A SIMULATION-GROUNDED VLM FRAMEWORK FOR AGENTIC WILDFIRE MONITORING AND REPORTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective wildfire monitoring requires relating visual evidence to physical fire dynamics, yet real videos with synchronized physical annotations are scarce and high-fidelity 3D simulation is costly. We present a simulation-grounded vision-language model (VLM) framework that automatically converts 2D wildfire simulations into labeled video episodes. A fixed Blender mapping produces low-detail 3D proxies aligned with simulator terrain, fuel layout, fire activity, and wind cues; controllable video generation supplies richer appearance. The proxies are intermediate representations rather than finely rendered final scenes. Generated videos and simulator labels form reusable multimodal memory for a training-free multi-agent VLM system that retrieves reference episodes, reconciles visual and memory-based predictions, and produces structured wildfire reports. On held-out generated episodes, video memory achieves 51.5% exact four-tag accuracy, compared with 22.6% for direct VLM querying and 16–17% for text-only memory; the complete system achieves 77.3% accuracy on six simulator-derived report fields. Component ablations, cross-generator tests, and three real-UAV evaluations assess retrieval, reporting, generator changes, and observable monitoring tasks. The framework connects automatic simulation-to-proxy conversion with memory-based VLM reasoning under scarce real-world physical annotations.

1 INTRODUCTION

Wildfire monitoring and reporting are fundamentally about connecting *what is seen* to *what is happening physically* in the wildland. Fire-spread simulators (e.g., Mandel et al. (2011); Finney (1998); Mallet et al. (2009); Doyle et al. (2024)) encode how fuel, slope, moisture, and wind shape propagation, and satellite, aerial, and UAV measurements provide partial visual observations of active incidents. Modern vision-language models (VLMs) need a framework to align two forms of evidence in a shared representation. Real wildfire videos are visually rich, yet labels for wind, moisture, slope, fuel condition, burn extent, and severe-fire behavior are usually absent. Simulators expose these variables directly, yet their native outputs are grid states, masks, and fire fronts, which differ from the aerial-view videos that VLMs are expected to interpret.

Though existing wildfire VLM work has advanced fire and smoke recognition, UAV monitoring, satellite risk analysis, and VLM-guided planning Du et al. (2025); Wu et al. (2025); Seidel et al. (2025); Ayanzadeh et al. (2026); Webb et al. (2026); Xie et al. (2025); Sandeep et al. (2026), temporally rich videos paired with physical fire-state labels remain scarce. Collecting synchronized wind, moisture, slope, fuel, and fire-behavior annotations is hazardous, expensive, and hard to control. Available UAV datasets provide valuable detection and segmentation annotations Shamsoshoara et al. (2021); Kularatne et al. (2024); Pesonen et al. (2025), but the annotations used here do not jointly describe these drivers. Fire-spread simulators expose such conditions without providing camera-view observations, while full 3D fire simulation Kokosza et al. (2024); Hädrich et al. (2021) incurs costs that limit coverage across fuels, terrains, weather, and cameras. We therefore study a scalable middle ground: efficient 2D simulations over real terrain and fuel maps are automatically lifted into lightweight Blender proxies, translated into controlled videos, and stored as physics-labeled memory for VLM reasoning.

Recent developments such as Code World Model (Chen et al., 2026b) and Blender scene authoring with GPT-6 Astra highlight the potential of executable representations for visual world construction. We explore this direction in a simulation-grounded VLM framework for wildfire monitoring and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reporting (Figure 1). Each episode starts with a 2D simulation over real LANDFIRE terrain and fuel data spanning approximately $3\,\mathrm{km}$. A fully automatic Blender procedure maps evolving simulator fields into terrain, vegetation, flame, and wind-driven smoke cues. This lightweight proxy is an *intermediate representation*: fixed conversion rules maintain simulator-to-scene correspondence, while depth, scribble, and prompt controls guide richer video appearance. Generated clips retain their prompts, simulator parameters, environmental tags, and fire-behavior labels as reusable multimodal memory. Retrieved episodes then complement direct visual reasoning for interpreting new wildfire videos without updating VLM weights. Section 6 further discusses the connections to code-based world construction and general-purpose scene authoring.

Generated videos with known simulator labels provide controlled evaluation targets. We use 500 geo-disjoint locations: 400 episodes populate memory and 100 are held out for evaluation (Appendix A). For each query, retrieved references complement direct visual reasoning, and a judge resolves reporter disagreements. Video memory achieves 51.5% exact four-tag accuracy versus 16–17% for text-only memory and 22.6% for direct VLM querying; reporting reaches 77.3% on six simulator-derived fields. Structured records retain predictions, evidence, and reasoning. Cross-generator and real-UAV tests additionally assess generator changes, smoke/fire detection, and visible fire-area assessment.

Our contributions are:

- **Automatic simulation-grounded visual evidence.** We connect efficient 2D wildfire simulation to controllable video generation through a fully automatic Blender proxy. A fixed mapping expresses terrain, fuel layout, fire activity, and wind cues as lightweight 3D controls, constructing labeled visual episodes under scarce real-world physical annotations.
- **Memory-augmented agentic reporting.** We organize generated videos and simulator context as reusable multimodal memory, and combine retrieval and reranking with complementary VLM reporters and selective disagreement adjudication. The training-free system produces environmental-driver estimates and structured wildfire reports without updating model weights.
- **Controlled evaluation with complementary real-UAV tests.** Video memory achieves 51.5% exact four-tag accuracy versus 22.6% for direct VLM querying and 16–17% for text-only memory; the full reporter reaches 77.3% on six simulator-derived fields. Ablations, cross-generator tests, and real-UAV perceptual evaluations examine the framework's components and transfer beyond the primary generated-video setting.

2 Related Work

Wildfire modeling and observation. Wildfire prediction combines spread models with sensing. Rothermel Rothermel (1972) underlies FARSITE Finney (1998); level-set methods handle fire-front topology changes Mallet et al. (2009); WRF-SFIRE couples spread with atmospheric dynamics Mandel et al. (2011). MODIS and VIIRS provide broad-area fire observations Justice et al. (2002); Giglio et al. (2016); Schroeder et al. (2014), complemented by UAV smoke/fire detection and front tracking Bailon-Ruiz et al. (2022); Raoufi et al. (2025); Fan et al. (2025). Rather than replacing simulators or sensors, we automatically translate simulator states into visual reference episodes for VLM monitoring and reporting.

VLMs for scientific and wildfire understanding. Scientific VLM benchmarks show that general-purpose multimodal models can be useful but struggle with domain-specific visual structure, terminology, and evaluation protocols Esparza et al. (2025); Roberts et al. (2024); Wu et al. (2024b); Kuckreja et al. (2024); Li et al. (2023). Wildfire VLM systems have been studied for scene recognition Du et al. (2025), smoke/fire detection Wu et al. (2025); Qi et al. (2026), UAV situational awareness Seidel et al. (2025), satellite-image risk reasoning Ayanzadeh et al. (2026), and VLM-guided reinforcement learning in wildfire digital twins Webb et al. (2026). These works motivate VLM-based decision support, but they also highlight the need for physically grounded video data and task-specific evaluation.

Code world models and simulation-conditioned video. WorldCoder Tang et al. (2024) constructs executable world models through interaction, while Code World Model (Chen et al., 2026b) connects coding-agent-driven state evolution to proxy-conditioned video generation. PhysGen Liu et al. (2024) combines rigid-body simulation with generative video refinement, and Cosmos-Transfer1 Alhaija et al. (2025) supports spatially controlled generation. Together, these approaches motivate separating structured dynamics and spatial controls from visual appearance. We examine this separation for

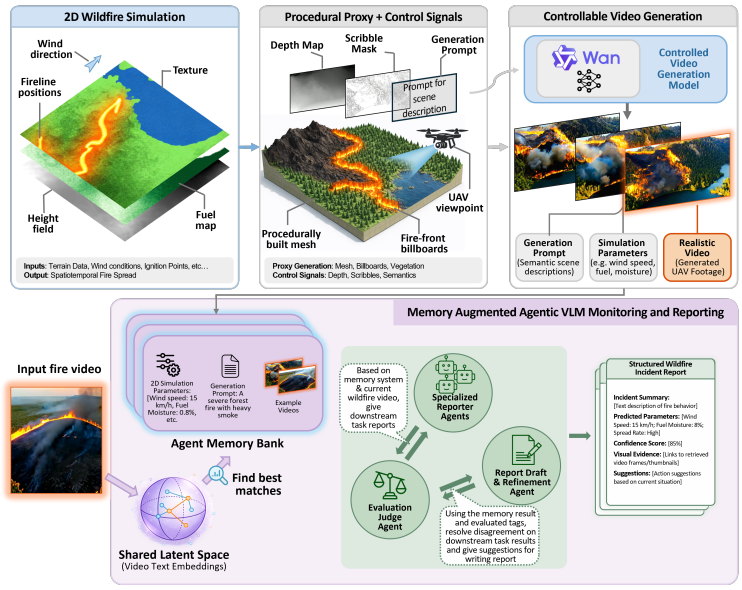


Figure 1: Overall simulation-grounded VLM framework. Top: 2D simulation and procedural 3D proxies provide depth, scribble, and prompt controls for synthetic aerial-view episodes with simulator-derived labels. Bottom: a memory-augmented multi-agent VLM reporter combines query frames, retrieved labeled references, and disagreement adjudication for wildfire monitoring and reporting.

simulation-grounded wildfire memory; Section 6 discusses its relationship to coding-agent world models and scene authoring.

Memory-augmented multimodal reasoning. Retrieval-augmented generation supplies external evidence without updating model weights Lewis et al. (2020). Agent memories support long-horizon reasoning Packer et al. (2023); Park et al. (2023), while multimodal RAG and video retrieval access visual, textual, and scene-level evidence Jeong et al. (2025); Luo et al. (2025); Gia et al. (2025); Zeng et al. (2026); Fu et al. (2026) using video–text embeddings Xu et al. (2021); Luo et al. (2022); Wang et al. (2022b); Ma et al. (2022). WorldMem Xiao et al. (2026) and PhysStream Chen et al. (2026a) use memory for consistent visual generation. Our memory instead pairs reference videos with simulator-controlled conditions and fire-behavior labels for inference across episodes, complementing direct observations and historical-data or context-aware wildfire systems Xie et al. (2025); Sandeep et al. (2026).

3 Method

Figure 1 gives the overall workflow. The framework has two main components: a simulation-grounded visual evidence construction method and a memory-augmented multi-agent VLM system for monitoring and reporting. The subsections below follow this order.

3.1 Simulation-Grounded Visual Evidence Construction

The first stage constructs labeled visual episodes to reduce reliance on scarce real footage with synchronized physical annotations. We retain the scalability and annotations of 2D wildfire simulation while making its outputs visually accessible to VLMs: diverse episodes over real terrain and fuel maps are lifted into procedural 3D proxies and translated into aerial-view clips. Here, *simulation-grounded* refers to the simulator-derived labels and controls linking each episode to its originating scenario; visual and temporal cue alignment is evaluated separately in Section 5.1.

3.1.1 Physics-Grounded 2D Wildfire Simulation

To balance physical grounding and generation throughput, we use SimFire Doyle et al. (2024) as our 2D wildfire simulation platform. SimFire follows the classical Rothermel fire-spread model Rothermel (1972) detailed in Appendix B.1, allowing us to generate large numbers of wildfire episodes with meaningful macroscopic spread behavior. For each episode, we randomly sample a geographic location specified by latitude and longitude, and query the LANDFIRE database to obtain real U.S.



Figure 2: Simulation-grounded synthetic wildfire episodes. Left: SimFire propagation under different settings and locations. Middle: procedural 3D billboard rendering from an aerial viewpoint. Right: generated videos conditioned on the proxy renderings. Episode labels come from the originating simulations.

fuel and terrain information. Fuel types are represented using the Anderson 13 fuel model Anderson (1982), while terrain elevation is used to derive slope and support later 3D proxy construction. Wind direction, wind-related parameters, and fuel-moisture settings are randomized to increase scenario diversity. Each simulation covers a region of approximately $3\,\mathrm{km}$ in spatial extent. We refer readers to Appendix B and C for implementation details.

Each episode is summarized using four physically meaningful factors: wind speed, moisture, slope angle, and vegetation density. Each factor is discretized into low and high levels, yielding a compact set of simulator-derived tags. These tags provide coarse environmental regimes for retrieval-based state estimation and agent reasoning, rather than continuous measurements of a query incident. Tag thresholds are specified in Appendix C; the six downstream report-field definitions appear in Appendix A.

3.1.2 Procedural 3D Proxy

As shown in Figure 1, the 2D simulator produces fire masks, fuel maps, terrain fields, and fire-front states, but these outputs are simple 2D projections onto the terrain topography and therefore do not provide a 3D visual image similar to UAV video footage. We therefore design a Blender pipeline that automatically construct a lightweight 3D proxy scene for each simulation. Elevation is converted into a height-field terrain mesh; Anderson 13 fuel categories are mapped to procedural vegetation and terrain textures; burning cells are augmented with flame billboards; and smoke billboards are oriented and animated according to wind speed and direction. The proxy is not intended to be photorealistic. Its role is to encode simulator-derived spatial structure—terrain shape, vegetation distribution, active fire location, and wind-conditioned smoke direction—in a form that can guide video generation. For details of fuel models mapping with billboard assets, we refer readers to Appendix B.2.

3.1.3 Controllable Video Generation

The proxy rendering encodes simulator structure but remains visually simple. We use controllable video generation to translate it into a visually richer wildfire clip. Following VACE Jiang et al. (2025), the generator is conditioned on a depth sequence Ranftl et al. (2020); Yang et al. (2024), a scribble sequence Chan et al. (2022), and a text prompt. The depth and scribble controls guide geometry, vegetation layout, fire-front location, and camera-consistent structure, while the prompt specifies appearance-level information such as smoke density, flame visibility, vegetation type, and wind-driven motion. The prompt is synthesized from simulation metadata and then refined by a VLM (see Appendix F for details). We also use parameter-conditioned negative prompts to suppress conflicting attributes, such as strong wind for low-wind cases or still smoke for high-wind cases. Each generated video retains its originating simulator labels. Its observable cues reflect both the proxy controls and metadata-conditioned prompts; cue agreement is assessed rather than assumed.

