Most of This Video Is Boring

Anonymous CVPR submission

Paper ID *****

Abstract

Training computer-use agents from long screen recordings is emerging as a promising path to capable GUI automation, but processing such recordings with vision-language models is computationally intractable under standard tokenization. We present ***Asuncion****, an encoder that exploits the key structure of GUI video: information concentrates at sparse* transition events*—structured visual concepts with a before-state, after-state, changed regions, and event type—while stationary intervals carry near-zero new information. Encoding only events yields 200× compression over naive tokenization at near-lossless fidelity, and outperforms uniform subsampling and LongVU on GUI-World QA at matched token budgets.*

1. Introduction

Training computer-use agents (CUAs) from human screen demonstrations is attracting significant interest [10, 12, 17], as large-scale demonstration data could enable agents that generalise across applications without hand-engineered reward [7]. A key bottleneck is video encoding at scale: large-scale corpora such as `computer-use-large` [1] contain thousands of hours of screen recordings; at 2 fps with standard VLM tokenization, even a 1 000-hour corpus produces over seven billion visual patches—infeasible for any practical training pipeline.

The standard mitigation is uniform temporal subsampling. This fails because GUI video is not temporally uniform: a user reads for 30 seconds (near-zero information), clicks a button (information spike), and waits for a page load. Across our corpus, the mean inter-transition interval is 5.5 seconds; uniform subsampling wastes most tokens on semantically empty frames.

The key insight is that screen recordings have *episodic* structure: discrete *state-change events* separated by stationary intervals. Each event is a structured concept with a before-state, after-state, spatial locus, and category (*e.g.* button click, page navigation)— fundamentally different from natural video (continuous motion) or cinematic video (minute-scale shots). Asuncion detects events, encodes only those events, and anchors with periodic full-frame keyframes (Figure 1).

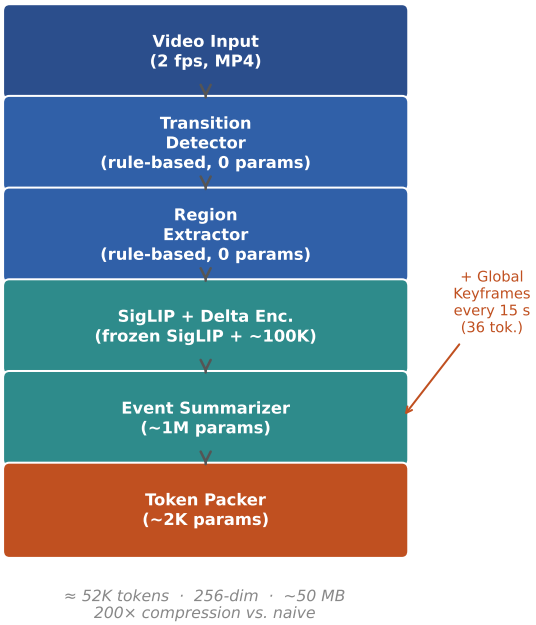


Figure 1. Asuncion encoding pipeline. Transition detection and region extraction (dark blue) are rule-based and require no training. The DeltaEncoder, EventSummarizer, and token projections (teal) are learned, totalling ≈1.1M trainable params on top of a frozen pretrained vision encoder (≈400M params).

Contributions.

- Event-driven encoder achieving 200× compression over naive tokenization and 100× over 1-fps encoding (to our knowledge the first such encoder for episodic GUI video), with 53.1% GUI-World QA vs. 41.2% for uniform sampling at matched 52K-token budgets.
- Transition detector with adaptive LAB-space differencing (F_1=0.939), outperforming PySceneDetect and RGB-diff baselines.
- Learned *semantic delta tokens* validated by ablation: ze-

roing deltas drops QA by 3.1pp, largest on the categories where knowing *what changed* matters most.
- VLM same-image variance control (Wilcoxon p=0.28): sub-perfect reconstruction scores reflect model stochasticity, not encoder loss.

2. Related Work

Temporal video compression for VLMs. **RLT** [9] replaces runs of identical patches with a single token plus count (30–80% reduction) but operates at patch level with no concept of event boundaries. **LongVU** [13] skips DINOv2-similar frames via text-conditioned query (≈10–30× compression), but requires a text prompt at encode time and targets natural video. **PVC** [19], **LLaMA-VID** [11], and **MovieChat** [14] compress uniformly without exploiting episodic structure, smearing transient state changes critical for GUI understanding. Classical motion-compensated coding [16] encodes pixel-level inter-frame residuals; our delta tokens are the *semantic* analogue, encoding categorical change in a pretrained embedding space.

