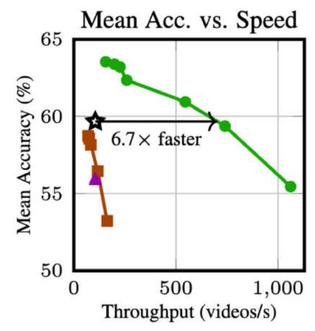
LookWhen? Fast Video Recognition by Learning When, Where, and What to Compute

⋆Ali Salamatian, ⋆Anthony Fuller*, Pritam Sarkar, James R. Green, Leonid Sigal*, Evan Shelhamer*
⋆co-first author, *co-advising author

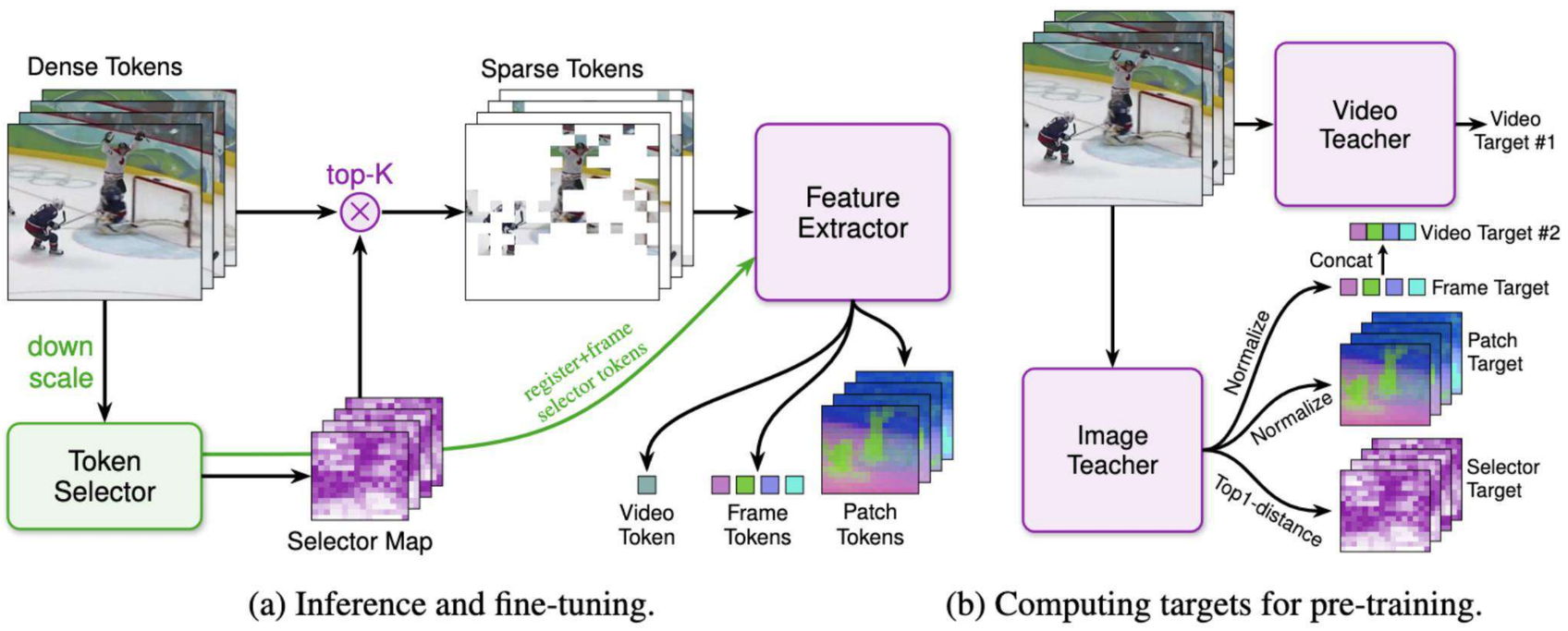
Paper

Motivation

- Video transformers process thousands of space-time tokens, making computation expensive and difficult to scale to long videos.
- Videos contain significant spatiotemporal redundancy, so dense processing of every token is often unnecessary.
- Existing adaptive methods often still compute all tokens before pruning or merging them, limiting real-world speedups despite lower FLOPs. Transformers naturally support sparse computation, creating an opportunity to process only the most informative tokens while preserving accuracy.
- LookWhen introduces a selector-extractor framework that learns when, where, and what to compute, achieving up to **6.7× higher throughput** than prior models at equal accuracy.



