Predicting the Needle in a Petabyte Scale Haystack: Open-Vocabulary Event Anticipation in Satellite Imagery

Shreelekha Revankar[1,2]*, **Mikhail Klassen**[1], **Creon Levit**[1], **Ash Hoover**[1], **Kavita Bala**[2], **Bharath Hariharan**[2]

[1]Planet Labs [2]Cornell University

Abstract

Satellite constellations now image the Earth daily, capturing the lifecycle of events from deforestation to urban construction. Anticipating these events before completion enables timely intervention, yet existing systems cannot jointly identify *where* a change occurs, *what* it will become, and *when* it will finish across an open vocabulary. Building such a model requires diverse event data, but events are like needles in a haystack, making manual annotation at global scale infeasible. We address this problem by representing change as the vector difference between vision-language model embeddings at distinct time points. This arithmetic approach allows us to search for semantic transformations directly in the latent space, enabling (1) an automated data engine for global event discovery without manual annotation and (2) the largest global satellite event dataset to our knowledge *SatEvents-21k* comprising 21,000+ locations across 64+ event types with 2M images from PlanetScope and Sentinel-2 spanning six continents, to train (3) a novel global-scale event anticipation model. Our model detects change with a 93.5% max F1 score, outperforms baselines 3.5× on predicted event retrieval, and forecasts completion dates with a 3-day median error, even nine months before completion. `https://github.com/ShreelekhaR/predicting-the-needle`

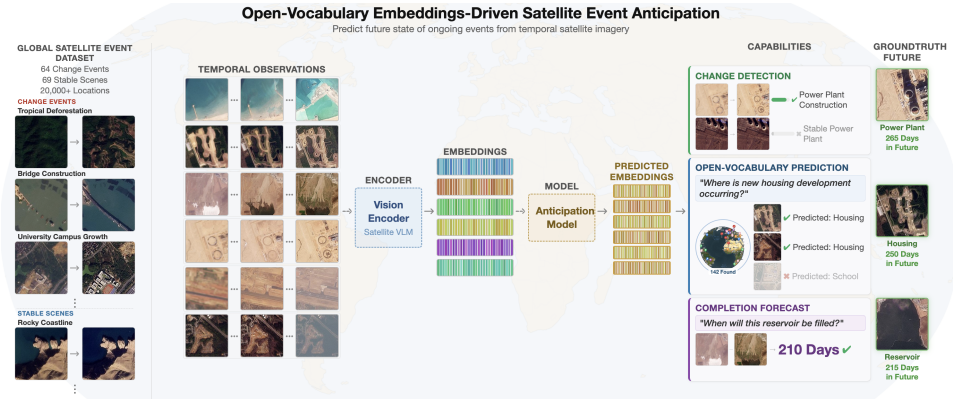


Figure 1: We present our latent-space world model. Trained on a large diverse set of global events, our model allows us to detect change events and predict their future states and completion dates.

*Corresponding Email: revankar@cs.cornell.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Satellite constellations now image the entire planet at a near-daily cadence, with public and commercial archives spanning nearly a decade of petabyte-scale records [Planet Labs PBC, 2018–2026, ESA, 2026]. These archive capture the full lifecycle of events, from urban development to deforestation. Anticipating these events prior to completion enables proactive intervention, such as alerting conservation organizations months before forest canopy is lost, or informing energy planners before a data center is constructed. Building a model capable of this anticipation requires answering three questions: *where* is a change event occurring globally, *what* will the completed state look like, and *when* will it complete, preferably in an open-vocabulary manner, without restricting to a fixed set of events or requiring task-specific annotations.

Currently no existing system satisfies all three goals, and the core challenge is data. Events we truly care about are extraordinarily rare. In a petabyte-scale archive, the majority of locations show stable, unchanging terrain. Training a model on random samples from the archive of satellite images would simply teach it to predict that nothing changes, because for most locations, nothing significant does. Manually curating annotated events at global scale is not feasible [Bergman et al., 1997, Bell et al., 2025]. For this reason existing multitemporal datasets and the models trained on them remain limited in diversity and geographic coverage [Revankar et al., 2025, Liu et al., 2022]. More general embedding-based change detection can flag that *something* changed [Klassen et al., 2025], but carries no semantic signal about *what* is changing or *where* it is headed.

To address these challenges, we propose an automated methodology to construct a diverse global event dataset without manual annotation. We leverage the fact that although we do not have open-vocabulary event detection models, we do have vision-language models (VLMs) that excel at recognizing *static* concepts in satellite imagery [Barzilai et al., 2025, Xiong et al., 2025, Li et al., 2023]. Our key insight is that we can detect and retrieve events by recognizing the “before" and “after". Concretely, similar to semantic vector arithmetic in other domains [Radford et al., 2021, Mikolov et al., 2013], we represent change as the vector difference between embeddings at two time points, where the latent space’s semantic structure replaces manual labels. For example, the difference between an *empty lot* and a *data center* captures the construction event itself.

With this dataset in hand, we draw on latent-space forecasting [Vondrick et al., 2016, LeCun et al., 2022, Assran et al., 2025], which has shown that future states can be predicted directly in embedding space with robustness to perceptual noise, a particularly valuable property for satellite imagery, which is inherently noisy [Komurcu and Petkevicius, 2026]. We build a satellite event world model that operates entirely within embedding space, ingesting a sequence of observations as a trajectory and predicting where it is headed and when it will complete. The VLM embedding space provides semantic grounding, enabling the model to answer arbitrary natural language queries about change without any task-specific fine-tuning.

In sum, our contributions are the following:

1. An **automated data engine** that discovers any described event at a global scale without manual annotation or external data sources, leveraging VLM embedding arithmetic to localize events.
2. **The largest global satellite event dataset** to our knowledge: $\sim$21,000+ locations across 64+ event types with $\sim$2M images from PlanetScope and Sentinel-2 spanning six continents. To be publicly released.
3. **A global-scale event anticipation model** trained on temporal observation sequences that detects change with 93.5% F1, outperforms baselines 3.6$\times$ on predicted event retrieval, and forecasts completion dates with a median 3-day error — up to 9 months before an event completes.

2 Related Works

Retrieval and Event Detection in Satellite Imagery Historically, satellite imagery retrieval has faced a “needle in a haystack” challenge [Bergman et al., 1997]. Various retrieval techniques were developed [Bell et al., 2025], but VLMs adapted for remote sensing [Liu et al., 2024, Barzilai et al., 2025, Mall et al., 2023, Li et al., 2023, Xiong et al., 2025] emerged as the popular solution. While most operate over static imagery, making change characterization difficult, a small number of spatiotemporal VLMs have been developed [Jain et al., 2025, Li et al., 2024]. Training depends on multitemporal datasets. Existing datasets often suffer from geographic constraints, limited classes [Mall et al., 2022, Chen and Shi, 2020], reliance on manual annotation, or dependence

on external sources like social media [Ahmad et al., 2019], databases [Uzkent et al., 2019], or news articles [Revankar et al., 2025]. Our work differs in that it requires no manual annotation, external sources, or geographic constraints, scales to arbitrary event types worldwide, and uniquely treats spatiotemporal change itself as a vector in a language-aligned latent space, queryable for open-vocabulary event retrieval.

Event Anticipation and World Models. While forecasting models like TAMMs [Guo et al., 2026] and Sat-JEPA-Diff [Komurcu and Petkevicius, 2026] explore future state prediction, they often rely on noisy, pixel-level visual reconstruction or fixed-duration and fixed-category forecasting [Irvin et al., 2024, Porta et al., 2025, Komurcu and Petkevicius, 2026]. Predicting future visual representations directly in feature space was proposed early on by Vondrick et al. [2016] to avoid the computational and noise-related pitfalls of pixel-level forecasting. A growing body of work has since explored prediction in latent space across vision and language modalities [Bain et al., 2021, LeCun et al., 2022, Jiang et al., 2024, Assran et al., 2025, Chen et al., 2025, Zhang et al., 2025, Zhong et al., 2025, Zou et al., 2026]. Most recently, Assran et al. [2023] and Chen et al. [2025] have demonstrated that models can learn high-level world dynamics by predicting masked or future embeddings. Komurcu and Petkevicius [2026] corroborate that this latent-only approach is particularly robust to the perceptual noise inherent to satellite imagery. However, these works largely operate on fixed event vocabularies or specific prediction horizons defined at training time. To our knowledge, none prior to us have used change vectors between scene embeddings as a signal for open-vocabulary event anticipation across arbitrary timescales.