3.2 Memory-Augmented Agentic VLM Monitoring and Reporting

The second stage uses the generated video episodes as physics-labeled visual memory for wildfire state estimation and reporting. The design goal is to keep the system training-free: new simulator episodes update the memory bank, not the VLM weights. As shown in Figure 3, given a query wildfire video, the system retrieves visually and semantically similar memory episodes, uses their

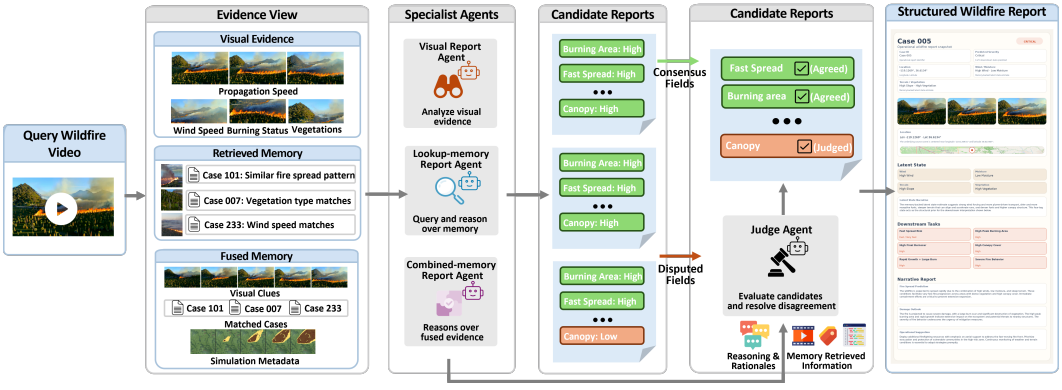


Figure 3: Agentic VLM reporting pipeline. A query wildfire video is analyzed using direct visual evidence and retrieved physics-labeled memory evidence. Agreement across reporters is written to the final report, while disagreement is routed to a judge. The resolved predictions are rendered as a structured HTML report.

simulator labels as reference priors, and combines these priors with direct visual reasoning in a multi-agent report.

3.2.1 Physics-Labeled Multimodal Memory

Each memory entry is a key–value item. The key is a multimodal embedding of the generated wildfire video and its simulation-derived text description. The value stores the generated video, the prompt used to produce it, the 2D simulation parameters, the environmental tags, and downstream fire-behavior labels. This structure makes the memory both searchable and interpretable: retrieval returns not only similar videos, but also the simulator conditions associated with those reference episodes, not ground-truth measurements of the query.

At query time, the same multimodal embedder maps the input video into the shared embedding space. We compute cosine similarity between the query embedding and each memory key and retrieve the top-k candidate episodes. The retrieved examples are then used as in-context evidence for estimating the query's wind, moisture, slope, and vegetation tags. In practice (see Section 5.2), neither the agent alone nor the embedder alone yields sufficiently accurate classification across these attributes. We therefore complement embedding-based retrieval with candidate reranking before using retrieved labels for query estimation.

3.2.2 Memory-Augmented Multi-Agent Reporting

We formulate downstream wildfire understanding as an *agentic VLM reporting* problem combining query observations with retrieved physics-labeled references. Given a query video, the system produces a structured report over fire-behavior fields such as spread risk, burn extent, vegetation condition, and severe-fire indicators. Rather than relying on a single model, we separate report generation into four stages: memory retrieval, multi-agent reporting, disagreement adjudication, and report rendering Lewis et al. (2020); Wu et al. (2024a); Chan et al. (2023).

For a query video q, we retrieve the top-k most relevant episodes from the memory split. Retrieved episodes supply environmental driver tags, physical summaries, and downstream labels. For each binary report field t, the retrieved examples define a score-weighted memory consensus for adjudication: $C_t(y \mid q) = \frac{\sum_{i=1}^{k} w_i \mathbf{1}[\ell_t(r_i)=y]}{\sum_{i=1}^{k} w_i}$, where r_i is a retrieved episode, w_i is its retrieval score, and $\ell_t(r_i)$ is its label for field t. This consensus summarizes the label favored by similar memory episodes and their score-weighted agreement; it is a reference prior rather than a calibrated probability for the query.

The reporting stage uses three agents with complementary evidence. The visual reporter receives only sampled query frames and focuses on directly observable cues. The lookup reporter receives the query frames and a train-derived lookup prior from the top retrieved tag bundle. The combined-memory reporter additionally receives the retrieved episode's simulator summaries. This decomposition keeps

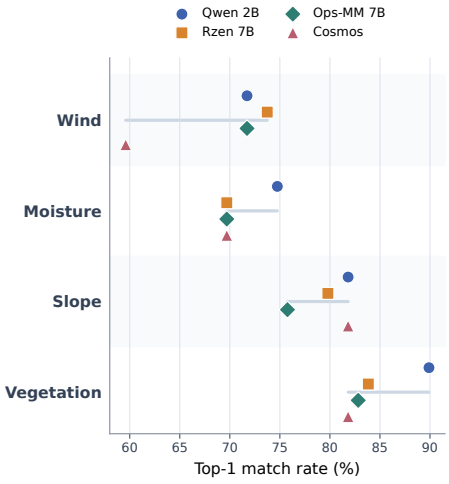


Figure 4: We show the effects of different multimodal embedding models on tag retrieval accuracy. From experiments, Qwen3-2B-VL gives the best retrieval accuracy in most cases which leads to our choice of it being our embedding model.

Table 1: Reporting-component ablation on six simulator-derived fields. Oracle Upper Bound counts a field as correct when any constituent reporter predicts its label correctly; it is not an executable selection rule.

Method	**Downstream Task Accuracy**
No Judge Agent (Majority Vote)	72.6%
No Lookup Memory Agent	74.3%
No Combined Memory Agent	74.1%
No Direct Visual Agent	73.6%
Only Direct Visual Agent	66.8%
Only Combined Memory Agent	72.2%
Only Lookup Memory Agent	67.8%
Full System	77.3%
Oracle Upper Bound	82.3%

Table 2: Multimodal versus text-only memory on held-out generated episodes. 4-Tags requires all four environmental driver tags to match.

Method	**4-Tags**	**Wind**	**Moist.**	**Slope**	**Veg.**
Caption video then query memory	16.2%	62.6%	51.5%	65.7%	68.7%
Video generation prompt as text memory	17.2%	68.7%	60.6%	66.7%	62.6%
Full Memory	**51.5%**	**74.7%**	**83.8%**	**82.8%**	**89.9%**

disagreements across evidence sources explicit across the reporting pipeline, in line with multi-agent LLM/VLM workflow designs Wu et al. (2024a); Chan et al. (2023).

Reporter agreement determines which fields require adjudication. If all reporters agree on a field, the shared prediction is copied into the final report. If they disagree, only that disputed field is routed to a judge agent Zheng et al. (2023); Wang et al. (2022a), which sees the query frames, reporter predictions, rationales, top-k memory consensus, and retrieved metadata summaries. The inputs and routing protocol are detailed in Appendix C.

Finally, the resolved predictions are compiled into an HTML report. The structured output records sampled visual evidence, predicted labels, routing status, reporter rationales, and retrieved memory summaries. A separate narrative stage summarizes the predicted tags and report fields. These records support inspection of the inference process; narrative response suggestions are illustrative outputs, not validated operational guidance.

4 Ablation Study

We analyze the main design choices behind the simulation-grounded VLM framework: the embedding model used for retrieval, the reranking step, the need for video memory rather than text-only memory, and the contribution of each reporting agent. Experiments in this section and Sec. 5 use 500 disjoint locations: 400 videos populate the memory bank and 100 videos are held out for evaluation. The distribution of downstream tasks and physics-labels are provided in Appendix A. We also refer readers to provided supplementary materials for more visual examples. Local experiments use an NVIDIA H200 GPU; hosted VLM evaluations use model APIs.



Figure 5: We show generated video frames with different simulation tags (2 video clips for each tag).

Table 3: Driver-tag accuracy on held-out generated episodes. Rows require all four, at least three, or at least two correct tags. Bold and underlined values mark the best and second-best results within each group.

	VLM Direct Query				Memory Retrieval	
	Gemini 2.5	Claude 4	GPT-5	GPT-4.1	w/o GPT-5 Rerank	w/ GPT-5 Rerank
All Tags	**22.6%**	22.2%	21.2%	16.2%	45.5%	**51.5%**
Three Tags	56.6%	63.4%	**63.6%**	49.5%	78.8%	**82.8%**
Two Tags	90.9%	**91.4%**	90.9%	85.9%	94.9%	**99.0%**

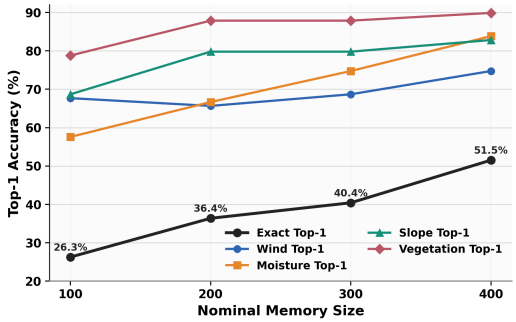


Figure 7: Larger memory improves accuracy for individual tags and all four tags.

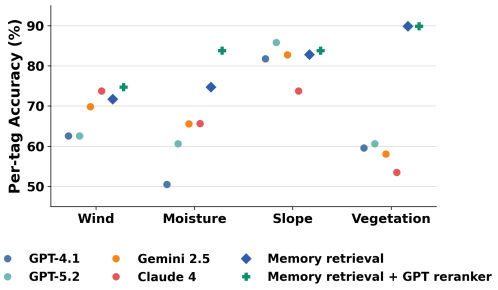


Figure 8: Reranked memory leads in most per-tag accuracy comparisons.

Multimodal embedding model. We compare Qwen3-2B-VL Li et al. (2026) with Cosmos-Embed1 NVIDIA et al. (2025), Rzen-v2-7B Jian et al. (2025), and Ops-MM-7B OpenSearch-AI. Figure 4 shows that Qwen3-2B-VL gives the strongest tag-retrieval accuracy in most cases, so we use it for the final memory bank.

Reranking retrieved candidates. GPT-5 reranking improves exact four-tag accuracy from 45.5% to 51.5% (Table 3). Increasing k from 1 to 4 raises four-tag hit rate from 51.52% to 69.70%, measuring retrieval coverage rather than final prediction accuracy. Reranker comparisons, latency, and retrieval settings appear in Appendices C, H.3, and H.4.

Importance of Video Memory. Table 2 compares full multimodal memory against two text-only variants: direct query-to-text matching and caption-then-text retrieval. Full memory improves exact four-tag accuracy from 16–17% to 51.5% and is stronger for every individual tag, supporting the use of stored video exemplars rather than prompt-only memory.

Multi-Agent Effects on Downstream Task Accuracy. The full system achieves 77.3% on six simulator-derived fields, versus 72.2% for the combined-memory reporter alone and 72.6% for majority vote (Table 1). The 82.3% oracle marks fields correct when any reporter is correct. These ablations compare the specified components, not context- or compute-matched agent configurations.

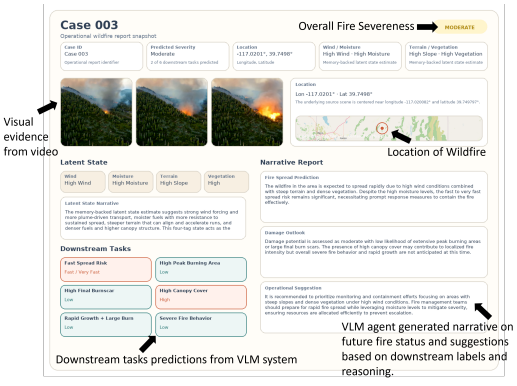


Figure 6: Illustrative reports show predictions, visual evidence, location, and response suggestions; these outputs are not operationally validated.

Table 4: Accuracy on six simulator-derived report fields for held-out generated episodes. Higher is better. Bold and underlined values mark the best and second-best results, respectively.

Task	Direct Query GPT	Multi-Agent + Combined Memory	ForestFireVLM	Qwen2-Wildfire-2B
Fast Spread Risk	<u>55.7%</u>	**75.3%**	58.8%	43.3%
High Peak Burning Area	<u>63.9%</u>	**77.3%**	54.6%	59.8%
High Final Burnscar	<u>64.9%</u>	**80.4%**	55.7%	60.8%
High Canopy Cover	57.7%	**73.2%**	<u>67.0%</u>	47.4%
Rapid Growth + Large Burn	<u>64.9%</u>	**79.4%**	55.7%	60.8%
Severe Fire Behavior	<u>66.0%</u>	**78.4%**	56.7%	59.8%

5 Validation and Results

We assess proxy–video alignment, environmental-driver prediction, and six-field reporting on generated episodes with known simulator targets, then test generator changes and real-UAV smoke/fire detection and visible fire-area assessment.

5.1 Simulation-to-Observation Results

Figures 2 and 5 show simulation-to-video construction and tag-conditioned appearances; additional examples appear in Figure 10 and the supplement. Against mismatched proxies, correct-pair SSIM and LPIPS win rates are 76.6% and 99.1%, with mean smoke-plume-axis and growth-front-axis errors of 7.6° and 35.8° (Appendix D). GPT-5.6-luna rates agreement across fire-front location, fuel distribution, burned area, and terrain at 75.00/100 (Appendix D.1). These results support partial visual-cue alignment, not preservation of physical spread rates or burned-area trajectories.

5.2 Memory-Based Environmental Driver Prediction

Table 3 compares exact four-tag accuracy and the proportions with at least three or two correct tags on held-out generated videos. Reranked memory reaches 51.5%, versus 22.6% for the strongest query-only VLM. With 16 labeled reference videos covering all tag combinations, the strongest example-augmented baseline achieves 27.27%; query simulator metadata remains withheld (Appendix G.5).

5.3 Memory Growth Effects

Increasing the memory bank from 100 to 400 episodes improves exact four-tag accuracy from 26.3% to 51.5% (Figure 7). This supports the training-free memory design: adding simulator-grounded episodes broadens the available reference coverage without updating VLM weights.

5.4 Downstream Fire-Behavior Prediction

With query simulator quantities withheld, Table 4 compares six simulator-derived report fields (Appendix A) against direct GPT querying, ForestFireVLM Seidel et al. (2025), and Qwen2-Wildfire-2B Du et al. (2025). Our system leads on all six fields, including Fast Spread Risk (75.3% versus 55.7% for direct GPT). Predictions, visual evidence, retrieved summaries, tags, and rationales support the structured report and separate narrative (Figure 6 and supplement).

5.5 Cross-Generator and Real-UAV Evaluation

Cross-generator evaluation. To test robustness to changes in the visual synthesis backbone, we replace VACE with LTX-2.3 + AVControl while retaining the simulation and proxy construction, and test both query–memory directions. Our method achieves 41.00% for LTX queries against VACE memory and 44.44% in the reverse direction, compared with 28.00% and 27.27% for the strongest direct VLM baselines (Table 5a).

Table 5: Generalization tests. (a) Cross-generator accuracy (%) on generated episodes; LTX denotes LTX-2.3 + AVControl. (b) Real-UAV smoke/fire detection and visible fire-area assessment: Boreal accuracy is a percentage; other scores are on a 0–1 scale. Best baseline is selected separately for each metric, not one common model. Bold includes ties. Full real-UAV results appear in Appendix L.

