TT1D	10.05	4.71	2.53	4.86	6.00	7.91	5.69	7.06	7.06	6.14	5.18	5.58
	TT3D	4.98	2.26	4.16	4.39	4.82	**18.28**	6.90	7.26	8.88	6.29	6.45	5.63

I.1 Motion-Inversion Probe

SSv2 contains class pairs that are exact mirrors under a pixel transform (Table 31): three pairs under horizontal flip (left and right), two under vertical flip (up and down), and four under time reversal (towards and away, closer and apart). Time reversal changes only the frame order. For each frozen encoder we train an attentive probe on clean features without flip or reversal augmentation, keep the validation clips the probe classifies correctly, transform the raw frames, re-encode them, and classify with the same probe. A flip is a move to the mirror class and shows the encoder re-read the motion. A stay keeps the original class and shows the decision came from appearance, since the content is

Table 22: Architecture-objective sweep on EK-100 verb (left) and EK-100 verb anticipation (right), top-1 accuracy (%) under all six frozen probes. **Bold** and underline mark the best and second-best cell within each probe and dataset. TT1D was not evaluated on anticipation.

		EK-100 verb						EK-100 verb anticipation					
Probe	Arch	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp	MAE	AdaAR	tjAR	AR	MAE-Diff	DiffComp
*k*NN	ViT3D	**31.01**	27.32	29.41	27.81	30.79	28.18	20.68	20.29	**21.83**	20.18	21.33	20.15
	DisMo	26.04	25.52	25.40	25.46	25.71	25.31	20.52	19.85	20.17	20.29	20.97	19.94
	TT1D	20.12	21.75	21.35	22.60	19.78	19.45	-	-	-	-	-	-
	TT3D	19.97	21.39	21.48	21.57	20.35	21.97	19.40	19.10	18.76	18.93	19.56	19.85
linear	ViT3D	**31.77**	23.20	23.75	22.36	30.67	26.85	22.09	21.81	22.61	22.13	**23.32**	22.62
	DisMo	25.34	21.64	22.70	21.87	26.94	22.70	21.15	22.68	21.70	22.62	21.49	20.76
	TT1D	23.32	20.78	19.68	20.21	21.71	21.35	-	-	-	-	-	-
	TT3D	21.34	19.66	19.83	20.14	21.43	25.22	21.19	20.47	20.28	20.42	21.67	20.64
linear_{1w}	ViT3D	**31.84**	22.97	25.42	22.37	31.08	26.48	22.96	23.07	22.42	22.23	**23.22**	22.76
	DisMo	27.85	23.20	22.68	21.57	26.14	23.12	22.25	22.23	21.67	22.66	21.89	21.39
	TT1D	23.75	20.71	19.86	20.10	21.59	21.18	-	-	-	-	-	-
	TT3D	21.66	19.82	19.87	20.12	21.91	25.81	21.57	20.49	20.64	20.73	21.20	20.92
MLP	ViT3D	**34.99**	25.82	26.32	25.22	33.69	29.11	22.66	22.92	22.59	22.46	**23.62**	23.07
	DisMo	27.80	25.39	24.43	23.68	29.78	24.40	22.34	23.13	22.26	22.53	22.05	22.55
	TT1D	23.42	22.36	20.11	22.35	22.20	21.46	-	-	-	-	-	-
	TT3D	21.42	20.30	20.51	22.30	22.60	27.13	20.93	20.96	20.93	21.12	21.52	20.08
MLP_{1w}	ViT3D	**34.54**	25.85	27.60	26.16	33.84	28.43	**23.31**	22.53	22.73	22.59	23.27	23.03
	DisMo	28.27	24.58	24.91	23.51	27.55	23.87	21.57	22.34	22.22	22.59	22.11	22.93
	TT1D	24.11	22.08	20.17	22.16	24.04	21.86	-	-	-	-	-	-
	TT3D	22.28	22.24	21.39	21.99	22.31	27.20	21.12	21.36	20.61	21.23	20.49	19.40
attentive	ViT3D	**35.62**	26.43	27.58	25.48	33.62	29.24	22.81	23.47	22.97	22.62	**23.53**	23.39
	DisMo	27.77	25.20	25.48	23.41	29.79	25.33	22.51	23.30	22.86	22.81	21.99	22.47
	TT1D	24.28	22.79	21.34	22.87	24.46	22.75	-	-	-	-	-	-
	TT3D	22.72	21.85	21.28	22.07	24.19	28.23	20.30	20.85	20.94	21.12	21.13	19.34