3 A Large-Scale Satellite Event Dataset: *SatEvents-21k*

To train a model capable of predicting future events from satellite imagery, we need a training dataset that captures the full diversity of events that may be of interest. Building such a training dataset is challenging, because in most locations of the planet at most times, nothing is changing. Therefore, if we simply sample locations and times uniformly at random, our dataset will be dominated by no change and many important events of interest will be missed.

We propose instead to start with a long list of diverse events, and retrieve locations and times where these events have occurred. To do so we leverage LLMs, using them to create a list of events, including text descriptions of the "before" and "after" states. This is done without any imagery and results in a list of events that *should be* visible through satellite imagery.

Unfortunately, retrieving locations and times where a particular kind of event has occurred is challenging because we do not have open-vocabulary event retrieval models with sufficiently diverse vocabularies. Instead, we propose to leverage satellite image VLMs that recognize *static* concepts [Liu et al., 2024, Barzilai et al., 2025]. Concretely, given an event description such as *Solar Farm Installation*, we use an LLM to produce a description of what the location would look like before: "*Open empty field*" and after: "*Grid of dark rectangular solar panels*". We can then use static VLMs to recognize the before and after descriptions. We consider two strategies to do so:

Strategy 1: Intersection Search. We retrieve locations that match both the before description at one time point and the after description at a later time point. In particular, we construct a database containing the embeddings e^i_{before} and e^i_{after} of each location i at two different time points, t_{before} and t_{after}. Given before and after text descriptions $(d_{\text{before}}, d_{\text{after}})$ for an event, we query the database separately at each timepoint:

$$\mathcal{R}_{\text{before}} = \text{top-}k\left(\{\text{sim}(M^t(d_{\text{before}}), e^i_{\text{before}})\}_{i=1}^n\right) \tag{1}$$

$$\mathcal{R}_{\text{after}} = \text{top-}k\left(\{\text{sim}(M^t(d_{\text{after}}), e^i_{\text{after}})\}\right) \tag{2}$$

where M^t is the text encoder. Locations appearing in both result sets are candidates:

$$\mathcal{R}_1 = \mathcal{R}_{\text{before}} \cap \mathcal{R}_{\text{after}} \tag{3}$$

This ensures retrieved locations both started in the expected initial state and ended in the expected final state. Thus, to retrieve locations where a ship entered the pier, we intersect a search for "*empty pier*" with "*pier with ship*" as shown in Figure 2.

This *intersection search* has a failure mode: when the same scene type appears in both descriptions, it may retrieve locations matching both without any change occurring. For example, querying "*park*

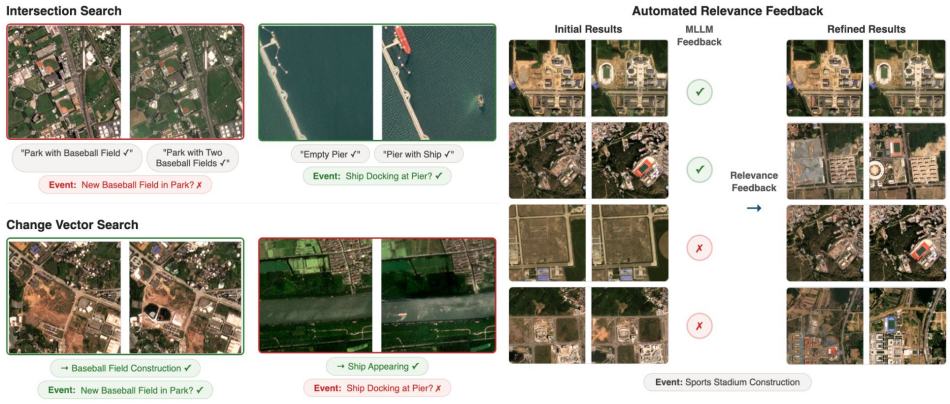


Figure 2: Change-Aware Retrieval and Automated Relevance Feedback. *(Left)* Intersection search enforces scene context at both timepoints but can produce false positives when a scene already matches the after-state without change. Change vector search finds the correct transformation but often loses context. *(Right)* ARF connects both: an MLLM scores initial intersection results, confirmed positives form averaged probe Δe_{probe} for refined re-retrieval.

with baseball field"→"*park with two baseball fields*" may surface a park with two fields at both timepoints, which is a false positive. We therefore consider a second strategy as well, outlined below.

Strategy 2: Direct Change Vector Search. Alternatively, we can mathematically represent the expected change using vector arithmetic:

$$\Delta e_{\text{query}} = M^t(d_{\text{after}}) - M^t(d_{\text{before}}) \tag{4}$$

This change vector encodes the semantic transformation. We query the database for similar changes:

$$\mathcal{R}_2 = \text{top-}k\left(\{\text{sim}(\Delta e_{\text{query}}, \Delta e^i)\}_{i=1}^n\right) \tag{5}$$

where $\Delta e^i = e^i_{\text{after}} - e^i_{\text{before}}$ is the change in embedding space at the i-th location between the before and after time points. This retrieves locations exhibiting similar semantic transformations regardless of specific visual appearance.

This change-vector search is more discriminative, but loses scene-level context: e.g., a "*ship appearing*" vector retrieves ships anywhere, not only at piers.

Thus both strategies have their flaws. We therefore propose to combine the two strategies using the idea of relevance feedback.

Automated Relevance Feedback (ARF) We combine the complementary strengths of the two strategies above as follows. We use intersection search to produce a list of candidates, and then use a commercial MLLM to identify confirmed positive examples $\mathcal{P}$ from the top-k intersection candidates. We then compute a mean change probe following the Rocchio algorithm [Rocchio Jr, 1971]:

$$\Delta e_{\text{probe}} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \Delta e^i \tag{6}$$

We then use this averaged change vector to perform change vector search (instead of constructing Δe_{query} from text descriptions alone). This averaged change vector captures the common semantic change pattern across confirmed positives, giving us a refined Δe_{probe} for re-retrieval.

This iterative refinement of retrieval probes using confirmed positive examples is known as *relevance feedback* [Rocchio Jr, 1971], which has more recently been shown effective for vision-language retrieval [Khaertdinov et al., 2026].

Dataset Construction With these retrieval strategies and ARF, our full pipeline for retrieving events at scale is as follows:

1. **Generate event descriptions:** MLLM generates event types with before/after descriptions $(d_{\text{before}}, d_{\text{after}})$

2. **Intersection retrieval:** Perform intersection search for top-k candidates
3. **ARF round 1:** MLLM identifies positive examples, compute Δe_{probe}
4. **Change vector retrieval:** Re-query using Δe_{probe} for refined candidates
5. **ARF round 2:** MLLM identifies final positive examples, save to dataset

This pipeline enables automated discovery across thousands of positive event types without manual annotation. We also apply this process to discover negative examples, i.e. stable locations where no significant change occurs, by setting $d_{\text{before}} = d_{\text{after}}$ (e.g., "dense forest" for both timepoints). This produces a balanced dataset of "positive" events and "negative" static scenes. We capture 26,039 locations spanning 64 event classes and 69 stable scene categories across 6 continents, filtered to 21,362 locations with a sufficient number of observations.

Event Completion Detection Our retrieval pipeline discovers locations that changed between two timepoints. However, many applications would require us to know or predict *when* the change completes. We define completion as the date beyond which anticipation provides no additional value, i.e. when the location can be retrieved reliably using its "after" state description. We devise an method to compute exactly that.

For each candidate location, we download and embed imagery between our previously retrieved two timepoints (which in our case is 1 year). Then we compute a differential change similarity signal:

$$S_\Delta(t) = \text{sim}(e_t, e_{\text{after}}) - \text{sim}(e_t, e_{\text{before}}) \tag{7}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, e_t is the embedding at time t, and e_{before}, e_{after} are the retrieved *before* and *after* embeddings from our event retrieval process. The metric $S_\Delta(t)$ is negative when the scene resembles the initial state and transitions to positive as the event completes. The zero-crossing of S_Δ indicates when the location appears more similar to the completed state than the initial state. Critically, this requires no additional supervision beyond the embeddings between the two retrieval timepoints. We visualize several examples in Figure 8, where the S_Δ signal exhibits sharp zero-crossings across diverse event types, confirming that our completion dates capture genuine perceptual transitions. Full implementation details and visualizations are in Appendix F.