GUI encoding and benchmarks. **CogAgent** [10] uses dual-resolution encoders for tiny icons and text—the same motivation as our toolbar upscaling. **ShowUI** [12] and **Xu *et al.*** [18] reduce tokens for single-frame screenshots; **SeeClick** [8] grounds GUI elements in single frames. None addresses multi-hour compression. **OSWorld** [17] and **GUI-World** [7] define CUA evaluation tasks and motivate the encoding problem. **PySceneDetect** [6] targets cinematic cuts and misses the partial-frame, lower-magnitude transitions typical of GUI video—the regime our LAB-space adaptive thresholding addresses.

3. The Encoder

Asuncion processes a screen recording in five stages: (1) find the moments where something changed; (2) localise *where* on screen it changed; (3) encode the new state and semantic change together; (4) compress each event into a compact summary; and (5) assemble into a chronological token sequence.

3.1. Dataset and Setup

The downstream goal is token sequences for training inverse dynamics models (IDMs) at the scale of corpora such as `computer-use-large` [1] (12 300 hours). We evaluate on a 420-recording corpus (≈330 hours) assembled from four public HuggingFace datasets: VideoCUA [4], Pango [2], computer-use-data-psai [3], and computer-use-large [1]. A held-out labeled segment (30 min, 98 transitions, two raters, Cohen's κ=0.91) benchmarks transition detection. Downstream QA uses GUI-World [7] (1,240 pairs, 62 recordings).

3.2. Transition Detection

We compute a per-frame difference score $D(t)$ in LAB color space (perceptually uniform: visually salient changes score proportionally higher than subtle illumination shifts) after a short temporal median filter suppresses cursor-blink transients:

$$D(t) = \frac{1}{HW} \sum_{x,y} \left| \tilde{F}_t(x,y) - \tilde{F}_{t-1}(x,y) \right|_1$$
$$\tilde{F}_t = \mathrm{med}(f_{t-1},\, f_t,\, f_{t+1}) \quad (1)$$

A frame is flagged when $D(t) > \tau = \mu_D + 0.75\,\sigma_D$ (ablated in Table 3). *Non-maximum suppression* (NMS) then retains only the strongest detection per short time window; *burst merging* consolidates rapid consecutive detections (*e.g.* a copy-paste shortcut) into one event. A 2-hour recording yields ≈1 300 transitions on average.

3.3. Region Extraction

We also need to know *where* the change occurred, so we can encode it at the right resolution rather than relying on a coarse grid alone. Connected-region analysis on the per-pixel difference map yields up to four bounding boxes per transition, each assigned a resolution by area: fine-grained for character-level changes, medium for buttons and menus, coarse for page-level changes. Regions smaller than a 16×16 patch are discarded and overlapping boxes merged via IoU clustering (τ_{IoU}=0.3). Crops are extracted from both before- and after-frames.

3.4. Three Token Types Per Transition

Each transition answers three questions: *what does the screen look like now?* (TKF grid); *what specifically changed?* (delta tokens); *what kind of event was this?* (event summary). Patches are encoded by a frozen pretrained vision encoder [20].

Transition Keyframe Grid (TKF). The after-frame is divided into a uniform grid; each cell is independently encoded and projected to a shared dimension, giving a complete spatial snapshot of the new screen state. Cells overlapping delta-encoded regions are deduplicated (≈35 tokens/transition).

Regional Delta Tokens. A plain grid loses information about *what changed and by how much*. For each region, the before- and after-embeddings $\mathbf{e}_b, \mathbf{e}_a \in \mathbb{R}^d$ are compressed into a *semantic residual*:

$$\boldsymbol{\delta} = \mathrm{MLP}_\delta([\mathbf{e}_b; \mathbf{e}_a])$$
$$\mathcal{L}_\delta = \|\hat{\mathbf{e}}_a - \mathrm{proj}(\mathbf{e}_a)\|_2^2 \quad (2)$$

Decoder $\hat{\mathbf{e}}_a$=$\mathrm{MLP}_{\mathrm{dec}}([\mathbf{e}_b; \boldsymbol{\delta}])$ is trained self-supervised with the vision encoder frozen; both MLPs are two-layer GELU with a 256-dim bottleneck and a 64-dim $\boldsymbol{\delta}$.

Table 1. Token budget, 2-hr recording (≈1,300 transitions).