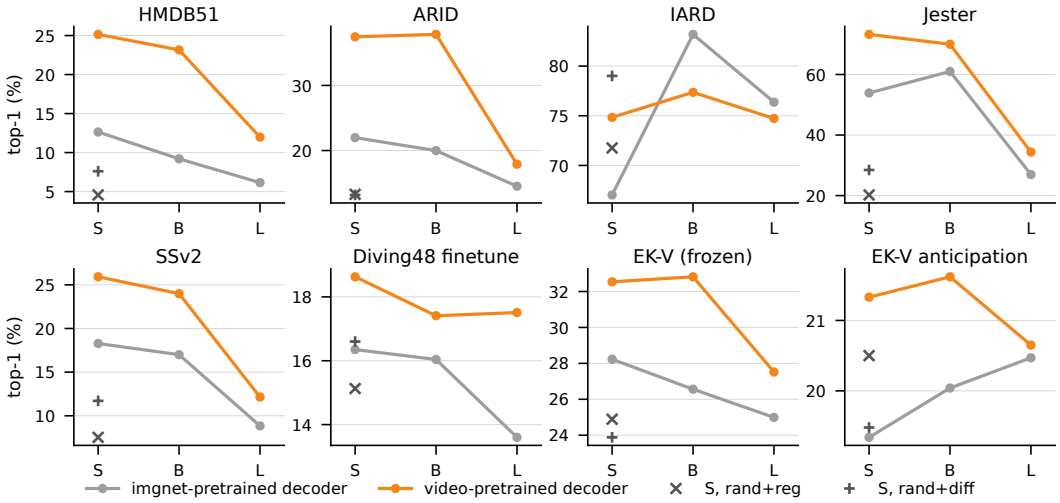

Figure 6: Decoder size and initialization on TT3D + Diff Compression, attentive probe (finetune for Diving48). Crosses are size-S decoders from random initialization.

otherwise identical. The single-frame DINOv3 probe is the appearance-only control. Only pretrained checkpoints are probed, since finetuning would mix in supervision. The TT-VidT and VideoMAE rows use matched probe data of 300 clips per class, and a second TT-VidT pretraining run reproduces the pattern.

TT-VidT keeps the original class on at most 1% of clips on every axis, while every baseline keeps it on 9% to 43%, most under time reversal. The control scores 0.0 flip and 100.0 stay under reversal, as a representation without access to frame order must. The shuffle column shows that TT-VidT and VideoMAE both use frame order, and the flip test shows that only TT-VidT reads its direction faithfully.

Table 23: Decoder ablation on TT3D + Diff Compression under all six frozen probes. Rows name the decoder size and its initialization (rand+reg and rand+diff are size S from random initialization with a regression or diffusion auxiliary loss). Best and second-best per probe and column marked.

Probe	Decoder	HMDB51	ARID	IARD	Jester	SSv2
kNN	rand+reg	3.90	9.60	56.48	6.93	1.46
	rand+diff	9.00	12.58	74.84	17.42	3.93
	S imgnet	9.79	16.21	**76.70**	18.21	4.63
	S video	**13.57**	**21.81**	62.20	**27.05**	**9.11**
	B imgnet	8.47	14.54	71.87	15.41	2.60
	B video	13.30	20.04	66.26	22.57	6.89
	L imgnet	4.96	12.38	73.30	11.76	1.95
	L video	8.93	13.63	71.10	14.69	2.58
linear	rand+reg	7.94	14.12	70.66	17.55	5.73
	rand+diff	10.52	13.92	71.98	23.89	8.11
	S imgnet	12.11	20.69	69.12	26.91	9.80
	S video	23.23	27.69	64.73	**38.82**	**14.44**
	B imgnet	10.46	15.77	76.04	28.14	7.83
	B video	**24.88**	**28.43**	70.44	35.36	13.19
	L imgnet	10.52	16.76	72.42	21.26	6.52
	L video	16.02	18.50	**78.46**	26.32	7.60
linear_{1w}	rand+reg	8.01	13.67	74.18	18.14	6.43
	rand+diff	12.97	12.13	72.64	25.57	8.48
	S imgnet	12.51	20.14	69.56	29.54	10.28
	S video	23.10	27.79	65.38	**42.73**	**15.22**
	B imgnet	10.59	16.52	77.14	31.20	8.31
	B video	**24.95**	**29.34**	69.89	39.05	13.86
	L imgnet	12.24	16.81	73.63	22.38	6.78
	L video	15.62	17.90	**79.23**	28.62	8.16

Probe	Decoder	HMDB51	ARID	IARD	Jester	SSv2
MLP	rand+reg	5.43	13.97	69.56	20.74	7.31
	rand+diff	7.68	13.13	76.48	26.91	10.53
	S imgnet	12.24	20.09	67.69	32.86	11.39
	S video	21.24	30.89	67.80	**49.39**	**18.68**
	B imgnet	9.46	17.48	**78.02**	37.55	10.72
	B video	**21.97**	**31.21**	75.93	46.37	17.26
	L imgnet	6.35	14.37	77.58	24.91	8.62
	L video	11.25	17.65	73.96	31.02	11.39
MLP_{1w}	rand+reg	5.03	13.23	71.32	20.93	7.79
	rand+diff	7.74	13.53	**79.23**	27.90	11.46
	S imgnet	12.57	19.34	68.35	35.49	11.69
	S video	20.98	29.45	67.58	**53.39**	**19.33**
	B imgnet	9.33	16.68	76.92	41.35	11.37
	B video	**22.57**	**31.43**	76.92	50.90	18.29
	L imgnet	6.29	13.03	76.15	27.05	8.55
	L video	9.73	18.10	72.97	34.11	11.51
attentive	rand+reg	4.57	13.28	71.76	20.21	7.53
	rand+diff	7.61	13.18	79.01	28.46	11.71
	S imgnet	12.64	21.98	67.03	53.89	18.28
	S video	**25.15**	37.47	74.84	**73.25**	**25.92**
	B imgnet	9.20	19.99	**83.19**	61.00	17.00
	B video	23.16	**37.84**	77.36	69.99	23.99
	L imgnet	6.15	14.52	76.37	26.95	8.84
	L video	11.98	17.90	74.73	34.37	12.15

Table 24: Decoder ablation on the downstream tasks: frozen attentive probe and end-to-end finetune (FT) on Diving48 and EPIC-Kitchens-100 verb classification, and frozen verb anticipation.