Human Verification. To quantify our data pipeline reliability and produce a human-certified test set, annotators reviewed georeferenced before-and-after image pairs and voted *correct*, *incorrect*, or *unsure*. Classes where fewer than a majority of samples received a *correct* vote were removed or re-prompted. The passing samples form our verified positive test set. Full details in Appendix G.

4 A Model for Anticipating Satellite Events

With our dataset in hand, we train a satellite event world model to predict the future at any location. Since multi-temporal satellite sequences labeled with open-vocabulary events are scarce, while static satellite VLMs are trained on abundant supervision and have a rich semantic space, we predict in the embedding space of a frozen VLM [Barzilai et al., 2025]. Because successive embeddings share a spatial coordinate, the trajectory between them represents the physical transformation at that location, and our model learns where that trajectory is heading.

We train a model M^{pred} that takes in a sequence of embeddings $\{e_0, \ldots, e_n\}$ as well as a time horizon h and predicts the embedding at that location h days from the last observation:

$$e_{\text{pred}}(h) = M^{\text{pred}}(\{e_0, \ldots, e_n\},\, h) \tag{8}$$

where e_0 is the embedding of the oldest observation, e_n is the most recent, and $h \in \mathbb{R}^+$ is the queried prediction horizon in days. This formulation allows the model to produce different predictions for different horizons from the same input sequence.

Predicting time-to-completion In addition to predicting what a location would look like at a given time in the future, for locations going through events, we may also want to know when the event will complete. To provide this additional capability, we also add an additional head to our model that predicts the time-to-completion, $t_{\text{pred-comp}}$. Thus, our model predicts both the future embedding at an input time horizon, as well as the predicted time to completion for any ongoing events:

$$e_{\text{pred}}(h), t_{\text{pred-comp}} = M^{\text{pred}}(\{e_0, \ldots, e_n\},\, h) \tag{9}$$

4.1 Model architecture

Our model M^{pred} is a causal transformer that ingests the full observation sequence prepended with a learnable before-state token: $[\texttt{BEFORE}, e_0, \ldots, e_n]$. Each token is augmented with temporal position encodings derived from its observation date. The transformer processes this sequence with causal self-attention, producing a context-aware representation of the location's temporal trajectory.

The queried future horizon h (in days) is encoded via a sinusoidal temporal encoding into a query token. This query token then attends to the encoded sequence through a cross-attention layer, producing a horizon-conditioned representation from which the final prediction e_{pred} is decoded via a feedforward head. The architecture supports arbitrary prediction horizons.

To make temporal dynamics explicit, we augment each embedding with its forward *velocity*—the change to the next observation:

$$\tilde{e}_i = W \cdot [e_i;\ e_{i+1} - e_i] + b \tag{10}$$

where $[\cdot;\cdot]$ denotes concatenation and W, b are learned projection parameters followed by layer normalization. This is analogous to providing optical flow [Horn and Schunck, 1981]: rather than requiring the model to infer rates of change implicitly from consecutive embeddings, we provide the change signal directly, allowing the model to distinguish accelerating from decelerating events.

4.2 Training Objective

We sample locations from our dataset, which includes both locations where events have occurred as well as "negative" locations where nothing changes. For each location, we sample a sequence of time points and corresponding satellite images before the event occurred, as well as a horizon h for which the model should predict the embedding. We train M^{pred} to predict the embedding at the input time horizon h, supervised against the actual observed embedding at that time horizon. We combine a directional prediction term (based on cosine similarity) with magnitude regularization:

$$\mathcal{L}_{\text{pred}} = \underbrace{(1 - \text{sim}(e_{\text{pred}}(h),\ e_{\text{gt}}(h)))}_{\text{completion state alignment}} + \lambda \cdot \underbrace{(\|e_{\text{pred}}(h)\| - \|e_{\text{gt}}(h)\|)^2}_{\text{magnitude regularization}} \tag{11}$$

where $e_{\text{gt}}(h)$ is the observed embedding at time horizon h and λ is a weighting coefficient. The cosine loss encourages directional alignment between predicted and ground-truth embeddings, while the magnitude loss preserves embedding scale.

Time-to-Completion Head. In addition to predicting *what* the completed state will look like, we jointly train a scalar regression head to predict *when* the event will complete:

$$\mathcal{L}_{\text{time}} = \text{MSE}\left(t_{\text{pred-comp}},\ t_{\text{comp}}\right) \tag{12}$$

where $t_{\text{pred-comp}}$ is the predicted time-to-completion (in days) and t_{comp} is the ground-truth lead time. This head branches from the same hidden representation as the embedding predictor, adding negligible parameters. The full training objective is:

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \mu \cdot \mathcal{L}_{\text{time}} \tag{13}$$

with $\mu = 0.1$. The time head is trained only on positive (event) samples; negative (stable) locations contribute only to $\mathcal{L}_{\text{pred}}$.

This allows us to directly output $t_{\text{pred-comp}}$, an estimate of the number of days until event completion. Combined with the known observation date, this yields an absolute predicted completion date.

4.3 Inference – Event anticipation

Once trained, we can use our model for event anticipation. Concretely, given an event description, we consider the task of retrieving all geographical locations where the event is *predicted to happen in the future* (in a time horizon h). Inference consists of two separate stages:

Stage 1: Anticipation. For every location i in our database, we predict the future embedding h days later using the observations till now, obtaining $e_{\text{pred}}(h)$ and $t_{\text{pred-comp}}$ for all locations. This stage takes **no text input**, so its outputs are query-independent and computed once per location and horizon.

Change detection: Of course, when anticipating events, we only want to retrieve locations where a change is actually taking place. We therefore pre-filter based on the predicted change vector

Method	Data	Embed	Seen			Unseen			Geo Seen			Geo Unseen		
			max F1	mAP5	mAP10	*max* F1	mAP5	mAP10	*max* F1	mAP5	mAP10	*max* F1	mAP5	mAP10
Oracle	S2	RemoteCLIP	—	.500	.481	—	.500	.463	—	.306	.286	—	.429	.420
Oracle	S2	RS-SigLIP	—	.489	.452	—	.700	.641	—	.333	.313	—	.714	.673
Oracle	PS	RemoteCLIP	—	.565	.537	—	.700	.680	—	.317	.286	—	.571	.559
Oracle	PS	RS-SigLIP	—	.715	.677	—	.800	.802	—	.656	.595	—	.714	.717
Static	S2	**RS-SigLIP**	—	.154	.125	—	.333	.275	—	.096	.079	—	.071	.042
Static	PS	**RS-SigLIP**	—	.155	.142	—	.389	.359	—	.126	.089	—	.000	.002
TeoChat	—	—	.293	.005	.006	.215	.011	.014	.225	.027	.015	.343	.057	.036
SAT-JEPA-DIFF	—	—	—	.035	.025	—	.133	.109	—	.014	.008	—	.000	.000
Gemini 3 Flash	—	—	.482	.110	.086	.273	.309	.266	—	.140	.085	—	.076	.080
Ours	S2	RemoteCLIP	.421	.385	.369	.174	.179	.164	.287	.027	.039	.151	.286	.275
Ours	S2	RS-SigLIP	.499	.148	.158	.222	.254	.266	.292	.071	.071	.153	.143	.129
Ours	PS	RemoteCLIP	.656	.503	.466	.389	**.520**	**.433**	.473	.166	.133	.355	.363	.344
Ours (w/o vel.)	PS	RS-SigLIP	**.954**	.511	.488	**.878**	.400	.382	**.927**	.256	.238	**.881**	.286	.271
Ours	**PS**	**RS-SigLIP**	.935	**.533**	**.515**	.859	.400	.380	.919	**.348**	**.323**	.868	**.429**	**.386**

Table 1: Change Detection and Event Anticipation Results. Evaluated on 43,533 test samples. Seen: 54 event classes, Unseen: 10 held-out classes, Geo: geographic holdout regions. Oracle is the upper bound with perfect change detection. Our model with PlanetScope(PS) imagery and RS-SigLIP embeddings achieves the strongest retrieval performance across all settings compared against our architecture using RemoteCLIP and Sentinel-2 (S2) variations.

