Attend Before Attention: Efficient and Scalable Video Understanding via Autoregressive Gazing

Baifeng Shi[1,4*] Stephanie Fu[1*] Long Lian[1] Hanrong Ye[4]
David Eigen[3] Aaron Reite[3] Boyi Li[1,4] Jan Kautz[4] Song Han[2,4]
David M. Chan[1†] Pavlo Molchanov[4†] Trevor Darrell[1†] Hongxu Yin[4†]

[1]UC Berkeley [2]MIT [3]Clarifai [4]NVIDIA

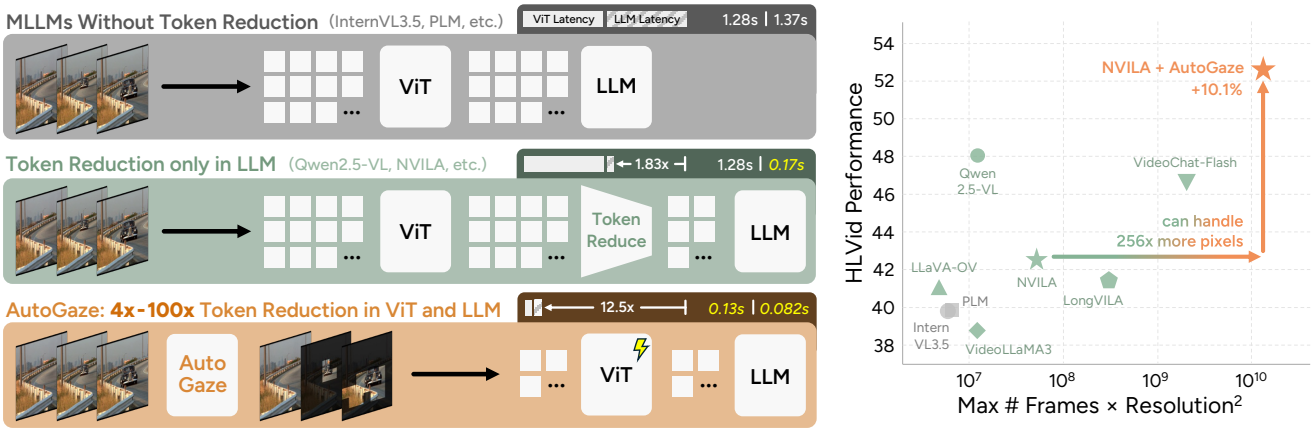

Figure 1. We propose **AutoGaze**, which **reduces the computational cost of video understanding** to **scale MLLMs to long, high-resolution videos**. *(Left)* Existing MLLMs either process all pixels which is inefficient, or prune tokens only in their LLMs, leaving ViTs the computational bottleneck. In contrast, AutoGaze eliminates redundant patches by up to 100× *before* ViTs, accelerating ViTs and MLLMs by up to 19×. *(Right)* This efficiency enables MLLMs with AutoGaze to scale to 1K-frame, 4K-resolution videos and achieve superior performance on HLVid, our new long, high-resolution video benchmark, surpassing prior MLLMs limited to short or low-resolution videos.

Abstract

Multi-modal large language models (MLLMs) have advanced general-purpose video understanding but struggle with long, high-resolution videos—they process every pixel equally in their vision transformers (ViTs) or LLMs despite significant spatiotemporal redundancy. We introduce ***AutoGaze****, a lightweight module that removes redundant patches before processed by a ViT or an MLLM. Trained with next-token prediction and reinforcement learning, AutoGaze autoregressively selects a minimal set of multi-scale patches that can reconstruct the video within a user-specified error threshold, eliminating redundancy while preserving information. Empirically, AutoGaze reduces visual tokens by 4×-100× and accelerates ViTs and MLLMs by up to 19×, enabling scaling MLLMs to 1K-frame 4K-resolution videos and achieving superior results on video benchmarks (e.g., 67.0% on VideoMME). Furthermore, we introduce* ***HLVid****: the first high-resolution, long-form video QA benchmark with 5-minute 4K-resolution videos, where an MLLM scaled with AutoGaze improves over the baseline by 10.1% and outperforms the previous best MLLM by 4.5%. Project page: https://autogaze.github.io/.*

1. Introduction

When observing a moving scene, humans don't process every detail equally. Our eyes dart around to moving objects, capture fine details, and skip over static backgrounds, efficiently understanding scenes by selectively attending to informative regions [2, 40, 41, 66]. This allows us to process high-FPS, high-resolution video streams in real time. In contrast, modern video understanding models (e.g., multi-modal large language models (MLLMs) [6, 38, 46, 53, 77]) still process

*Equal contribution. †Equal advising.

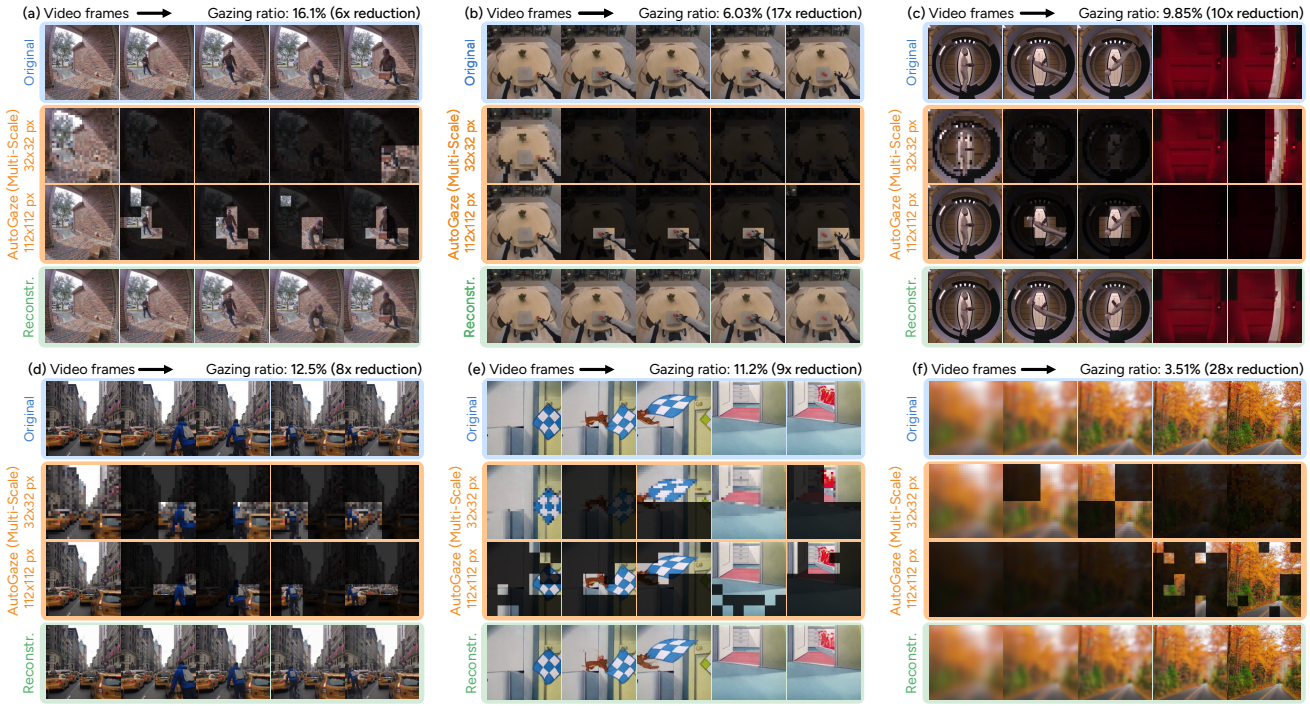

Figure 2. **What is AutoGaze paying attention to?** For each example, we show the original video, multi-scale gazed patches, and reconstructed video. Note that we only show gazing on two scales to save space while AutoGaze actually uses four. In general, AutoGaze can **1)** focus on moving objects while removing redundancy in static regions (a-e), **2)** adapt to scene changes by selecting more patches (c, e), and **3)** distribute attention with different granularity based on detailedness (c, f). This allows AutoGaze to select a small ratio of patches (gazing ratio) without much information loss, as reflected by the reconstruction quality.

every pixel in every frame equally, wasting computation due to spatiotemporal redundancy in videos [13, 45, 74, 91, 92]. For example, in Fig. 2 (top-left), the static background only needs to be viewed once. Thus, these models cannot scale to *long-form* and *high-resolution* videos crucial for real-world applications [19, 20, 31, 65, 100] due to computational cost.

Recent work attempts to reduce video redundancy in MLLMs, but typically prunes tokens only in the LLM while the vision transformer (ViT) still processes all pixels, creating a huge efficiency bottleneck that prevents scaling to longer, higher-resolution videos [49, 69, 70, 75, 99] (Fig. 1). Moreover, these methods either rely on heuristics such as attention scores which underperforms learned approaches [72] or involves heavy search and reasoning that adds overhead and further limits scalability [83, 86, 101, 103].

To this end, we propose **AutoGaze**, a 3M-parameter lightweight model that attends to informative patches and removes redundant ones *before* a ViT. Specifically, AutoGaze perceives each frame and autoregressively selects a minimal set of *multi-scale* patches which, along with the selected patches from previous frames, can reconstruct the current frame within a user-specified reconstruction loss threshold. This model, pre-trained with next-token-prediction on a curated dataset of gazing sequences and post-trained with RL on reconstruction rewards, learns to focus only on newly emerged content while ignoring repeated information, and use multi-scale patches to cover broad areas coarsely, zooming in on fine details where needed. For example, Fig. 2 shows AutoGaze removing redundant patches in static regions and selecting coarser scales in low-detail areas. By only processing the selected multi-scale patches, both ViTs and LLMs are substantially sped up, unlocking efficient processing of long, high-resolution videos (Fig. 1).

Empirically, AutoGaze reduces the number of patches by 4×-100× for videos with different FPS and resolution (e.g., 1% patches for 30-FPS 4K-resolution videos) while maintaining downstream MLLM performance. This leads to up to 19× and 10× speedup for ViTs and MLLMs. Leveraging this efficiency, we scale an MLLM (NVILA [53]) to 1K-frame 4K-resolution videos, demonstrating consistent improvements on various benchmark (e.g., 67.0% on VideoMME [30]) and outperforming strong MLLMs such as Qwen2.5-VL [6]. We also show that AutoGaze generalizes to videos with out-of-distribution styles and semantics.

Furthermore, noticing that existing benchmarks only focus on long videos but not high resolution [30, 54, 85, 93, 108], we introduce **HLVid**, the first high-resolution, long-form video QA benchmark, to stress-test AutoGaze's scalability. It consists of 268 QAs about details in up to 5-minute, 4K-resolution videos, requiring visual perception at 1K - 2K

resolution to solve. We show that scaling an MLLM [53] to 1K frames and 4K resolution via AutoGaze significantly improves its performance from 42.5% to 52.6%, outperforming the previous best MLLM [49] by 4.5% (Fig. 1).

2. Related Work

Video understanding and Long-Context MLLMs. Classical video understanding has long been driven by supervised or self-supervised video encoders including 3D-ConvNets and early transformers [3, 12, 27, 28], and pre-training algorithms such as masked auto-encoding [4, 7, 29, 78, 82], predictive coding [35, 62, 80], and large-scale vision-language pre-training [10, 87–89, 95, 96]. Recent MLLMs have extended these encoders to general-purpose video QA and captioning [6, 38, 46, 53, 77, 105]. However, these models usually operate on short, low-resolution clips due to costs of scaling to higher spatiotemporal resolution. While new long-video benchmarks [54, 93, 108] and models [15, 16, 49, 58, 106] emphasize extended temporal understanding, they remain limited to low resolutions due to inefficient whole-video processing, leaving a gap for methods and benchmarks that support *both* thousand-frame context and 4K-resolution detail under realistic compute constraints.
Token Reduction and Compression. A rapidly growing line of work has targeted ViT and MLLM efficiency by reducing input tokens. Spatial methods [9, 11, 48, 57, 63, 72, 98, 104] compress tokens or select informative patches based on attention scores or task relevance. Temporal methods reduce frame redundancy via sub-sampling [81], segment-level pooling [26, 64], or learned keyframe selection [76, 109]; spatiotemporal schemes such as STORM [42], FastVID [69], LongVU [70], and VideoChat-Flash [49], either simply pool tokens or use the ViT features to prune or aggregate tokens. However, all of these models only prune tokens inside the model or between the ViT and LLM, leaving part of the model still processing the full video at high cost. In contrast, AutoGaze removes redundant patches *before* the ViT, significantly improving efficiency. Other works on adaptive tokenization lean where to allocate tokens rather than using a fixed uniform grid [5, 24, 25, 97, 102]. However, their large tokenizer adds additional computational overhead and the tokenization is not adaptive to pre-trained ViTs.

3. AutoGaze for Efficient Video Understanding

Given a video, AutoGaze selects a minimal set of patches (i.e., "gazing") which can reconstruct the video within a reconstruction loss threshold. Formally, for a T-frame video $\boldsymbol{X}^{1:T}$ where $\boldsymbol{X}^t$ is the t-th frame and each frame contains V patches, AutoGaze outputs a set of patch indices:

$$\texttt{AutoGaze} : \boldsymbol{X}^{1:T} \rightarrow p^1_{1:N^1}, \ldots, p^T_{1:N^T}, \tag{1}$$

where $p^t_k \in \{1, \ldots, V\}$ is the index of the k-th patch selected at frame t, and N^t is the number of selected patches (or "gazing length") at frame t.