(a) Cross-generator evaluation

Query → Memory	Ours	GPT	Claude	Gemini
LTX → VACE	**41.00**	19.00	24.00	28.00
VACE → LTX	**44.44**	27.27	23.23	26.26

(b) External UAV evaluation

Dataset	Metric	Ours	Best baseline
Boreal Forest Fire	Accuracy (%)	**100.0**	**100.0**
FireMan-UAV-RGBT	Balanced accuracy	**0.9625**	0.8625
	Accuracy	**0.9423**	0.9231
FLAME	Mean AP	0.950	**0.952**
	Largest-fire Hit@1	**0.850**	**0.850**
	Fire-area NDCG	**0.984**	0.973

The advantage therefore persists when queries and memory use different video generators, although accuracy is below the within-generator result.

External UAV footage. We evaluate real footage using the annotations available in three public UAV datasets: smoke detection on Boreal Forest Fire Pesonen et al. (2025), binary fire presence on FireMan-UAV-RGBT Kularatne et al. (2024), and segmentation-derived visible fire-area assessment on FLAME Shamsoshoara et al. (2021). These tests complement the controlled physical-label experiments by assessing observable aspects of wildfire monitoring. Table 5b shows the highest balanced accuracy on FireMan-UAV-RGBT and NDCG on FLAME; our method ties the best FLAME Hit@1 but not mean AP. Several methods reach 100% on Boreal Forest Fire. The annotations used here do not jointly specify all four environmental drivers, so these results assess real-UAV perception rather than physical-driver recovery. Full results and evaluation descriptions appear in Appendix L.

6 Discussion

Relationship to Code World Model. Code World Model (Chen et al., 2026b) and our framework combine explicit state evolution, spatial proxies, and generative visualization. Code World Model uses a coding agent to maintain persistent state and revise mechanisms for interaction. We instead map prescribed wildfire trajectories to aligned aerial-view episodes through a fixed simulator-to-proxy converter, using these labeled episodes as multimodal memory for interpreting query videos.

Fixed conversion versus scene authoring. GPT-6 Astra can author Blender scenes (**?**) and could help develop our converter. New episodes, however, change inputs rather than code. The proxy prioritizes simulator alignment and low rendering overhead, leaving appearance detail to video generation. Reuse avoids per-episode scene-authoring calls and token costs; other inference costs remain. Like Code World Model, we separate high-level reasoning from repeated execution: the distinction concerns where reasoning is needed, not a measured speed or cost advantage.

7 Limitations and Future Work

Limitations. Real-incident environmental-driver estimation still requires validation against synchronized measurements: our real-UAV tests assess smoke/fire detection and visible fire area, whereas physical-label targets come from simulation. The fixed Blender mapping is domain-specific; new simulator fields or scene types may require extending its rules and assets. Spatial correspondence does not make billboard smoke a 3D physical solution or ensure that generated videos preserve proxy dynamics. Metadata-conditioned prompts may introduce tag-specific cues, and footage may omit sensor noise, camera motion, long-horizon evolution, or rare extremes. The coarse simulator and tags omit coupled fire–atmosphere dynamics, spotting, suppression, and changing weather. Retrieval depends on memory coverage, reporter agreement does not guarantee correctness, and reports require expert validation before operational use.

Future Work. Future work should calibrate generated observations against physically annotated real footage, expand scenario coverage, and involve wildfire experts. Further tests should isolate prompt conditioning, compare reporting with matched evidence and compute, and measure stage-wise costs against agent-authored proxy alternatives. Integration with hazard-adaptation and UAV planning frameworks Xie et al. (2025); Hyun et al. (2025); Webb et al. (2026) requires task-specific validation.

8 Conclusion

We presented a simulation-grounded agentic VLM framework for wildfire monitoring and reporting. A fully automatic Blender conversion expresses 2D wildfire states as lightweight, simulation-aligned proxies; controlled video generation produces labeled episodes for multimodal memory. Retrieved references and complementary reporters support environmental-driver estimation and structured reporting without updating VLM weights. Controlled, cross-generator, and real-UAV perceptual tests support this use of simulation experience under scarce physical annotations. Real-world physical calibration and expert validation remain important next steps.

Ethics Statement

This work supports wildfire-monitoring research using synthetic episodes and existing UAV datasets. Generated videos should be clearly identified as synthetic, and reports require expert review and real-world validation before operational use. Limited memory coverage and simulation-to-real discrepancies may produce inaccurate predictions across environments. Data reuse and dissemination should respect applicable licenses, attribution requirements, and privacy protections.

Reproducibility Statement

The main text describes the pipeline, evaluation splits, and ablations. Appendices B and C document simulation, rendering, generation, and retrieval settings. Appendix A provides label definitions, while Appendices E–J document model roles and prompts. Additional alignment and generalization evaluations appear in Appendices D and L. We plan to release the code upon acceptance.

LLM Usage Statement

We used LLM to assist with drafting and revising manuscript sections. We also used LLMs to obtain critical feedback on the methodology and evaluation, and to assist with implementation through code generation and debugging. These research and writing assistance roles are distinct from the LLM/VLM components used within the proposed framework and experiments, whose roles and prompts are documented in the main text and appendix. The authors are responsible for reviewing and validating LLM-assisted text and code and take full responsibility for all methodological decisions, references, reported results, and conclusions.

References

Hassan Abu Alhaija, Jose Alvarez, Maciej Bala, Tiffany Cai, Tianshi Cao, Liz Cha, Joshua Chen, Mike Chen, Francesco Ferroni, Sanja Fidler, et al. Cosmos-transfer1: Conditional world generation with adaptive multimodal control. *arXiv preprint arXiv:2503.14492*, 2025.

Hal E Anderson. Aids to determining fuel models for estimating fire behavior. *The Bark Beetles, Fuels, and Fire Bibliography*, pp. 143, 1982.

Aydin Ayanzadeh, Prakhar Dixit, Sadia Kamal, and Milton Halem. Wildfirevlm: Ai-powered analysis for early wildfire detection and risk assessment using satellite imagery. *arXiv preprint arXiv:2602.13305*, 2026.

Rafael Bailon-Ruiz, Arthur Bit-Monnot, and Simon Lacroix. Real-time wildfire monitoring with a fleet of uavs. *Robotics and Autonomous Systems*, 152:104071, 2022.

Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7915–7925, 2022. URL https://arxiv.org/abs/2203.12691.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Chuhao Chen, Peter Wonka, Chaoyang Wang, Chen Wang, Qiao Feng, Sergey Tulyakov, and Lingjie Liu. Physstream: Streaming physics-grounded video generation with structured scene memory and fine-grained motion control. *arXiv preprint arXiv:2609.17521*, 2026a.

Duowen Chen, Zhiqiang Lao, Yu Guo, and Heather Yu. Fluid composer: Fluid detail composition and rendering using video diffusion models. In *Computer Graphics Forum*, pp. e70300. Wiley Online Library, 2025.

Yiwen Chen, Guosheng Lin, and Chi Zhang. Code world model: Coding agent as world brain. *arXiv preprint arXiv:2608.25927*, 2026b.

Michael Doyle, Michael Threet, Marissa Dotter, Chris Kempis, Alex Tapley, and Tim Welsh. SimFire, July 2024. URL https://github.com/mitrefireline/simfire.

Shiyan Du, Jiacheng Li, and Masato Noto. Wildfire scene recognition based on qwen2-wildfire. In *Proceedings of the 2025 8th international conference on software engineering and information management*, pp. 254–262, 2025.

Miguel Esparza, Archit Gupta, Kai Yin, Yiming Xiao, and Ali Mostafavi. Automated wildfire damage assessment from multi view ground level imagery via vision language models. *arXiv preprint arXiv:2509.01895*, 2025.

Xijian Fan, Fan Lei, and Kun Yang. Real-time detection of smoke and fire in the wild using unmanned aerial vehicle remote sensing imagery. *Forests*, 16(2):201, 2025.

Mark Arnold Finney. *FARSITE, Fire Area Simulator–model development and evaluation*. Number 4. The Station, 1998.

Honghao Fu, Miao Xu, Yiwei Wang, Dailing Zhang, Liu Jun, and Yujun Cai. Videostir: Understanding long videos via spatio-temporally structured and intent-aware rag. *arXiv preprint arXiv:2604.05418*, 2026.

Bao Tran Gia, Khiem Le, Tien Do, Tien-Dung Mai, Thanh Duc Ngo, Duy-Dinh Le, and Shin'ichi Satoh. Vrag: Retrieval-augmented video question answering for long-form videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3728–3737, 2025.

Louis Giglio, Wilfrid Schroeder, and Christopher O Justice. The collection 6 modis active fire detection algorithm and fire products. *Remote sensing of environment*, 178:31–41, 2016.

Torsten Hädrich, Daniel T Banuti, Wojtek Pałubicki, Sören Pirk, and Dominik L Michels. Fire in paradise: Mesoscale simulation of wildfires. *ACM Transactions on Graphics (TOG)*, 40(4):1–15, 2021.

Jonathan Hyun, Nicholas R Waytowich, and Boyuan Chen. Crew-wildfire: Benchmarking agentic multi-agent collaborations at scale. *arXiv preprint arXiv:2507.05178*, 2025.

Soyeong Jeong, Kangsan Kim, Jinheon Baek, and Sung Ju Hwang. Videorag: Retrieval-augmented generation over video corpus. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21278–21298, 2025.

Weijian Jian, Yajun Zhang, Dawei Liang, Chunyu Xie, Yixiao He, Dawei Leng, and Yuhui Yin. Rzenembed: Towards comprehensive multimodal retrieval. *arXiv preprint arXiv:2510.27350*, 2025.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

CO Justice, Louis Giglio, S Korontzi, J Owens, JT Morisette, D Roy, J Descloitres, Samuel Alleaume, F Petitcolin, and Y Kaufman. The modis fire products. *Remote sensing of Environment*, 83(1-2): 244–262, 2002.

Andrzej Kokosza, Helge Wrede, Daniel Gonzalez Esparza, Milosz Makowski, Daoming Liu, Dominik L Michels, Soren Pirk, and Wojtek Palubicki. Scintilla: Simulating combustible vegetation for wildfires. *ACM Transactions on Graphics (TOG)*, 43(4):1–21, 2024.

Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 27831–27840, 2024.

SDMW Kularatne, Constantino Álvarez Casado, Janne Rajala, Tuomo Hänninen, Miguel Bordallo López, and Le Nguyen. Fireman-uav-rgbt: A novel uav-based rgb-thermal video dataset for the detection of wildfires in the finnish forests. In *2024 IEEE 29th international conference on emerging technologies and factory automation (ETFA)*, pp. 1–8. IEEE, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36: 28541–28564, 2023.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv*, 2026.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. Physgen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision*, pp. 360–378. Springer, 2024.

Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508: 293–304, 2022.

Yongdong Luo, Xiawu Zheng, Guilin Li, Shukang Yin, Haojia Lin, Chaoyou Fu, Jinfa Huang, Jiayi Ji, Fei Chao, Jiebo Luo, and Rongrong Ji. Video-RAG: Visually-aligned retrieval-augmented long video comprehension. In *Advances in Neural Information Processing Systems*, 2025.

Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *Proceedings of the 30th ACM international conference on multimedia*, pp. 638–647, 2022.

Vivien Mallet, David E Keyes, and FE Fendell. Modeling wildland fire propagation with level set methods. *Computers & Mathematics with Applications*, 57(7):1089–1101, 2009.

Jan Mandel, Jonathan D Beezley, and Adam K Kochanski. Coupled atmosphere-wildland fire modeling with wrf 3.3 and sfire 2011. *Geoscientific Model Development*, 4(3):591–610, 2011.

NVIDIA, Francesco Ferroni, Prithvijit Chattopadhyay, Greg Heinrich, Mike Ranzinger, Roberto Amoroso, Alice Luo, Andrew Wang, and Ming-Yu Liu. Cosmos-embed1: A joint video-text embedder for physical ai, 2025. URL `https://research.nvidia.com/labs/dir/cosmos-embed1/`.

OpenSearch-AI. Ops-mm-embedding-v1-7b. `https://huggingface.co/OpenSearch-AI/Ops-MM-embedding-v1-7B`. Hugging Face model card, accessed 2026-04-16.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: towards llms as operating systems. 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Julius Pesonen, Anna-Maria Raita-Hakola, Jukka Joutsalainen, Teemu Hakala, Waleed Akhtar, Niko Koivumäki, Lauri Markelin, Juha Suomalainen, Raquel Alves de Oliveira, Ilkka Pölönen, et al. Boreal forest fire: Uav-collected wildfire detection and smoke segmentation dataset. *Scientific Data*, 12(1):1419, 2025.

Tianye Qi, Weihao Li, and Nick Barnes. Smokebench: Evaluating multimodal large language models for wildfire smoke detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1043–1053, 2026.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

Mona Raoufi, Akbar Telikani, Tieling Zhang, and Jun Shen. Fire front path planning and tracking control of uncrewed aerial vehicles using deep reinforcement learning. *Robotics and Autonomous Systems*, 193:105076, 2025.

Jonathan Roberts, Kai Han, Neil Houlsby, and Samuel Albanie. Scifibench: Benchmarking large multimodal models for scientific figure interpretation. *Advances in Neural Information Processing Systems*, 37:18695–18728, 2024.

Richard C Rothermel. *A mathematical model for predicting fire spread in wildland fuels*, volume 115. Intermountain Forest & Range Experiment Station, Forest Service, US . . . , 1972.

Ashen Sandeep, Sithum Jayarathna, Sunera Sandaruwan, Venura Samarappuli, Dulani Meedeniya, and Charith Perera. Context-aware multi-agent architecture for wildfire insights. *Sensors*, 26(3): 1070, 2026.

Wilfrid Schroeder, Patricia Oliva, Louis Giglio, and Ivan A Csiszar. The new viirs 375 m active fire detection data product: Algorithm description and initial assessment. *Remote Sensing of Environment*, 143:85–96, 2014.

Leon Seidel, Simon Gehringer, Tobias Raczok, Sven-Nicolas Ivens, Bernd Eckardt, and Martin Maerz. Advancing early wildfire detection: integration of vision language models with unmanned aerial vehicle remote sensing for enhanced situational awareness. *Drones*, 9(5):347, 2025.

Alireza Shamsoshoara, Fatemeh Afghah, Abolfazl Razi, Liming Zheng, Peter Z Fulé, and Erik Blasch. Aerial imagery pile burn detection using deep learning: The flame dataset. *Computer Networks*, 193:108001, 2021.

Hao Tang, Darren Key, and Kevin Ellis. Worldcoder, a model-based llm agent: Building world models by writing code and interacting with the environment. *Advances in Neural Information Processing Systems*, 37:70148–70212, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022a.

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*, 2022b.

Chris Webb, Mobin Habibpour, Mayamin Hamid Raha, Ali Reza Tavakkoli, Janice Coen, and Fatemeh Afghah. Fire-vlm: A vision-language-driven reinforcement learning framework for uav wildfire tracking in a physics-grounded fire digital twin. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1493–1502, 2026.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*, 2024a.