Methodology: Architecture Overview

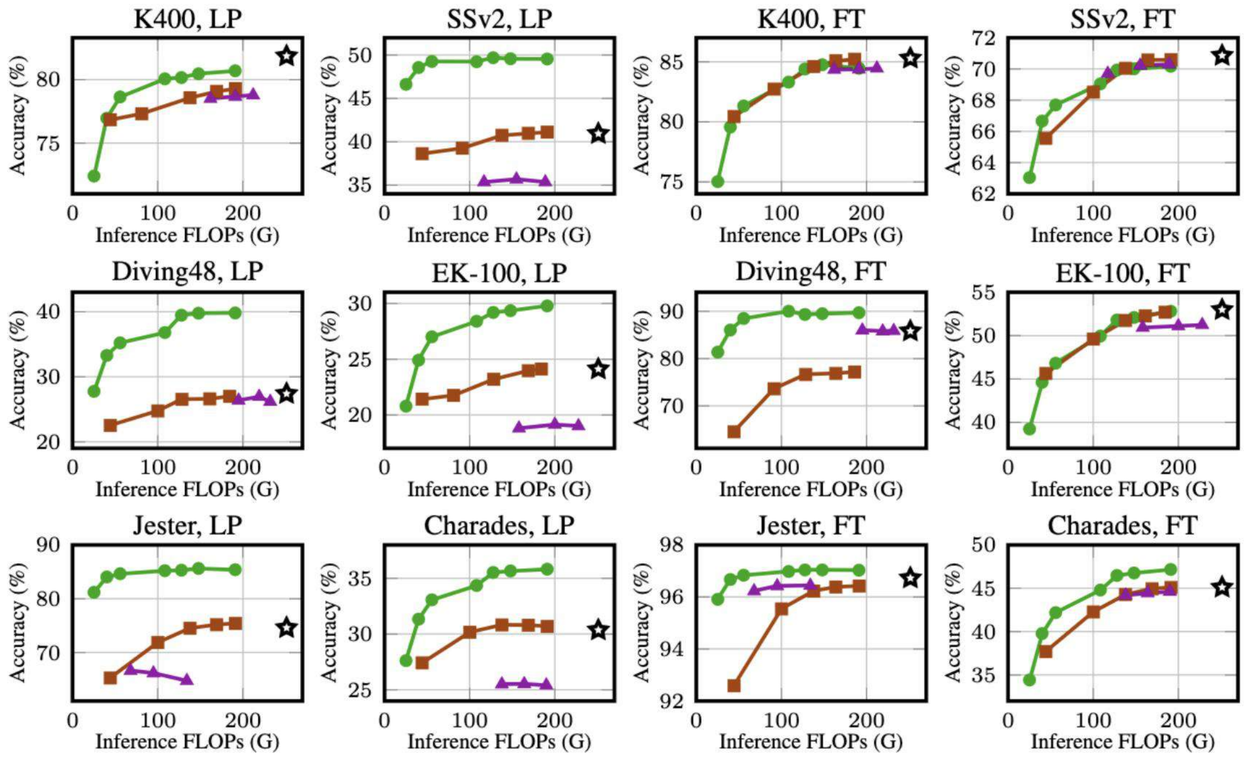


(a) Inference and fine-tuning. (b) Computing targets for pre-training.

LookWhen's shallow selector gets a downscaled video and scores tokens on their feature uniqueness (left). Target uniqueness is from our "top1-distance" algorithm, which computes each patch's distance to its nearest neighbor in an image teacher's feature space (bottom right). LookWhen's extractor gets the top-K input tokens for sparse and deep processing. Target features are from a video teacher (top right) and an image teacher (bottom right); we normalize to emphasize within-video change. Teachers are only needed during pre-training, so inference and fine-tuning is efficient.

Quantitative Results

1) Better accuracy-FLOPs trade-off than upgraded baselines



2) Better accuracy-computation trade-off than existing models on Kinetics-400 and SSv2

Model	Params M	Kinetics-400 FLOPs G×T×S	Kinetics-400 Top-1 % acc.	SSv2 FLOPs G×T×S	SSv2 Top-1 % acc.
Larger models for reference only, not for direct comparisons					
V-JEPA-2 [25]	355	935×8×3	85.1	935×2×3	73.7
InternVideo2 [18]	1020	2500×4×3	89.4	2500×2×3	69.7
ViT-B, Swin-B, or Mamba-M models					
UMT-B800e [29]	87	180×4×3	85.7	180×2×3	70.8
VideoMAE [3]	87	180×5×3	81.5	180×2×3	70.8
VideoMAEv2 [5]	87	180×5×3	81.5	180×2×3	71.2
VideoMamba-M800e [30]	74	202×4×3	83.4	202×2×3	71.0
VideoMambaPro [31]	72	392×4×3	84.0	183×4×3	69.4
VideoSwin + STTS ($T_0^{0.6}$) [32]	89	181×4×3	81.4	190×1×3	68.1
VideoMAE + LITE (K=0.3) [26]	87	46×5×3	78.4	46×2×3	68.3
VideoMAE + ToMe (r=64) [10, 8]	87	131×4×3	80.0	131×4×3	69.7
VideoMAE + RLT (τ=0.1) [8]	87	120×4×3	80.1	120×4×3	70.2
VideoMAE + vid-TLDR [11]	87	—	—	57×unk	69.6
LookWhen (90% sparse)	106	40×4×3	82.6	40×2×3	69.3
LookWhen (70% sparse)	106	108×4×3	84.6	108×2×3	72.0

3) Unique token selection beats attention-based selection.