To select the minimal set satisfying the threshold, AutoGaze is able to select patches that minimize reconstruction loss under *any* $N^{1:T}$ and find the smallest $N^{1:T}$ satisfying the threshold. Formally, given any $N^{1:T}$, AutoGaze can predict patch indices $p^1_{1:N^1}, \ldots, p^T_{1:N^T}$ that optimize

$$\min_{p^1_1, \ldots, p^T_{N^T}} L(\boldsymbol{X}^{1:T}, \texttt{Recon}(\boldsymbol{X}^1[p^1_1], \ldots, \boldsymbol{X}^T[p^T_{N^T}])), \tag{2}$$

where $\boldsymbol{X}^t[p^t_k]$ is the p^t_k-th patch in frame t, $\texttt{Recon}(\cdot)$ is a model that reconstructs the original video from the gazed patches, and $L(\cdot, \cdot)$ is a distance function between the original and the reconstructed videos. We instantiate $\texttt{Recon}(\cdot)$ as a custom VideoMAE [78] with block-causal attention, and $L(\cdot, \cdot)$ as a weighted sum of pixel reconstruction loss and perceptual loss [43, 107] (see Appendix A for details). At the same time, AutoGaze can identify the smallest $N^{1:T}$ that satisfies $L^*(N^{1:T}) < \epsilon$ where $L^*(N^{1:T})$ is the optimal reconstruction loss under gazing lengths $N^{1:T}$ (Eq. 2) and ϵ is a user-specified loss threshold.

To achieve this, we build AutoGaze to **autoregressively select patch indices** that optimize reconstruction loss for any gazing length, while **automatically deciding the smallest gazing length** by predicting reconstruction loss on the fly and stopping once it falls below the threshold. Below, we introduce model design (Sec. 3.1), training pipeline (Sec. 3.2), how to apply it to videos of any duration and resolution, and integrate it into any ViT (Sec. 3.3), and a new benchmark to stress-test scalability (Sec. 3.4).

3.1. Model Design

Fig. 3 (Middle) illustrates AutoGaze's lightweight design: a convolutional encoder and autoregressive transformer decoder, totaling 3M parameters.
Autoregressive gazing. Given a video, AutoGaze interleaves frame encoding and patch gazing. It starts by encoding the first frame with the convolutional encoder, passing the features to the decoder, and autoregressively decoding patch indices. The decoding process mirrors LLMs except the vocabulary contains only patch indices $\{1, \ldots, V\}$ instead of words. Next, AutoGaze encodes the second frame and decodes its patch indices based on *the features of both frames and the gazed patch indices of the first frame*. This lets the model avoid redundant patches by referring to frame and gazing history. The process repeats for subsequent frames.
Automatically deciding the gazing length. To identify the smallest N^t satisfying the reconstruction loss threshold, we add a head on the decoder that, when decoding every p^t_k, predicts the loss of reconstructing frame t from the patches gazed up to that step, i.e., $\{p^1_1, \ldots, p^t_k\}$. Once the predicted loss falls below the threshold, it stops gazing for that frame.
Multi-scale gazing. Considering that not all regions need full resolution (e.g., solid-colored regions can be stored losslessly in low resolution), AutoGaze supports multi-scale

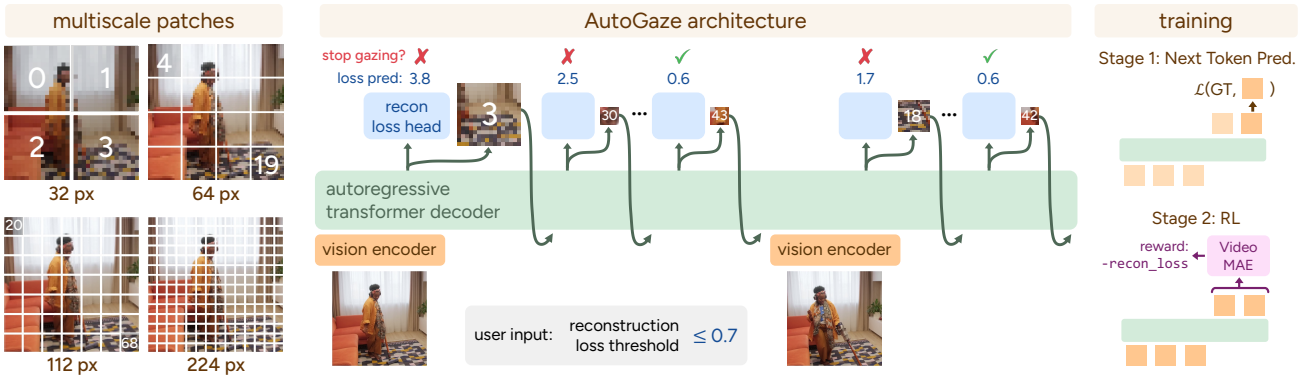

Figure 3. **Architecture and training pipeline of AutoGaze.** *(Left & Middle)* Given a video, AutoGaze processes each frame and autoregressively decodes indices of multi-scale patches based on the history of frames and selected patches. Once it believes the previously-gazed patches are sufficient to reconstruct the current frame, it automatically stops gazing and moves to the next frame. *(Right)* AutoGaze is trained in two stages: next-token-prediction pre-training on collected gazing sequences, and RL post-training with reconstruction reward.

gazing. The decoder's vocabulary includes patches from multiple scales (Fig. 3 (Left)), letting the decoder select different scales for regions with different level of detail, reducing patches while preserving reconstruction quality (Sec. 4.5). This also requires the downstream ViT to accept multi-scale patches as input, which we detail in Sec. 3.3.

Multi-token prediction. We adopt multi-token prediction [32] by using multiple heads to output multiple patch indices and corresponding reconstruction losses at once, speeding up gazing with little performance loss (Sec. 4.5).

3.2. Training Pipeline

AutoGaze is trained to decode patch indices that minimize reconstruction loss at any gazing length and predict reconstruction loss at each step for automatic stopping. Inspired by modern LLM training [1, 34, 56, 60], we train AutoGaze in two stages (Fig. 3 (Right)). First, we pre-train with next-token prediction (NTP) on videos paired with ground-truth gazing sequences that are collected via greedy search to approximately minimize reconstruction loss. Next, since the pre-trained gazing quality is bounded by the sub-optimal gazing data, we further post-train AutoGaze using RL with reconstruction reward to discover gazing sequences with lower reconstruction loss. We also train reconstruction loss prediction in both stages to enable automatic stopping.

Pre-training with next-token-prediction (NTP). Given a dataset with pairs of video $\boldsymbol{X}^{1:T}$, gazing sequences $\{\widetilde{p}_1^1, \ldots, \widetilde{p}_{N^T}^T\}$ that approximately minimize reconstruction loss under random gazing length $N^{1:T}$, and $\{\widetilde{l}_1^1, \ldots, \widetilde{l}_{N^T}^T\}$ where $\widetilde{l}_k^t$ records reconstruction loss of frame t after gazing at $\widetilde{p}_k^t$, we pre-train AutoGaze with NTP cross-entropy loss

$$L_{NTP} = -\sum_{t=1}^{T}\sum_{k=1}^{N^t} \log \pi_\theta(\widetilde{p}_k^t \mid \boldsymbol{X}^{1:t}, \widetilde{p}_{1:N^1}^1, \ldots, \widetilde{p}_{1:k-1}^t), \quad (3)$$

where π_θ is the model and $\pi_\theta(\widetilde{p}_k^t \mid \boldsymbol{X}^{1:t}, \widetilde{p}_{1:N^1}^1, \ldots, \widetilde{p}_{1:k-1}^t)$ is the probability of decoding $\widetilde{p}_k^t$ based on previous frames and gazing. We also supervise reconstruction loss prediction with an ℓ_2 loss using $\{\widetilde{l}_1^1, \ldots, \widetilde{l}_{N^T}^T\}$. AutoGaze thus learns sub-optimal gazing at different gazing length and learns to predict reconstruction loss at each decoding step.

Post-training with RL. Since the pre-training data only contains sub-optimal gazing, we further improve AutoGaze with RL post-training, using a simplified, on-policy GRPO [52, 68] algorithm with reconstruction loss as reward:

$$L_{GRPO} = -\sum_{t=1}^{T}\sum_{k=1}^{N^t} \frac{\pi_\theta(p_k^t)}{\pi_{\theta_{detached}}(p_k^t)} A_k^t, \quad (4)$$

where $\pi_\theta(p_k^t)$ is short for the decoding probability of patch index p_k^t as in Eq. 3, $\pi_{\theta_{detached}}$ is π_θ without gradient, and advantage A_k^t is the return G_k^t normalized within the group of GRPO where $G_k^t = \sum_{\tau=t}^{T} \gamma^{N^t - k + \sum_{s=t+1}^{\tau} N^s} \cdot (-l_{N^\tau}^\tau)$, i.e., sum of negative reconstruction loss of future frames discounted by γ. Additionally, we supervise reconstruction loss prediction at the last patch of each frame (i.e., $l_{N^t}^t$) using the actual reconstruction loss at frame t.

Training data curation. The training pipeline above requires raw videos and paired gazing sequences for pre-training. We first collect a set of 800K videos spanning egocentric, exocentric, natural, and text-rich videos. Each video is sampled at 16 frames and 224 resolution. We then collect gazing sequences that approximately minimize reconstruction loss for 250K videos using greedy search. Specifically, we start from the first patch of the first frame and exhaustively find which patch gives the lowest reconstruction loss. We repeat this until reaching the first frame's gazing length, then proceed to the second frame and so on. We also record reconstruction loss at each step to supervise loss prediction. See Appendix B for details.

3.3. Downstream Usage of AutoGaze

Inference on videos with any resolution and duration. Despite being trained on 16-frame 224×224 videos, Au-

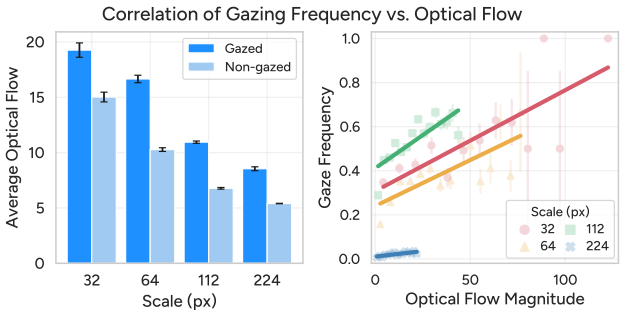

Figure 4. **AutoGaze targets patches with higher optical flow.** *(Left)* AutoGaze uses coarser scales to capture higher optical flow. *(Right)* Across all scales, AutoGaze more frequently selects patches with higher optical flow. Error bars represent SEM.

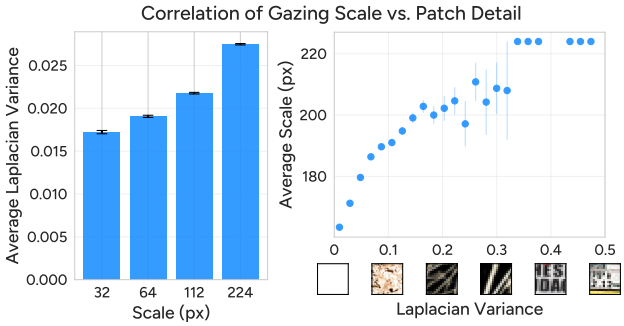

Figure 5. **Gazing scale correlates with patch detail.** *(Left)* At finer scales, AutoGaze selects more detailed patches (measured as Laplacian variance). *(Right)* With increasing detail, AutoGaze uses finer scales ($\rho = .12, p < 0.001$). Sample patches with Laplacian variances are shown below the x-axis. Error bars represent SEM.

toGaze processes videos of any resolution and duration without additional training. Inspired by any-resolution MLLMs [17, 51, 71], we split the video into 16×224×224 spatiotemporal tiles, run AutoGaze on each tile, and merge the gazed positions back together, allowing AutoGaze to scale to 1K-frame and 4K-resolution videos (Sec. 4).

Integrating AutoGaze into ViTs and MLLMs. Current MLLMs typically encode each full frame using an image ViT [6, 53, 84]. To integrate AutoGaze, we make two changes. First, we allow ViTs to take multi-scale patch input by interpolating each frame and positional embeddings to different scales, running patch embedding on each scale separately, and then feeding embedded tokens from all scales to the ViT. Second, we repurpose image ViTs into video ViTs by letting them process tokens from all 16 frames in the same sequence. With these changes, AutoGaze selects multi-scale patches for a video, encodes them with a ViT, and the encoded tokens can be fed into MLLMs as usual.

3.4. HLVid: A High-Res, Long Video Benchmark

Although AutoGaze enables efficient understanding of long, high-resolution videos, benchmarks to evaluate this capability are still missing—current benchmarks [54, 73, 93] only focus on long videos with several minutes of duration but not high resolution. To this end, we propose HLVid, the first long-form, high-resolution video QA benchmark featuring 268 QA pairs on up to 5-minute, 4K-resolution videos. Each question is manually reviewed to ensure high resolution is required. Details are deferred to Appendix C, and some examples from the benchmark are visualized in Fig. 13. We find that an MLLM scaled to 1K frames and 4K resolution via AutoGaze achieves significant improvement and unlocks state-of-the-art performance on HLVid (Sec. 4.3).

4. Experiments

We evaluate AutoGaze 's behavior, efficiency, and performance. Sec. 4.1 examines which patches AutoGaze selects or ignores and tests its generalization to unseen video styles and semantics. Sec. 4.2 measures its efficiency gains for ViTs and MLLMs. Leveraging the efficiency, Sec. 4.3 shows that AutoGaze enables higher-resolution and longer video processing in MLLMs with improved performance. Sec. 4.4 compares AutoGaze against gazing and MLLM token-reduction baselines, and Sec. 4.5 ablates training and modeling choices. We use SigLIP2-SO400M [79] and NVILA-8B-Video [53] as the ViT and MLLM by default.

4.1. What is AutoGaze paying attention to?

AutoGaze's efficiency comes from selecting only a small fraction of patches — but does it make principled decisions about *which* patches to select and at *what* scale? We examine the factors that influence the behavior of AutoGaze and its generalization to videos with unseen styles and semantics.

AutoGaze gazes more at moving patches. Motion is a primary source of new information across video frames, and thus should intuitively be selected (examples are shown in Fig. 2). As illustrated in Fig. 4, AutoGaze does indeed prioritize motion: tested on pairs of videos and flow data from FlyingChairs [22, 39], we find that across all scales, it more frequently selects patches with higher optical flow.