	Diving48		EK-V		Antic.
Decoder	frozen	FT	frozen	FT	frozen
rand+reg	5.53	15.13	24.89	41.68	20.50
rand+diff	6.70	16.60	23.88	**43.18**	19.48
S imgnet	5.63	16.35	28.23	42.49	19.34
S video	8.17	**18.63**	32.54	40.68	21.33
B imgnet	6.09	16.04	26.56	41.76	20.04
B video	**9.64**	17.41	**32.82**	42.31	**21.62**
L imgnet	5.69	13.60	24.99	40.60	20.47
L video	7.16	17.51	27.52	39.42	20.65

Table 25: TT3D decoder initialization for two objectives: the size-S decoder pretrained on ImageNet against random initialization with a diffusion or regression loss.

Probe	Objective	Decoder init	HMDB51	ARID	IARD	Jester	SSv2
kNN	DiffComp	pretrained	9.79	**16.21**	**76.70**	**18.21**	**4.63**
	DiffComp	rand+diff	9.00	12.58	74.84	17.42	3.93
	DiffComp	rand+reg	3.90	9.60	56.48	6.93	1.46
	AdaAR	pretrained	9.00	6.32	34.62	5.46	1.08
	AdaAR	rand+diff	**11.18**	9.60	33.63	4.14	1.31
	AdaAR	rand+reg	10.92	8.35	50.44	6.47	1.90
attentive	DiffComp	pretrained	12.64	**21.98**	67.03	**53.89**	**18.28**
	DiffComp	rand+diff	7.61	13.18	**79.01**	28.46	11.71
	DiffComp	rand+reg	4.57	13.28	71.76	20.21	7.53
	AdaAR	pretrained	13.50	17.65	39.45	12.85	2.26
	AdaAR	rand+diff	17.41	12.33	34.07	9.76	3.29
	AdaAR	rand+reg	**18.20**	10.69	57.47	11.07	5.63

Table 26: DisMo-style dual augmentation (encoder and decoder see different spatial augmentations) applied to every architecture, size-S ImageNet decoder. Each cell is no augmentation / dual augmentation. (a) and (b) are the two TT3D runs of Table 30.

Probe	Arch	Objective	HMDB51	ARID	IARD	Jester	SSv2
kNN	ViT3D	AdaAR	10.46 / 16.02	9.30 / 9.83	40.99 / 46.59	6.38 / 11.14	1.44 / 2.51
	ViT3D	tjAR	13.24 / 15.95	9.95 / 14.38	61.21 / 48.68	5.55 / 10.90	1.61 / 2.82
	DisMo	AdaAR	12.05 / 16.74	11.09 / 12.56	57.25 / 68.68	8.68 / 12.27	1.67 / 2.91
	DisMo	tjAR	12.11 / 16.41	13.58 / 11.81	49.45 / 55.82	9.76 / 12.58	1.69 / 2.70
	TT1D	AdaAR	9.53 / 6.75	5.72 / 14.22	28.68 / 24.40	4.63 / 5.53	1.12 / 1.45
	TT1D	tjAR	8.93 / 6.95	8.50 / 7.75	24.62 / 27.36	4.34 / 4.93	1.10 / 1.10
	TT3D	AdaAR (a)	9.00 / 12.51	6.32 / 10.96	34.62 / 53.63	5.46 / 5.59	1.08 / 1.44
	TT3D	AdaAR (b)	9.00 / 11.25	6.32 / 11.28	34.62 / 45.71	5.46 / 5.52	1.08 / 1.48
	TT3D	tjAR	9.07 / 11.78	7.41 / 9.67	33.19 / 36.15	4.47 / 5.73	1.10 / 1.31
attentive	ViT3D	AdaAR	16.68 / 22.96	13.58 / 17.64	53.41 / 70.77	12.85 / 32.62	3.99 / 9.78
	ViT3D	tjAR	19.32 / 20.78	15.66 / 17.85	62.86 / 72.86	13.01 / 30.61	4.86 / 10.05
	DisMo	AdaAR	17.87 / 22.57	15.46 / 22.45	56.81 / 89.89	16.46 / 46.95	5.29 / 13.33
	DisMo	tjAR	17.74 / 24.49	14.87 / 21.49	58.79 / 85.60	15.94 / 48.23	4.35 / 13.10
	TT1D	AdaAR	13.24 / 10.85	13.03 / 20.26	38.46 / 20.00	11.70 / 15.39	4.71 / 5.14
	TT1D	tjAR	13.83 / 8.93	12.73 / 14.11	40.00 / 20.00	10.45 / 12.25	2.53 / 2.39
	TT3D	AdaAR (a)	13.50 / 19.52	17.65 / 16.62	39.45 / 65.71	12.85 / 30.16	2.26 / 10.38
	TT3D	AdaAR (b)	13.50 / 18.33	17.65 / 17.42	39.45 / 51.43	12.85 / 28.67	2.26 / 9.26
	TT3D	tjAR	13.10 / 19.52	13.92 / 17.80	24.07 / 57.36	10.42 / 27.48	4.16 / 9.40

Table 27: Decoder size and initialization for DisMo + AdaAR with dual augmentation, all six probes.

