Streaming VideoLLMs for Real-Time Procedural Video Understanding

Supplementary Material

1. Temporal Variance Computation

To compute temporal variance in high-dimensional spaces, we follow the methodology of [21]. Language-aligned visual features, particularly CLIP, are observed to have low norms, so all features (`[CLS]` tokens) are normalized onto a unit hypersphere [16]. The temporal variance is given by:

$$\sigma_{T_c} = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{1}{T}\sum_{t=1}^{T}\|z_{i,t}-\mu_i\|_2\,, \quad \mu_i = \frac{1}{T}\sum_{t=1}^{T} z_{i,t} \tag{1}$$

Here, σ_{T_c} denotes the temporal variance for class c, computed as the average L_2-distance of each normalized feature $z_{i,t}$ from its temporal centroid μ_i. The temporal centroid μ_i is the mean of $z_{i,t}$ across all time steps T for instance i. For Fig. 3 of the main paper, we use $T = 8$ for representing a fine-grained segment for both datasets.

2. Ablations

Scaling ProVideLLM. We evaluate three variants of ProVideLLM: the efficiency-focused ProVideLLM-1B/5, the performance-focused ProVideLLM-8B/11, and the enhanced ProVideLLM-8B/11+ for tasks beyond daily egocentric footage. In Fig. 1, we report their performance on EgoExo4D [5] fine-grained keystep recognition (validation) and COIN [15] step recognition (test). Across both datasets, ProVideLLM-8B/11+ outperforms ProVideLLM-8B/11, which in turn significantly surpasses ProVideLLM-1B/5. Notably, ProVideLLM-8B/11+ improves COIN performance by 4.2% compared to ProVideLLM-8B/11 but shows marginal gains on EgoExo4D. This is attributed to COIN's noisy videos, which frequently omit hands, include transitions to irrelevant frames, and feature third-person views where actions are obscured. Leveraging SigLIP's `[CLS]` token, ProVideLLM-8B/11+ benefits from its alignment with large-scale web images, reducing the domain gap and improving performance on COIN.

Caching long-term observations: Visual or Text? In Tab. 1, we demonstrate that verbalizing the long-term past into compressed text tokens outperforms vision tokens in both accuracy and efficiency. We present a vision-only ProVideLLM baseline in row 3 but ProVideLLM does not have a two-step visual memory like LSTR [18]. To ensure a fair comparison, we extend our short-term visual cache to match LSTR's full temporal span of $m_S + m_L = 16 + 128 = 144$ seconds and use no textual long-term tokens. This vision-only baseline increases inference latency and memory usage due to redundant visual tokens for representing longer temporal span. Compressing 128 seconds of effective long-term past as text tokens, yields superior accuracy and runtime efficiency.

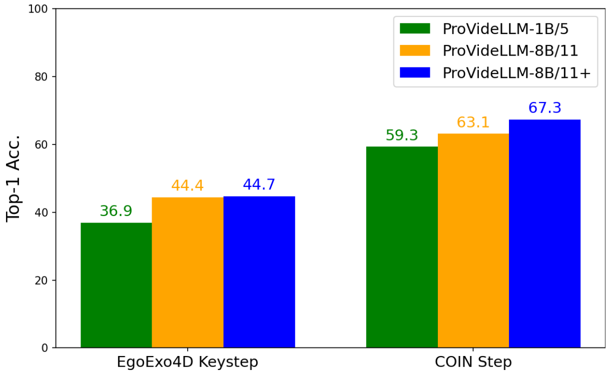


Figure 1. **Scaling ProVideLLM** on EgoExo4D [5] and COIN [15] for fine-grained step recognition. Larger models improve accuracy, with ProVideLLM-8B/11+ excelling on COIN due to reduced domain gap from SigLIP's alignment with large-scale web images.