AutoGaze uses finer scales for more detailed patches. Regions with different detailedness should be represented with different scales, as illustrated in Fig. 2. To verify this, we measure the relationship between gazing scale and patch detail by convolving 2,000 ImageNet images [21] with a Laplacian kernel and computing variance over each patch (higher values indicate more detail). Fig. 5 (left) shows that at finer scales, AutoGaze tends to select more detailed patches. Fig. 5 (right) confirms that AutoGaze gazes at higher resolutions to capture fine detail.

AutoGaze generalizes to OOD videos. We test whether AutoGaze transfers beyond its training distribution to unseen semantics and styles, as shown in Fig. 6. First, we show that AutoGaze behavior holds in unconventional scenarios including a CCTV footage, a robot video, and a video [61] that

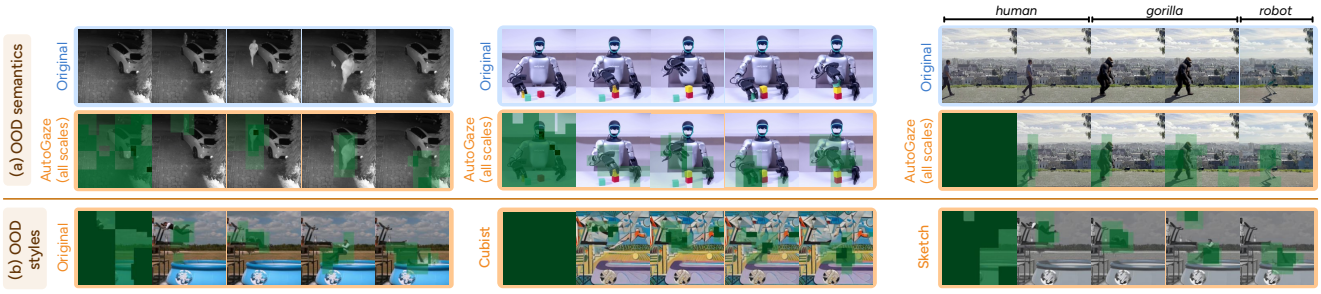

Figure 6. **Generlizability of AutoGaze to OOD videos.** *(a)* We show model behavior on videos with OOD semantics, including a CCTV clip (left), robot grasping demo (middle), and a video with object swapping (right). In each example, AutoGaze still robustly tracks the changing parts despite the unseen semantics, object categories, and unexpected changes. *(b)* We show AutoGaze output on the same video with different style transfer. AutoGaze consistently tracks the falling person regardless of visual style, texture and global illumination.

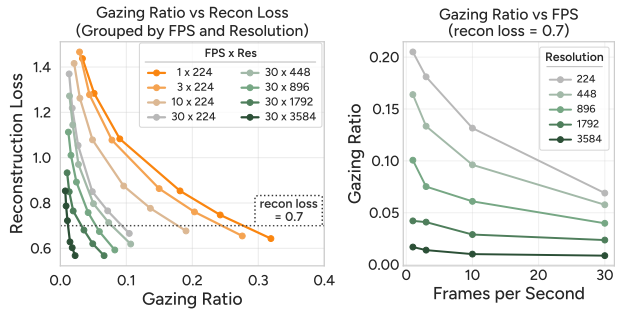

Figure 7. **What gazing ratio do we need for different video types?** *(Left)* There is a trade-off between gazing ratio and reconstruction loss: videos with higher FPS or resolution need lower gazing ratio to reach the same reconstruction quality. *(Right)* Gazing ratios required to reach a loss of 0.7 for videos with different FPS and resolution. 30-FPS, 4K-res videos only need ∼1% patches.

constantly swaps its foreground object between human, gorilla, and humanoid robot (created with Luma's Ray2 Flash). In each example, AutoGaze successfully tracks changing regions despite the novel semantics or unexpected changes. Next, we test on unseen styles by style-transferring a video with TokenFlow [59] to vary texture and global illumination. Across styles, AutoGaze maintains consistent gazing patterns, continuing to track the falling subject.

4.2. Efficiency of ViT and MLLM with AutoGaze

We now study how efficient ViTs and MLLMs can be by selecting fewer patches via AutoGaze. To answer this question, we first analyze the number of patches required to represent a video with AutoGaze, and then benchmark the latency of ViT and MLLM when only the selected patches are processed.

How many patches do we need to represent a video? The number of patches needed depends on both the required reconstruction loss and the level of redundancy (e.g., different FPS and resolution) in the video. We first pinpoint the reconstruction loss that leads to minimal performance drop in downstream MLLMs, and find that a threshold of 0.7 usually leads to less than 0.5% performance degradation across benchmarks (see detailed results in Appendix E). Next, we analyze how many patches are needed to represent videos with varying FPS and resolutions in order to achieve a reconstruction loss of 0.7. Fig. 7 (Left) shows the reconstruction loss for different gazing ratio and different FPS and resolution. We can see the gazing ratio required for a certain loss decreases with higher FPS and resolution. Fig. 7 (Right) shows complete results of gazing ratios required to reach a loss of 0.7 for different videos. Usually a video can be represented with 4×-100× fewer patches. Specifically, only ∼1% patches are needed for 30-FPS, 4K videos.

How much faster are ViTs and MLLMs with AutoGaze? With a target reconstruction loss of 0.7, we analyze the efficiency gains by testing wall-clock ViT and MLLM latency when processing one second of video. We use FP32 and disable flash attention for all models. We report the aggregated latency of AutoGaze and ViT / MLLM, and compare to the baseline without gazing in Fig. 8. The ViT baseline quickly runs out of memory around 30 FPS and 896 resolution, and the MLLM baseline can only encode 30 FPS and 224 resolution. In contrast, AutoGaze helps efficiently process videos with lower gazing ratios. When using the gazing ratio required for a reconstruction loss of 0.7, it achieves up to 19× and 10× speedup for ViTs and MLLMs respectively, enabling scaling to 4K resolution.

4.3. Scaling MLLMs with AutoGaze

Leveraging AutoGaze's efficiency, we scale MLLMs to longer, higher-resolutions videos and achieve state-of-the-art performance on video benchmarks.

Scaling properties. We compare performance and efficiency when scaling MLLMs at test time to longer and high-resolution videos with or without AutoGaze, and report results in Fig. 9. We first scale the number of frames, identify the best frame count for each benchmark, then scale resolution. Starting from 64 frames and 448 resolution, MLLM with AutoGaze has slightly worse performance than the baseline while using ∼4× fewer tokens. This performance drop vanishes after scaling to 256 frames. When further scaling

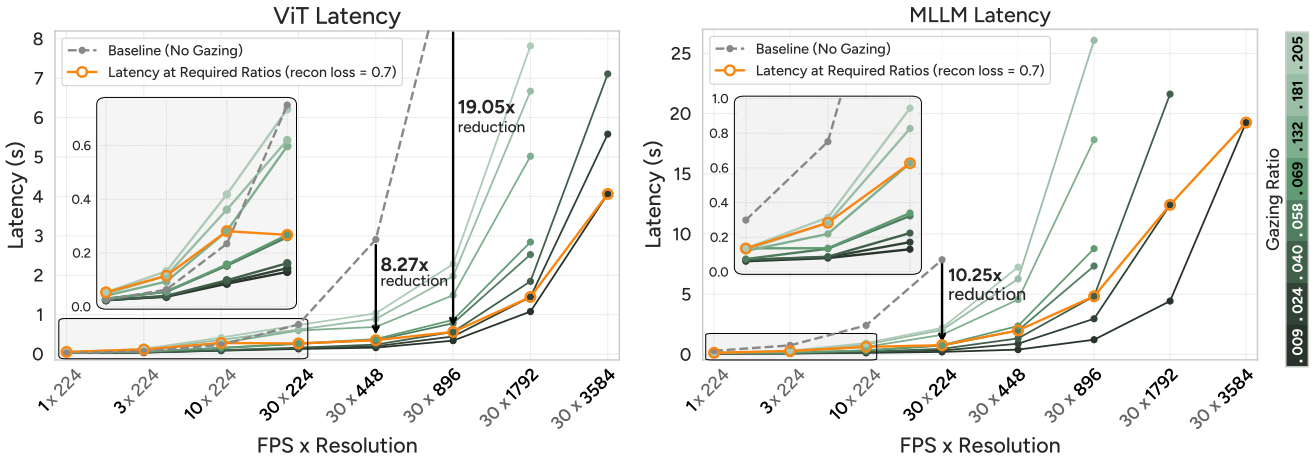

Figure 8. **Efficiency gain on ViTs and MLLMs with AutoGaze.** We benchmark the ViT and MLLM latency of encoding one second of video with varying FPS and resolution. AutoGaze can select different numbers of patches to vary latency depending on user needs. When using the gazing ratio required for a reconstruction loss of 0.7, AutoGaze reduces the ViT and MLLM latency by up to 19× and 10×.

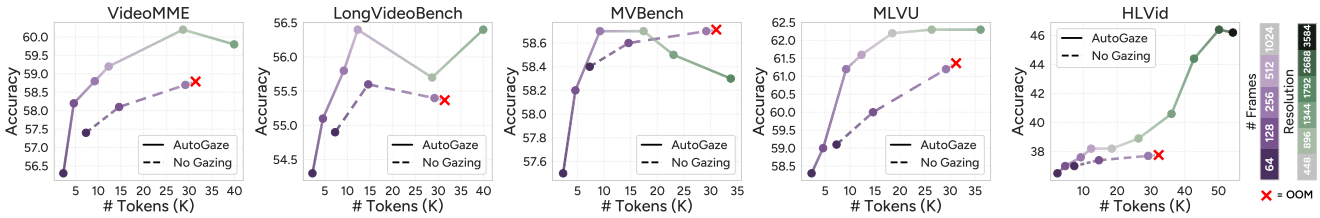

Figure 9. **Scaling MLLMs to more video tokens.** AutoGaze enables scaling to longer videos and higher resolution, while the baseline runs out of memory beyond 256 frames. Performance boosts are especially clear with HLVid, where high-resolution video processing is required.

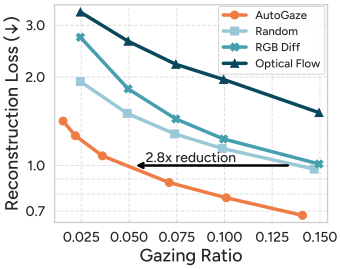

Figure 10. **Comparison to baseline gazing methods.** AutoGaze can use smaller gazing ratios to reach the same reconstruction loss, compared to other heuristics-based gazing approaches.

video duration and resolution, the baseline runs out of memory while AutoGaze enables scaling to 1K frames and 4K resolution with consistent improvements. Note that on some benchmarks, using too long or too high-resolution videos is detrimental, likely because those benchmarks require neither, while scaling to 4K resolution significantly improves performance on HLVid, verifying it does require high resolution.

Comparing to state-of-the-art MLLMs. We train NVILA-8B-Video [53] with AutoGaze at 256 frames and 896 resolution, scale up to 1K frames and 4K resolution at test time, and compare to existing MLLMs in Tab. 1. Using AutoGaze, it consistently improves over the base NVILA-8B-Video on all benchmarks. Notably, performance on HLVid improves by 10.1%, surpassing all open-source MLLMs like Qwen2.5-VL-7B [6] and proprietary models like GPT-4o [38] despite much smaller model size or training dataset. NVILA-8B-Video with AutoGaze also outperforms others on VideoMME, though falls short on MVBench and LongVideoBench likely due to differing training recipes.

4.4. Comparing to Token Pruning Baselines

Comparing to baseline gazing approaches. In Fig. 10, we compare AutoGaze to heuristic-based gazing approaches that select patches randomly (**Random Gaze**), with the largest RGB difference (**RGB-Diff Gaze**) or with the largest optical flow using SEA-RAFT [90] (**Optical-Flow Gaze**). We left-pad a blank image when computing differences for the first frame. AutoGaze greatly improves efficiency — for example, reaching reconstruction loss 1.0 with 5% patches versus 15% for Random Gaze. We find RGB-Diff Gaze and Optical-Flow Gaze to be worse than Random Gaze because they fixate on the first frame (an abrupt change from the padding) and ignore other frames, causing information loss.

Comparing to MLLM token reduction baselines. We compare AutoGaze to existing MLLM token reduction methods [9, 14, 37, 42, 49, 50, 69, 70, 75, 98] and three simple baselines: spatial, temporal, and spatiotemporal pooling (S-Pool, T-Pool, and ST-Pool). These approaches process the whole video in the ViT, then reduce visual tokens in the

Table 1. **Comparison to state-of-the-art MLLMs.** NVILA-8B-Video with AutoGaze is scaled to 1K-frame, 4K-resolution videos, achieving competitive performance on general and long video benchmarks and state-of-the-art result on HLVid.