Probe	Decoder	HMDB51	ARID	IARD	Jester	SSv2
kNN	S imgnet	16.74	12.56	**68.68**	12.27	**2.91**
	S video	**17.47**	**12.67**	52.31	12.16	2.78
	B imgnet	15.09	10.15	52.53	**12.37**	2.70
	B video	17.01	7.27	43.85	11.27	2.47
linear	S imgnet	23.69	16.57	75.82	20.74	6.59
	S video	**24.95**	**18.23**	67.58	19.65	6.09
	B imgnet	20.38	14.86	**79.45**	**22.63**	**7.05**
	B video	20.12	14.11	50.66	20.21	5.92
linear$_{1w}$	S imgnet	23.69	16.52	76.70	22.58	6.79
	S video	**25.15**	**18.07**	67.80	20.97	6.27
	B imgnet	20.85	14.86	**80.33**	**23.22**	**7.20**
	B video	20.25	14.11	51.76	21.87	6.03

Probe	Decoder	HMDB51	ARID	IARD	Jester	SSv2
MLP	S imgnet	22.57	19.99	**87.03**	34.17	9.91
	S video	**26.21**	**20.52**	70.99	30.90	10.05
	B imgnet	25.22	19.78	83.52	**34.98**	**10.89**
	B video	22.30	19.99	67.69	33.40	9.90
MLP$_{1w}$	S imgnet	23.83	**22.61**	**87.47**	35.31	10.73
	S video	**26.74**	21.17	71.54	33.31	10.56
	B imgnet	25.08	19.29	83.74	**35.96**	**11.21**
	B video	22.30	18.65	67.25	34.93	10.21
attentive	S imgnet	22.57	**22.45**	**89.89**	46.95	13.33
	S video	**25.35**	19.35	79.01	47.31	13.23
	B imgnet	23.03	20.84	85.60	**47.90**	**13.91**
	B video	23.49	18.17	76.92	46.28	12.52

Table 28: TT3D + Diff Compression with Kinetics-710 added to the pretraining mixture (6 epochs on the larger mixture against 8 epochs on OpenVid and Moments-in-Time).

Probe	Decoder	Data	HMDB51	ARID	IARD	Jester	SSv2
kNN	S imgnet	OpenVid+MiT	9.79	16.21	**76.70**	18.21	4.63
	S imgnet	+K710	7.02	13.08	73.08	12.55	2.71
	S video	OpenVid+MiT	**13.57**	**21.81**	62.20	**27.05**	**9.11**
	S video	+K710	11.25	17.11	73.74	21.33	7.49

Probe	Decoder	Data	HMDB51	ARID	IARD	Jester	SSv2
attentive	S imgnet	OpenVid+MiT	12.64	21.98	67.03	53.89	18.28
	S imgnet	+K710	7.28	13.67	**82.64**	31.28	11.88
	S video	OpenVid+MiT	**25.15**	**37.47**	74.84	**73.25**	**25.92**
	S video	+K710	15.88	28.79	79.89	44.13	18.43

Table 29: Final comparison under all six frozen probes. V-JEPA 2 is evaluated with 8 frames at 6 fps and 16 frames at 12 fps. The single-frame DINOv3 row is the appearance reference (attentive probe only). Best and second-best per probe and column marked, the reference excluded.

Probe	Method	HMDB51	ARID	IARD	Jester	SSv2
kNN	V-JEPA 2 (8f)	19.26	7.91	61.76	11.24	3.94
	V-JEPA 2 (16f)	18.07	10.04	**73.52**	9.76	3.85
	VideoMAE	**20.19**	17.50	63.96	14.30	4.83
	DisMo (dual aug)	16.74	12.56	68.68	12.27	2.91
	TT-VidT	13.57	**21.81**	62.20	**27.05**	**9.11**
linear	V-JEPA 2 (8f)	25.88	16.51	61.98	21.98	9.33
	V-JEPA 2 (16f)	25.94	17.01	74.73	22.61	9.82
	VideoMAE	**29.65**	21.98	72.31	22.61	9.94
	DisMo (dual aug)	23.69	16.57	**75.82**	20.74	6.59
	TT-VidT	23.23	**27.69**	64.73	**38.82**	**14.44**
linear$_{1w}$	V-JEPA 2 (8f)	24.88	15.66	70.55	23.00	9.69
	V-JEPA 2 (16f)	25.74	18.50	76.37	23.20	9.89
	VideoMAE	**29.12**	22.48	71.10	22.67	10.41
	DisMo (dual aug)	23.69	16.52	**76.70**	22.58	6.79
	TT-VidT	23.10	**27.79**	65.38	**42.73**	**15.22**