Teacher	Method	K400-20K LP	K400-20K FT	SSv2-20K LP	SSv2-20K FT	Diving48 LP	Diving48 FT	EK100-20K LP	EK100-20K FT	Jester-20K LP	Jester-20K FT	Charades LP	Charades FT
DINOv3	top1-dist	72.9	73.6	31.5	45.2	**29.7**	**83.2**	**11.6**	**22.2**	**63.4**	**94.7**	**29.3**	**36.7**
DINOv3	attn	72.1	72.8	30.1	41.1	28.0	77.7	9.1	17.6	53.8	90.1	26.6	33.2
DINOv3	Δattn	**73.5**	**74.2**	**31.7**	**45.6**	27.6	76.0	11.0	21.3	**63.4**	94.4	28.6	35.7
InternVideo2	top1-dist	64.7	64.7	27.0	31.2	15.8	44.6	7.3	13.4	56.8	89.2	21.9	26.0
InternVideo2	attn	68.6	68.9	28.6	35.4	26.2	72.9	9.3	15.2	58.2	89.9	24.4	30.4
none	random	72.5	72.7	30.4	42.8	18.3	57.2	10.2	20.3	57.2	91.8	25.9	30.8

4) Jointly predicting video tokens from InternVideo2 and DINOv3 improves representations.

IntVid2	DINOv3 Vid	DINOv3 Frame	DINOv3 Patch	DINOv3 Norm	K400-20K LP	K400-20K FT	SSv2-20K LP	SSv2-20K FT	Diving48 LP	Diving48 FT	EK100-20K LP	EK100-20K FT	Jester-20K LP	Jester-20K FT	Charades LP	Charades FT
✔	✔	✘	✘	✔	**73.3**	**74.0**	**37.9**	48.0	29.9	81.9	**12.8**	**23.3**	69.0	94.6	**30.3**	**37.4**
✔	✔	✘	✘	✘	72.1	73.3	31.5	45.1	**32.1**	84.2	11.3	22.4	63.9	94.9	29.5	36.9
✔	✘	✘	✘	✘	**73.3**	73.9	32.1	45.6	27.9	83.1	11.0	22.0	61.6	94.4	29.0	36.3
✘	✔	✘	✘	✔	17.9	61.1	26.7	**48.8**	26.7	**85.7**	2.6	18.7	**70.4**	**95.2**	14.0	30.5

1) Our LookWhen (•) outperforms the baselines in controlled settings. Gains are largest for linear probing (LP), sometimes surpassing the dense InternVideo2 (⋆). We make these upgraded baselines by applying the sparsification methods vid-TLDR (■) and RLT (▲) to the SOTA ViT-B InternVideo2.

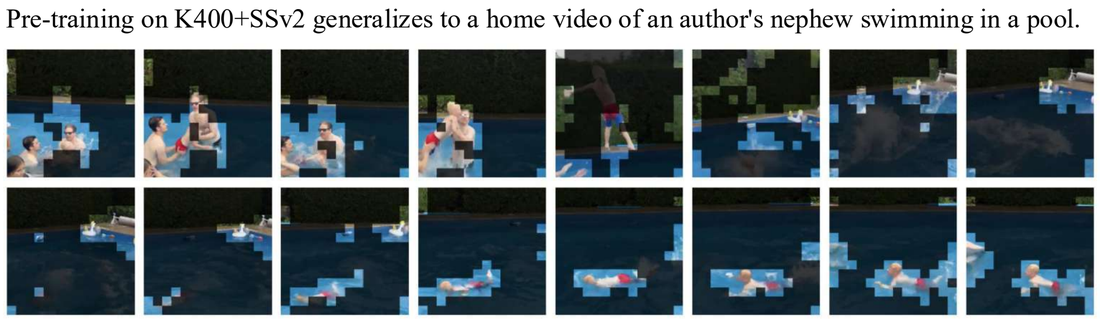
2) LookWhen achieves a better accuracy-computation trade-off on K400 and SSv2. Despite having more parameters, its sparse computation also leads to a stronger overall accuracy-throughput trade-off.

3) Training to select unique tokens (top1-dist) beats highly-attended tokens, if we have suitable teacher features (e.g. DINOv3). The change in DINOv3's attention between successive frames (Δ) beats InternVideo2's space-time attention (which contains artifacts) and DINOv3's space-only attention.

4) Predicting both InternVideo2's video token and a video token we make from DINOv3's frame-wise class tokens improves representations. We time-normalize each dimension before concatenating DINOv3 tokens to learn what changes. Frame and patch losses are disabled to isolate video-token supervision.

Qualitative Results: LookWhen Generalizes Well

Pre-training on K400+SSv2 generalizes to a home video of an author's nephew swimming in a pool.



Example from Jester.