Models	Max #F	Max Res.	*general video* VideoMME (w/o sub)	VideoMME (w/ sub)	MVBench (test)	NExT-QA (mc)	*long video* L-VidBench (val)	EgoSchema (test)	MLVU (m-avg)	*high-res & long* HLVid (test)
Gemini 1.5-Pro [77]	-	-	75.0	81.3	60.5	-	64.0	71.2	-	-
Gemini 2.5 Flash-Lite [18]	-	-	65.0	-	-	-	-	-	69.3	52.2
GPT-4o [38]	-	-	71.9	77.2	64.6	-	66.7	72.2	64.6	49.3
LLaVA-OV-8B [46]	32	384	58.2	61.5	56.7	79.4	56.5	60.1	64.7	41.1
LongVILA-7B [15]	2048	384	60.1	65.1	67.1	80.7	57.1	-	-	41.4
LongVILA-R1-7B [16]	8192	448	65.1	71.1	-	81.5	58.0	-	-	42.2
Apollo-7B [110]	2FPS	384	61.3	63.3	-	-	58.5	-	70.9	-
VideoLLaMA3-7B [105]	180	384	66.2	70.3	69.7	**84.5**	59.8	63.3	73.0	38.8
VideoChat-Flash [49]	10000	448	65.3	69.7	**74.0**	-	**64.7**	-	**74.7**	46.6
InternVL3.5-8B [84]	64	448	66.0	68.6	72.1	-	62.1	-	70.2	39.9
Qwen2.5-VL-7B [6]	48	896	65.1	71.6	69.6	-	56.0	65.0	70.2	48.1
NVILA-8B-Video [53]	256	448	64.2	70.0	68.1	82.2	57.7	-	70.1	42.5
+ AutoGaze	1024	3584	**67.0**	**71.8**	69.7	82.8	61.0	**66.9**	71.6	**52.6**
(vs. NVILA-8B-Video)	*(×4)*	*(×8)*	*(+2.8)*	*(+1.8)*	*(+1.6)*	*(+0.6)*	*(+3.3)*	*-*	*(+1.5)*	*(+10.1)*

Table 2. **Comparing AutoGaze with MLLM token reduction methods.** We compare to baselines of spatial/temporal/spatiotemporal (S-/T-/ST-) token reduction approaches that are either prompt-agnostic (PA) or prompt-dependent (PD). All the methods select 6.25% visual tokens on average.

Type	Method	ViT lat.	LLM lat.	V-MME (w/o sub)	L-Vid (val)
-	No Reduction	2.20s	1.42s	53.4	51.1
S-PA	S-Pool	2.20s	0.18s	51.5	47.2
	ToMe [9]	2.23s	0.11s	51.5	49.3
	VisionZip [98]	2.22s	0.15s	50.7	48.5
S-PD	FastV [14]	2.23s	0.38s	**53.0**	46.3
T-PA	T-Pool	2.20s	0.13s	52.2	50.0
T-PD	AKS [75]	3.27s	0.12s	50.8	49.5
ST-PA	ST-Pool	2.19s	0.13s	52.0	49.8
	STORM [42]	2.18s	0.15s	52.7	**51.5**
	FastVID [69]	2.34s	0.12s	52.4	50.3
	F-16 [50]	2.20s	0.18s	51.8	50.0
ST-PD	LongVU [70]	2.17s	0.12s	52.2	50.1
	PruneVID [37]	2.52s	0.15s	50.3	48.0
	VChat-Flash [49]	2.21s	0.15s	52.4	49.9
ST-PA	AutoGaze	**0.55s**	**0.10s**	52.3	50.3

LLM. We use 128-frame videos and 6.25% selection ratio and report results in Tab. 2. The baseline without token reduction has high ViT and LLM latency, with ViT latency slightly higher than LLM latency despite its smaller size due to LLMs' token shuffling [6, 53]. While baseline methods improve LLM latency by 3.7×-13.4×, the ViT latency remains unchanged. In contrast, AutoGaze significantly improves ViT latency by 4× in addition to the LLM latency improvement. Beyond speedups, both the token-reduction baselines and AutoGaze retain performance comparable to the no-reduction baseline.

Table 3. **Ablation on AutoGaze training pipeline.** Both NTP pre-training and RL post-training helps with the performance.

Pre-Train	Post-Train	Recon Loss	Gazing Ratio
✗	✗	0.7	0.263
✓	✗	0.7	0.102
✗	✓	0.7	0.209
✓	✓	0.7	**0.094**

Table 4. **Ablations of AutoGaze model designs.**

Multi-Token Pred.	Multi-Scale Gazing	Recon Loss	Gazing Ratio	Latency
1	✓	0.7	0.074	0.949s
5	✓	0.7	0.078	0.246s
10	✓	0.7	**0.094**	**0.193s**
20	✓	0.7	0.109	0.156s
10	✓	0.7	**0.094**	**0.193s**
10	✗	0.7	0.220	0.467s

4.5. Ablations

Training pipeline of AutoGaze. We analyze the effect of the two-stage training pipeline by comparing the gazing ratio required to reach reconstruction loss 0.7 when removing different training stages. As shown in Tab. 3, with only pre-training or post-training, efficiency improves over no training, while pre-training contributing more. With both stages, we obtain the lowest gazing ratio, achieving ∼10% improvement over pre-training only.

Model designs of AutoGaze. We analyze the effect of multi-token prediction and multi-scale gazing on the model performance and efficiency (Tab. 4). We report the gazing ratio required to reach reconstruction loss 0.7 and the corresponding latency when processing a 1-second video at 10 FPS and 224 resolution. We find decoding more tokens at

a time leads to lower latency but higher gazing ratio, with decoding 10 tokens balancing both factors well. On the other hand, multi-scale gazing reduces the gazing ratio and improves the efficiency by 2.3×.

5. Conclusion

We introduce AutoGaze, a lightweight framework that removes redundant video patches to improve ViT and MLLM efficiency. Trained via NTP on gazing sequences collected through a greedy algorithm and RL with reconstruction reward, AutoGaze learns to select a minimal set of multi-scale patches that reconstructs the video within a user-specified threshold. Empirically, AutoGaze reduces visual tokens by 4×–100× and accelerates ViTs and MLLMs by up to 19× and 10×, enabling 1024-frame 4K-resolution video understanding and improving performance on video benchmarks. We further introduce HLVid, the first long-form (5-minute), high-resolution (4K) video QA benchmark, where an MLLM with AutoGaze outperforms previous SOTA model by 4.5%.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] John Robert Anderson and Jane Crawford. Cognitive psychology and its implications. 1995.

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[4] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[5] Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025.

[6] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[7] 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. *arXiv preprint arXiv:2404.08471*, 2024.

[8] Ali Furkan Biten, Ruben Tito, Lluis Gomez, Ernest Valveny, and Dimosthenis Karatzas. Ocr-idl: Ocr annotations for industry document library dataset. In *European Conference on Computer Vision*, pages 241–252. Springer, 2022.

[9] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

[10] Daniel Bolya, Po-Yao Huang, Peize Sun, Jang Hyun Cho, Andrea Madotto, Chen Wei, Tengyu Ma, Jiale Zhi, Jathushan Rajasegaran, Hanoona Rasheed, et al. Perception encoder: The best visual embeddings are not at the output of the network. *arXiv preprint arXiv:2504.13181*, 2025.

[11] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. PuMer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 12890–12903, 2023.

[12] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[13] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021.

[14] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[15] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024.

[16] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025.

[17] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024.

[18] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[19] Kellie Corona, Katie Osterdahl, Roderic Collins, and Anthony Hoogs. Meva: A large-scale multiview, multimodal video dataset for activity detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1060–1068, 2021.

[20] Adrien Deliege, Anthony Cioppa, Silvio Giancola, Meisam J Seikavandi, Jacob V Dueholm, Kamal Nasrollahi, Bernard

Ghanem, Thomas B Moeslund, and Marc Van Droogenbroeck. Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4508–4519, 2021.

[21] 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, 2009.

[22] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[23] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[24] Shivam Duggal, Phillip Isola, Antonio Torralba, and William T. Freeman. Adaptive length image tokenization via recurrent allocation. In *Scalable Optimization for Efficient and Adaptive Foundation Models (NeurIPS Workshop)*, 2024.

[25] Shivam Duggal, Sanghyun Byun, William T Freeman, Antonio Torralba, and Phillip Isola. Single-pass adaptive image tokenization for minimum program search. *arXiv preprint arXiv:2507.07995*, 2025.

[26] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6824–6835, 2021.

[27] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[28] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3299–3309, 2021.

[29] Christoph Feichtenhofer, Yanghao Li, Kaiming He, et al. Masked autoencoders as spatiotemporal learners. *Advances in neural information processing systems*, 35:35946–35958, 2022.

[30] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025.

[31] Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. Magicdrive-v2: High-resolution long video generation for autonomous driving with adaptive control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28135–28144, 2025.

[32] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024.

[33] 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.

[34] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[35] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019.

[36] 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*, pages 16000–16009, 2022.

[37] Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. *arXiv preprint arXiv:2412.16117*, 2024.

[38] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[39] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[40] Laurent Itti and Christof Koch. Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203, 2001.

[41] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 2002.

[42] Jindong Jiang, Xiuyu Li, Zhijian Liu, Muyang Li, Guo Chen, Zhiqi Li, De-An Huang, Guilin Liu, Zhiding Yu, Kurt Keutzer, et al. Storm: Token-efficient long video understanding for multimodal llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5830–5841, 2025.

[43] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[44] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[45] Didier Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34 (4):46–58, 1991.

[46] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[47] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[48] Ling Li, David Thorsley, and Joseph Hassoun. SaiT: Sparse vision transformers through adaptive token pruning. *CoRR*, abs/2210.05832, 2022.

[49] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024.

[50] Yixuan Li, Changli Tang, Jimin Zhuang, Yudong Yang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. Improving llm video understanding with 16 frames per second. *arXiv preprint arXiv:2503.13956*, 2025.

[51] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024.

[52] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[53] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4122–4134, 2025.

[54] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244, 2023.

[55] 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.

[56] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[57] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogério Feris, and Aude Oliva. IA-RED^2: Interpretability-aware redundancy reduction for vision transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 24898–24911, 2021.

[58] Ziqi Pang and Yu-Xiong Wang. Mr. video:" mapreduce" is the principle for long video understanding. *arXiv preprint arXiv:2504.16082*, 2025.

[59] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation, 2025.

[60] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[61] Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. *Advances in neural information processing systems*, 37:79307–79324, 2024.

[62] Jathushan Rajasegaran, Ilija Radosavovic, Rahul Ravishankar, Yossi Gandelsman, Christoph Feichtenhofer, and Jitendra Malik. An empirical study of autoregressive pre-training from videos. *arXiv preprint arXiv:2501.05453*, 2025.

[63] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13937–13949, 2021.

[64] Shuhuai Ren, Sishuo Chen, Shicheng Li, Xu Sun, and Lu Hou. TESTA: Temporal-spatial token aggregation for long-form video-language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 932–947, 2023.

[65] Weiming Ren, Huan Yang, Jie Min, Cong Wei, and Wenhu Chen. Vista: Enhancing long-duration and high-resolution video understanding by video spatiotemporal augmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3804–3814, 2025.

[66] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

[67] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020.

[68] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[69] Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models. *arXiv preprint arXiv:2503.11187*, 2025.

[70] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

[71] Baifeng Shi, Ziyang Wu, Maolin Mao, Xin Wang, and Trevor Darrell. When do we not need larger vision models? In *European Conference on Computer Vision*, pages 444–462. Springer, 2024.

[72] Baifeng Shi, Boyi Li, Han Cai, Yao Lu, Sifei Liu, Marco Pavone, Jan Kautz, Song Han, Trevor Darrell, Pavlo Molchanov, et al. Scaling vision pre-training to 4k resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9631–9640, 2025.

[73] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024.

[74] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.

[75] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 29118–29128, 2025.

[76] Xi Tang, Jihao Qiu, Lingxi Xie, Yunjie Tian, Jianbin Jiao, and Qixiang Ye. Adaptive keyframe sampling for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29118–29127, 2025.

[77] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[78] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems*, 35:10078–10093, 2022.

[79] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[80] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016.

[81] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36, 2016.

[82] 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*, pages 14549–14560, 2023.

[83] Shihao Wang, Guo Chen, De-an Huang, Zhiqi Li, Minghan Li, Guilin Li, Jose M Alvarez, Lei Zhang, and Zhiding Yu. Videoitg: Multimodal video understanding with instructed temporal grounding. *arXiv preprint arXiv:2507.13353*, 2025.

[84] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[85] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. Lvbench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025.

[86] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024.

[87] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[88] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023.

[89] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *European Conference on Computer Vision*, pages 396–416. Springer, 2024.

[90] Yihan Wang, Lahav Lipson, and Jia Deng. Sea-raft: Simple, efficient, accurate raft for optical flow. In *European Conference on Computer Vision*, pages 36–54. Springer, 2024.

[91] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[92] Chao-Yuan Wu, Nayan Singhal, and Philipp Krahenbuhl. Video compression through image interpolation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 416–431, 2018.

[93] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857, 2024.

[94] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786, 2021.

[95] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training

for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021.

[96] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022.

[97] Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. Elastictok: Adaptive tokenization for image and video. *arXiv preprint arXiv:2410.08368*, 2024.

[98] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802, 2025.

[99] Hanrong Ye, Haotian Zhang, Erik Daxberger, Lin Chen, Zongyu Lin, Yanghao Li, Bowen Zhang, Haoxuan You, Dan Xu, Zhe Gan, et al. Mm-ego: Towards building egocentric multimodal llms for video qa. *arXiv preprint arXiv:2410.07177*, 2024.

[100] Hanrong Ye, Chao-Han Huck Yang, Arushi Goel, Wei Huang, Ligeng Zhu, Yuanhang Su, Sean Lin, An-Chieh Cheng, Zhen Wan, Jinchuan Tian, et al. Omnivinci: Enhancing architecture and data for omni-modal understanding llm. *arXiv preprint arXiv:2510.15870*, 2025.

[101] Jinhui Ye, Zihan Wang, Haosen Sun, Keshigeyan Chandrasegaran, Zane Durante, Cristobal Eyzaguirre, Yonatan Bisk, Juan Carlos Niebles, Ehsan Adeli, Li Fei-Fei, et al. Rethinking temporal search for long-form video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8579–8591, 2025.

[102] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

[103] Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-voyager: Learning to query frames for video large language models. *arXiv preprint arXiv:2410.03226*, 2024.

[104] Xin Yuan, Hongliang Fei, and Jinoo Baek. Efficient transformer adaptation with soft token merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3658–3668, 2024.

[105] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[106] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024.

[107] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[108] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Shitao Xiao, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: A comprehensive benchmark for multi-task long video understanding. *arXiv e-prints*, pages arXiv–2406, 2024.

[109] Zirui Zhu, Hailun Xu, Yang Luo, Yong Liu, Kanchan Sarkar, Zhenheng Yang, and Yang You. FOCUS: Efficient keyframe selection for long video understanding. *CoRR*, abs/2510.27280, 2025.