Shanjunxia Wu, Yuming Qiao, Sen He, Jiahao Zhou, Zhi Wang, Xin Li, and Fei Wang. Fireclip: Enhancing forest fire detection with multimodal prompt tuning and vision-language understanding. *Fire*, 8(6):237, 2025.

Siwei Wu, Yizhi Li, Kang Zhu, Ge Zhang, Yiming Liang, Kaijing Ma, Chenghao Xiao, Haoran Zhang, Bohao Yang, Wenhu Chen, et al. Scimmir: Benchmarking scientific multi-modal information retrieval. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12560–12574, 2024b.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.

Yangxinyu Xie, Bowen Jiang, Tanwi Mallick, Joshua Bergerson, John K Hutchison, Duane R Verner, Jordan Branham, M Ross Alexander, Robert B Ross, Yan Feng, et al. Marsha: multi-agent rag system for hazard adaptation. *npj Climate Action*, 4(1):70, 2025.

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. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 6787–6800, 2021.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Nianbo Zeng, Haowen Hou, F Richard Yu, Si Shi, and Ying Tiffany He. Scenerag: Scene-level retrieval-augmented generation for video understanding. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12122–12126. IEEE, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

APPENDIX CONTENTS

1 Introduction **1**

2 Related Work **2**

3 Method **3**
3.1 Simulation-Grounded Visual Evidence Construction . . . 3
3.2 Memory-Augmented Agentic VLM Monitoring and Reporting . . . 4

4 Ablation Study **6**

5 Validation and Results **8**
5.1 Simulation-to-Observation Results . . . 8
5.2 Memory-Based Environmental Driver Prediction . . . 8
5.3 Memory Growth Effects . . . 8
5.4 Downstream Fire-Behavior Prediction . . . 8
5.5 Cross-Generator and Real-UAV Evaluation . . . 8

6 Discussion **9**

7 Limitations and Future Work **9**

8 Conclusion **9**

A Downstream Task and Physics Labels **17**

B Wildfire Models **17**
B.1 Rothermel Surface Fire Spread Model . . . 17
B.2 Anderson 13 Fuel Profiles . . . 20

C Implementation Details **21**

D Simulation to Observation Alignment Numerical Metrics **22**
D.1 VLM-Based Observable Cue Agreement . . . 23

E Model-to-Prompt Mapping **24**

F Video Generation Prompt **24**
F.1 Video Generation Prompt From Simulation Data . . . 24
F.2 Prompt Refinement for Video Generation . . . 26

G Direct Four-Tag Evaluation Prompts **27**
G.1 OpenAI Query-Only Tag Prompt . . . 27
G.2 Gemini / Claude External Query-Only Prompt Wrapper . . . 28
G.3 Local VLM Query-Only Prompt for Qwen2-Wildfire-2B and ForestFireVLM . . . 28
G.4 OpenAI Candidate-Reranking Prompt . . . 29
G.5 Baselines with Labeled Reference Videos . . . 30

H Qwen-Embedding Retrieval and Hybrid Rerank Prompts **30**
H.1 Our Reranking Method . . . 30
H.2 Optional Qwen3-VL Reranker Instruction . . . 30
H.3 Reranker Accuracy and Latency . . . 31
H.4 Retrieval-Size Ablation . . . 31

I Downstream Six-Task Specialist Prompts **31**
I.1 Downstream Task List . . . 32
I.2 Visual Report Agent . . . 32
I.3 Lookup- and Combined-Memory Specialist Shared System Prompt . . . 33
I.4 Lookup-Memory Specialist Context . . . 33
I.5 Combined-Memory Agent Context . . . 34
I.6 Specialist Response Schema . . . 35
I.7 Downstream Task Prediction Prompt for GPT, Qwen2-Wildfire-2B and ForestFireVLM . . . 36
I.8 Multi-Agent Judge Prompts . . . 37

J HTML Report Generation Prompt **41**
J.1 System Prompt for Report Generation 41
J.2 User Prompt Template for Report Generation 41
J.3 Structured Report Example 42

K Structured Output Contracts **42**
K.1 Four-Tag Classification Schema 42
K.2 Candidate-Reranking Schema 43
K.3 Downstream Specialist Agent / Judge Schema 43

L Cross-Generator and External UAV Evaluation **43**
L.1 Cross-Generator Evaluation 43
L.2 External UAV Evaluation 43

M Additional Visual Results **44**

A Downstream Task and Physics Labels

Downstream Task and Label Definitions We evaluate downstream wildfire understanding using six binary report fields. Each field is defined from simulator-derived quantities associated with an episode. Let $v_{\max}$ denote the maximum radial spread rate in meters per second, A_{peak} the maximum burning area during the episode in square meters, A_{final} the final burn-scar area in square meters, and c the mean canopy-cover percentage. For a held-out query, these quantities define evaluation targets and are withheld from inference. Retrieved memory entries may contain the corresponding quantities for their own reference episodes. The fields describe episode-level conditions and outcomes, not validated operational forecasts. Definitions and label construction follow Section 3.2.2; the four environmental-tag thresholds are given in Appendix C.

Task and Label Distributions We report the class distributions of the four simulation-derived physical labels and the six downstream tasks used in our evaluation. We also include three simple reference baselines: *Majority*, which always predicts the most frequent class in the evaluation set; *Random*, which predicts uniformly at random; and *Stratified*, which samples predictions according to the empirical class prior.

Task	Class balance	Majority	Random	Stratified
Wind	high: 49.5%, low: 50.5%	50.5%	50.0%	50.0%
Moisture	high: 49.5%, low: 50.5%	50.5%	50.0%	50.0%
Slope	high: 44.4%, low: 55.6%	55.6%	50.0%	50.6%
Vegetation	high: 47.5%, low: 52.5%	52.5%	50.0%	50.1%

Task	Class balance	Majority	Random	Stratified
Fast Spread Risk	fast: 56.7%, slow/moderate: 43.3%	56.7%	50.0%	50.9%
High Peak Burning Area	high: 40.2%, low: 59.8%	59.8%	50.0%	51.9%
High Final Burnscar	high: 39.2%, low: 60.8%	60.8%	50.0%	52.3%
High Canopy Cover	high: 44.3%, low: 55.7%	55.7%	50.0%	50.6%
Rapid Growth + Large Burn	high: 39.2%, low: 60.8%	60.8%	50.0%	52.3%
Severe Fire Behavior	high: 40.2%, low: 59.8%	59.8%	50.0%	51.9%

B Wildfire Models

For completeness, in this section, we list the fire simulation model we used for generating 2D fire simulations together with the definition of the Anderson 13 Fuel Profile with corresponding billboard assets.

B.1 Rothermel Surface Fire Spread Model

For completeness, we provide the Rothermel surface fire spread model used for 2D simulation which is the backend model that SimFire uses. It is a semi-empirical model that predicts the forward rate of spread of a surface fire from local fuel, moisture, wind, and terrain conditions Rothermel (1972). In our simulations, this model provides a directional rate of spread from an actively burning grid cell to each candidate neighboring cell. All quantities follow the SimFire implementation.

Let cell i be an actively burning cell and cell j be a candidate neighboring cell. The directional rate of spread from i to j is

$$R_{i\to j} = \max\left(0,\ \frac{I_R \xi\,(1+\phi_w+\phi_s)}{\rho_b \epsilon Q_{\text{ig}}}\right), \tag{1}$$

where $R_{i\to j}$ is the directional rate of spread from cell i to cell j, I_R is the reaction intensity, ξ is the propagating flux ratio, ϕ_w and ϕ_s are the wind and slope correction factors, ρ_b is the oven-dry bulk density, ϵ is the effective heating number, and Q_{ig} is the heat of preignition.

Table 8: Binary simulator-derived report-field definitions. Here $v_{\max}$ is the maximum radial spread rate, A_{peak} is the peak burning area, A_{final} is the final burn-scar area, and c is the mean canopy-cover percentage.

Task	Meaning	Positive Label Definition
Fast Spread Risk	The fire spreads at a fast or very fast rate.	Positive if $v_{\max} \geq 0.30$ m/s. We use four spread classes: slow for $v_{\max} < 0.15$, moderate for $0.15 \leq v_{\max} < 0.30$, fast for $0.30 \leq v_{\max} < 0.45$, and very fast for $v_{\max} \geq 0.45$.
High Peak Burning Area	The episode reaches a large active burning area.	Positive if $A_{\text{peak}} \geq 90{,}000$ m^2.
High Final Burn Scar	The episode leaves a large final burned area.	Positive if $A_{\text{final}} \geq 300{,}000$ m^2.
High Canopy Cover	The episode occurs in a relatively high-canopy environment.	Positive if c is at or above the median canopy cover in the training set. In our data, this threshold is 2.61%.
Rapid Growth + Large Burn	The episode combines rapid fire spread with a large final burn scar.	Positive if $v_{\max} \geq 0.30$ m/s and $A_{\text{final}} \geq 300{,}000$ m^2.
Severe Fire Behavior	The episode satisfies multiple indicators of severe fire behavior.	Positive if at least two of the following three conditions hold: $v_{\max} \geq 0.30$ m/s, $A_{\text{peak}} \geq 90{,}000$ m^2, and $A_{\text{final}} \geq 300{,}000$ m^2.

For each target cell, It first computes the fuel packing and damping terms. Let w_0 denote oven-dry fuel load, δ fuel-bed depth, M_x moisture of extinction, σ surface-area-to-volume ratio, h low heat content, S_T total mineral content, S_e effective mineral content, ρ_p oven-dry particle density, and M_f environmental fuel moisture. The net fuel load, bulk density, packing ratio, and optimal packing ratio are

$$w_n = w_0(1 - S_T), \qquad \rho_b = \frac{w_0}{\delta}, \qquad \beta = \frac{\rho_b}{\rho_p}, \qquad \beta_{\text{op}} = 3.348\sigma^{-0.8189}. \tag{2}$$

Here, w_n is the net fuel load after discounting the total mineral content S_T, ρ_b is the oven-dry bulk density of the fuel bed, β is the fuel packing ratio, and β_{op} is the optimum packing ratio determined by the surface-area-to-volume ratio σ.

The moisture and mineral damping coefficients are

$$r_M = \min\left(\frac{M_f}{M_x}, 1\right), \qquad \eta_M = 1 - 2.59r_M + 5.11r_M^2 - 3.52r_M^3, \tag{3}$$

$$\eta_S = \min\left(0.174S_e^{-0.19}, 1\right). \tag{4}$$

Here, r_M is the moisture ratio between the environmental fuel moisture M_f and the moisture of extinction M_x, η_M is the moisture damping coefficient, and η_S is the mineral damping coefficient determined by the effective mineral content S_e.

The maximum reaction velocity and reaction velocity are

$$\Gamma'_{\max} = \frac{\sigma^{1.5}}{495 + 0.0594\sigma^{1.5}}, \qquad A = 133\sigma^{-0.7913}, \tag{5}$$

$$\Gamma' = \Gamma'_{\max}\left(\frac{\beta}{\beta_{\text{op}}}\right)^A \exp\left[A\left(1 - \frac{\beta}{\beta_{\text{op}}}\right)\right]. \tag{6}$$

Here, $\Gamma'_{\max}$ is the maximum reaction velocity, A is an empirical exponent controlled by the surface-area-to-volume ratio σ, and Γ' is the reaction velocity after accounting for the ratio between the actual packing ratio β and the optimum packing ratio β_{op}.

The reaction intensity is then

$$I_R = \Gamma' w_n h \eta_M \eta_S. \tag{7}$$

Here, the reaction intensity I_R combines the reaction velocity Γ', net fuel load w_n, low heat content h, moisture damping coefficient η_M, and mineral damping coefficient η_S.

The propagating flux ratio is

$$\xi = \frac{\exp\left[(0.792 + 0.681\sigma^{1/2})(\beta + 0.1)\right]}{192 + 0.2595\sigma}. \tag{8}$$

Here, ξ is the propagating flux ratio, which depends on the surface-area-to-volume ratio σ and the packing ratio β.

Wind is incorporated by projecting the wind velocity onto the direction from the burning cell to the candidate cell. Let θ_{ij} be the travel direction from cell i to cell j, and let θ_U be the wind direction in mathematical coordinates. SimFire converts wind direction from degrees clockwise from north and uses

$$U_{\parallel} = \max\left(0,\, U\cos(\theta_U - \theta_{ij})\right), \tag{9}$$

where U is the wind speed and $U_{\parallel}$ is its nonnegative component projected onto the candidate spread direction. Thus, only the wind component aligned with the candidate spread direction increases the spread rate. The wind factor is

$$\phi_w = cU_{\parallel}^{b}\left(\frac{\beta}{\beta_{\mathrm{op}}}\right)^{-e}, \tag{10}$$

where ϕ_w is the wind correction factor, and c, b, and e are empirical coefficients determined by the fuel surface-area-to-volume ratio and packing properties, with

$$c = 7.47\exp(-0.133\sigma^{0.55}), \qquad b = 0.02526\sigma^{0.54}, \qquad e = 0.715\exp(-3.59\times 10^{-4}\sigma). \tag{11}$$

Here, c, b, and e are functions of σ, and they control how strongly the projected wind speed $U_{\parallel}$ modifies the spread rate.

Terrain slope is also made directional by projecting the local slope onto the travel direction. Let s_{mag} be the slope magnitude and θ_s the direction of steepest ascent. SimFire computes

$$s_{\parallel} = -s_{\mathrm{mag}}\cos(\theta_s + \theta_{ij}), \tag{12}$$

where $s_{\parallel}$ is the signed slope component along the candidate spread direction. It then uses a signed slope correction

$$\phi_s = 5.275\beta^{-0.3}\,\mathrm{sgn}(s_{\parallel})s_{\parallel}^{2}, \tag{13}$$

where ϕ_s is the slope correction factor, $\mathrm{sgn}(s_{\parallel}) = 1$ for uphill spread and -1 otherwise. This makes uphill spread faster and downhill spread slower in the directional grid update.

Finally, the effective heating number and heat of preignition are

$$\epsilon = \exp\left(-\frac{138}{\sigma}\right), \qquad Q_{\mathrm{ig}} = 250 + 1116M_f. \tag{14}$$

Here, ϵ is the effective heating number, which depends on the surface-area-to-volume ratio σ, and Q_{ig} is the heat of preignition, which increases with environmental fuel moisture M_f. Substituting these terms into Eq. (1) gives the local rate of spread $R_{i\to j}$.