Method	1w–1mo	1–3mo	3–6mo	6–9mo	9mo+
Oracle (PS, RS-SigLIP)	.522	.582	.603	.542	.472
Oracle (PS, RemoteCLIP)	.483	.536	.517	.448	.239
Static (PS, RS-SigLIP)	.113	.085	.053	.039	.014
Static (PS, RemoteCLIP)	.239	.247	.166	.111	.024
TeoChat	.000	.008	.004	.002	.001
SAT-JEPA-DIFF	.024	.014	.006	.005	.001
Gemini 3 Flash	.051	.060	.048	.029	.008
Ours (PS, RemoteCLIP)	.275	.300	.286	.244	.144
Ours	**.342**	**.351**	**.354**	**.330**	**.245**

Table 2: Retrieval Performance by Lead Time (mAP@10). Our model maintains strong anticipation even at 9+ month horizons where all baselines collapse.

Method	r	Mean	Med.	≤7d	≤14d	≤30d	≤60d
TeoChat	−0.133	97d	80d	11.1%	16.9%	30.9%	58.7%
Gemini 3 Flash	−0.308	122d	102d	6.7%	9.9%	20.6%	31.9%
Ours (S2, RemoteCLIP)	0.970	17d	8d	44.3%	68.4%	83.5%	93.0%
Ours (S2, RS-SigLIP)	0.992	10d	6d	54.7%	75.4%	94.8%	99.7%
Ours (PS, RemoteCLIP)	0.996	7d	5d	72.1%	88.7%	99.0%	99.9%
Ours (PS, RS-SigLIP)	**0.997**	**5d**	**3d**	**76.8%**	**92.1%**	**98.9%**	**99.9%**

Table 3: Time-to-Completion Prediction. Pearson correlation (r), mean/median absolute error, and percentage within day-error windows. Our model predicts completion dates with median 3-day error and r=0.997.

magnitude $\|\Delta e^i_{\text{pred}}\|$. Locations where this predicted change magnitude is less than a threshold ϵ are not considered for retrieval.

$$\Delta e^i_{\text{pred}}(h) = e^i_{\text{pred}}(h) - e^i_n \tag{14}$$

Stage 2: Retrieval. For any set of events, we use an LLM to produce "after" text descriptions d_{after}. With these descriptions, we compute their text embeddings $M^t(d_{\text{after}})$ via the frozen VLM's text encoder. We can now score and rank each remaining location based on the cosine similarity between the predicted image embedding and query "after" text embedding:

$$\mathcal{R}_{pred} = \text{top-}k\left(\{\text{sim}(M^t(d_{\text{after}})), e^i_{\text{pred}}(h)\}\right) \tag{15}$$

This is a nearest-neighbor lookup in the shared VLM embedding space. Retrieving a new event category costs only one text encoding and needs no rerun of Stage 1.

5 Experimental Setup

5.1 Embedding Models

We use two satellite Vision-Language Models as frozen feature extractors:

1. **RS-SigLIP** [Barzilai et al., 2025] (1152-dim): A SoTA satellite VLM from Google Research Earth AI Remote Sensing Foundations suite, available to researchers.
2. **RemoteCLIP Fine-tuned** [Liu et al., 2024] (768-dim): Off-the-shelf public satellite VLMs performed poorly on our event taxonomy. We fine-tuned RemoteCLIP with LoRA for 2 epochs on before/after image-text pairs to bring it to a usable retrieval baseline(Table 4).

Neither VLM receives sequential data, both are trained only on static image-text pairs. All temporal reasoning is learned by our anticipation model over frozen embeddings. We additionally compare several open-source satellite VLMs on the task of event retrieval in Appendix C.

5.2 Event Generation

We utilize Gemini 3.0 Pro [Gemini Team, Google, 2023] to generate a diverse, open-vocabulary taxonomy of satellite events. For each event, the model generates natural language descriptions of the pre- and post-conditions, which serve as the semantic targets for our latent-space arithmetic.

5.3 Prediction Models

- **Satellite Event Anticipator (Ours):** A transformer-based architecture trained to predict the future latent $e_{\text{pred}}(h)$ from partial observations $\{e_0, \ldots, e_n\}$ and a queried horizon h. Ablation study on backbone and additional implementation details in Appendices D and H.
- **TeoChat [Irvin et al., 2024]:** Satellite-specific multimodal LLM.
- **Gemini 3 Flash [Gemini Team, Google, 2023]:** A frontier multimodal LLM.
- **Sat-JEPA-Diff [Komurcu and Petkevicius, 2026]:** A diffusion-based satellite image prediction model.
- **Static (No-Change):** Predicts that nothing has changed. It uses the most recent observation as the future prediction ($e_{\text{pred}} = e_n$).
- **Oracle:** This represents the upper-bound for prediction. It uses the true future embedding (e_{gt}) for locations where an event occurs, combined with perfect change detection.

For the MLLMs, Gemini and TEOchat are prompted to identify whether change is occurring, classify event type, and estimate days to completion. We embed the classification outputs to calculate retrieval metrics using RS-SigLIP text encoder. For SAT-JEPA-DIFF, we provide the most recent observation in each sequence and embed its predicted image with RS-SigLIP image encoder for retrieval comparison. Unlike our model, it accepts no temporal context, only a single input frame.

5.4 Benchmark

Train/Test Splits. Our dataset *SatEvents-21k* contains 177,241 samples from 21,362 unique locations, split by S2 [Veach and Google, 2017] cell ID with zero location overlap between splits. Training uses 132,908 samples (64,608 positive, 68,300 stable). Testing uses 43,533 samples: 25,368 from seen classes (11,275 positive, 14,093 stable) plus 18,965 out-of-vocabulary samples (3,187 positive from 10 randomly sampled held-out event classes) to evaluate generalization to unseen event types. Three geographic holdout regions containing 517 locations (4,510 samples) test spatial generalization. We include a map visualizing the dataset and its subsets in Figure 7, Appendix E.

Temporal Event Sampling. For each positive location, we sample at four fixed lead times before event completion: 1, 3, 6, and 9 months, with 5 additional random lead times, ensuring coverage across different stages of event progression. Each sample consists of the weekly embedding sequence up to the cutoff date, with a median of 19 input embeddings per sample with PlanetScope (PS) MCLA. We use the same locations and lead times with Sentinel-2 (S2); however, due to S2's lower revisit cadence (5 days vs daily for PS), the median sequence length is 14 embeddings per sample. At test time, all models are evaluated on the same 43,533 samples with identical locations and lead times across both sensors, with S2 naturally exhibiting gaps for some observation periods.

5.5 Metrics

We evaluate using max F1 scores for change detection (thresholding predicted change magnitude $\|e_0 - e_{\text{pred}}\|$ at $\epsilon \approx 0.245$), mean average precision (mAP@k) for event and predicted event retrieval (ranking test samples by cosine similarity to *after* text queries, with undetected samples scored $-\infty$), and Pearson correlation r for completion date estimation.

To test robustness and statistical significance of our results we perform a bootstrapping experiment by subsampling within our test set 1,000 times, and a permutation test on the per-class AP between the static baseline and our model using 10,000 permutations. We also report false-positive rate (FPR), the fraction of stable samples whose predicted change magnitude exceeds the max-F1 threshold. We assess this per prediction horizon, and per land-cover class, and assess expected false alarms per 1000 km^2.

6 Results

Open Vocabulary Event Detection and Anticipation Performance. As seen in Table 1, our best model achieves a detection F1 of 0.935 and mAP@5 of 0.533, a 3.4$\times$ improvement over the Static baseline (0.155). At global monitoring scale, the vast majority of locations are stable; our change magnitude gating achieves F1 of 0.935, effectively filtering negatives before retrieval. The model generalizes both to held-out event types (10 unseen classes) and unseen geographic regions (Geo Unseen mAP@5 of 0.429). To verify robustness, we bootstrap 1000 test set subsamples and find the 90% confidence interval spans ±0.092 mAP across all 64 seen and unseen classes, and our permutation test of 10,000 permutations on the per-class AP between the static baseline and our model confirms statistical significance at $p < 0.001$. SAT-JEPA-Diff performs below the Static

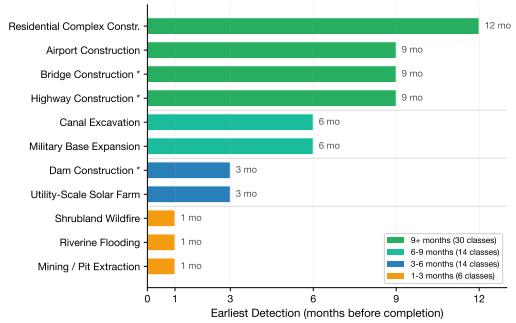


Figure 3: Earliest detection horizon per event class. * denotes classes unseen during training.