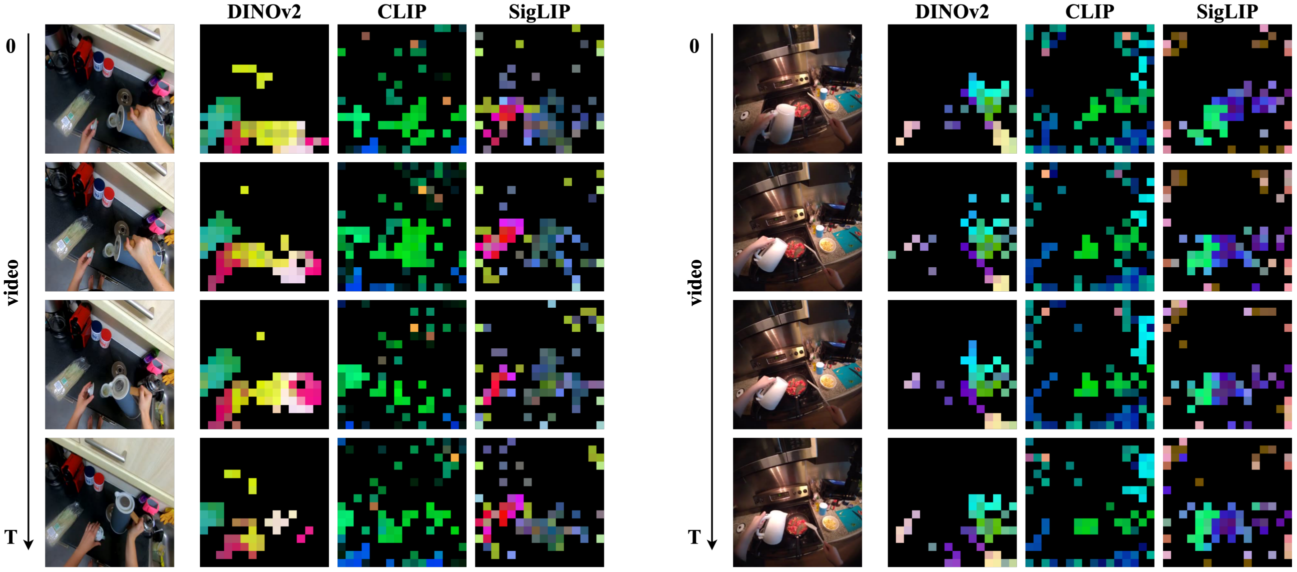
3. Additional Patch Visualizations

In Fig. 2, we extend the visualizations from Fig. 2 of the main paper, comparing fine-grained steps in egocentric datasets Ego4D Goal-Step [14] and EgoExo4D [5] for the activity step *"add water"*. For each visualization, we uniformly sample 16 frames from the video (consistent with the fine-grained evaluation setup) and compute the top-3 PCA components across the frames, thresholded by the first component [12]. Each component is matched to a different color channel. Results are shown for every 4th frame within the 16-frame window.

In Fig. 2, DINOv2 consistently identifies hands and objects-in-contact as primary factors of variation in Ego4D Goal-Step [14] and EgoExo4D [5], while language-aligned encoders like CLIP and SigLIP fail to highlight meaningful patterns. DINOv2 patch tokens also demonstrate gradual transformations of hands and objects-in-contact (e.g., yellow tokens in Ego4D Goal-Step), resulting in higher temporal variance. However, occasional activation of background elements (*e.g.* a *packet of noodles* in Ego4D Goal-Step or a *chopping board* in EgoExo4D) is observed, which can be partially attributed to ego-motion. These cases underscore the importance of our hand and object-in-contact detection loss in mitigating such distractions.

Model	Short-term Span (secs)	Long-term Span (secs)	Long-term Token type	per-frame mAP (Val)	FPS (↑)	Memory (GB) (↓)
LSTR [18]	16	128	Visual	8.9	-	-
ProVideLLM-1B/5	16 + 128 = 144	-	Visual	12.4	5.3	2.8
ProVideLLM-1B/5 (+long-term verbalization)	16	128	Text	**13.0**	**9.1**	**2.0**

Table 1. **Vision vs. text tokens for representing long-term observations** in Ego4D Goal-Step [14]. Verbalizing long-term past into compressed language tokens improves accuracy, reduces memory usage, and enhances inference speed compared to vision-only baseline.



(a) Ego4D Goal-Step

(b) EgoExo4D

Figure 2. **Visualizations of visual encoder patches** for egocentric videos (*Ego4D Goal-Step [14]* and *EgoExo4D* [5]) for the step *"add water"* highlight that DINOv2 patch tokens effectively capture hand-object interactions and their gradual transformations over time, unlike language-aligned ones CLIP and SigLIP, which fail to highlight any recognizable pattern.