In SimFire, this continuous rate of spread is applied on a discrete grid. For each active burning cell, the simulator enumerates neighboring candidate cells, using eight-connected neighborhoods. For each candidate cell, SimFire evaluates Eq. (1), multiplies the resulting rate by the simulation time step Δt, and accumulates a burn distance:

$$B_j^{t+\Delta t} = B_j^{t} + R_{i\to j}\Delta t. \tag{15}$$

Here, B_j^t is the accumulated burn distance at candidate cell j at time t, and Δt is the simulation time step. A candidate cell ignites once its accumulated burn amount exceeds the physical cell width Δx:

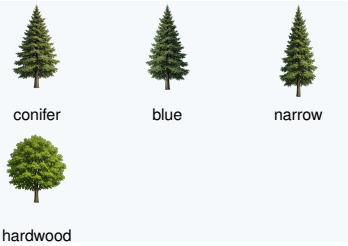
$$B_j^{t+\Delta t} > \Delta x. \tag{16}$$

Here, Δx is the physical width of one grid cell. Cells with zero fuel load are assigned zero spread rate, and negative rates are clipped to zero. It can additionally attenuate spread through mitigation cells such as firelines, scratchlines, and wetlines by reducing or zeroing the computed rate of spread before the burn amount is accumulated.

B.2 Anderson 13 Fuel Profiles

In this section, we show the the billboards we used for different Anderson 13 fuel code. We also summarizes the runtime assets based on different fuel families.

Code	Name	Base billboard mix in renderer
1	Short grass	`grass_short` 94%, `shrub_green` 6%
2	Timber (grass + understory)	`grass_short` 62%, `shrub_green` 26%, `tree_conifer` 12%
3	Tall grass	`grass_tall` 94%, `tree_hardwood` 6%
4	Chaparral	`shrub_green` 65%, `shrub_dry` 20%, `grass_tall` 15%
5	Brush	`shrub_green` 42%, `shrub_dry` 33%, `grass_short` 25%
6	Dormant brush / hardwood slash	`shrub_dry` 55%, `grass_short` 20%, `slash` 25%
7	Southern rough	`shrub_dry` 40%, `shrub_green` 28%, `grass_tall` 20%, `tree_hardwood` 12%
8	Closed timber litter	`tree_conifer` 58%, `shrub_green` 22%, `grass_short` 20%
9	Hardwood litter	`tree_hardwood` 58%, `shrub_green` 22%, `grass_tall` 20%
10	Timber (litter + understory)	`tree_conifer` 32%, `tree_hardwood` 12%, `grass_short` 40%, `grass_tall` 16%
11	Light logging slash	`slash` 38%, `tree_conifer` 17%, `grass_short` 32%, `shrub_dry` 13%
12	Medium logging slash	`slash` 42%, `shrub_dry` 35%, `grass_short` 23%
13	Heavy logging slash	`slash` 48%, `shrub_dry` 26%, `grass_tall` 26%

Family / effect	Preview variants	Anderson 13 codes	Notes
Short grass		1, 2, 5, 6, 8, 10, 11, 12	Expanded from the short-grass foliage archetype.
Tall grass		3, 4, 7, 9, 10, 13	Expanded from the tall-grass foliage archetype.
Green shrub		1, 2, 4, 5, 7, 8, 9	Expanded from the green-shrub foliage archetype.
Dry shrub		4, 5, 6, 7, 11, 12, 13	Expanded from the dry-shrub foliage archetype.
Shared tree pool		2, 3, 7, 8, 9, 10, 11	Current student pipeline expands tree references into this shared active pool.
Slash		6, 11, 12, 13	Expanded from the slash foliage archetype.

Family / effect	Preview variants			Anderson 13 codes	Notes
Fire and smoke effects	 flame	 ember	smoke	–	Effect billboards for fire or smoke rendering; not tied to Anderson fuel codes.

C Implementation Details

In this section, we provide additional implementation details of the proposed framework.

2D Simulation. For the 2D simulation stage, we use SimFire Doyle et al. (2024). All simulations are conducted on a $3\,\mathrm{km}$ domain, with each grid cell corresponding to approximately $30\,\mathrm{m}$. The simulation horizon is set to represent one hour of wildfire propagation. For the four major factors affecting the simulations, we define low wind speed as below $3\,\mathrm{mph}$ and high wind speed as above $3\,\mathrm{mph}$. Low moisture is defined in the range $[0.0001, 0.005)$, while high moisture is defined in the range $[0.005, 0.01]$. For vegetation coverage and slope, we first query real-world fuel maps and terrain data for each location, and then perform post hoc categorization based on the resulting measurements. Under this procedure, the dataset is approximately divided into low vegetation coverage for values below 6.96%, and high vegetation coverage for values between 6.96% and 70.77%. Similarly, low slope corresponds to values between $0°$ and $9.10°$, while high slope corresponds to values between $9.10°$ and $34.94°$. We additionally show the queried 500 physical locations in Figure 9.

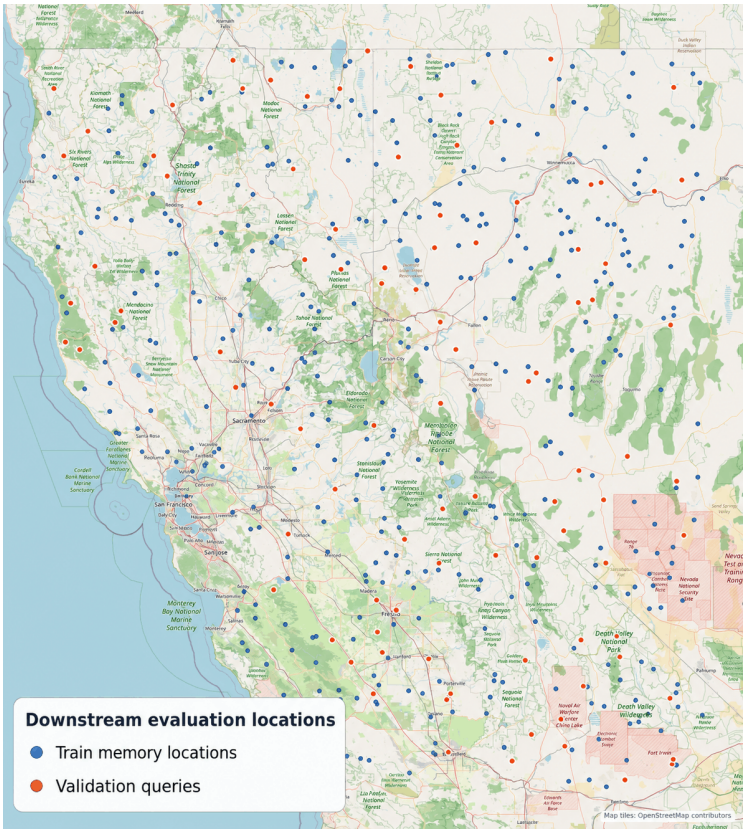


Figure 9: Geographic locations for memory construction and held-out evaluation; no VLM weights are updated.

3D Rendering. We use a fixed, fully automatic Blender procedure for 2D-to-3D proxy conversion. Once the asset library and conversion rules are defined, each episode is constructed without manual scene editing or an LLM generating new scene code. The proxy is an intermediate visualization of simulator state, not an independent 3D fire simulation. For geometry, the elevation fields queried from LANDFIRE are converted into 3D terrain surfaces. For vegetation, following the Anderson 13 fuel model, we collect multiple vegetation and terrain textures and represent different fuel types using billboards. If a grid cell is labeled as burning in the 2D simulation, we randomly sample fire billboards within that cell and additionally render tilted smoke billboards according to the wind speed and wind direction. For camera placement, we choose aerial viewpoints that can observe the initial ignition region while resembling footage captured by a UAV. Finally, all proxy videos are rendered in Blender using Eevee. The same field-to-asset rules are reused across episodes; random billboard sampling varies appearance within those rules rather than changing the mapping itself. The design emphasizes simulator-driven terrain, fuel layout, burning-cell activity, and wind cues rather than production-quality assets or volumetric rendering.

Scope of automation and computational cost. The fixed Blender converter is executed for each episode without a scene-authoring model call. This scope is separate from prompt refinement, video generation, reranking, and reporting, which retain their own inference costs. Fixed conversion rules also differ from fixed scene contents: simulator fields and randomized billboard placement vary within the same procedure. Section 6 discusses how this organization relates to general-purpose scene authoring with GPT-6 Astra and coding-agent-driven world evolution in Code World Model.

3D Generation. We use VACE Jiang et al. (2025) as the video generation backbone. Following the preprocessing strategy of Chen et al. (2025), we compute both depth and scribble control videos from proxy videos rendered in Blender. To better align the controls with the domain expected by VACE, we estimate depth using Yang et al. (2024) and extract scribble signals using Chan et al. (2022). To screen depth alignment, we measure the Spearman correlation between the generated video depth and the proxy-rendered depth. If the score is below 0.35, we retry generation up to three times; if all retries fail, we resample a new location and repeat the full generation pipeline.

Memory System. We use Qwen3-2B-VL as the embedding model. As shown in Sec. 4, this model provides more accurate results than the other embedding models considered in our setting. We use GPT-5 as the reranker; the corresponding prompt is provided in Appendix G.4.

Multi-Agent Reporting The memory split supplies reference episodes and train-derived lookup priors without updating VLM weights. The top $k = 5$ retrieved episodes are aggregated to compute task-wise consensus scores and metadata summaries for adjudication, whereas four-tag prediction in Table 3 uses the top $k = 1$ retrieved episode. The three reporters correspond to different prompt contexts: the visual reporter receives only sampled query frames; the lookup reporter receives query frames and the top retrieved tag-bundle prior; and the combined-memory reporter receives query frames, the lookup prior, and retrieved simulator-derived summaries. For each report field, unanimous reporter outputs are copied directly into the final prediction vector, while disputed fields are sent to the judge. The judge prompt includes the sampled frames, reporter predictions, short rationales, locked unanimous fields, top-k memory consensus, and retrieved metadata summaries, and is constrained to revise only disputed fields. The resolved predictions, routing decisions, rationales, and retrieval summaries are serialized and rendered into an HTML report for each query.

D Simulation to Observation Alignment Numerical Metrics

Here, we quantitatively assess whether the VACE generation pipeline follows the provided proxy render. We evaluate this by testing whether the generated video is more consistent with its paired render than with a mismatched render. The evaluation metrics are summarized in Table 11, and the corresponding results are reported in Table 12. Higher agreement with paired than mismatched renders indicates sensitivity to the intended controls. These relative image-alignment measurements do not establish exact physical equivalence between generated videos and simulator trajectories.

Table 11: Proxy–generated alignment metrics.

Metric	Directions	Meaning
SSIM	↑	Grayscale structural similarity between aligned VACE and render frames.
Edge F1 / Edge-control F1	↑	Canny-edge overlap between VACE and render frames with a small dilation tolerance.
RGB L1	↓	Mean absolute RGB difference over aligned frames.
LPIPS	↓	AlexNet LPIPS perceptual distance averaged over aligned frames.
Correct-pair win rate	↑	Fraction of samples for which the paired render is closer to the VACE output than a mismatched render.

Table 12: Proxy–generated alignment and correct-pair win rates.

SSIM	SSIM Win Rate	Edge F1	Edge F1 Win Rate	RGB L1	RGB L1 Win Rate	LPIPS	LPIPS Win Rate
0.334	76.6%	0.612	81.3%	0.189	72.0%	0.516	99.1%

We also compare two image-space directional cues between generated and proxy videos. **Smoke plume axis** Smoke-colored pixels are extracted in each sampled frame. We then use PCA to give the dominant plume axis. Axes are averaged over frames and compared as undirected axes. **Growth-front fire-spread axis** Flame pixels from the first third of the clip are unioned into an early-fire mask. Flame pixels from the last third are unioned into a late-fire mask. Pixels already near the early mask are removed from the late mask, giving a late-growth region. The vector from the early-fire centroid to the late-growth centroid estimates spread direction. We then compare the generated and rendered videos fire spread axis.

Metric	$\leq 30^\circ$	$\leq 45^\circ$	$\leq 60^\circ$	Median	Mean
Smoke plume axis	95.8%	97.4%	99.2%	3.7	7.6
Growth-front fire-spread axis	47.1%	67.3%	79.0%	32.7	35.8

The mean errors are 7.6° for the smoke-plume axis and 35.8° for the growth-front fire-spread axis. The larger growth-direction deviation limits a claim of full dynamic preservation. These image-space cues do not measure physical spread rate or burned-area trajectories.

D.1 VLM-Based Observable Cue Agreement

We use GPT-5.6-luna to assess agreement between sampled frames from a proxy video and its corresponding conditioned generated video. Each pair is scored from 0 to 100 for fire-front location, fuel distribution, burned area, and terrain structure. Table 14 reports the average scores. Identical proxy–proxy pairs score approximately 95, while identical generated–generated pairs score approximately 90. These identity comparisons contextualize the ratings; the scores are not calibrated percentages of physical fidelity. This assessment complements the numerical alignment metrics above rather than establishing exact physical equivalence.

Table 14: VLM-assessed agreement on a 0–100 scale. Identity-pair values are approximate overall averages, not per-cue scores.

Cue or comparison	Score
Fire-front location	79.17
Fuel distribution	68.75
Burned area	70.83
Terrain structure	81.25
Overall	75.00
Identical proxy–proxy pairs	≈ 95
Identical generated–generated pairs	≈ 90

E MODEL-TO-PROMPT MAPPING

Reading guide. Blue boxes denote system prompts, green boxes user prompts, orange boxes schemas and output contracts, purple or teal boxes task and wrapper prompts, red boxes negative prompts, and gray boxes assembly or payload snippets.

Model family	Evaluation role	Prompt family
GPT-4.1 / GPT-5.2	direct query-only four-tag inference	shared query-only tag prompt in Section G.1
Gemini 2.5 Pro	direct query-only four-tag inference	shared query-only prompt plus Gemini/Claude JSON wrapper in Section G.2
Claude Sonnet 4	direct query-only four-tag inference	shared query-only prompt plus Gemini/Claude JSON wrapper in Section G.2
Qwen2-Wildfire-2B	local VLM four-tag inference; six-task downstream conditioned on its own predicted tags	local VLM binary-tag prompt in Section G.3; downstream VLM prediction in Section I.7
ForestFireVLM-7B	local VLM four-tag inference; six-task downstream conditioned on its own predicted tags	local VLM binary-tag prompt in Section G.3; downstream VLM prediction in Section I.7
GPT-5-mini	downstream specialist inference	downstream specialist system + context blocks in Section I
GPT-5-mini family	multi-agent disputed-task resolution	multi-agent judge prompt in Section I.8
GPT-5-mini	final HTML report narrative generation	report-generation prompt in Section J
GPT-5-mini	video-generation prompt rewriting	prompt-refinement template in Section F.2
Qwen3-VL embedding / reranker	hybrid retrieval and optional reranking	Qwen retrieval/rerank formatting in Section H

F VIDEO GENERATION PROMPT

The following templates document the controls used to construct the reference episodes. Environmental tags influence both positive and negative prompts, so generated appearance reflects the prompt design as well as the simulated state and proxy geometry. The templates are reproduced as experimental records; requested realism and motion characteristics are generation objectives, not guarantees of physical fidelity.