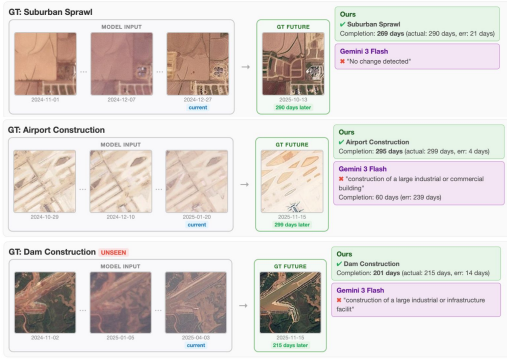


Figure 4: Qualitative examples of our model. Additional examples in Appendix B.

baseline at all horizons, demonstrating that a single input frame cannot capture the trajectory of an ongoing event. Gemini 3 Flash, despite processing the full image sequence, also underperforms, demonstrating that even frontier MLLMs struggle with the dynamics of satellite events.

False Positive Rates. Our overall FPR is 5.56%. In PlanetScope imagery, a 1000 km² area involves 2,200 image chips. With our average FPR rate of 5.56% at our reported max F1 threshold, this translates to 123 false alarms per 1000 km².

To assess FPR as a function of prediction horizon (time). A natural concern is that longer-horizon predictions would produce more false alarms because uncertainty compounds. To test this, we ran inference on 5,000 randomly sampled stable-scene test samples at four query offsets: 30, 90, 180, and 270 days into the future, holding the input sequence fixed. FPR is essentially invariant to horizon: 5.56% at 30 days, 5.56% at 90 days, 5.56% at 180 days, and 5.58% at 270 days, with only 1 of 5,000 samples flipping across the threshold from the shortest to the longest lookahead. Extending the anticipation horizon does not inflate the false-alarm rate on stable land, because the model was trained to predict "no change" for stable scenes regardless of the queried date.

To assess FPR as a function of stable-scene land-cover class. We computed per-class FPR on the negative samples pool. Classes with visually uniform content achieve 0.0% FPR (Open Ocean, Stable Forest, Salt Flats, Seasonal Grassland). The highest per-class FPR is Coastal Beach at 43.2%. We note this may partly reflect a reasonable model behavior rather than a pure error, since beaches change across tidal cycles, seasonal erosion, and storms.

Forecasting Horizon and Lead Time. Table 2 analyzes retrieval across various lead times. Our model maintains a stable predictive signal across all horizons: at 6-9 months, it achieves 8.5× the performance of the static baseline. Even beyond 9 months, the model retains a meaningful signal (0.245). Both SAT-JEPA-Diff and Gemini degrade to near-zero at all horizons, performing below the Static baseline.

How early we can detect an event depends heavily on event type. As seen in Figure 3), large-scale civil infrastructure is detectable up to 12 months before completion, while rapid-onset natural events such as wildfires and flooding allow only 1 month of anticipation. This trend holds even for novel classes unseen during training (Bridge and Highway Construction, Dam Construction).

Time-to-Event Accuracy. A robust world model should not only predict *what* will happen, but *when*. Our model additionally predicts event completion dates via a scalar regression head (Table 3). On PlanetScope imagery, it achieves a Pearson correlation of r=0.997 with median absolute error of 3 days and 92.1% of predictions within 14 days. On Sentinel-2, with sparser temporal coverage, it still achieves r=0.992 with median error of 6 days.

MLLM Performance & Scalability. While commercial MLLMs demonstrate competitive classification accuracy on detected changes, they achieve substantially lower retrieval mAP because they produce a single text answer per location rather than a rankable embedding. They also fail to anticipate satellite events at long horizons and cannot predict completion timelines. Practically, MLLMs also require inference with the full sequence per location per query, whereas our approach retrieves across all locations via vector similarity, making repeated queries feasible at scale.

Robustness to Noise. We apply no explicit cloud-masking, atmospheric correction, or pixel-level denoising to the model's inputs. Our sequences have inconsistent gaps in observation cadence, and include visible snow cover, cloud haze, and partial occlusions. By operating in VLM embedding space rather than pixel space, our model avoids modeling pixel-level noise. Images with noise or sensor artifacts still produce meaningful embeddings, and within a temporal sequence, noisy frames are naturally downweighted relative to clearer observations. This robustness to low-level noise is a well-studied property of joint-embedding methods. JEPA and JEPA-family objectives have been shown to prioritize abstract, slowly-varying semantic features over unpredictable pixel-level detail [Sobal et al., 2022, Littwin et al., 2024].

Limitations. Our data engine bypasses common geographic and categorical limits but currently assumes a single dominant event between endpoints. Consequently, intermediate transitions, such as an empty lot becoming a parking area before becoming a shopping mall, are not captured. Furthermore, while this study uses a one-year timespan, the same process can scale from daily to multi-year cadences as embeddings archives grow. Additionally we recognize that the term "completion" can be ambiguous. We compare our embeddings based completion detection against human completion labels in Appendix J.Importantly, the world anticipation model's performance is capped by its VLM embeddings, as concepts foreign to the VLM remain unrecognized in our label-free training. However, we show in Appendix C that finetuning the VLM on our pipeline's image-caption pairs improves retrieval, suggesting our data pipeline may help mitigate the very limitations the model inherits. We also assess how well we detect, retrieve, and estimate events not collected via embeddings we created an independent unbiased test set by labeling a small dataset of logging/deforestation events recorded in publicly available state logging permits records Washington State Department of Natural Resources [2025]. These assessments are included in Appendix I.

Broader Impact. This research transitions Earth observation towards predictive intelligence. Such foresight provides a crucial window for intervention, allowing humanitarian and environmental groups to act before events fully escalate. However, the forecasting capability afforded by this model carries risks of dual-use. While our dataset is curated to exclude sensitive content, the underlying methodology could be repurposed for surveillance. To mitigate these risks, we emphasize that the scaling of such datasets must be accompanied by ethical governance frameworks that prevent the exposure of sensitive activities.

7 Conclusion

For the first time in history, we are capturing a daily snapshots of the entire Earth. With this abundance of information, we want to know *where*, *what*, and *when* global events of interest will occur. We achieve this through a novel data engine that uses vector arithmetic on VLM embeddings to discover change events at scale. This produced the largest and most diverse satellite event dataset to date featuring over 21,000 locations and 64 event types across six continents. This dataset enables us to build a first-of-its-kind global scale event anticipation model, capable of **event detection** with 93.5% max F1 score, outperforms baselines 3.6× on **predicted event retrieval**, and **predicts completion** dates with a 3-day median error-even nine months before completion. Our automated data pipeline lowers the barrier for future work in satellite event understanding and our event model shifts Earth observation from reactive detection to proactive anticipation of the events that shape our world.

Acknowledgments This work was fully funded by Planet Labs. We thank Kiruthika Devaraj, Leela Prabhu, Megan Van Welie, and Michael Thomas for their support.

References

K. Ahmad, K. Pogorelov, M. Riegler, N. Conci, and P. Halvorsen. Social media and satellites: Disaster event detection, linking and summarization. *Multimedia Tools and Applications*, 78(3): 2837–2875, 2019.

M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *arXiv preprint arXiv:2301.08243*, 2023.

M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, M. Muckley, A. Rizvi, C. Roberts, K. Sinha, A. Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

M. Bain, A. Nagrani, G. Varol, and A. Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1728–1738, 2021.

A. Barzilai, Y. Gigi, A. Helmy, V. Silverman, Y. Refael, B. Jaber, T. Shekel, G. Leifman, and G. Beryozkin. A recipe for improving remote sensing vlm zero shot generalization. *arXiv preprint arXiv:2503.08722*, 2025.

A. Bell, A. Aides, A. Helmy, A. Muslim, A. Barzilai, A. Slobodkin, B. Jaber, D. Schottlander, G. Leifman, J. Paul, et al. Earth ai: unlocking geospatial insights with foundation models and cross-modal reasoning. *arXiv preprint arXiv:2510.18318*, 2025.

L. D. Bergman, V. Castelli, and C.-S. Li. Progressive content-based retrieval from satellite image archives. *D-Lib Magazine*, 1997.

D. Chen, M. Shukor, T. Moutakanni, W. Chung, J. Yu, T. Kasarla, A. Bolourchi, Y. LeCun, and P. Fung. Vl-jepa: Joint embedding predictive architecture for vision-language. *arXiv preprint arXiv:2512.10942*, 2025.

H. Chen and Z. Shi. A spatial-temporal attention-based method and a new dataset for remote sensing image change detection. *Remote sensing*, 12(10):1662, 2020.