[110] Orr Zohar, Xiaohan Wang, Yann Dubois, Nikhil Mehta, Tong Xiao, Philippe Hansen-Estruch, Licheng Yu, Xiaofang Wang, Felix Juefei-Xu, Ning Zhang, et al. Apollo: An exploration of video understanding in large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18891–18901, 2025.

A. Additional Details of AutoGaze Model Design

Model architecture. AutoGaze contains a convolutional vision encoder, a visual connector, and a transformer decoder. The convolutional encoder contains one 2D convolutional layer with spatial kernel size of 16 to embed each patch, and one 3D convolutional layer with spatial and temporal kernel sizes of 3 to extract the spatiotemporal vision features of each frame based on the current frame and previous two frames. Note that the vision encoder is causal. The visual connector bridges between the vision encoder output and the transformer decoder input, adding positional embeddings to the output visual features from the vision encoder and passing them to the decoder. The positional embeddings are added in each frame separately, such that each token in each frame is aware of its spatial position in the frame. The transformer decoder uses the same architecture design as LLaMA 3 [23] but with only four layers. The decoder takes in each frame's visual tokens and decodes the patch indices. It also predicts the reconstruction loss of the current frame at each step using a linear decoding head. The vocabulary of the decoder only contains all the possible patch indices in a frame. We use four scales, i.e., 32×32, 64×64, 112×112, and 224×224. Since the patch size is 16, the number of all possible patches (i.e., the vocabulary size of the decoder) is 4 + 16 + 49 + 196 = 265. The hidden dimension of the whole model is 192.

Instantiation of the reconstruction objective. AutoGaze is trained to select as few patches as possible while keeping a certain level of reconstruction loss (Eq. 2). The reconstruction objective includes a distance function $L(\cdot, \cdot)$ and a video reconstruction model $\texttt{Recon}(\cdot)$. We obtain the reconstruction model by taking an MAE [36] pre-trained on images and fine-tuning it on videos with the same masked auto-encoding objective such that it can reconstruct a video from partially observed patches. The resulting model is similar to VideoMAE [78] except that the self-attention layers are block-causal, i.e., the model reconstructs each frame based on only the current and previous frames. This is important because the gazing model is also causal and we should not train it to optimize a reconstruction loss that depends on the future. This also allows us to calculate the reconstruction loss of each frame based on previous gazed patches up until any step at that frame, which we use to supervise the reconstruction loss prediction at each step. For the distance function, we use a combination of pixel reconstruction loss and perceptual loss [43, 107], i.e., a weighted sum of ℓ_1 loss in the pixel space and ℓ_2 loss on the frame-wise DINOv2 [55] and SigLIP2 [79] embeddings between the reconstructed and the original video. The weights for ℓ_1 loss, DINOv2 embedding loss, and SigLIP2 embedding loss are 1, 0.3, and 0.3 respectively.

B. Additional Details of AutoGaze Training Pipeline

NTP pre-training. We pre-train AutoGaze on about 250K videos with paired gazing sequences. We train for 150 epochs, with a batch size of 256 and a learning rate of 5e-4.

RL post-training. We use GRPO algorithm that is a simplified and on-policy version of the original version, i.e., we remove the KL regularization term and use the same policy at each step as the old policy in the original algorithm. We use a group size of 12. When calculating the advantage for each gazing step, we count in the negative reconstruction loss for the current frame as well as the subsequent frames. The reconstruction loss for each frame is discounted based on the number of gazing steps between the current gazing and the last gazing of that frame. The discounting factor we use is 0.995. We also follow Dr. GRPO to remove the std normalization and the sequence length normalizer in GRPO. During RL training, we anneal the temperature when rolling out the gazing prediction from 1 to 0.01. At each step of training, instead of running VideoMAE on all the frame to get the reconstruction reward for every frame, we instead just randomly sample 2 frames to reconstruct and only use the reconstruction rewards for those two frames for RL training in order to improve training efficiency. When rolling out gazing prediction during training, we set a reconstruction loss threshold of 0.7, let AutoGaze to predict one gazing sequence based on the threshold, record the number of gazing for each frame, and then roll out a group of gazing sequences with the same number of gazing for each frame for GRPO loss. This ensures every gazing sequence in the same GRPO group has the same number of gazing for each frame, such that the reconstruction rewards between sequences can be fairly compared. We train for 3 epochs, with a batch size of 256 and a learning rate of 5e-4.

Training data. For the training data, we first collect videos from datasets including Ego4D [33], 100DoH [67], and InternVid [88], covering both exocentric and egocentric natural videos. We also create artificial videos that simulates camera motions by placing a window on an large image, slide the window to random directions, and take the content within the sliding window as a video. We create videos like this from high-resolution image datasets including SA-1B [44] and IDL [8], covering both natural and text-rich videos. In total, we end up with collecting ∼800K videos. For pre-training, we collect gazing sequences for a subset of ∼250K videos. Specifically, for each video, we first randomly sample the gazing ratios for each frame, and then search the best gazing sequence that optimizes the reconstruction loss under the sampled gazing ratio, using the greedy search algorithm described in Sec. 3.2. When sampling the gazing ratio, we first randomly sample the average gazing ratio of the whole video from an exponential distribution between 0.02 and 0.2.

Then, given the average gazing ratio of all the frames, we sample the gazing ratio for each frame using a Dirichlet distribution with alpha of 10 for the first frame and 3 for the rest of the frames, such that the gazing ratio is biased towards the first frame. This is to simulate the real distribution of gazing where the first frame is usually gazed at more since it contains completely new information without any history context.

C. Additional Details of HLVid Benchmark

We collect 268 QA on videos of autonomous driving and household scenarios scraped from YouTube. Each video has 4K resolution and a large portion of them have 5 minutes of duration. We design the QAs such that every question needs high-resolution perception at at least 1K - 2K resolution to solve. We also review the QAs such that questions from the same video are asking about different details, and the answer to each question is not ambiguous, i.e., there is only one correct answer and there is no other correct answers appearing in other frames in the video. As as comparison, existing benchmarks mostly only focus of long videos but not high resolution. For example, LongVideoBench [93] and EgoSchema [54] contain videos with an average duration of 473s and 180s separately, but their questions mostly only need low resolution to solve. Our benchmark, instead, requires both long and high-resolution video understanding to solve. See Figure 13 for examples of HLVid's videos and questions.

D. Additional Details of Analyses

Section 4.1 explores what AutoGaze pays attention to in videos. Below, we provide further implementation details for the optical flow and patch detail analyses.

Optical flow. To analyze whether AutoGaze selects moving patches more often than static patches, we use the image pairs and optical flow map in the Flying Chairs dataset [22]. Since the data only contains forward optical flow, it only assigns object motion to the pixels of the first frame, not the second frame. Therefore, we compute backward flow by finding each source pixel's resulting location and assigning the inverse optical flow there as well (occluded/invalid values are set to zero). The final flow map that we use is a pixel-wise maximum over the forward and backward optical flows. Given this map, we crop all possible patches at each gazing scale and compute the maximum optical flow in that patch.

Patch detail. We compute patch detail for Section 4.1 using the variance of the Laplacian for each image. Specifically, we compute the patch "detailedness" by convolving each video frame with a 3x3 Laplace filter

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

and computing the variance of the values in each possible patch. Lower variance corresponds to smoother/more constant coloring, while high variance corresponds to busier textures (e.g., stripes, text).

E. Additional Quantitative Results

How large of a reconstruction loss is tolerable for downstream video understanding? In Sec. 4.2, we benchmark the efficiency of ViTs and MLLMs with AutoGaze. However, since the efficiency depends on how many patches to select for each video, which further depends on the reconstruction threshold, we first need to pinpoint the appropriate reconstruction loss threshold, i.e., the reconstruction loss threshold that leads to little or no downstream performance drop. To do this, we set different reconstruction loss thresholds and compare the MLLM performance. Tab. 5 shows results averaged over four benchmarks [30, 47, 93, 94]. Compared to the baseline without gazing, reconstruction loss below 0.7 leads to performance drop consistently less than 0.5% across videos with different numbers of frames. This is supported by reconstruction visualization under different losses (Fig. 11), where a loss > 0.7 usually results in visible artifacts. Balancing performance and efficiency, we choose 0.7 as the threshold for MLLM experiments.

Table 5. **What reconstruction loss is tolerable for MLLM performance?** We find a loss of 0.7 has little performance drop.

Recon. Loss	64 Frames	128 Frames	256 Frames
No Gaze	59.1	60.1	60.5
0.6	59.0	60.0	60.7
0.7	58.6	59.7	60.3
0.8	57.8	58.4	59.0
1.0	56.3	56.7	57.2

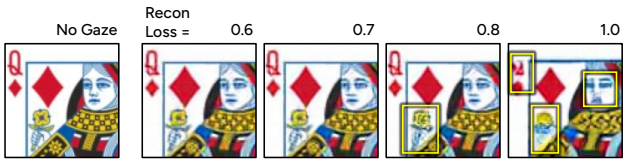

Figure 11. **Reconstruction quality under varying loss thresholds.** Outlined in `recon_loss`$= 0.8, 1.0$ are noticeable visual defects.

Efficiency gain of ViTs and MLLMs on streaming videos. In Sec. 4.2, we benchmark the ViT and MLLM efficiency with AutoGaze. However, the benchmarking is conducted on static videos, i.e., the models can see the full video beforehand such that it can split each video into multiple clips and process in a batchified way. However, lots of real-world applications calls for streaming video understanding, i.e., the ViT and MLLM needs to process frames one by one. To

this end, we also benchmark the ViT and MLLM efficiency under this situation, where they process frames one by one and we measure the maximum FPS they can process frames at. Results are shown in Fig. 12. We test on videos with different FPS and resolution, and compare the maximum FPS of ViT/MLLM with or without AutoGaze. We can see that ViTs and MLLMs with AutoGaze usually achieve an FPS that is up to $\sim$16$\times$ higher. This enables real-time processing for ViTs and MLLMs which is infeasible without AutoGaze (e.g., real-time ViT encoding of 10FPS videos at resolution higher than 500, and real-time MLLM processing of 3 FPS videos at 1K resolution).

F. Additional Qualitative Results

Figs 14 - 27 showcase examples of AutoGaze applied to various video domains and OOD use cases, including lectures, sports live stream, cartoons, film clips, picture-in-picture videos, fisheye lens security footage, warehouse surveillance camera, nighttime driving, black-and-white movies, robot arm demonstrations, and split-view videos. Finally, we provide another example of AutoGaze continuing to track moving objects even when an object is swapped in the middle of the video (Fig. 28). Refer to each figure caption for particular capabilities demonstrated in that example.

G. Limitations

We identify two main limitations with AutoGaze. First, it does not account for most camera motion; as a scene pans in some direction, it will still subsample patches but not necessarily ignore patches that are redundant up to a shift. See Fig. 29 for an example of this limitation. Second, our model cannot anticipate future frames according to knowledge of physics; though our VideoMAE is causal, it is not trained to have "intuitive physics" knowledge (e.g., knowledge that a falling ball will keep falling in the next frame). We illustrate this limitation in Fig. 30 by providing several full video frames of a ball in free fall, and visualizing the VideoMAE's reconstruction of subsequent frames.

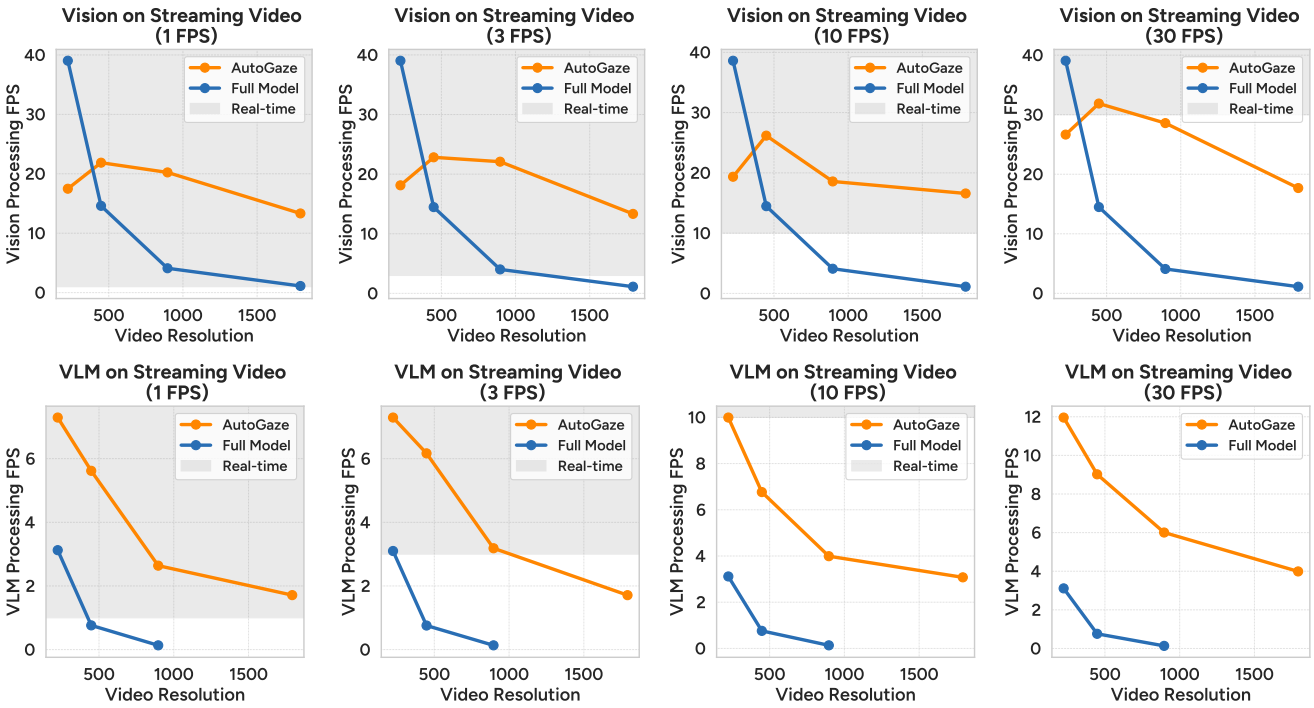

Figure 12. **Efficiency on streaming videos.** We compare the ViT and MLLM efficiency on streaming videos with or without AutoGaze. We measure the maximum FPS of processing different types of videos (i.e., different FPS and resolution). For each plot, the gray area denotes the region where real-time processing of the video is achieved.