F.1 VIDEO GENERATION PROMPT FROM SIMULATION DATA

Raw Prompt Template The raw prompt uses a fixed sentence template with simulation-conditioned slots:

```
Photorealistic aerial drone video from a distant, stabilized stand-off
 viewpoint
over a broad natural landscape, showing <duration> of wildfire
propagation with
the terrain, smoke plume, and fire perimeter kept readable in a wide
frame while
the active flames remain visible. Surface cover should read as <
burnable cover>,
with <barriers> acting as fire breaks; fuel continuity is <continuity>
 and
```

```
barrier influence is <barrier influence>. <vegetation density text>
<built environment text> Use <wind strength> winds moving toward <wind
 direction>,
<moisture>, and <terrain>. <flame visibility>. Depict a <run size>
fire with
<fire motion>, <smoke motion>, realistic flank slowdowns at fragmented
 fuels and
barrier edges and <spotting>. Natural
daylight, cinematic realism, and physically grounded motion.
```

Fuel and Barrier Description The fuel map is summarized by pixel count and percentage for each fuel code. Anderson-13 burnable fuel codes are mapped to natural-language land-cover names, and common non-burnable codes are mapped to barrier names.

Input	Decision rule	Prompt effect
Fuel percentage by code	The top four burnable land-cover groups are selected after grouping by semantic cover type.	Forms the phrase "Surface cover should read as ..."
Non-burnable percentage by code	The top three non-burnable groups are selected after grouping by semantic barrier type.	Forms the phrase "with ... acting as fire breaks."
Cover percentage p	$p \geq 60$: dominant; $30 \leq p < 60$: major; $10 \leq p < 30$: moderate; $3 \leq p < 10$: minor; $p < 3$: trace.	Adds qualitative size words such as "major brushy shrubland" or "minor open water."
Burnable percentage b	$b < 60$: fragmented; $60 \leq b < 85$: partially interrupted; $b \geq 85$: mostly continuous.	Sets the fuel-continuity phrase.
Non-burnable percentage n	$n \geq 25$: substantial; $10 \leq n < 25$: noticeable; $n < 10$: limited.	Sets the barrier-influence phrase.

Urban Visibility The city overlay count controls how strongly the prompt asks for visible urban context. Urban pixels are not duplicated in the generic barrier list when the city is already visible, clear, or prominent.

Wind, Smoke, and Spotting Wind speed and direction are read from simulation metadata. Direction in degrees is converted to one of sixteen compass directions. Wind speed controls the wind-strength phrase, smoke behavior, fire-motion wording, flame-visibility wording, spotting behavior, and the negative prompt. These are appearance controls: in particular, spotting language does not mean that ember trajectories or spotting dynamics are simulated by the current surface-spread model.

Fire-motion text combines wind speed with spread intensity. For very light wind, fast or very fast spread is described as "steady to locally fast flame-front growth driven mainly by fuels and slope rather than wind-whipped runs." For moderate wind, the wording becomes "noticeable downwind push." For strong or very strong wind, the wording becomes "wind-driven flame-front growth with pronounced downwind push."

Wind speed	Wind phrase	Smoke phrase	Spotting phrase
< 0.75 m/s	very light	Mostly vertical smoke column with gentle drift and limited sideways shear.	Only sparse short-range ember lofting where continuity supports it.
0.75–< 2.0 m/s	light	Mostly upright smoke plume with mild downwind drift and restrained lateral shear.	Limited ember lofting and short-range spotting where continuity supports it.
2.0–< 5.0 m/s	moderate	Clear downwind smoke advection with moderate plume tilt and lateral drift.	Occasional ember spotting where continuity supports it.
5.0–< 8.0 m/s	strong	Strong downwind smoke transport with a clearly bent plume and sustained lateral advection.	Frequent downwind ember transport and intermittent spotting where continuity supports it.

Wind speed	Wind phrase	Smoke phrase	Spotting phrase
≥ 8.0 m/s	very strong	Strong downwind smoke transport with a clearly bent plume and sustained lateral advection.	Frequent downwind ember transport and intermittent spotting where continuity supports it.

Moisture, Terrain, and Vegetation

Feature	Decision rule	Prompt effect
Moisture tag	`low_moisture`: very dry surface fuels; `high_moisture`: comparatively moister surface fuels.	Tags override numeric moisture wording.
Numeric moisture m	$m < 0.03$: very dry; $0.03 \leq m < 0.06$: dry; $0.06 \leq m < 0.10$: moderately dry; $m \geq 0.10$: moist.	Used when no explicit moisture tag is present.
Elevation relief r	$r < 100$ m: gentle; $100 \leq r < 300$ m: rolling; $300 \leq r < 600$ m: rugged; $r \geq 600$ m: very rugged.	Sets the terrain phrase. Default wording says spread accelerates on favorable slopes and eases where aspect is less favorable.
Slope tag	`high_slope`: emphasizes steeper local slopes; `low_slope`: emphasizes broader gentler local slopes.	Modifies terrain wording beyond the numeric relief class.
Vegetation tag	`high_vegetation`: denser vegetation, continuous canopy, fewer gaps; `low_vegetation`: more open, sparse vegetation, patchier canopy.	Adds a separate vegetation-density sentence.

Negative Prompt Generation The negative prompt is also generated from the same simulation metadata. It always includes "camera shake" and "rapid viewpoint changes." Wind, moisture, slope, and vegetation tags then add exclusions intended to discourage contradictory visual behavior.

Condition	Additional negative phrases
$w < 0.75$ m/s	strong wind; very strong wind; gale-force smoke; wind-whipped flames; violent lateral smoke; long horizontal smoke bands; storm-like plume bending.
$0.75 \leq w < 2.0$ m/s	strong wind; gale-force smoke; wind-whipped flames; extreme lateral smoke advection; fully horizontal smoke plume.
$2.0 \leq w < 5.0$ m/s	hurricane-force wind; perfectly still air; fully vertical plume with no drift.
$w \geq 5.0$ m/s	still air; calm smoke column; perfectly vertical plume with no downwind drift.
`low_moisture`	rain-soaked fuels; wet ground everywhere.
`high_moisture`	ultra-dry fuels; explosive firestorm behavior from uniformly desiccated vegetation.
`low_slope`	sheer mountainsides; cliff-dominated terrain.
`high_slope`	perfectly flat plain; table-flat ground.
`low_vegetation`	dense closed-canopy forest everywhere; jungle-thick vegetation.
`high_vegetation`	nearly barren terrain; treeless empty plain.

F.2 Prompt Refinement for Video Generation

As mentioned in 3.1.3. This stage rewrites an existing wildfire video prompt into a cleaner text-to-video prompt while preserving the intended wildfire scenario.

F.2.1 System Prompt for Prompt Refinement

```
You are an expert text-to-video prompt writer for realistic
environmental simulations.
Rewrite user prompts to improve visual clarity, motion continuity,
physical realism, and generation stability.
Keep the environmental meaning faithful.
Preserve explicit cues about visible city buildings or developed
structures, aerial drone capture, distant stand-off viewpoint,
elapsed fire duration such as one hour, blackened burned land or burn
scars, and a clearly visible active flame front when they are present.
Preserve explicit wind-strength cues. For low-wind or very-light-wind
cases, keep smoke mostly upright with only gentle drift and do not
rewrite them into wind-whipped or strongly horizontal plumes.
Do not let the rewrite imply that smoke fully hides the flames when
the source prompt says the active fireline must stay readable.
For strong-wind cases, preserve clearly downwind smoke transport
rather than weakening the wind effects.
Keep the rewrite compact and high-signal, ideally about 90 to 140
words in a single paragraph.
Avoid coordinates, fuel model names, numeric fuel codes, brittle shot-
by-shot instructions, and markdown.
Return exactly one concise paragraph.
```

F.2.2 User Prompt Template for Prompt Refinement

```
Refine this wildfire video generation prompt for a modern text-to-
video model.
Preserve the scenario and realism constraints while making it clearer
and more producible:

<original prompt text>
```

Message assembly.

```
[system]  <prompt-refinement system prompt above>
[user]    <prompt-refinement user prompt above>
```

Output contract. The response is constrained to exactly one concise paragraph, with low verbosity and reasoning effort in the OpenAI Responses API call.

G Direct Four-Tag Evaluation Prompts

G.1 OpenAI Query-Only Tag Prompt

The VLM direct query baselines in Table 3 use a shared instruction shown below. The OpenAI request is assembled as one user message containing this instruction block, followed by the query frames. In later sections, we refer this prompt as `DEFAULT_QUERY_ONLY_CLASSIFICATION_INSTRUCTIONS`.

```
Classify the query wildfire video into four binary tags using only
visible evidence from the frames.
Decide each field independently.
Use smoke and flame drift or lean for wind, vegetation greenness or
dryness and burn behavior for moisture, terrain incline and relief for
 slope, and plant density or fuel coverage for vegetation.
```

```
For each field, return a label of low or high and one short reason
based on visible evidence.
```

Message assembly.

```
[input_text]  <instruction block above>
[input_text]  Query video to classify.
[input_text]  Query frame 1.
[input_image] <frame 1>
...
[input_text]  Query frame n.
[input_image] <frame n>
```

G.2 GEMINI / CLAUDE EXTERNAL QUERY-ONLY PROMPT WRAPPER

Gemini and Claude use the same core direct query instruction, but the text prompt is wrapped with an explicit JSON-format reminder.

```
<DEFAULT_QUERY_ONLY_CLASSIFICATION_INSTRUCTIONS>
Return only a JSON object with exactly four keys: wind_regime_tag,
moisture_regime_tag, slope_regime_tag, and vegetation_regime_tag. Each
 key must map to an object with exactly two keys: label and reason.
The label must be either low or high. The reason must be one short
sentence based only on visible evidence. Do not return markdown.
```

Gemini message assembly.

```
[text]        <wrapped instruction block above>
[text]        Query video to classify.
[inline_data] <frame 1>
...
[inline_data] <frame n>
```

Claude message assembly.

```
[text]   <wrapped instruction block above>
[text]   Query video to classify.
[image]  <frame 1>
...
[image]  <frame n>
```

The default provider models in this evaluator are:

- Gemini: `gemini-2.5-pro`
- Claude: `claude-sonnet-4-20250514`

G.3 LOCAL VLM QUERY-ONLY PROMPT FOR QWEN2-WILDFIRE-2B AND FORESTFIREVLM

For wildfire finetuned VLM baselines shown in Table 4, we first use the follow prompt to get the 4 tag prediction and then use this together with the prompts in Appendix I.7 to perform downstream task prediction. For simulation tag prediction, the prompt used is shown below:

```
Classify this wildfire video into four binary tags from visible
evidence only.
```

```
For wind use smoke/flame drift, for moisture use dryness and burn
behavior,
for slope use terrain incline/relief, for vegetation use fuel density/
coverage.
This benchmark is strictly binary. There is no medium or moderate
class.
If evidence is mixed, choose the closer of low or high and explain
briefly.
Return only JSON with keys wind_regime_tag, moisture_regime_tag,
slope_regime_tag, vegetation_regime_tag.
Each key may be either a string label or an object with label and
reason.
Every label must be exactly low or high.
```

Message assembly.

```
[text]   <prompt or prompt + retry suffix>
[text]   Frame 1.
[image]  <frame 1>
...
[text]   Frame n.
[image]  <frame n>
```

G.4 OpenAI Candidate-Reranking Prompt

Here we show the prompt we used for GPT reranking which correspond to the result shown in Table 3.

```
Score how well each candidate video's known four-tag bundle matches
the query wildfire video.
Use only visible evidence from the query frames and the candidate
frames.
The candidate tags are the known labels for that candidate, and the
candidate video is a visual anchor for those labels.
Score each candidate independently on a 0 to 100 scale, where 100
means the candidate's full four-tag bundle is a very strong match to
the query and 0 means it is a very poor match.
Consider wind, moisture, slope, and vegetation together.
Return one numeric score and one short reason for each candidate.
```

Candidate text template.

```
Candidate video <index>. Return item_id=<candidate_id> exactly in your
 response.
Known labels: wind_regime_tag=<wind>,
moisture_regime_tag=<moisture>,
slope_regime_tag=<slope>,
vegetation_regime_tag=<vegetation>.
```

Message assembly.

```
[input_text]  <candidate-scoring instruction block above>
[input_text]  Query video.
[input_text]  You must return exactly one scored entry for every
candidate item_id shown below. Do not omit any candidate. Use the
short item_id tokens exactly as written, such as C01, C02, and C03.
[input_text]  Query frame 1.
[input_image] <query frame 1>
```

```
...
[input_text]  Candidate video 1. Return item_id=C01 exactly in your
response. Known labels: ...
[input_text]  Candidate 1 frame 1.
[input_image] <candidate 1 frame 1>
...
```

G.5 Baselines with Labeled Reference Videos

We augment direct VLM inference with 16 labeled reference videos per query, one for each combination of the four binary environmental driver tags. Reference frames are accompanied by ground-truth tags and descriptions; the query's simulator metadata is withheld. Table 20 reports exact four-tag accuracy and the proportions of queries with at least three or two correct tags. The strongest exact-match baseline is GPT-5 mini at 27.27%, compared with 51.5% for the full memory system in Table 3. Model versions differ between some query-only and example-augmented baselines, so cross-group comparisons are not a matched-model ablation of adding examples.

Table 20: Environmental driver prediction with 16 labeled reference videos per query. All entries are percentages.

Model	All four	At least three	At least two
GPT-4.1 mini	24.27	57.65	91.92
GPT-5 mini	27.27	66.67	95.96
Claude Sonnet 4.5	23.23	63.64	95.96
Gemini 2.5 Pro	26.26	75.76	97.98

H Qwen-Embedding Retrieval and Hybrid Rerank Prompts

H.1 Our Reranking Method

The main hybrid retrieval build referenced throughout the downstream pipeline is the Qwen-embedding-backed retrieval setup, followed by GPT candidate scoring over the retrieved shortlist.

H.2 Optional Qwen3-VL Reranker Instruction

We also tested an optional Qwen3-VL-2B reranker (instead of using GPT API) with the default instruction:

```
Given a search query, retrieve relevant candidates that answer the
query.
```

H.2.1 Qwen3-VL Reranker Message Formatting

When the Qwen3-VL reranker is enabled, each query-document pair is formatted as follows.

System message.

```
Judge whether the Document meets the requirements based on the Query
and the Instruct provided.
Note that the answer can only be "yes" or "no".
```

User message template.

```
<Instruct>: Given a search query, retrieve relevant candidates that
answer the query.
```

```
<Query>:
<query video and/or query text>

<Document>:
<candidate video and optional candidate prompt text>
```

Practical note. The reranker comparison in Appendix H.3 reports the added model-choice and timing evaluation.

H.3 Reranker Accuracy and Latency

Table 21 compares reranker accuracy and scoring latency. GPT-5 and Gemini 2.5 Pro both achieve 51.52% exact four-tag accuracy, with lower mean scoring latency for GPT-5 in the tested configurations. Qwen3-VL-Reranker-2B runs locally on an H200 and is faster but less accurate. API-based and local execution are distinct deployment conditions, so these measurements are not a hardware-controlled comparison of model efficiency. Preparation plus scoring is reported separately from scorer-only latency; neither is the end-to-end multi-agent pipeline runtime.