ESA. Copernicus sentinel-2, 2026.

Gemini Team, Google. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. URL `https://arxiv.org/abs/2312.11805`.

Z. Guo, Y. Wang, P. Jian, C. Li, X. Chen, Z. Yang, and E. E. TAMMs: Change understanding and forecasting in satellite image time series with temporal-aware multimodal models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=PXDFTeIqMd`.

B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

J. A. Irvin, E. R. Liu, J. C. Chen, I. Dormoy, J. Kim, S. Khanna, Z. Zheng, and S. Ermon. Teochat: A large vision-language assistant for temporal earth observation data. *arXiv preprint arXiv:2410.06234*, 2024.

P. Jain, D. Marcos, D. Ienco, R. Interdonato, and T. Berchoux. Timesenclip: A time series vision–language model for remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 2025. URL `https://api.semanticscholar.org/CorpusID:284057771`.

Z. Jiang, R. Meng, X. Yang, S. Yavuz, Y. Zhou, and W. Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024.

B. Khaertdinov, M. Popa, and N. Tintarev. A little more like this: Text-to-image retrieval with vision-language models using relevance feedback. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3825–3834, 2026.

M. Klassen, R. Nair, A. Hoover, B. Neuberg, and C. Levit. Seeing the difference: Vision transformer-powered multitemporal generic change detection. In *IGARSS 2025 - 2025 IEEE International Geoscience and Remote Sensing Symposium*, pages 933–937, 2025. doi: 10.1109/IGARSS55030.2025.11243584.

K. Komurcu and L. Petkevicius. Sat-jepa-diff: Bridging self-supervised learning and generative diffusion for remote sensing. *arXiv preprint arXiv:2603.13943*, 2026.

Y. LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

X. Li, C. Wen, Y. Hu, and N. Zhou. Rs-clip: Zero shot remote sensing scene classification via contrastive vision-language supervision. *International Journal of Applied Earth Observation and Geoinformation*, 124:103497, 2023.

Y. Li, W. Xu, G. Li, Z. Yu, Z. Wei, J. Wang, and M. Peng. Unirs: Unifying multi-temporal remote sensing tasks through vision language models. *ArXiv*, abs/2412.20742, 2024. URL https://api.semanticscholar.org/CorpusID:275133775.

E. Littwin, O. Saremi, M. Advani, V. Thilak, P. Nakkiran, C. Huang, and J. Susskind. How jepa avoids noisy features: The implicit bias of deep linear self distillation networks. *Advances in Neural Information Processing Systems*, 37:91300–91336, 2024.

C. Liu, R. Zhao, H. Chen, Z. Zou, and Z. Shi. Remote sensing image change captioning with dual-branch transformers: A new method and a large scale dataset. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–20, 2022. doi: 10.1109/TGRS.2022.3218921.

C. Liu, K. Chen, R. Zhao, Z. Zou, and Z. Shi. Text2earth: Unlocking text-driven remote sensing image generation with a global-scale dataset and a foundation model. *IEEE Geoscience and Remote Sensing Magazine*, pages 2–23, 2025. doi: 10.1109/MGRS.2025.3560455.

F. Liu, D. Chen, Z. Guan, X. Zhou, J. Zhu, Q. Ye, L. Fu, and J. Zhou. Remoteclip: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–16, 2024.

U. Mall, B. Hariharan, and K. Bala. Change event dataset for discovery from spatio-temporal remote sensing imagery. *Advances in Neural Information Processing Systems*, 35:27484–27496, 2022.

U. Mall, C. P. Phoo, M. K. Liu, C. Vondrick, B. Hariharan, and K. Bala. Remote sensing vision-language foundation models without annotations via ground remote alignment. *arXiv preprint arXiv:2312.06960*, 2023.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Planet Labs PBC. Planet application program interface: In space for life on earth, 2018–2026. URL https://api.planet.com.

H. Porta, E. Dalsasso, J. L. McCarty, and D. Tuia. Canadafiresat: Toward high-resolution wildfire forecasting with multiple modalities. *arXiv preprint arXiv:2506.08690*, 2025.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

S. Revankar, U. Mall, C. P. Phoo, K. Bala, and B. Hariharan. Monitrs: Multimodal observations of natural incidents through remote sensing. *arXiv preprint arXiv:2507.16228*, 2025.

J. J. Rocchio Jr. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, 1971.

V. Sobal, J. SV, S. Jalagam, N. Carion, K. Cho, and Y. LeCun. Joint embedding predictive architectures focus on slow features. *arXiv preprint arXiv:2211.10831*, 2022.

B. Uzkent, E. Sheehan, C. Meng, Z. Tang, M. Burke, D. Lobell, and S. Ermon. Learning to interpret satellite images in global scale using wikipedia. *arXiv preprint arXiv:1905.02506*, 2019.

E. Veach and Google. S2 geometry library. https://github.com/google/s2geometry, 2017.

C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106, 2016.

Washington State Department of Natural Resources. Providing GIS data for forest practices activities throughout Washington. https://dnr.wa.gov/forest-practices/providing-gis-data-forest-practices-activities-throughout-washington, 2025. Accessed: 2026-11-07.

Z. Xiong, Y. Wang, W. Yu, A. J. Stewart, J. Zhao, N. Lehmann, T. Dujardin, Z. Yuan, P. Ghamisi, and X. X. Zhu. Dofa-clip: Multimodal vision-language foundation models for earth observation. *arXiv preprint arXiv:2503.06312*, 2025.

K. Zhang, K. Ge, X. Chi, R. Zhang, S. Shi, Z. Dong, S. Han, and S. Zhang. Can world models benefit vlms for world dynamics? *arXiv preprint arXiv:2510.00855*, 2025.

S. Zhong, W. Ruan, M. Jin, H. Li, Q. Wen, and Y. Liang. Time-vlm: Exploring multimodal vision-language models for augmented time series forecasting. *arXiv preprint arXiv:2502.04395*, 2025.

S. Zou, X. Tian, L. Wesemann, F. Waschkowski, Z. Yang, and J. Zhang. Unlocking vision-language models for video anomaly detection via fine-grained prompting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4223–4233, 2026.