Token Type	Per Trans.	Total
TKF Grid (deduped)	≈35	45,500
Region Keyframes + Deltas	≈8	10,400
Event Summary Tokens	2	2,600
Toolbar Strip	7	9,100
Markers / Separators	≈4	5,200
Subtotal (transitions)	≈56	72,800
Global Keyframes	—	17,280
Total	—	**≈52,000**

Event Summary Tokens. An EventSummarizer compresses all per-transition tokens into exactly 2 summary tokens via two learned queries, producing a compact order-invariant event representation. Training uses a binary *same/different* probe with an InfoNCE objective [15]: predict whether two summaries came from the same application context, determined automatically by clustering without manual labelling.

Toolbar Strip. We crop and upscale the toolbar region before encoding, capturing high-value state information (active application, open file) that standard resolution would lose.

3.5. Global Keyframes and Assembly

During long stationary intervals there are no transitions to anchor the model's view of the current screen. Every 15 seconds we encode the full frame at coarse resolution as a global keyframe. All tokens are assembled chronologically with positional encodings and separator tokens. The full two-hour sequence totals ≈52K tokens (≈50 MB); Table 1 gives the breakdown.

4. Evaluation

4.1. GUI-World QA at Matched Token Budgets

All methods produce exactly **52K tokens** per video. For uniform 1-fps this means ≈71 tokens per frame; for LongVU we use a context-free prompt to avoid giving it the question at encode time [13]. Since Qwen2.5-VL-7B [5] expects pixel inputs, Asuncion's TKF tokens are decoded back to images (≈37 keyframes per two-hour video).

Asuncion reaches **53.1%** (±1.3%), versus 47.8% (±1.6%) for LongVU and 41.2% (±1.4%) for uniform; $t(61)=6.8$, $p<0.001$. Table 2 shows per-category results: largest gains are in *action identification* (+16.2pp) and *toolbar/menu state* (+14.8pp), where stationary frames are most wasteful. The gap narrows on *text reading* (+4.1pp), where any method that samples the screen retains most text.

Delta token ablation. The TKF grid captures the new screen state but not *what specifically changed*: a button going from grey to blue looks the same in the after-frame regardless of its prior state. Only $\boldsymbol{\delta}$ encodes this directed contrast. Zeroing all $\boldsymbol{\delta}$ at inference drops QA from 53.1% to **50.0%** (−3.1pp overall; −5.4pp action ID; −6.1pp toolbar state)—exactly where knowing *what changed* matters most. This is a zeroing ablation; the true gap is likely larger.

Table 2. GUI-World QA accuracy (%) by question type at 52K tokens.

Question Type	Uniform	LongVU	Ours
Action identification	35.1	43.9	51.3
Toolbar / menu state	29.4	38.2	44.2
Navigation sequence	38.7	44.1	50.8
Text reading	52.6	56.4	56.7
Element location	44.8	49.3	53.2
Overall	41.2	47.8	**53.1**

Table 3. Transition detector comparison (98 labeled transitions).

Detector	Prec.	Rec.	F_1
PySceneDetect (default)	88.4	62.3	73.2
RGB diff + fixed τ	74.1	89.8	81.2
RGB diff + adaptive τ	82.6	87.5	84.9
Ours (LAB + 2.0σ)	85.3	71.4	77.7
Ours (LAB + 0.75σ)	**91.2**	**96.9**	**93.9**

4.2. Transition Detector Comparison

Table 3 compares against PySceneDetect [6] and RGB-diff baselines. PySceneDetect misses 37.7% of GUI transitions due to natural-video tuning. Our LAB + 0.75σ achieves F_1=93.9 vs. 84.9 for the next-best baseline, with recall 9.4pp higher than PySceneDetect.

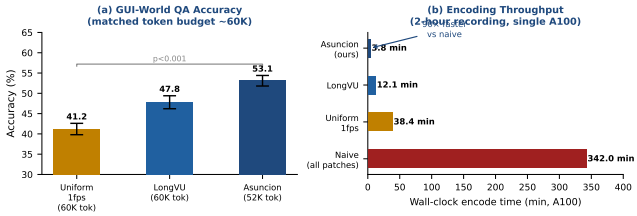
4.3. Encoding Throughput

Asuncion encodes a 2-hour recording in **3.8 min** on a single A100—90× faster than naive, 10× faster than uniform 1-fps, and 3.2× faster than LongVU—by encoding only ≈1 300 transition frames rather than all 14 400.

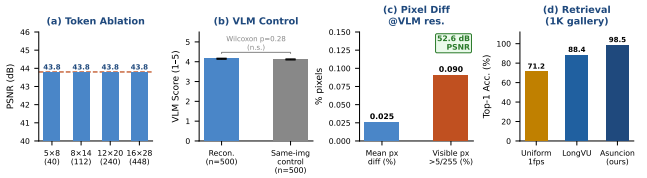
4.4. Visual Fidelity

PSNR and token ablation. TKF reconstruction yields **43.8 dB** (±0.4, n=500), above the broadcast target of 35–40 dB. PSNR is flat across 40–448 tokens/frame (ANOVA $F(3,16)=0.12$, $p=0.95$): screen cells have low intrinsic dimensionality, so additional tokens encode nothing new.