Table 21: Reranker ablation. Latencies are in seconds and exact four-tag accuracy is a percentage. Mean and median refer to scorer latency in seconds.

Reranker	Execution	Mean	Median	Prep. + scorer	Four-tag accuracy
GPT-5	API	9.18	8.60	11.80	51.52
Claude Sonnet 4.5	API	12.30	11.97	14.59	46.46
Gemini 2.5 Pro	API	22.24	21.59	24.53	51.52
Qwen3-VL-Reranker-2B	Local H200	5.79	5.16	7.97	40.40

H.4 Retrieval-Size Ablation

Table 22 reports the exact four-tag hit rate and per-attribute Hit@k as the number of retrieved videos increases from one to four. Exact four-tag hit rate increases from 51.52% to 69.70%. These are retrieval-coverage measurements, not final predictions from a multi-agent system using larger retrieval sets.

Table 22: Retrieval-size ablation. Hit rates are percentages and marginal gains are percentage-point increases in exact four-tag hit rate. These are retrieval metrics, not downstream prediction accuracies.

k	Exact four-tag hit	Marginal gain	Wind	Moisture	Slope	Vegetation
1	51.52	–	74.75	83.84	82.83	89.90
2	57.58	+6.06	79.80	88.89	88.89	91.92
3	64.65	+7.07	81.82	94.95	92.93	92.93
4	69.70	+5.05	87.88	95.96	95.96	97.98

I Downstream Six-Task Specialist Prompts

Here we show the system prompts and context blocks used for the downstream six-task wildfire behavior inference agents in Section 3.2.2. The fixed multi-agent pipeline keeps three specialists: a direct visual specialist, a lookup-memory specialist, and a combined-memory specialist. Unanimous specialist fields pass through directly, and the GPT judge resolves only disputed fields in the final fixed run.

I.1 Downstream Task List

```
Predict all of these task keys:
- spread_fast_bin: Fast Spread Risk: choose `fast_or_very_fast` if the
 query likely shows fast or very fast spread. Otherwise choose `
slow_or_moderate`.
- peak_high_bin: High Peak Burning Area: choose `high` if peak burning
 area is likely at least 90,000 m^2, else `low`.
- burnscar_high_bin: High Final Burnscar: choose `high` if final
burnscar area is likely at least 300,000 m^2, else `low`.
- vegetation_mean_canopy_cover_pct_median_bin: High Canopy Cover:
choose `high` if canopy cover is likely above the train-set median
(~2.61%), else `low`.
- rapid_growth_and_large: Rapid Growth + Large Burn: choose `high`
only if the query likely has fast or very fast spread and also likely
ends with final burnscar area at least 300,000 m^2. Otherwise choose `
low`.
- severe_fire_behavior: Severe Fire Behavior: choose `high` if at
least 2 of these 3 are likely true: fast/very_fast spread, peak
burning area at least 90,000 m^2, final burnscar area at least 300,000
 m^2. Otherwise choose `low`.
```

I.2 Visual Report Agent

Visual Report Agent System Prompt

```
You are the direct visual specialist in a wildfire multi-agent
downstream inference pipeline. Use only the query video frames and the
 task rubric; do not assume the retrieved memory prior is correct, and
 do not use train labels. Your role is to provide calibrated visual
evidence, not to agree with memory specialists. Be conservative for
thresholded area tasks: high labels require broad, continuous, visible
 evidence. Do not infer final burnscar or high canopy from smoke
density alone. Return only JSON matching the requested schema.
```

Visual Report Agent Task Rubrics

```
Predict all downstream task keys from visual evidence only:
- spread_fast_bin: Fast Spread Risk: choose `fast_or_very_fast` only
when the active flame front visibly expands or translates
substantially over the sampled frames. Choose `slow_or_moderate` for
compact, slowly changing, or ambiguous fronts.
- peak_high_bin: High Peak Burning Area: choose `high` only when
active burning visibly covers a broad continuous landscape-scale area
likely >= 90,000 m^2. Choose `low` when the active flaming footprint
is localized or uncertain.
- burnscar_high_bin: High Final Burnscar: choose `high` only when
later frames show a large continuous blackened/burned footprint likely
 >= 300,000 m^2. Do not infer high burnscar from smoke alone.
- vegetation_mean_canopy_cover_pct_median_bin: High Canopy Cover:
choose `high` only for clear forest/tree canopy or repeated dense tree
 patches. Choose `low` for mostly grass, shrub, bare, or sparse
isolated trees. Avoid overcalling high from small tree clusters.
- rapid_growth_and_large: Rapid Growth + Large Burn: choose `high`
only when both visible fast spread and a large final burned footprint
are supported. If either condition is uncertain, choose `low`.
- severe_fire_behavior: Severe Fire Behavior: choose `high` only if at
 least two visual indicators are supported: fast spread, large active
burning area, large final burnscar. Choose `low` if only one or none
are visually supported.
```

```
Output a label and a short visual reason for each task.
```

Message assembly.

```
[input_text]  <visual specialist system prompt>
[input_text]  <visual specialist task rubric>
[input_text]  Query video to evaluate.
[input_text]  Query frame 1.
[input_image] <query frame 1, detail=low>
...
[input_text]  Query frame n.
[input_image] <query frame n, detail=low>
```

I.3 Lookup- and Combined-Memory Specialist Shared System Prompt

The lookup-memory and combined-memory specialists use the shared memory-specialist prompt. The memory prior is treated as soft context rather than ground truth, and query frames remain primary evidence.

```
Estimate downstream wildfire behavior attributes for the query video.
Use the query frames as primary evidence. You are also given a hybrid-
memory prior from a retrieved similar case, including retrieved latent
 tags and metadata. Treat the memory prior as soft context, not ground
 truth for the query. If the memory prior conflicts with visible query
 evidence, trust the query frames. Return only JSON matching the
requested schema.
```

I.4 Lookup-Memory Specialist Context

```
Predict all of these task keys:
- spread_fast_bin: Fast Spread Risk: choose `fast_or_very_fast` if the
 query likely shows fast or very fast spread. Otherwise choose `
slow_or_moderate`.
- peak_high_bin: High Peak Burning Area: choose `high` if peak burning
 area is likely at least 90,000 m^2, else `low`.
- burnscar_high_bin: High Final Burnscar: choose `high` if final
burnscar area is likely at least 300,000 m^2, else `low`.
- vegetation_mean_canopy_cover_pct_median_bin: High Canopy Cover:
choose `high` if canopy cover is likely above the train-set median
(~2.61%), else `low`.
- rapid_growth_and_large: Rapid Growth + Large Burn: choose `high`
only if the query likely has fast or very fast spread and also likely
ends with final burnscar area at least 300,000 m^2. Otherwise choose `
low`.
- severe_fire_behavior: Severe Fire Behavior: choose `high` if at
least 2 of these 3 are likely true: fast/very_fast spread, peak
burning area at least 90,000 m^2, final burnscar area at least 300,000
 m^2. Otherwise choose `low`.

Hybrid lookup prior from the top-1 retrieved tag bundle follows.
The lookup prior is train-derived from the retrieved tags. Use it as a
 soft prior only.
If the query frames clearly disagree, trust the query frames over the
lookup prior.
{
  "retrieved_item_id": "<top1 item_id>",
  "retrieval_score": <top1 score>,
```

```
  "retrieved_tags": {
    "wind_regime_tag": "<low|high>",
    "moisture_regime_tag": "<low|high>",
    "slope_regime_tag": "<low|high>",
    "vegetation_regime_tag": "<low|high>"
  },
  "lookup_prior_from_retrieved_tags": {
    "spread_fast_bin": "<label>",
    "peak_high_bin": "<label>",
    "burnscar_high_bin": "<label>",
    "vegetation_mean_canopy_cover_pct_median_bin": "<label>",
    "rapid_growth_and_large": "<label>",
    "severe_fire_behavior": "<label>"
  }
}
```

Message assembly.

```
[input_text]  <shared memory-specialist system prompt>
[input_text]  <lookup specialist context block>
[input_text]  Query video to evaluate.
[input_text]  Query frame 1.
[input_image] <query frame 1, detail=low>
...
[input_text]  Query frame n.
[input_image] <query frame n, detail=low>
```

I.5 Combined-Memory Agent Context

The combined-memory specialist receives the query frames together with the combined-memory context block and the lookup-memory context block. The combined block includes the retrieved prompt, environmental metadata, and fire-behavior summaries shown below.

```
Predict all of these task keys:
- spread_fast_bin: Fast Spread Risk: choose `fast_or_very_fast` if the
 query likely shows fast or very fast spread. Otherwise choose `
slow_or_moderate`.
- peak_high_bin: High Peak Burning Area: choose `high` if peak burning
 area is likely at least 90,000 m^2, else `low`.
- burnscar_high_bin: High Final Burnscar: choose `high` if final
burnscar area is likely at least 300,000 m^2, else `low`.
- vegetation_mean_canopy_cover_pct_median_bin: High Canopy Cover:
choose `high` if canopy cover is likely above the train-set median
(~2.61%), else `low`.
- rapid_growth_and_large: Rapid Growth + Large Burn: choose `high`
only if the query likely has fast or very fast spread and also likely
ends with final burnscar area at least 300,000 m^2. Otherwise choose `
low`.
- severe_fire_behavior: Severe Fire Behavior: choose `high` if at
least 2 of these 3 are likely true: fast/very_fast spread, peak
burning area at least 90,000 m^2, final burnscar area at least 300,000
 m^2. Otherwise choose `low`.

Hybrid memory prior from the top-1 retrieved similar case follows.
Use it as a soft prior only. Do not copy it blindly if the query
frames disagree.
{
  "retrieved_item_id": "<top1 item_id>",
  "retrieval_score": <top1 score>,
  "retrieved_tags": {
```

```
    "wind_regime_tag": "<low|high>",
    "moisture_regime_tag": "<low|high>",
    "slope_regime_tag": "<low|high>",
    "vegetation_regime_tag": "<low|high>"
  },
  "retrieved_prompt": "<retrieved training prompt>",
  "retrieved_environment_metadata": {
    "wind_speed_mean_mph": <float>,
    "wind_dir_mean_cardinal": "<cardinal direction>",
    "environment_moisture": <float>,
    "slope_mean_deg": <float>,
    "vegetation_mean_canopy_cover_pct": <float>,
    "elevation_relief_m": <float>,
    "burnable_fraction": <float>
  },
  "retrieved_fire_behavior_metadata": {
    "peak_burning_area_m2": <float>,
    "final_burnscar_area_m2": <float>,
    "burnscar_ratio": <float>,
    "max_radial_spread_mps": <float>
  }
}
```

Message assembly.

```
[input_text]  <shared memory-specialist system prompt>
[input_text]  <combined-memory specialist context block>
[input_text]  <lookup specialist context block>
[input_text]  Query video to evaluate.
[input_text]  Query frame 1.
[input_image] <query frame 1, detail=low>
...
[input_text]  Query frame n.
[input_image] <query frame n, detail=low>
```

I.6 Specialist Response Schema

All three specialists return a label and a short reason for each downstream task.

```
{
  "type": "json_schema",
  "name": "hybrid_downstream_prediction",
  "schema": {
    "type": "object",
    "properties": {
      "spread_fast_bin": {
        "type": "object",
        "properties": {
          "label": {"type": "string", "enum": ["slow_or_moderate", "
fast_or_very_fast"]},
          "reason": {"type": "string"}
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      },
      "peak_high_bin": {
        "type": "object",
        "properties": {
          "label": {"type": "string", "enum": ["low", "high"]},
          "reason": {"type": "string"}
```

```
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      },
      "burnscar_high_bin": {
        "type": "object",
        "properties": {
          "label": {"type": "string", "enum": ["low", "high"]},
          "reason": {"type": "string"}
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      },
      "vegetation_mean_canopy_cover_pct_median_bin": {
        "type": "object",
        "properties": {
          "label": {"type": "string", "enum": ["low", "high"]},
          "reason": {"type": "string"}
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      },
      "rapid_growth_and_large": {
        "type": "object",
        "properties": {
          "label": {"type": "string", "enum": ["low", "high"]},
          "reason": {"type": "string"}
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      },
      "severe_fire_behavior": {
        "type": "object",
        "properties": {
          "label": {"type": "string", "enum": ["low", "high"]},
          "reason": {"type": "string"}
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      }
    },
    "required": [
      "spread_fast_bin",
      "peak_high_bin",
      "burnscar_high_bin",
      "vegetation_mean_canopy_cover_pct_median_bin",
      "rapid_growth_and_large",
      "severe_fire_behavior"
    ],
    "additionalProperties": false
  }
}
```

I.7 Downstream Task Prediction Prompt for GPT, Qwen2-Wildfire-2B and ForestFireVLM

This section shows the prompt used in Table 4 for the three VLMs (Direct GPT Query, Qwen2-Wildfire-2B and ForestFireVLM) on six-task downstream prediction when their own predicted four-tag state is provided as soft context together with the video frames. Here, the task list variable is the string from Appendix I.2.

```
Estimate downstream wildfire behavior attributes for the query video.
Use the query frames as primary evidence.
You are also given a predicted latent 4-tag wildfire state from an
earlier classifier for this same query.
Treat those predicted tags as soft context only, not ground truth.
If the predicted tags conflict with visible evidence, trust the query
frames.
Return only JSON with the requested task keys.

Predict all of these task keys:
<task list above>

Allowed labels:
- spread_fast_bin: slow_or_moderate, fast_or_very_fast
- peak_high_bin: low, high
- burnscar_high_bin: low, high
- vegetation_mean_canopy_cover_pct_median_bin: low, high
- rapid_growth_and_large: low, high
- severe_fire_behavior: low, high

Predicted latent 4-tag state for this query (soft context only):
{"wind_regime_tag":"...","moisture_regime_tag":"...","slope_regime_tag
":"...","vegetation_regime_tag":"..."}

Return only JSON with exactly these keys and string labels.
{"spread_fast_bin":"slow_or_moderate","peak_high_bin":"low","
burnscar_high_bin":"low","vegetation_mean_canopy_cover_pct_median_bin
":"low","rapid_growth_and_large":"low","severe_fire_behavior":"low"}
```

Message assembly.

```
[text]   <downstream-conditioned prompt or prompt + retry suffix>
[text]   Frame 1.
[image]  <frame 1>
...
[text]   Frame n.
[image]  <frame n>
```

I.8 Multi-Agent Judge Prompts

The templates below mention both disagreement-only and audit-all use. The reported fixed pipeline follows the disagreement-only routing described in Appendix C: only disputed task keys are requested, while unanimous fields are retained. Terms such as "calibrated" in these prompts express requested behavior, not an empirically validated uncertainty calibration. The prompts are retained verbatim to document the evaluated system.