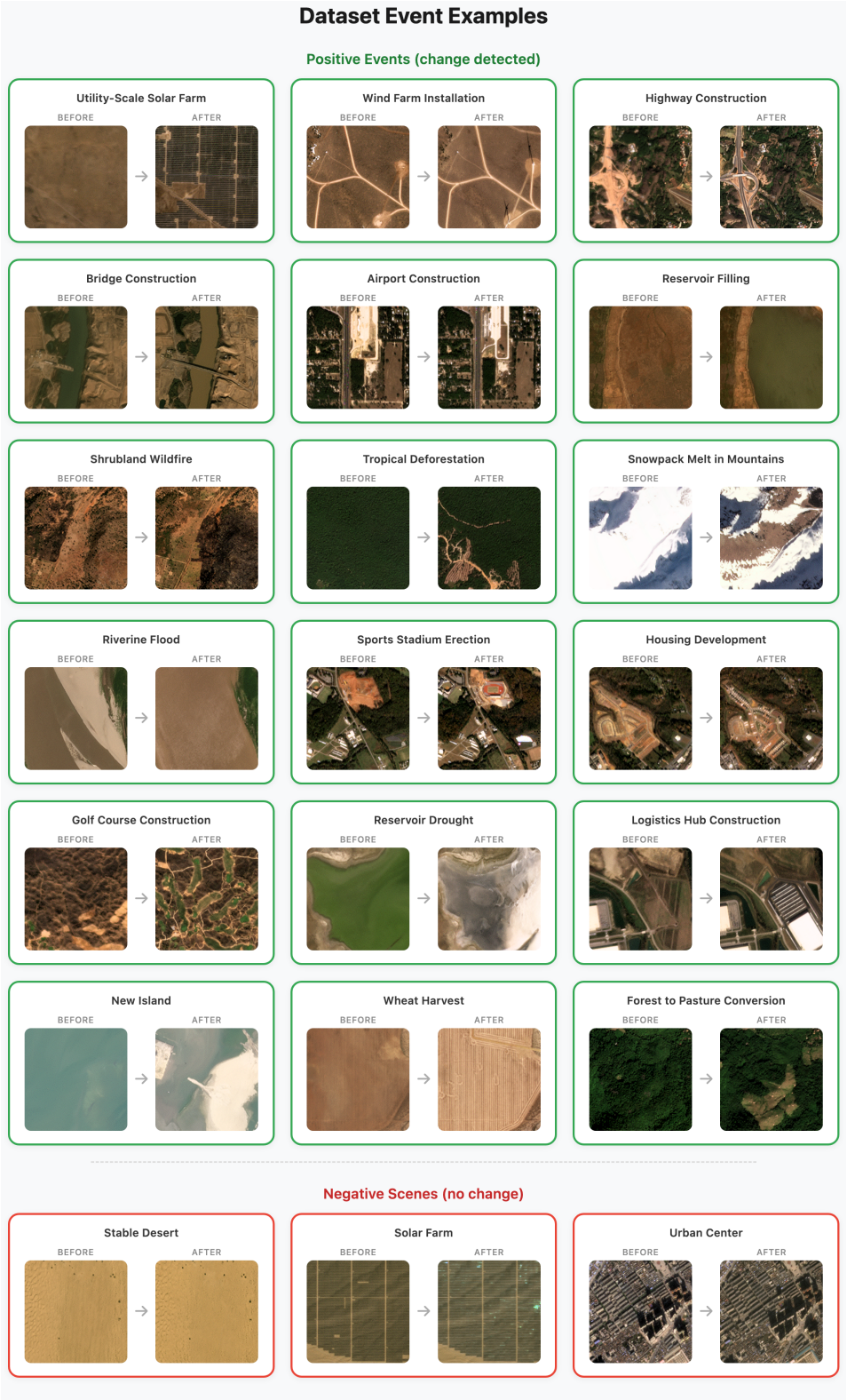


Figure 5: Dataset Event Examples. Representative before and after image pairs from our automatically-discovered dataset. *Top (green):* Positive events showing diverse changes. *Bottom (red):* Negative scenes showing stable locations with no significant change over time. In total we have 64 positive event classes and 69 stable scene classes.

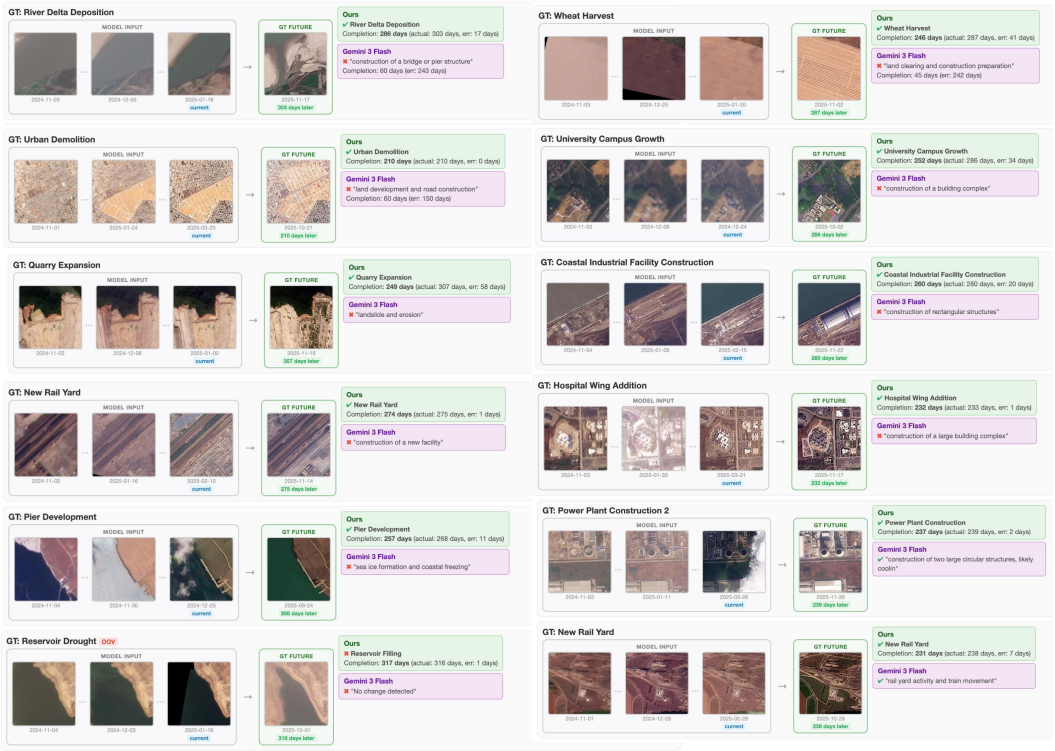


Figure 6: Qualitative prediction examples. Each card shows samples from the multi-temporal input sequences as well as the future state of the event with days until completion. To the right we include our model's prediction vs Gemini 3 Flash

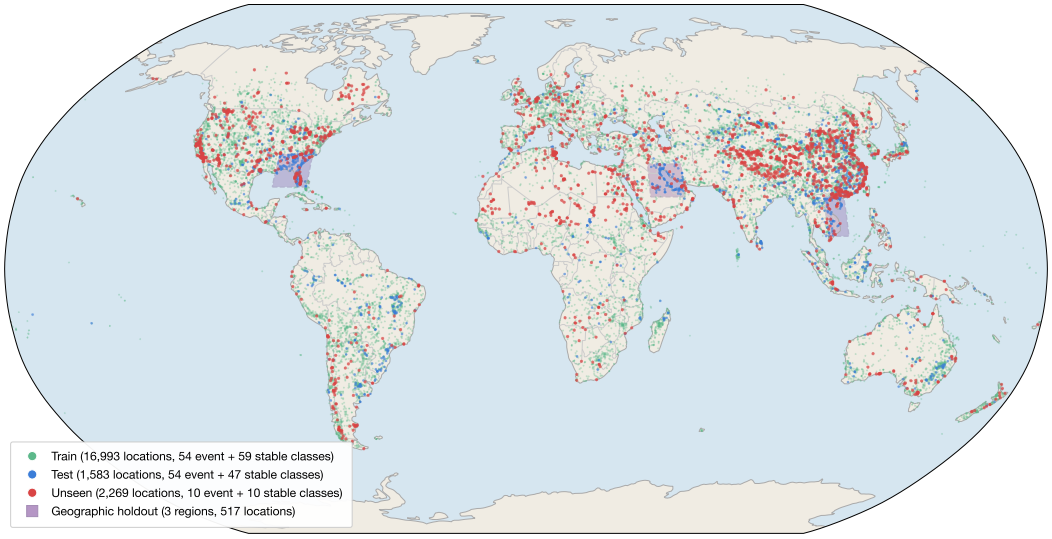


Figure 7: Dataset Geographic Distribution. Our global event dataset comprises over 21,000+ locations. We explicitly partition the data into a training set (green), test set (blue), and an unseen class set (red) to evaluate zero-shot event anticipation, and three geographic holdout regions (purple) in North America, the Middle East, and East Asia to test regional robustness.