Probe	Method	HMDB51	ARID	IARD	Jester	SSv2
MLP	V-JEPA 2 (8f)	23.83	13.97	68.90	25.99	9.09
	V-JEPA 2 (16f)	22.70	14.77	83.08	26.42	10.04
	VideoMAE	**27.93**	22.13	74.73	31.94	12.59
	DisMo (dual aug)	22.57	19.99	**87.03**	34.17	9.91
	TT-VidT	21.24	**30.89**	67.80	**49.39**	**18.68**
MLP$_{1w}$	V-JEPA 2 (8f)	22.57	14.07	63.85	25.50	9.22
	V-JEPA 2 (16f)	22.10	14.87	81.87	26.59	10.22
	VideoMAE	**26.74**	21.63	76.48	32.04	12.54
	DisMo (dual aug)	23.83	22.61	**87.47**	35.31	10.73
	TT-VidT	20.98	**29.45**	67.58	**53.39**	**19.33**
attentive	V-JEPA 2 (8f)	21.38	13.67	68.35	27.31	9.71
	V-JEPA 2 (16f)	22.70	16.01	79.34	27.41	10.59
	VideoMAE	**27.73**	24.37	80.44	39.47	14.95
	DisMo (dual aug)	22.57	22.45	**89.89**	46.95	13.33
	TT-VidT	25.15	**37.47**	74.84	**73.25**	**25.92**
	DINOv3 1 frame	57.51	22.92	59.56	62.37	-

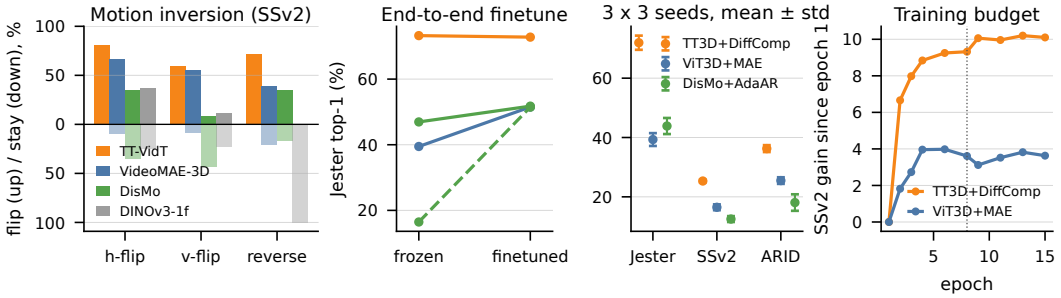

Figure 7: From left: the motion-inversion probe on SSv2 (flip above the axis, stay below), frozen probe against end-to-end finetuning on Jester (dashed: DisMo without augmentation), the canonical cells over three pretraining and three probe seeds, and the cumulative SSv2 gain over a 15-epoch continuation (the dotted line marks the 8-epoch budget).

I.2 Attribution Controls

Table 34 isolates the two ingredients TT-VidT shares with other work. A 3D ViT initialized from DINOv3 lands on the scratch ViT3D + MAE row for every motion-heavy dataset and moves only IARD, so the DINOv3 initialization does not create the motion regime. The initialization covers 12 of the 24 layers, the depth of the DINOv3 ViT-B teacher. A VTok-style motion token computed as an explicit feature difference [Wang et al., 2026] already beats every non-TT configuration on Jester,

Table 30: Every pretrained run of this work, frozen attentive probe on every task and end-to-end finetune (FT) where run. Best and second-best per column marked. Decoder is size S pretrained on ImageNet unless named. TT3D + AdaAR with dual augmentation was trained twice, with (b) and without (a) a final LayerNorm in the decoder.