Judge System Prompt

```
You are the judge agent in a multi-agent wildfire downstream inference
 pipeline. Three specialist agents have already analyzed the query: a
direct visual specialist, a lookup-prior specialist, and a combined-
memory specialist. Your job is to audit and resolve the requested
downstream tasks. These may be only disputed tasks, or an audit-all
pass. Use the query frames as primary evidence for directly visible
spread pace and obvious footprint extent. Use the top-k hybrid-memory
consensus and retrieved metadata as calibrated soft prior evidence,
not ground truth. Do not merely vote between specialists. Apply task-
specific reliability: the direct visual specialist is useful for
spread and for conservative negative evidence, but area-threshold
```

```
outcomes and canopy cover should lean more on strong top-k memory
consensus. You may override unanimous specialist outputs when they
conflict with strong memory consensus or clear visual evidence. For
high labels on peak area, final burnscar, rapid growth, or severe
behavior, require either broad visible evidence or strong memory
support. Return only JSON matching the requested schema.
```

Judge Context Block The judge received only the task keys where the three specialists disagreed. All six labels and reasons from every specialist were still included as context.

```
Audit and resolve these requested tasks:
- <task_key_1>: <task description from TASK_SPECS>
- <task_key_2>: <task description from TASK_SPECS>
...

Previously agreed tasks, if any. Treat them as context, not immutable
labels during audit-all runs:
<json.dumps(agreed_labels, ensure_ascii=True, sort_keys=True)>

Specialist outputs follow. These are inputs, not votes:
{
  "Direct visual specialist": {
    "predictions": {
      "spread_fast_bin": "<label>",
      "peak_high_bin": "<label>",
      "burnscar_high_bin": "<label>",
      "vegetation_mean_canopy_cover_pct_median_bin": "<label>",
      "rapid_growth_and_large": "<label>",
      "severe_fire_behavior": "<label>"
    },
    "reasons": {
      "spread_fast_bin": "<reason>",
      "peak_high_bin": "<reason>",
      "burnscar_high_bin": "<reason>",
      "vegetation_mean_canopy_cover_pct_median_bin": "<reason>",
      "rapid_growth_and_large": "<reason>",
      "severe_fire_behavior": "<reason>"
    }
  },
  "Lookup-prior specialist": {
    "predictions": {...},
    "reasons": {...}
  },
  "Combined-memory specialist": {
    "predictions": {...},
    "reasons": {...}
  }
}

Top-k hybrid-memory consensus follows. This is more reliable than any
single retrieved item, but still only a soft prior:
{
  "candidate_count": 5,
  "task_consensus": {
    "spread_fast_bin": {
      "winner": "<label>",
      "confidence": <float>,
      "distribution": {
        "slow_or_moderate": <float>,
        "fast_or_very_fast": <float>
      }
```

```
    },
    "peak_high_bin": {
      "winner": "<label>",
      "confidence": <float>,
      "distribution": {
        "low": <float>,
        "high": <float>
      }
    },
    "burnscar_high_bin": {
      "winner": "<label>",
      "confidence": <float>,
      "distribution": {
        "low": <float>,
        "high": <float>
      }
    },
    "vegetation_mean_canopy_cover_pct_median_bin": {
      "winner": "<label>",
      "confidence": <float>,
      "distribution": {
        "low": <float>,
        "high": <float>
      }
    },
    "rapid_growth_and_large": {
      "winner": "<label>",
      "confidence": <float>,
      "distribution": {
        "low": <float>,
        "high": <float>
      }
    },
    "severe_fire_behavior": {
      "winner": "<label>",
      "confidence": <float>,
      "distribution": {
        "low": <float>,
        "high": <float>
      }
    }
  },
  "weighted_metric_means": {
    "peak_burning_area_m2": <float>,
    "final_burnscar_area_m2": <float>,
    "max_radial_spread_mps": <float>,
    "vegetation_mean_canopy_cover_pct": <float>
  },
  "top_candidates": [
    {
      "item_id": "<train/sample_id>",
      "hybrid_score": <float>,
      "retrieved_tags": {...},
      "lookup_prior": {
        "spread_fast_bin": "<label>",
        "peak_high_bin": "<label>",
        "burnscar_high_bin": "<label>",
        "vegetation_mean_canopy_cover_pct_median_bin": "<label>",
        "rapid_growth_and_large": "<label>",
        "severe_fire_behavior": "<label>"
      },
      "environment": {...},
      "fire_behavior": {...}
```

```
    }
  ]
}

Decision guidance: prefer the query frames for visible spread evidence
; use strong memory consensus for less directly visible outcomes such
as canopy and final burnscar; do not default to any single specialist.
 If the direct visual specialist is conservative but memory strongly
supports a high threshold outcome, explain why you accept or reject
that memory prior.
```

Message assembly.

```
[input_text]  <judge system prompt>
[input_text]  Audit and resolve these requested tasks:
              - <only task keys where the three specialists disagree>
[input_text]  Previously agreed tasks, if any. Treat them as context,
not immutable labels during audit-all runs: <agreed_labels JSON>
[input_text]  Specialist outputs follow. These are inputs, not votes:
<specialist_payload JSON>
[input_text]  Top-k hybrid-memory consensus follows. This is more
reliable than any single retrieved item, but still only a soft prior:
<memory_consensus JSON>
[input_text]  Decision guidance: prefer the query frames for visible
spread evidence; use strong memory consensus for less directly visible
 outcomes such as canopy and final burnscar; do not default to any
single specialist. If the direct visual specialist is conservative but
 memory strongly supports a high threshold outcome, explain why you
accept or reject that memory prior.
[input_text]  Query video to adjudicate.
[input_text]  Query frame 1.
[input_image] <query frame 1, detail=low>
...
[input_text]  Query frame n.
[input_image] <query frame n, detail=low>
```

Judge Response Schema

```
{
  "type": "json_schema",
  "name": "multiagent_downstream_resolution",
  "schema": {
    "type": "object",
    "properties": {
      "<requested_task_key>": {
        "type": "object",
        "properties": {
          "label": {
            "type": "string",
            "enum": <TASK_SPECS[task_key]["label_enum"]>
          },
          "reason": {
            "type": "string"
          }
        },
        "required": ["label", "reason"],
        "additionalProperties": false
      }
    },
    "required": ["<all requested task keys>"],
```

```
    "additionalProperties": false
  }
}
```

I.8.1 LEAVE-ONE-OUT JUDGE PROMPT VARIANT

For the ablations studies shown in Table 1, the judge prompt was changed only in the specialist roster sentence. The remainder of the prompt stayed the same.

```
You are the judge agent in a multi-agent wildfire downstream inference
 pipeline. <Two specialist agents have / Specialist agents have>
already analyzed the query: <joined list of the remaining specialist
names>. Your job is to audit and resolve the requested downstream
tasks. These may be only disputed tasks, or an audit-all pass. Use the
 query frames as primary evidence for directly visible spread pace and
 obvious footprint extent. Use the top-k hybrid-memory consensus and
retrieved metadata as calibrated soft prior evidence, not ground truth.
 Do not merely vote between specialists. Apply task-specific
reliability: the direct visual specialist is useful for spread and for
 conservative negative evidence, but area-threshold outcomes and
canopy cover should lean more on strong top-k memory consensus. You
may override unanimous specialist outputs when they conflict with
strong memory consensus or clear visual evidence. For high labels on
peak area, final burnscar, rapid growth, or severe behavior, require
either broad visible evidence or strong memory support. Return only
JSON matching the requested schema.
```

J HTML REPORT GENERATION PROMPT

To generate the narrative in Figure 6, we use GPT to summarize the predicted environmental tags and six report-field labels. The structured inference records retain retrieval and routing information, while the narrative prompt below deliberately omits these internal details. Its use of "operational" describes the requested writing style, not an established deployment capability. Reports and response suggestions are illustrative research outputs requiring expert validation before operational use.

J.1 SYSTEM PROMPT FOR REPORT GENERATION

```
You write concise operational wildfire reports. Follow the user's
schema exactly and output valid JSON only.
```

J.2 USER PROMPT TEMPLATE FOR REPORT GENERATION

The user prompt is instantiated per case with the predicted location, severity label, four latent tags, and six downstream task labels.

```
Write a concise wildfire operational report for one case.

Use the downstream task predictions and four-tag latent state exactly
as the evidence source.
Do not mention ground truth, validation, agents, routing, retrieval,
memory, or model internals.
Do not mention confidence scores.
Write in a polished, professional style suitable for a final user-
facing wildfire report.
```

```
Location: Longitude <lon>, Latitude <lat>
Severity: <severity_label>

Predicted 4-tag state:
- Wind: <tag label> (confidence <c>)
- Moisture: <tag label> (confidence <c>)
- Terrain: <tag label> (confidence <c>)
- Vegetation: <tag label> (confidence <c>)

Downstream task predictions:
- Fast Spread Risk: <prediction text>
- High Peak Burning Area: <prediction text>
- High Final Burnscar: <prediction text>
- High Canopy Cover: <prediction text>
- Rapid Growth + Large Burn: <prediction text>
- Severe Fire Behavior: <prediction text>

Return strict JSON with exactly these keys:
- fire_spread_prediction
- damage_prediction
- suggestion

Each value must be one compact paragraph of 2-4 sentences.
```

Message assembly.

```
[system]  You write concise operational wildfire reports. Follow the
user's schema exactly and output valid JSON only.
[user]    <instantiated report-generation prompt above>
```

Output contract.

```
{
  "fire_spread_prediction": "<compact 2-4 sentence paragraph>",
  "damage_prediction": "<compact 2-4 sentence paragraph>",
  "suggestion": "<compact 2-4 sentence paragraph>"
}
```

J.3 Structured Report Example

Figure 6 shows a representative report from the complete pipeline, combining visual evidence, predicted environmental tags, downstream predictions, retrieved metadata, natural-language reasoning, and response suggestions. These reports illustrate the output format rather than establish operational validity.

K Structured Output Contracts

K.1 Four-Tag Classification Schema

```
{
  "wind_regime_tag":        {"label": "low|high", "reason": "<short
visible-evidence sentence>"},
  "moisture_regime_tag":    {"label": "low|high", "reason": "<short
visible-evidence sentence>"},
  "slope_regime_tag":       {"label": "low|high", "reason": "<short
visible-evidence sentence>"},
```

```
  "vegetation_regime_tag":  {"label": "low|high", "reason": "<short
visible-evidence sentence>"}
}
```

K.2 Candidate-Reranking Schema

```
{
  "candidates": [
    {"item_id": "C01", "match_score": <0-100 number>, "reason": "<
short reason>"},
    {"item_id": "C02", "match_score": <0-100 number>, "reason": "<
short reason>"},
    ...
  ]
}
```

K.3 Downstream Specialist Agent / Judge Schema

```
{
  "spread_fast_bin": {"label": "slow_or_moderate|fast_or_very_fast", "
reason": "<short reason>"},
  "peak_high_bin": {"label": "low|high", "reason": "<short reason>"},
  "burnscar_high_bin": {"label": "low|high", "reason": "<short reason
>"},
  "vegetation_mean_canopy_cover_pct_median_bin": {"label": "low|high",
 "reason": "<short reason>"},
  "rapid_growth_and_large": {"label": "low|high", "reason": "<short
reason>"},
  "severe_fire_behavior": {"label": "low|high", "reason": "<short
reason>"}
}
```

L Cross-Generator and External UAV Evaluation

L.1 Cross-Generator Evaluation

We use LTX-2.3 + AVControl as a video-generation backbone independent of the original VACE generator. We evaluate both LTX-generated queries against VACE-generated memory and VACE-generated queries against LTX-generated memory. Results are reported in Table 5a. Our method reaches 41.00% and 44.44% accuracy, respectively, compared with 28.00% and 27.27% for the strongest direct VLM baselines in the two query sets. This tests changing the video-generation backbone, not changing the physical simulator or procedural proxy renderer. It isolates a visual-generator change from the separate real-UAV tests below.

L.2 External UAV Evaluation

Real wildfire videos jointly annotated with environmental drivers and fire behavior are scarce, motivating the simulation-based construction of labeled reference episodes. We complement the controlled physical-label evaluation with real-UAV tests using available annotations: smoke detection, fire presence, and visible fire-area assessment. These tests assess observable monitoring tasks; the annotations used here do not jointly specify the four environmental drivers needed to validate physical-driver recovery.

Boreal Forest Fire. We evaluate smoke detection using the smoke-detection labels in Boreal Forest Fire Pesonen et al. (2025). Table 23 reports accuracy, precision, recall, and F1. Our method, Claude

4, and ForestFireVLM-7B reach 100% on all four metrics; this evaluation does not distinguish among those methods.

Table 23: Boreal Forest Fire smoke detection. All entries are percentages.

Method	Accuracy	Precision	Recall	F1
Ours	100.0	100.0	100.0	100.0
GPT-5	94.4	90.0	100.0	94.7
Claude 4	100.0	100.0	100.0	100.0
ForestFireVLM-7B	100.0	100.0	100.0	100.0
Qwen2-Wildfire-2B	94.4	100.0	88.9	94.1

FireMan-UAV-RGBT. We evaluate binary fire presence on FireMan-UAV-RGBT Kularatne et al. (2024). Overall accuracy is accompanied by balanced accuracy, which assigns equal weight to fire and no-fire classes in this imbalanced evaluation. Table 24 reports the complete per-method results.

Table 24: FireMan-UAV-RGBT fire-presence detection. Scores are on a 0–1 scale. Bold indicates the highest score.

Method	Balanced accuracy	Accuracy
GPT-5 mini	0.7646	0.8173
Claude Sonnet 4.6	0.8563	0.9135
Gemini 2.5 Pro	0.8125	0.9135
ForestFireVLM-7B	0.8625	0.9231
Qwen2-Wildfire-2B	0.8375	0.7500
Ours	**0.9625**	**0.9423**

FLAME. For FLAME Shamsoshoara et al. (2021), we derive reference visible fire area by summing the fire segmentation regions. In the ranking task, each prompt provides ten videos and asks the model to rank them from the largest to the smallest fire region. Table 25 reports mean AP, largest-fire Hit@1, and fire-area NDCG. Our method has the highest NDCG and ties Claude Sonnet 4.6 on Hit@1; Gemini 2.5 Pro has the highest mean AP. These annotations concern visible fire regions, not a complete physical characterization of fire severity.

Table 25: FLAME fire-area evaluation. Scores are on a 0–1 scale. Bold indicates the highest score, including ties.

Method	Mean AP	Largest-fire Hit@1	Fire-area NDCG
GPT-5	0.930	0.658	0.947
Claude Sonnet 4.6	0.946	**0.850**	0.973
Gemini 2.5 Pro	**0.952**	0.564	0.891
ForestFireVLM-7B	0.939	0.767	0.957
Qwen2-Wildfire-2B	0.844	0.547	0.883
Ours	0.950	**0.850**	**0.984**

M Additional Visual Results

Figure 10 provides additional generated-video examples. Figure 8 reports the per-attribute comparison complementing Table 3 in the main text.



Figure 10: Additional generated wildfire video frames conditioned on 2D simulations through procedural 3D proxies.