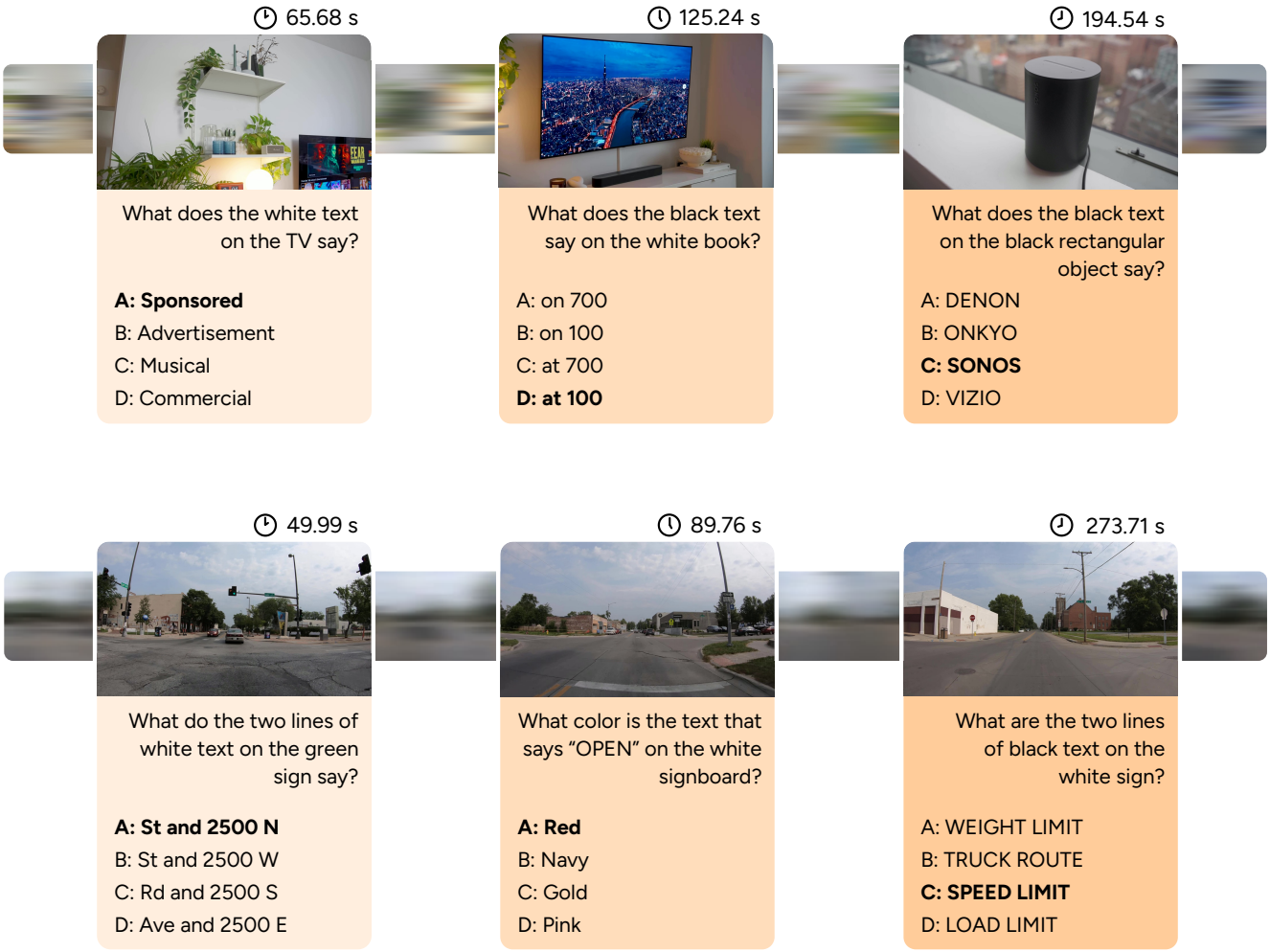

Figure 13. **Examples of HLVid benchmark.** Each sample from HLVid consists of a 5-minute long video at 4K resolution, along with multiple-choice questions that require high-resolution video encoding to answer. The answers to HLVid's questions can be found at any spatiotemporal location, emphasizing long-context video understanding. Our video content is diverse and includes house tours (top), city driving (bottom), nature videos, etc.

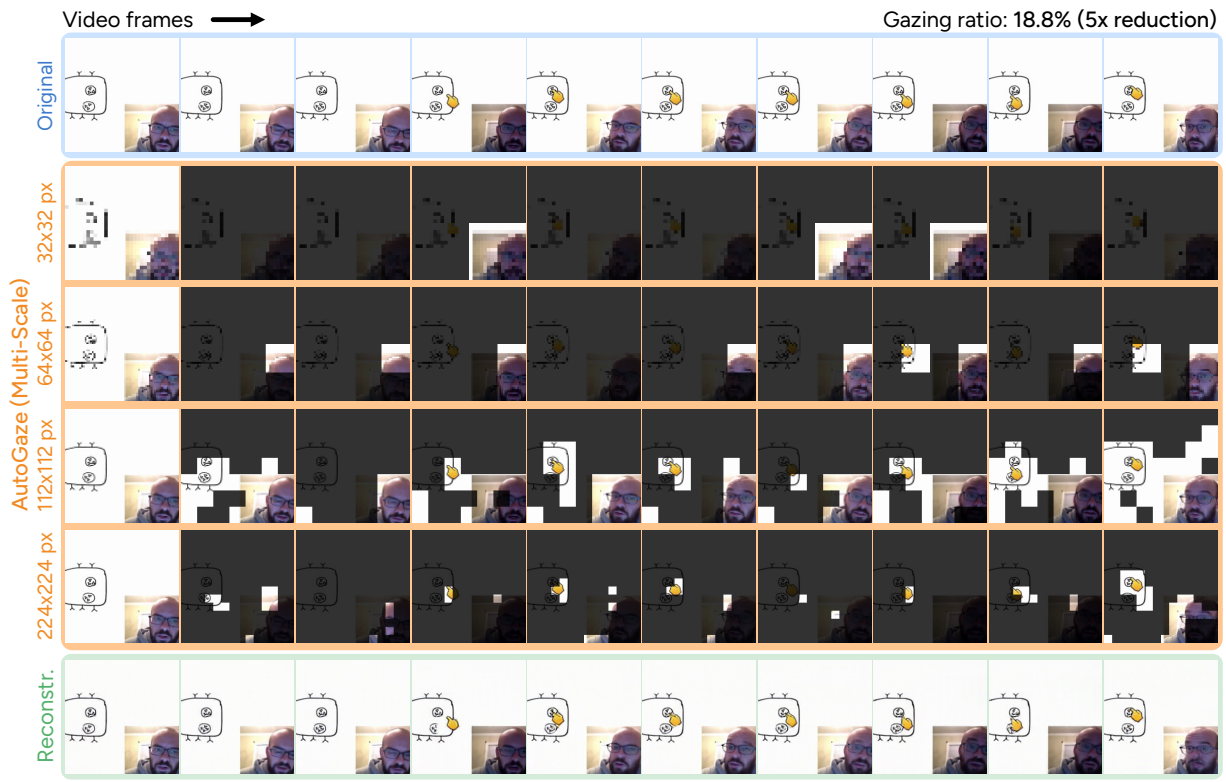

Figure 14. **Picture-in-picture whiteboard lecture.** After the first frame, AutoGaze focuses on the moving cursor and the lecturer's face.

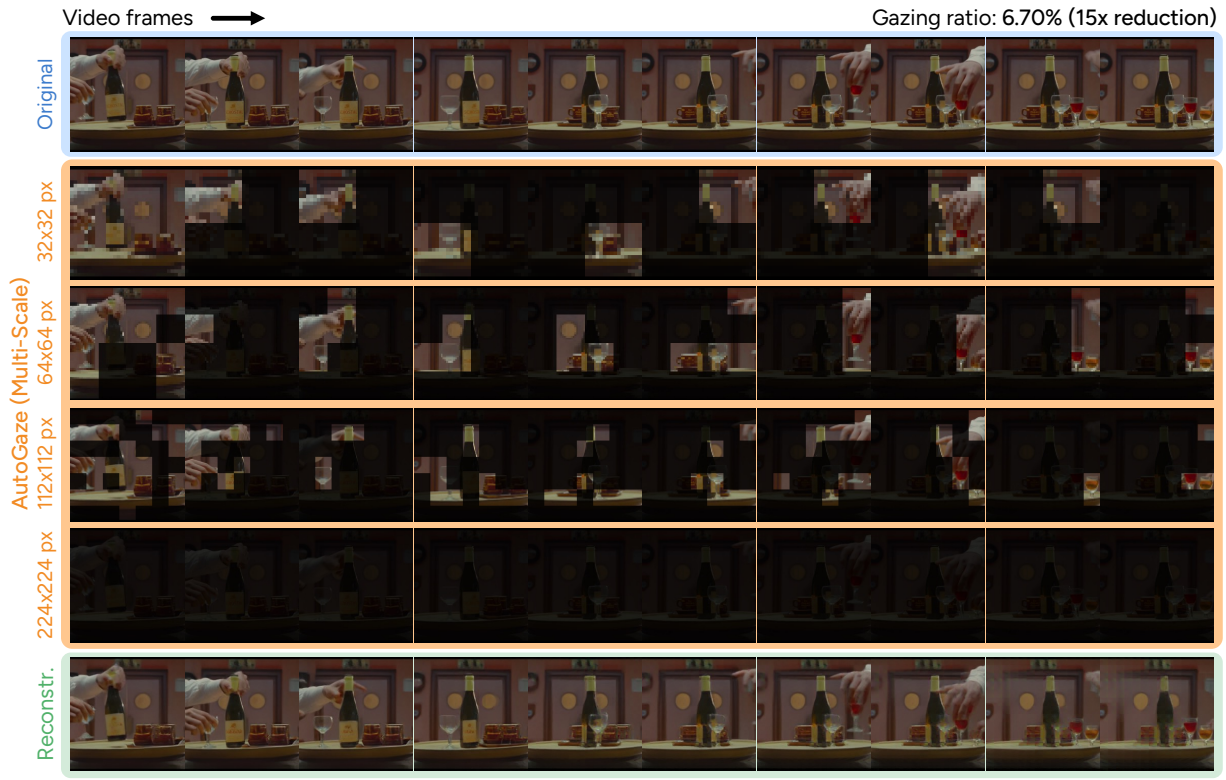

Figure 15. **Film clip.** AutoGaze selects minimal patches to track the drink bottles and glasses as they are moved by hands a rotating plate.

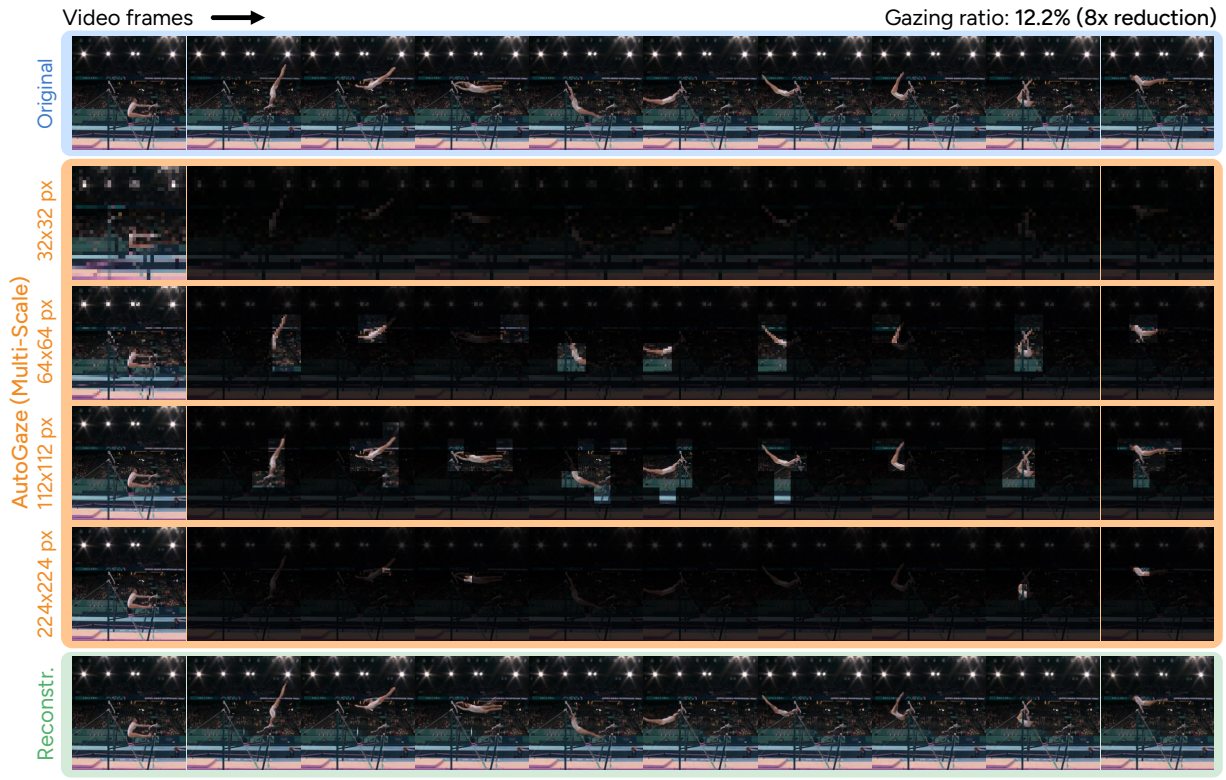

Figure 16. **Gymnastics clip.** AutoGaze tracks the gymnast across the uneven bars, using finer scales when appropriate.