Run	HMDB51	ARID	IARD	Jester	SSv2	D48	D48 FT	EK-V	EK-V FT	Antic.
ViT3D, MAE	27.73	24.37	80.44	39.47	14.95	8.68	8.43	**35.62**	-	22.81
ViT3D, AdaAR	16.68	13.58	53.41	12.85	3.99	7.11	6.29	26.43	-	23.47
ViT3D, AdaAR, dual aug	22.96	17.64	70.77	32.62	9.78	9.34	7.77	28.62	-	23.33
ViT3D, tjAR	19.32	15.66	62.86	13.01	4.86	7.87	6.45	27.58	-	22.97
ViT3D, tjAR, dual aug	20.78	17.85	72.86	30.61	10.05	8.43	8.73	29.78	-	23.59
ViT3D, AR	18.46	14.12	54.95	12.85	3.93	8.83	7.66	25.48	-	22.62
ViT3D, MAE-Diff	**27.93**	18.75	74.95	31.68	11.73	8.93	7.16	33.62	-	23.53
ViT3D, DiffComp	20.19	17.70	77.03	19.82	6.57	7.16	5.84	29.24	-	23.39
DisMo, MAE	17.54	18.15	73.19	29.71	11.20	8.17	5.28	27.77	-	22.51
DisMo, AdaAR	17.87	15.46	56.81	16.46	5.29	9.19	6.29	25.20	-	23.30
DisMo, AdaAR, dual aug	22.57	22.45	**89.89**	46.95	13.33	9.19	8.12	31.97	-	22.72
DisMo, AdaAR, B imgnet, dual aug	23.03	20.84	85.60	47.90	13.91	9.44	7.11	32.75	-	23.70
DisMo, AdaAR, B video, dual aug	23.49	18.17	76.92	46.28	12.52	7.51	6.60	30.93	-	22.60
DisMo, AdaAR, S video, dual aug	25.35	19.35	79.01	47.31	13.23	**10.15**	7.72	31.69	-	**23.77**
DisMo, tjAR	17.74	14.87	58.79	15.94	4.35	8.17	6.45	25.48	-	22.86
DisMo, tjAR, dual aug	24.49	21.49	85.60	48.23	13.10	8.48	8.88	32.00	-	23.62
DisMo, AR	16.48	14.02	57.25	13.54	3.54	8.58	6.70	23.41	-	22.81
DisMo, MAE-Diff	18.86	15.71	54.84	21.66	7.89	8.32	5.43	29.79	-	21.99
DisMo, DiffComp	15.42	16.51	59.12	12.98	5.53	6.70	6.65	25.33	-	22.47
TT1D, MAE	10.06	17.65	62.42	26.65	10.05	5.69	10.91	24.28	37.92	-
TT1D, AdaAR	13.24	13.03	38.46	11.70	4.71	7.06	9.04	22.79	37.82	-
TT1D, AdaAR, dual aug	10.85	20.26	20.00	15.39	5.14	7.16	8.22	21.69	33.97	-
TT1D, tjAR	13.83	12.73	40.00	10.45	2.53	7.06	8.58	21.34	36.07	-
TT1D, tjAR, dual aug	8.93	14.11	20.00	12.25	2.39	5.94	5.69	19.64	26.14	-
TT1D, AR	13.24	13.38	28.35	11.86	4.86	6.14	7.66	22.87	38.54	-
TT1D, MAE-Diff	6.88	14.52	57.14	15.68	6.00	5.18	13.71	24.46	39.24	-
TT1D, DiffComp	8.74	16.41	76.48	24.32	7.91	5.58	10.61	22.75	38.23	-
TT3D, MAE	7.61	11.49	24.07	11.07	4.98	6.90	15.38	22.72	42.78	20.30
TT3D, AdaAR	13.50	17.65	39.45	12.85	2.26	7.26	9.04	21.85	36.61	20.85
TT3D, AdaAR, S rand+diff	17.41	12.33	34.07	9.76	3.29	6.50	14.62	21.72	40.84	20.62
TT3D, AdaAR, S rand+reg	18.20	10.69	57.47	11.07	5.63	7.97	13.20	24.14	38.92	21.35
TT3D, AdaAR, dual aug, (a)	19.52	16.62	65.71	30.16	10.38	7.87	15.28	26.14	**45.38**	22.08
TT3D, AdaAR, dual aug, (b)	18.33	17.42	51.43	28.67	9.26	8.07	16.45	24.28	43.42	22.01
TT3D, tjAR	13.10	13.92	24.07	10.42	4.16	8.88	11.42	21.28	37.29	20.94
TT3D, tjAR, dual aug	19.52	17.80	57.36	27.48	9.40	7.31	16.24	25.04	43.61	22.20
TT3D, AR	12.84	15.76	23.96	11.83	4.39	6.29	10.30	22.07	36.19	21.12
TT3D, MAE-Diff	6.68	7.71	32.09	10.32	4.82	6.45	15.99	24.19	43.80	21.13
TT3D, DiffComp	12.64	21.98	67.03	53.89	18.28	5.63	16.35	28.23	42.49	19.34
TT3D, DiffComp, +K710	7.28	13.67	82.64	31.28	11.88	5.74	15.08	26.87	41.19	19.53
TT3D, DiffComp, B imgnet	9.20	19.99	83.19	61.00	17.00	6.09	16.04	26.56	41.76	20.04
TT3D, DiffComp, B video	23.16	**37.84**	77.36	69.99	23.99	9.64	17.41	32.82	42.31	21.62
TT3D, DiffComp, L imgnet	6.15	14.52	76.37	26.95	8.84	5.69	13.60	24.99	40.60	20.47
TT3D, DiffComp, L video	11.98	17.90	74.73	34.37	12.15	7.16	17.51	27.52	39.42	20.65
TT3D, DiffComp, S rand+diff	7.61	13.18	79.01	28.46	11.71	6.70	16.60	23.88	43.18	19.48
TT3D, DiffComp, S rand+reg	4.57	13.28	71.76	20.21	7.53	5.53	15.13	24.89	41.68	20.50
TT3D, DiffComp, S video	25.15	37.47	74.84	**73.25**	**25.92**	8.17	**18.63**	32.54	40.68	21.33
TT3D, DiffComp, S video, +K710	15.88	28.79	79.89	44.13	18.43	8.53	17.51	32.76	41.44	20.82

Table 31: The nine SSv2 class pairs of the motion-inversion probe (18 classes, SSv2 label ids in parentheses). The transform maps a clip of one class onto the motion of the other.

Transform	Class A	Class B
Horizontal flip	Pushing something from left to right (93) Pulling something from left to right (86) Turning the camera left while filming something (166)	Pushing something from right to left (94) Pulling something from right to left (87) Turning the camera right while filming something (167)
Vertical flip	Moving something up (45) Turning the camera upwards while filming something (168)	Moving something down (43) Turning the camera downwards while filming something (165)
Time reversal	Moving something towards the camera (44) Moving something and something closer to each other (37) Moving something closer to something (42) Approaching something with your camera (0)	Moving something away from the camera (41) Moving something and something away from each other (36) Moving something away from something (40) Moving away from something with your camera (32)

which credits the keyframe factorization, and the learned token of TT-VidT improves on it on all five datasets.