4. Additional Dataset Details

In this section, we describe the four datasets used for our state-of-the-art benchmarking:

Ego4D Goal-Step [14] is a subset of Ego4D [4] featuring 430 hours of egocentric footage across 86 goals, including cooking, harvesting, and housekeeping. For online step detection, we combine steps and substeps, resulting in 514 categories and 47.7k segments. Each segment averages 32.5 seconds, while videos containing a single goal average just over 30 minutes with approximately 20 step segments per goal. Following the original paper [14], we evaluate online step detection with per-frame mean average precision.

EgoExo4D [5] is a recently introduced multi-view dataset with 1286 hours of egocentric and exocentric footage. For fine-grained keystep recognition, we use the Keystep benchmark, which covers 689 categories across 17 activities, including cooking, bike repair, and healthcare, with an average segment length of 11 seconds. All our fine-grained experiments and ablations on this dataset uses egocentric views for both training and evaluation. Fine-grained keystep recognition is evaluated using Top-1 accuracy.

COIN [15] is a widely used instructional video dataset containing 476 hours of YouTube videos spanning 180 daily tasks. COIN features 778 step categories with videos averaging 3.91 step segments and a segment duration of 14.91 seconds. While ProVideLLM is designed for continuous egocentric footage [5, 14], COIN videos often include transitions, cuts, and irrelevant frames within step segments. To ensure compatibility and fair comparison with SoTAs [1, 9, 17], we use the SigLIP `[CLS]` token. Following [1, 11, 17], we evaluate on five benchmarks: *(i)* step recognition, *(ii)* task recognition, *(iii)* (next) step forecasting, *(iv)* (long-term) procedure forecasting, and *(v)* (long-term) procedure forecasting when task is specified, using Top-1 Accuracy as a metric for all benchmarks.

Assembly101 [13] is a large-scale multi-view procedural dataset comprising 513 hours of assembling and disassembling 101 take-apart toy vehicles annotated with 1380 fine-grained steps. Each video averages 7.1 minutes and 236.7

steps where each step lasts 1.7 seconds on average. For forecasting, we follow the dataset authors' anticipation split, excluding steps like *"attempt to"*, resulting in 1064 categories. From the 12 available views, we use `v4` for both training and evaluation. The anticipation task is to forecast (anticipate) steps $\tau_a = 1$ seconds before it occurs. To address prediction uncertainty, we adopt class-mean Top-5 recall [3], as used in the original paper.

5. Additional Implementation Details

This section provides supplementary implementation details to ensure reproducibility. All training, evaluation, and benchmarking were conducted on 8 V100 32GB GPUs. Only runtime results (Tab. 10 of main paper and Tab. 1) are reported on a single A6000 GPU, excluding flash attention [2] for fair comparison. Hyperparameters for training are listed in Tab. 2. We set $\lambda_1 = 1$ for Stage-1 pre-training. Consistent with comparisons to VideoLLM-online [1], we applied LoRA [6] with $(r = 128, \alpha = 256)$ to fine-tune the LLM during Stage-2. Stage-2 training takes 20 epochs on average over all datasets for supervised loss convergence. For classification tasks, generated outputs were matched to action categories using edit distance, again following VideoLLM-online for consistency. For multi-task learning (Tab. 8 of the main paper), we uniformly sampled 16 frames from the current step segment without incorporating additional past context for all tasks.

Hyperparameters	Stage-1	Stage-2
batch size	128	128
learning rate (lr)	1e-3	1.5e-4
lr schedule decay	cosine	cosine
lr warmup ratio	0.03	0.05
weight decay	0	0
optimizer	AdamW [10]	AdamW [10]
epochs	5	20

Table 2. **Training Hyperparameters.**

DETR-QFormer. For our two main variants, ProVideLLM-1B/5 and ProVideLLM-8B/11, we employ two distinct DETR-QFormer architectures. The smaller model features a 4-layer cross-attention transformer decoder with 8 heads, a hidden dimension of 128, 2 visual queries, 2 hand queries, and 2 object queries. Hand and object queries are average-pooled separately before being passed to the LLM, resulting in 5 tokens per frame (including `[CLS]`). The larger model features a 6-layer cross-attention transformer decoder with 8 heads, a hidden dimension of 512, 4 visual queries, 2 hand queries, and 4 object queries, producing 11 tokens per frame, including `[CLS]`. The design of DETR-QFormer prioritizes token efficiency, as demonstrated in Tab. 8 of the main paper. While increasing the token count per frame could potentially enhance accuracy, it would come at the cost of additional computational overhead.