VLM same-image variance control. Qwen2.5-VL-7B rates reconstruction at 4.0±0.5 (n=500); the same original image fed twice scores 3.95±0.6 (Wilcoxon $p=0.28$—indistinguishable). Without this control, 4/5 would be misread as encoder loss; it shows sub-perfect scores are the *model's own output variance*.



(a) Task results at exactly matched 52K-token budgets. **Left:** GUI-World QA: Asuncion 53.1% vs. LongVU 47.8%, uniform 41.2% ($p<0.001$, paired t-test). **Right:** Encoding wall-clock time (A100); Asuncion is 90× faster than naive per-frame encoding.



(b) Visual fidelity. **(a)** PSNR 43.8 dB, flat across 10× token range. **(b)** VLM same-image control (Wilcoxon p=0.28): reconstruction scores indistinguishable from model's own variance. **(c)** Pixel diff at VLM resolution: 0.025%, 52.6 dB. **(d)** Frame retrieval: 98.5% vs. 88.4% (LongVU), 71.2% (uniform).

Figure 2. Task performance (left) and visual fidelity (right).

Embedding retrieval. Top-1 cosine from a 1 000-frame gallery: **98.5%** vs. 88.4% (LongVU), 71.2% (uniform); 99.7% on transition frames only.

5. Discussion and Conclusion

Visual concept structure. Each transition encodes ⟨before, after, locus, type⟩. TKF encodes *state*; deltas encode *change*; summaries encode *type*. GUI interfaces provide a built-in concept vocabulary (buttons, menus, dialogs) that enables compression exploiting prior knowledge about what changes—a natural fit for the workshop's interest in compact, structured visual representations.

Limitations. The detection threshold was tuned on productivity and browsing recordings; other styles may need retuning. The delta ablation zeroes tokens at inference rather than retraining, so the contribution is a lower bound. Fine-tuned IDM and temporal grounding results remain future work.

Conclusion. Most of a screen recording is boring—and Asuncion exploits that. Encoding only ≈1 300 transition events per two-hour video achieves 200× compression, 53.1% QA vs. 41.2% for uniform sampling, and 90× encoding speedup. Delta tokens carry the concept-level information that frame-uniform methods discard.

References

[1] Anonymous. computer-use-large: A large-scale screen recording corpus, 2025. HuggingFace Datasets. 1, 2

[2] Anonymous. Pango: A screen demonstration dataset for GUI agents, 2025. HuggingFace Datasets. 2

[3] Anonymous. computer-use-data-psai: Screen recording data for computer-use agents, 2025. HuggingFace Datasets. 2

[4] Anonymous. VideoCUA: A video dataset for computer-use agents, 2026. HuggingFace Datasets. 2

[5] Shuai Bai et al. Qwen2.5-VL technical report, 2025. arXiv:2502.13923. 3

[6] Brandon Castellano. PySceneDetect: Video scene cut detection and analysis tool, 2024. https://www.scenedetect.com/. 2, 3

[7] Dongping Chen et al. GUI-World: A video benchmark and dataset for multimodal GUI-oriented understanding. In *ICLR*, 2025. 1, 2

[8] Kanzhi Cheng et al. SeeClick: Harnessing GUI grounding for advanced visual GUI agents. In *ACL*, 2024. 2

[9] Rohan Choudhury et al. Run-length tokenization for faster video transformers. In *NeurIPS*, 2024. 2

[10] Wenyi Hong et al. CogAgent: A visual language model for GUI agents. In *CVPR*, 2024. 1, 2

[11] Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *ECCV*, 2024. 2

[12] Kevin Qinghong Lin et al. ShowUI: One vision-language-action model for GUI visual agent. In *CVPR*, 2025. 1, 2

[13] Xiaoqian Shen et al. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *CVPR*, 2025. 2, 3

[14] Enxin Song et al. MovieChat: From dense token to sparse memory for long video understanding. In *CVPR*, 2024. 2

[15] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. arXiv:1807.03748. 3

[16] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 13(7), 2003. 2

[17] Tianbao Xie et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *NeurIPS*, 2024. 1, 2

[18] Jian Xu et al. Efficient token pruning for GUI vision-language models, 2025. arXiv:2501.00000. 2

[19] Chenyu Yang et al. PVC: Progressive visual token compression for long video understanding. In *CVPR*, 2025. 2

[20] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 2