I.3 End-to-End Finetuning

Table 33 finetunes the whole encoder on Jester and SSv2. The baselines gain 12 to 35 points from supervision and converge near 51.5 on Jester, while TT-VidT stays at its frozen accuracy and leads by 21.0 points on Jester and 12.9 on SSv2.

Table 32: Motion-inversion probe on SSv2 validation clips, flip / stay (%). A flip moves the prediction to the mirror class after the input transform, a stay keeps the original class. Shuffle drop is the relative accuracy drop of a probe retrained on temporally shuffled features. DINOv3-1f is the single-frame appearance control.

Model	Horizontal flip	Vertical flip	Time reversal	Shuffle drop
TT-VidT	**81.2 / 0.6**	**59.5 / 0.9**	**71.4 / 1.0**	50.9%
VideoMAE-3D	66.9 / 9.7	55.1 / 9.1	38.6 / 20.9	50.0%
DisMo	34.5 / 35.0	8.4 / 43.2	34.5 / 16.4	31.9%
DINOv3-1f	37.1 / 24.5	11.8 / 22.7	0.0 / 100.0	3.1%

Table 33: Frozen attentive probe against 30 epochs of end-to-end finetuning on Jester and SSv2, one shared recipe with identical encoder and head learning rates.

	Jester		SSv2	
Model	frozen	FT	frozen	FT
ViT3D+MAE	39.47	51.50	14.95	13.40
DisMo+AdaAR (no aug)	16.46	51.46	5.29	12.31
DisMo+AdaAR (dual aug)	46.95	51.76	13.33	14.85
TT-VidT	73.25	72.79	25.92	27.72

Table 34: Attribution controls, frozen attentive probe. Top: initialization control, a 24-layer 3D ViT whose first 12 layers are initialized from DINOv3 ViT-B (the depth of the teacher) with 3D RoPE, trained exactly as ViT3D + MAE. Bottom: a VTok-style explicit feature-difference motion token against the learned token under the same encoder, decoder and recipe.

Model	HMDB51	ARID	IARD	Jester	SSv2
ViT3D+MAE (scratch)	27.73	24.37	80.44	39.47	14.95
ViT3D+MAE (DINOv3 init)	21.58	24.86	86.70	39.53	14.00
TT-VidT	25.15	37.47	74.84	73.25	25.92
TT3D, explicit feature diff	20.3	26.5	50.5	64.7	24.8
TT3D, learned token (TT-VidT)	25.15	37.47	74.84	73.25	25.92

Table 35: Continuation from 8 to 15 epochs under the same recipe with a fresh warmup and cosine schedule for both cells, mean $\pm$ standard deviation over 3 probe seeds. The 8-epoch rows re-probe the paper checkpoints, so they differ slightly from the 9-measurement means of Table 36.

Model	Epochs	Jester	SSv2	ARID
TT3D+DiffComp	8	74.23 ± 0.59	25.86 ± 0.16	36.22 ± 0.82
	15	75.20 ± 0.24	26.73 ± 0.14	36.60 ± 0.81
ViT3D+MAE	8	40.52 ± 0.59	15.34 ± 0.23	24.50 ± 0.80
	15	43.62 ± 0.75	16.99 ± 0.26	24.68 ± 0.48

I.4 Seeds and Training Budget

Table 36 repeats the three canonical cells with three pretraining seeds and three probe seeds each. On Jester, SSv2 and ARID the worst TT3D + Diff Compression measurement lies above the best ViT3D + MAE measurement. Probe-seed variance is about as large as pretraining-seed variance. On IARD the variance comes mostly from which of the five actors is held out, and under one fixed actor split all models land at a similar level. Tables 35 and 37 continue the two cells from 8 to 15 epochs. The ordering is unchanged, the Jester gap stays above 31 points, and every TT3D interval after epoch 6 changes SSv2 by less than one point. All conclusions of this paper are stated for the matched budget of Section 4.1.

Table 36: The three canonical cells over 3 pretraining seeds $\times$ 3 probe seeds, mean $\pm$ standard deviation over the 9 measurements, frozen attentive probe. The last row compares the worst TT3D + Diff Compression measurement with the best ViT3D + MAE measurement on the three motion-heavy benchmarks.

Model	Jester	SSv2	ARID	HMDB51	IARD
TT3D+DiffComp	71.95 ± 2.39	25.33 ± 0.44	36.26 ± 1.16	24.59 ± 1.14	87.16 ± 3.59
ViT3D+MAE	39.31 ± 2.16	16.48 ± 0.98	25.48 ± 1.11	27.08 ± 0.68	87.22 ± 6.67
DisMo+AdaAR	43.87 ± 2.71	12.51 ± 0.99	18.07 ± 2.78	22.04 ± 1.70	79.90 ± 6.52
worst TT3D / best ViT3D+MAE	67.78 / 43.25	24.83 / 17.94	34.34 / 27.20	-	-

Table 37: SSv2 attentive-probe change (points) between consecutive measured epochs of the 15-epoch continuation.