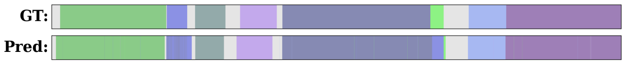


Figure 3. **Sample visualization of Online Step Detection** of a video from Ego4D Goal-Step [14]. The video contains 8 activity steps from which 7 were correctly identified.

Instruction Prompts. For experiments on COIN [15], we adopt the instruction prompts from VideoLLM-online [1]. For EgoExo4D fine-grained keystep recognition, we use the prompt: *"Describe the activity step being performed in the video. Format your answer concisely. No extra text output."*. For Ego4D Goal-Step online action detection, our prompt is: *"Please output the corresponding action of each frame. If a frame does not show any action, output background."*. When using multimodal interleaved cache, we append: *"Any previous actions performed are prepended with* `<L>` *and interleaved with visual frames."*. Here, `<L>` can be single token from the LLM's vocabulary. For cross-dataset generalization benchmarks, we provide the set of action (step) categories in the prompt, prepended with: *"Categorize the following video into one of the following classes. Classes are written in the following form: <index of the output class>: <name of the output class>. Reply with just the <name of the output class> and nothing else."*.

Cross-Dataset Generalization Benchmark. Here, we provide additional details omitted from the main paper due to space constraints. The objective of the cross-dataset generalization benchmark is to evaluate the ability of current VideoLLMs, trained on large-scale image-text and video-text datasets, to recognize procedural steps they were not explicitly trained on. We term this benchmark *cross-dataset* rather than *zero-shot*, as the majority of activity steps were seen by the VideoLLMs in some context. For the Ego4D Goal-Step to EgoExo4D transfer, approximately 30 actions were shared between the datasets. The two strong VideoLLM baselines in this benchmark are VideoLLaVA [8], which uses LanguageBind [22] as the vision encoder and Vicuna v1.5 as the language decoder, and Llava-OneVision [7], which uses SigLIP [20] as the vision encoder and Qwen-2 [19] as the language decoder. Both baselines rely on a 2-layer MLP with GELU activation as their connector. For evaluation, we assume the goal is known, framing the task as a classification problem restricted to relevant steps within the goal. As with all our experiments, ProVideLLM's generated outputs are matched with step categories using minimum edit distance, whereas the VideoLLMs leverage a separate language expert, Llama-3.1-8B-Instruct, to map their outputs to the categories. However, ProVideLLM is only trained for 2 epochs on Ego4D Goal-Step to reduce overfitting.

References

[1] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024. 2, 3

[2] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022. 3

[3] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 3

[4] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2

[5] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 1, 2

[6] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 3

[7] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 3

[8] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 3

[9] Han Lin, Tushar Nagarajan, Nicolas Ballas, Mido Assran, Mojtaba Komeili, Mohit Bansal, and Koustuv Sinha. Vedit: Latent prediction architecture for procedural video representation learning. *arXiv preprint arXiv:2410.03478*, 2024. 2

[10] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[11] Medhini Narasimhan, Licheng Yu, Sean Bell, Ning Zhang, and Trevor Darrell. Learning and verification of task structure in instructional videos. *arXiv preprint arXiv:2303.13519*, 2023. 2

[12] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[13] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 2

[14] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. *Advances in Neural Information Processing Systems*, 36, 2023. 1, 2, 3

[15] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 1, 2, 3

[16] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020. 1

[17] Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. *arXiv preprint arXiv:2408.16730*, 2024. 2

[18] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34:1086–1099, 2021. 1, 2

[19] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 3

[20] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 3

[21] Shihao Zhang, Linlin Yang, Michael Bi Mi, Xiaoxu Zheng, and Angela Yao. Improving deep regression with ordinal entropy. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[22] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, WANG HongFa, Yatian Pang, Wenhao Jiang, Junwu Zhang, Zongwei Li, Cai Wan Zhang, Zhifeng Li, Wei Liu, and Li Yuan. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *The Twelfth International Conference on Learning Representations*, 2024. 3