Figure 17. **Claymation cartoon.** AutoGaze captures small movements (blinking), scene changes, and enough patches to reconstruct text.

Figure 18. **Claymation cartoon.** AutoGaze skips the finest scale as it is not needed to achieve the specified reconstruction loss.

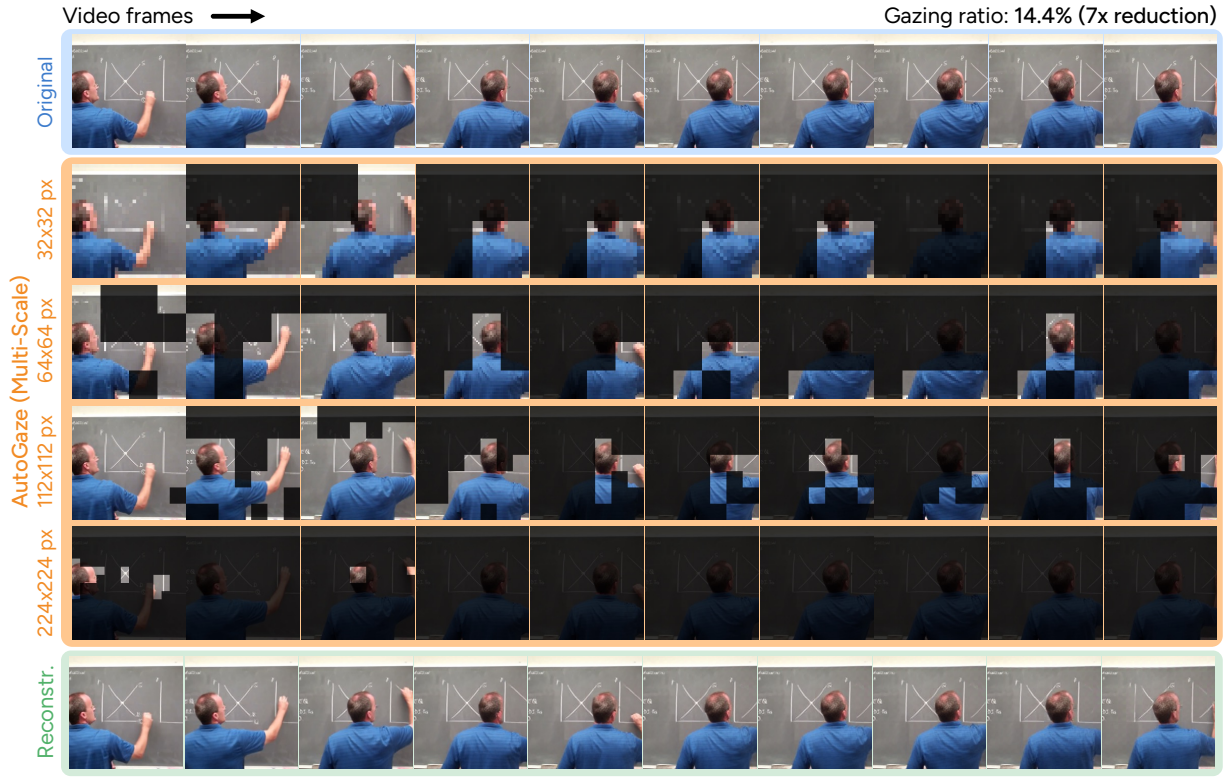

Figure 19. **Chalkboard lecture.** The minimal patches are selected to reconstruct both the lecturer's movement and the chalkboard writing.

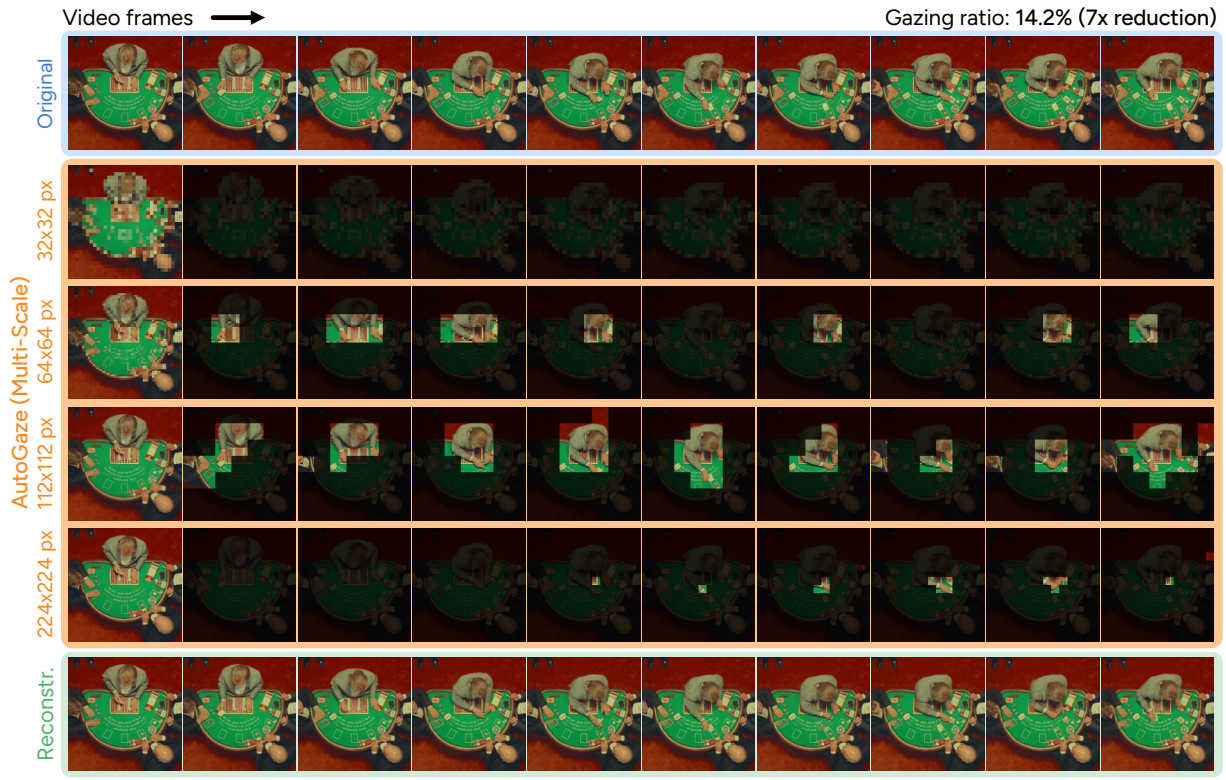

Figure 20. **Film clip.** In this game of Blackjack, AutoGaze uses finer scales to capture hand movements and cards.

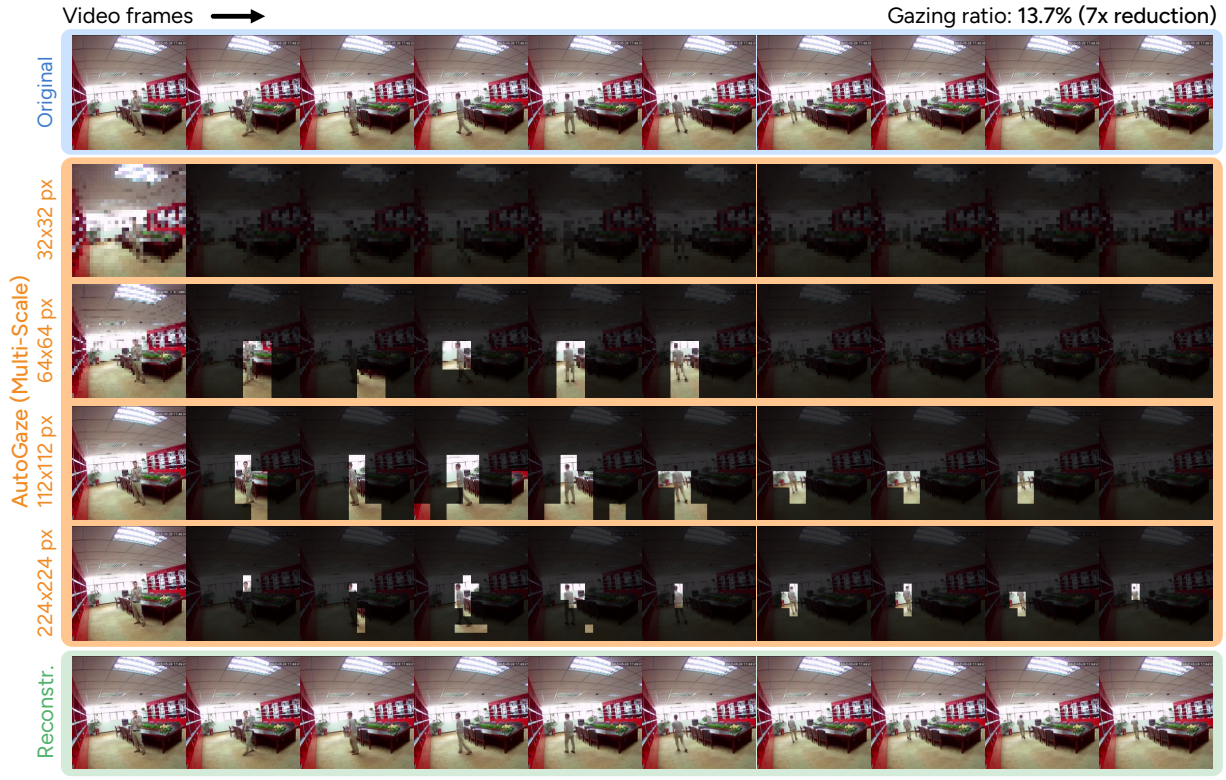

Figure 21. **Fisheye lens.** AutoGaze can select patches that track moving objects with appropriate scales even with lens distortion.

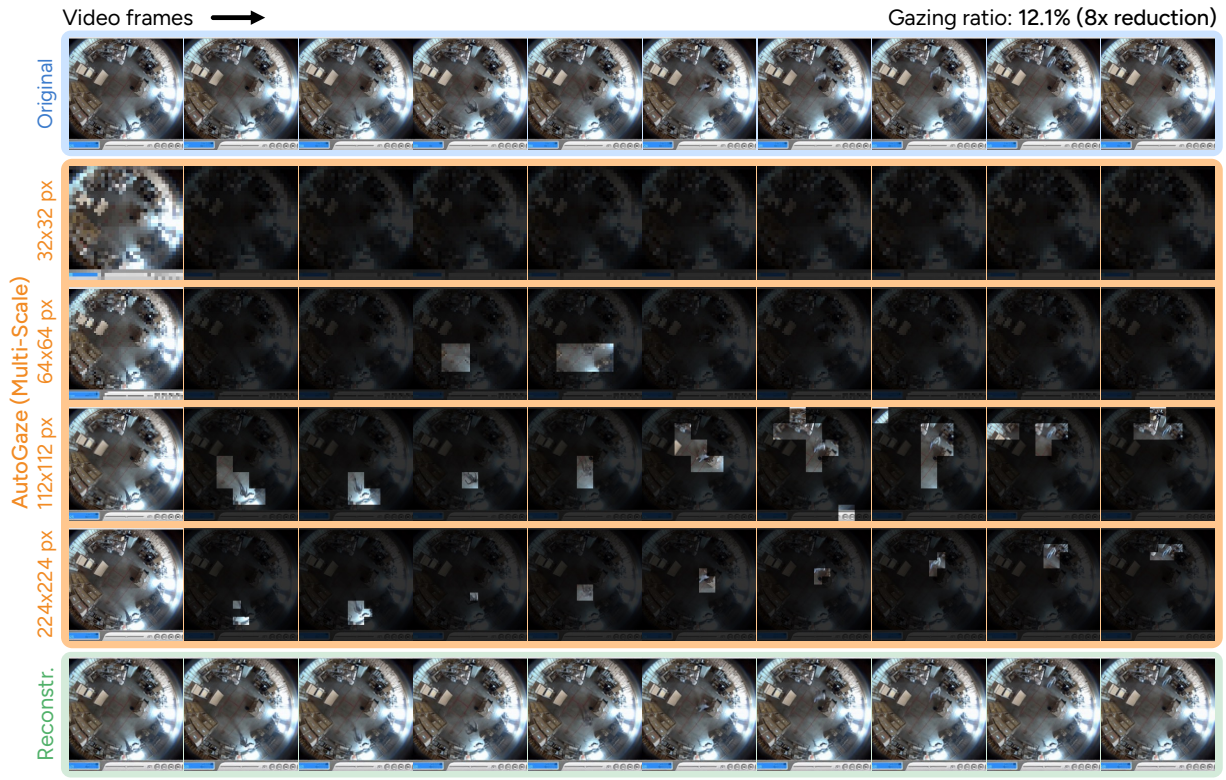

Figure 22. **Fisheye lens.** In this overhead video with lens distortion, AutoGaze tracks the walking person and ignores the static warehouse.