Interval	1→2	2→3	3→4	4→6	6→8	8→9	9→11	11→13	13→15
TT3D+DiffComp	+6.66	+1.32	+0.86	+0.41	+0.07	+0.74	−0.10	+0.24	−0.10
ViT3D+MAE	+1.82	+0.91	+1.23	+0.02	−0.37	−0.49	+0.40	+0.30	−0.19

J Broader Impact

TT-VidT is a representation-learning study rather than a deployed video understanding system. Its positive impact is mainly methodological: a more controlled way to test whether video SSL models

use temporal change can support better benchmarks, more efficient video encoders, and downstream applications where motion cues matter, such as robotics, assistive perception, sports or skill analysis, and scientific video understanding. The same improvements could also be misused in surveillance-sensitive settings, because action recognition can reveal behavioral information about people even when identity is not the target. Our experiments do not introduce a new dataset of people, do not release a high-risk deployed system, and emphasize diagnostic limitations; any deployment should still account for privacy, consent, fairness across capture conditions, and domain-specific failure modes.

K Limitations and Future Work Discussion

The main limitation is scale and statistical coverage. We study one matched pretraining mixture, one roughly 170M$\sim$190M encoder scale, and single-run sweep cells with multi-seed checks on the canonical ones; this is sufficient for a controlled architecture-objective comparison but not for claiming that the same profile will persist at much larger data, model, or schedule scales. Future work should test whether TT-VidT's motion-prioritized behavior survives longer pretraining, stronger data mixtures, and multi-seed evaluation. Another natural direction is to combine TT3D with DisMo-style dual augmentation, since our current comparison evaluates them as separate operating points. The appearance-vs-motion diagnostic is also intentionally lightweight: it uses a single-frame DINOv3 attentive probe as an appearance reference, so alternative appearance baselines such as DINOv2 or CLIP may shift dataset positions. Finally, selective unfreezing of the spatial encoder, richer temporal readouts, and broader evaluation on untrimmed or egocentric video tasks may clarify where compact motion channels help and where appearance-rich representations remain preferable.

L Data, Licenses, and Release

We use existing datasets and pretrained components rather than introducing a new data source. OpenVid-1M [Nan et al., 2025] is distributed for research use under CC-BY-4.0 while also requiring users to respect upstream video licenses; Moments-in-Time v2 [Monfort et al., 2020] and the downstream benchmarks are used through their original academic or public access channels and are not redistributed by this work. Our code and reproduction instructions are released at `https://github.com/KohakuBlueleaf/TTVidT` under the Apache License 2.0, while raw datasets should be obtained from their original providers under the corresponding terms.

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: The abstract and introduction state the matched-recipe scope, architecture–objective comparison, and boundary cases where TT-VidT is not dominant.

   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: The conclusion and appendix discuss the small-recipe scope, single-seed limitation, scale limitation, diagnostic assumptions, and future work directions.

   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: [N/A]

Justification: The paper is empirical and does not include theoretical theorems or formal proofs.

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: The method, experiment setup, and appendix specify the architecture, objectives, datasets, optimization recipe, probing protocol, finetuning recipe, and compute footprint needed to reproduce the reported results.

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: Code and reproduction instructions are publicly released at `https://github.com/KohakuBlueleaf/TTVidT` under the Apache License 2.0 (Appendix L), while datasets are obtained from their original public or academic access channels rather than redistributed.

   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: The main experiment section and appendix report the pretraining data, sampling, optimizer, schedules, batch sizes, resolutions, probe settings, finetuning settings, and normalization choices.

   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 three canonical configurations are repeated with three pretraining seeds and three probe seeds each, and Appendix I reports the mean and standard deviation over these nine measurements (Table 36, Figure 7) together with the worst-versus-best comparison. The 24-cell architecture-objective sweep is single-run, as stated in Section 4.2 and the limitations.

   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: [Yes]

   Justification: The appendix reports GPU type, per-run wall-clock/GPU-hour estimates for encoder and decoder pretraining, finetuning and probing costs, and the total compute footprint.

   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: The work uses existing academic/public datasets, reports controlled experiments, and does not introduce procedures that deviate from the NeurIPS Code of Ethics.

   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: The appendix discusses methodological benefits and potential negative uses in surveillance-sensitive video understanding settings, along with privacy and fairness considerations for deployment.

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: The paper does not release a new dataset or a high-risk deployed model requiring special access safeguards; the released research code carries no such risk.

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: Existing datasets and pretrained components are cited and discussed in the appendix; OpenVid-1M is CC-BY-4.0 for research/non-commercial use with upstream-license obligations, and other datasets are used through their original access terms.

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: [Yes]

Justification: The only new asset is the research code, released at `https://github.com/KohakuBlueleaf/TTVidT` under the Apache License 2.0 with documentation and reproduction instructions in the repository; no new dataset, benchmark, or model checkpoint is introduced.

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: The work uses existing video datasets and does not involve new crowdsourcing or direct human-subject experiments by the authors.

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: The work does not involve new human-subject recruitment or intervention by the authors, so IRB approval is not applicable.

    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: [N/A]

    Justification: LLMs are not used as a core methodological component of the research.

    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.