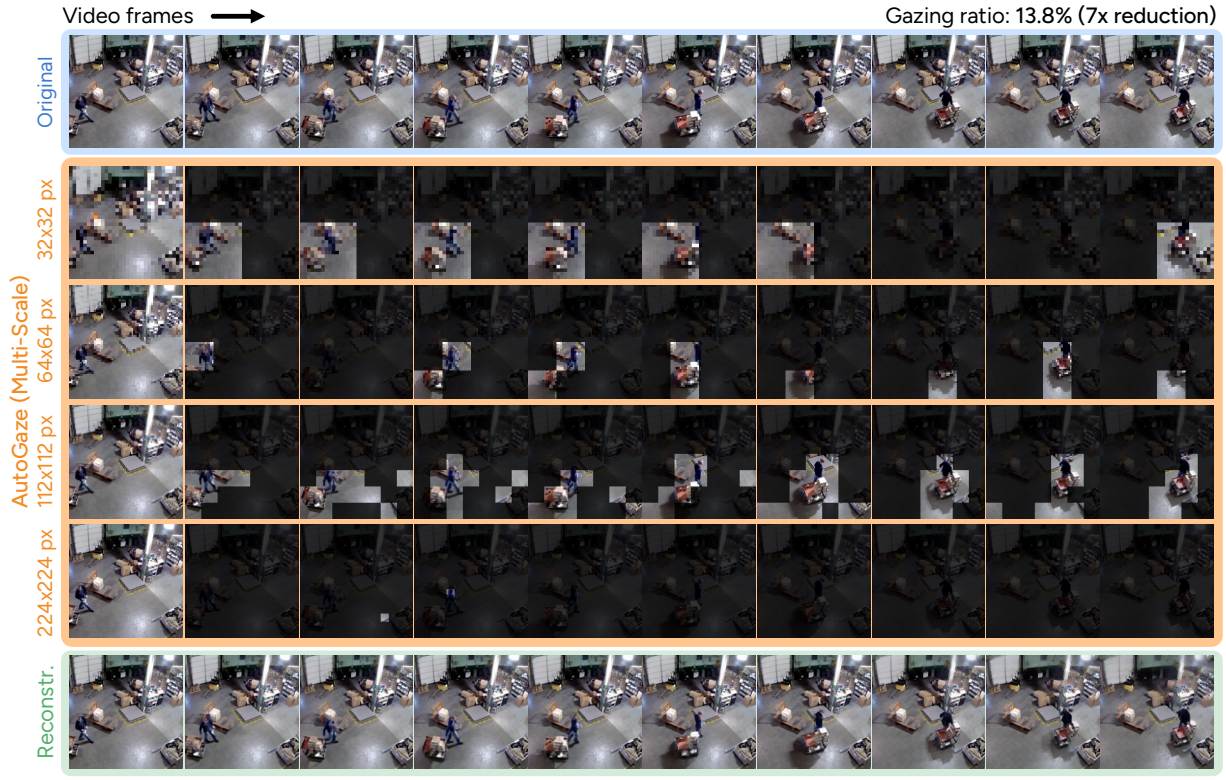

Figure 23. **Warehouse example.** In this video, AutoGaze selects just enough patches to reconstruct the moving person and cart.

Figure 24. **Nighttime driving.** AutoGaze can be used on nighttime videos such as this one, capturing pedestrians and passing cars.

Figure 25. **Robot arm video.** AutoGaze uses different scales to gaze at the robot arm and marker.

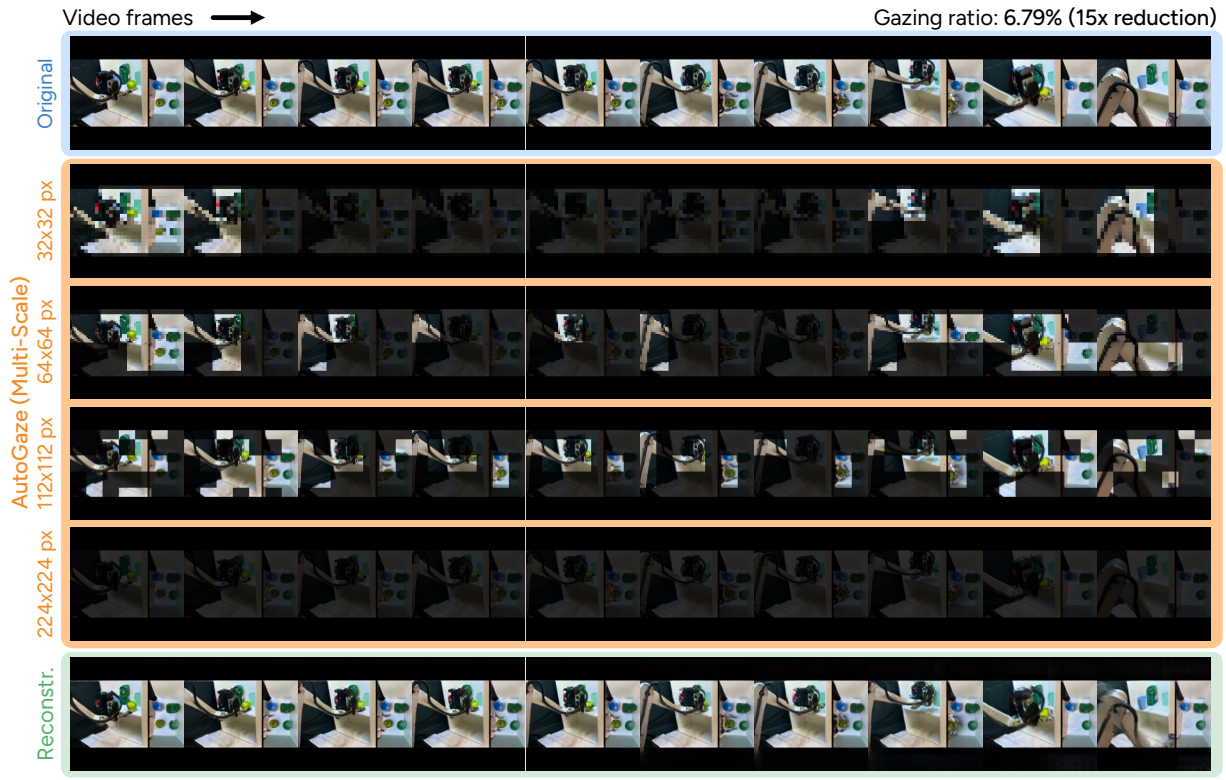

Figure 26. **Multiple perspectives.** Given two side-by-side videos, AutoGaze selects patches in both halves to reconstruct the video.

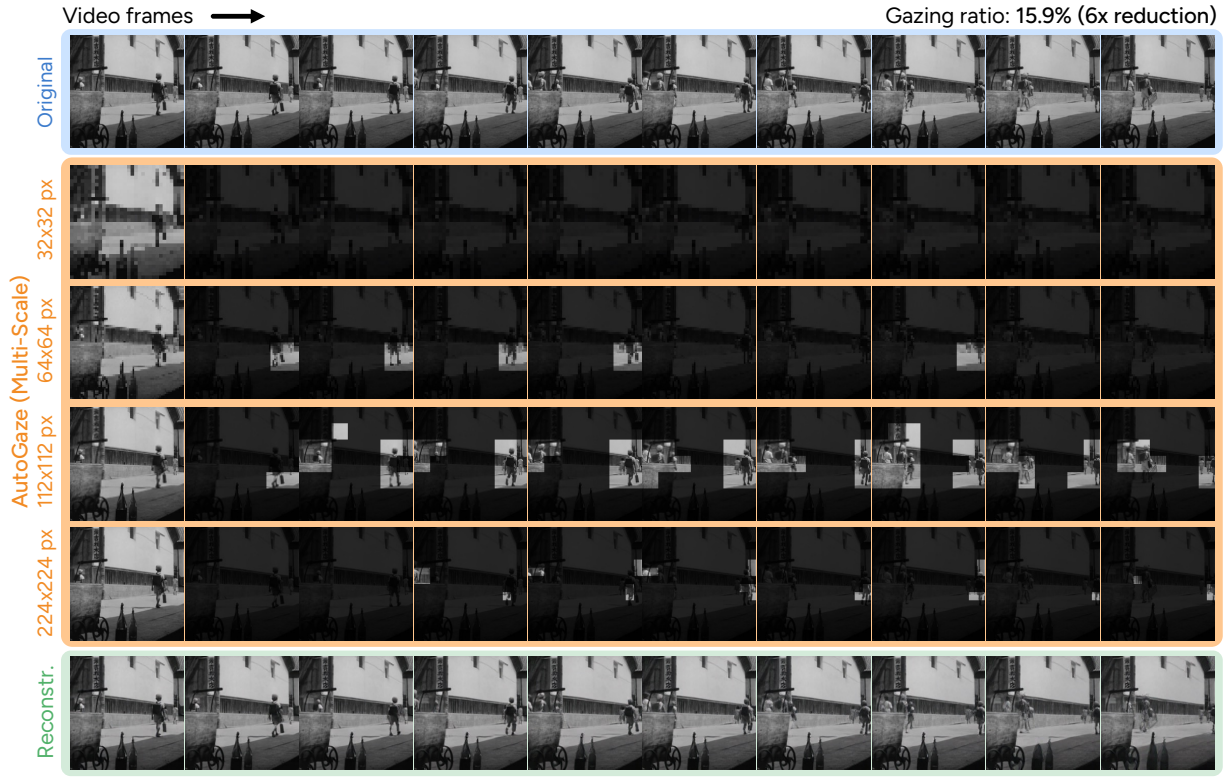

Figure 27. **Black-and-white film.** In this clip where people are walking, AutoGaze uses finer scales to select people as they get smaller.

Figure 28. **Swapping object.** In an OOD scenario where the skateboard swaps out halfway through the video, AutoGaze continues to track the moving parts and properly reconstruct each video frame.

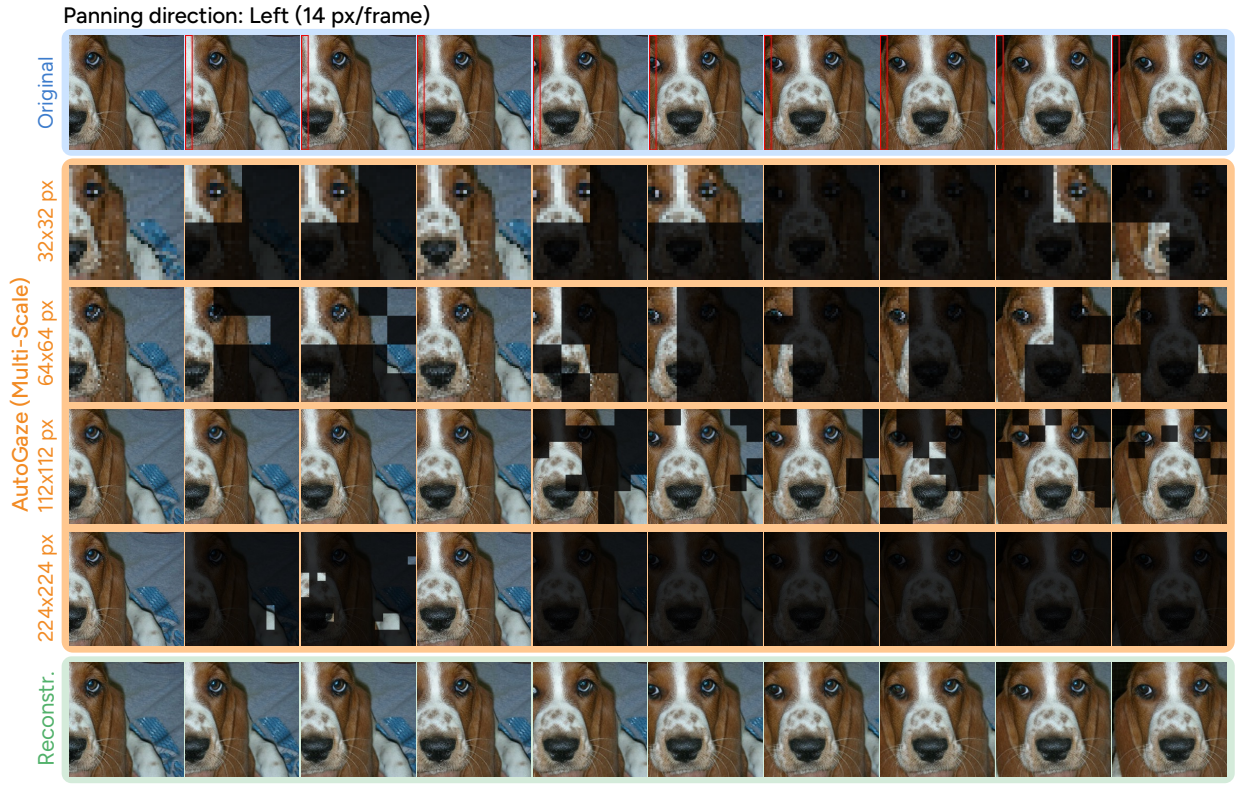

Figure 29. **Panning over a static image.** As the image slides to the left (new pixels highlighted with a red border), AutoGaze does not perfectly track movement; it can select regions that were in different parts of previous frames.

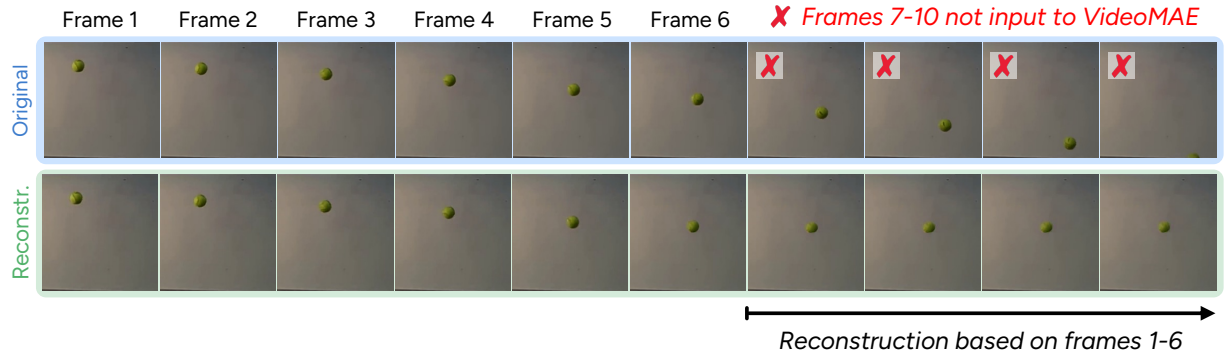

Figure 30. **Predicting next frames from physics knowledge.** Given the first 6 frames of a ball falling along a parabolic path, VideoMAE is unable to reconstruct the next frames to reflect the continued falling motion.