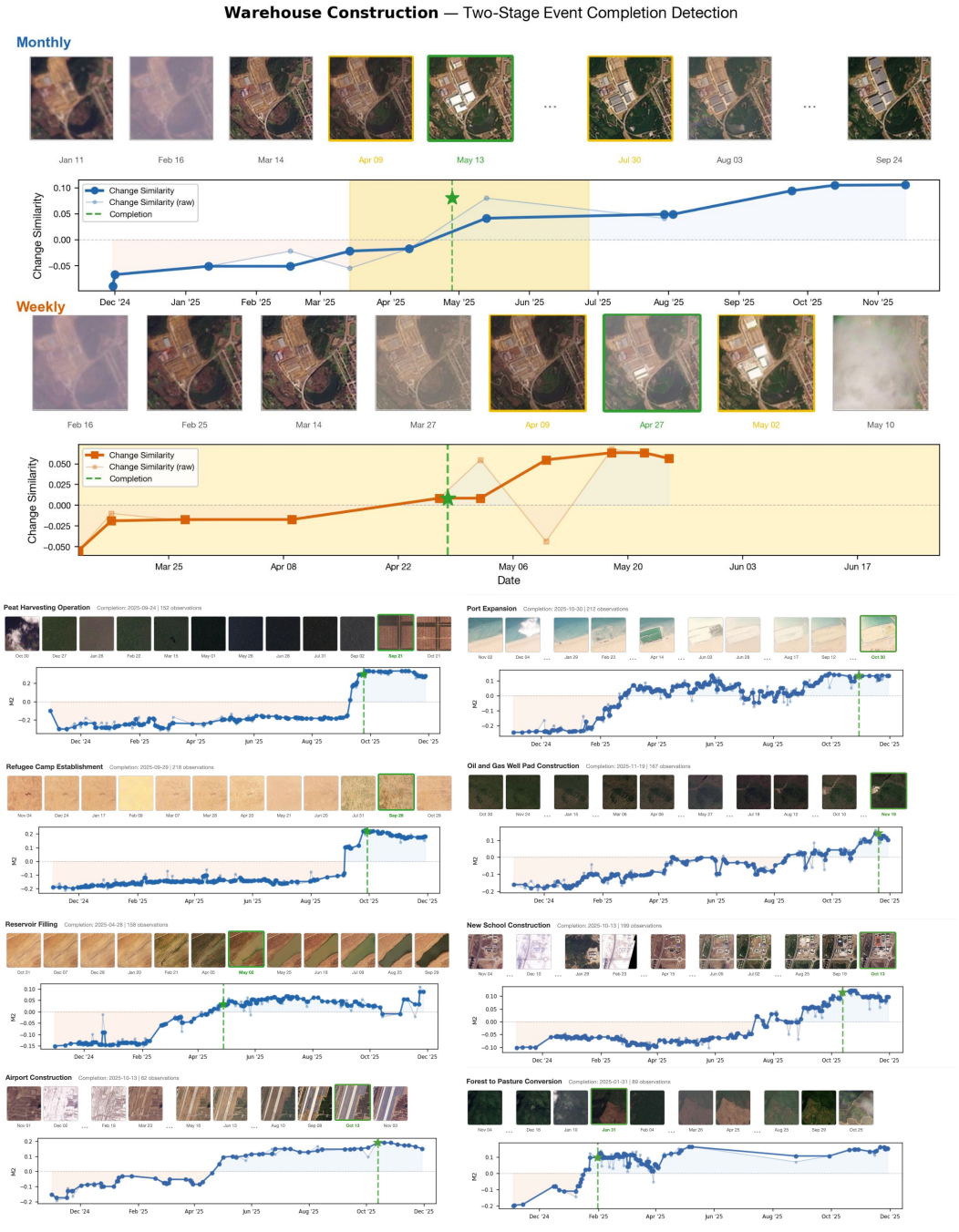


Figure 8: Event Completion Detection. We demonstrate our two-phased completion detection process for a *Warehouse Construction* event. The visualization shows the transition from a monthly to a weekly analysis level to increase temporal resolution. Section. 3 and Appendix F. Below, additional examples illustrate completion detection across diverse events.



Figure 9: Screenshots of the human evaluation interface. The top row shows two example events, *highway construction* and *mangrove deforestation*, followed by the corresponding maps and news results for the mangrove example for reference. Evaluators used these 4 context pages to label the event with "correct", "incorrect" or "unsure". Sampling with multiple "correct" labels were used to create our test sets.

A Examples of Satellite Event Dataset

We include some examples of the events from our dataset in Figure 5 showcasing its diversity.

B Additional Qualitative Results of Predicted Event Retrieval

We visualize a variety of qualitative results from our test set, starting with samples where our model predicts events several months before completion, as well as samples where our models fall short in Figure 6. Each sample shows the input sequences followed by the future state to be predicted.

C Model Comparison for Event Retrieval

	Change mAP@K			**Intersection mAP@K**		
Method	@5	@10	@20	@5	@10	@20
RemoteCLIP FT	**0.752**	**0.760**	**0.764**	**0.822**	**0.835**	**0.840**
RS-SigLIP Barzilai et al. [2025]	0.688	0.709	0.710	0.691	0.716	0.720
GIT-RS-CLIP Liu et al. [2025]	0.400	0.418	0.431	0.422	0.459	0.478
DOFA-CLIP-B Xiong et al. [2025]	0.318	0.338	0.356	0.350	0.380	0.396
RemoteCLIP Liu et al. [2024]	0.262	0.287	0.294	0.327	0.365	0.380

Table 4: Zero-shot Event Retrieval. Comparative evaluation of satellite VLMs performance using Intersection Search and Change Vector Search as described in Sec. 3.

We evaluate several popular satellite VLMs for event retrieval to determine which models we should use to train our world model. From Table. 4, one can see that the performance of RS-Siglip was much higher than the open source models. For this reason we finetuned RemoteCLIP to ensure that we had an opensource model that was able to recognize the *before* and *after* states of our events.

While this improved its understanding visually of the event, we see that it improved the *Intersection* retrieval far more than the *Change Vector* retrieval.To better test the change understanding of the models, we calculate the *Intra-Event Consistency*, asking For events with 2+ samples, how similar are their change vectors? Here we calculate the mean pairwise cosine similarity between change vectors of the same event, from this we found that the RS-Siglip model held a 0.5606 vs the finetuned RemoteCLIP models carried a 0.4695 and off-the-shelf RemoteCLIP held 0.3121.

The fine-tuned RemoteCLIP learned better representations of the individual states (before/after), which helped it find relevant images via intersection retrieval. But when you look at the direction of change (after minus before), RS-SigLIP's change vectors for the same event type are more tightly aligned with each other. We hypothesize that this consistency, i.e. the degree to which a model produces a stable change direction for a given event type directly affects prediction quality. Supporting this, Table 1 shows that RS-SigLIP embeddings yield stronger prediction performance, particularly on unseen events (mAP5 .254 vs .200), where consistent change geometry would matter most for generalization.

D Ablation Study on Anticipation Model Architectures

To test whether the anticipation gains depend on the transformer backbone, we replace it with an LSTM and a GRU while keeping the inputs (velocity-augmented embeddings), horizon conditioning, training objective, and data fixed. All three architectures outperform the Static baseline at each k.

Method	max F1	mAP@5	mAP@10	mAP@20
Oracle (PS, RS-SigLIP)	–	.734	.705	.642
Static (PS, RS-SigLIP)	–	.208	.191	.153
TeoChat	.275	.005	.006	.006
SAT-JEPA-DIFF	–	.057	.044	.032
Gemini 3 Flash	.435	.155	.126	.100
Ours (LSTM)	.766	.361	.315	.270
Ours (GRU)	.839	.401	.348	.269
Ours (Transformer)	**.949**	**.503**	**.485**	**.415**

Table 5: Anticipation Architecture Ablation. Evaluated over the seen and unseen event classes combined. All three backbones outperform every baseline, and the transformer is best on every metric.

E Global Distribution of Event Dataset

Figure. 7 visualizes the global distribution of our event dataset, as well as the locations of the test set, unseen classes, and geographic hold out regions.

F Implementation Details of Event Completion Detection

Noise filtering. Before analyzing the S_Δ curve, we detect and exclude cloudy or noisy images. For each embedding, we compute cosine similarity to both temporal neighbors and set an adaptive threshold based on the median neighbor similarity across all images. Images with low similarity to both neighbors are flagged as outliers and excluded from subsequent analysis. We then apply a median filter to the S_Δ series to suppress remaining noise while preserving endpoints.

Coarse-to-fine completion detection. At monthly resolution, we detect the first sustained zero-crossing—the earliest timepoint where the signal transitions from negative to positive and remains positive for at least k subsequent observations. Small epsilon margins (ϵ^-, ϵ^+) avoid false detections from noise. If no clear crossing exists, we fall back to the temporal window with the highest mean change similarity, and finally to the maximum first-order difference.

To refine from $\sim$30-day to $\sim$7-day precision, we download weekly imagery ($\sim$1 image/week) for the detected completion month plus one month before and after. We repeat the noise filtering and zero-crossing analysis on this weekly S_Δ series, yielding a precise completion estimate to the week. We visualize this process for a warehouse construction event in Figure 8. The completion timestamps are a crucial part of a temporal event dataset and these annotations are included for each event in the dataset.

As shown in Figure 8, the S_Δ signal exhibits sharp zero-crossings across diverse event types, confirming that our completion dates capture genuine perceptual transitions rather than gradual drift.

G Human Verification of Test Data

Screenshots from our web-based labeling interface where annotators classify satellite before/after image pairs as showing the described change event or not. Annotators get to see beyond the boundaries of the single image for both time points, as well as map and search results for the event type in the area for additional context.

H Implementation Details of Anticipation Model

Our model uses a 4-layer causal transformer with 8 attention heads, feed-forward dimension 2048, and dropout 0.2. Input sequences are padded to a maximum of 53 tokens. We train with AdamW (lr $= 10^{-4}$, weight decay 10^{-3}) and cosine annealing for 200 epochs with batch size 32 and gradient clipping at 1.0. The magnitude regularization weight is $\lambda = 0.1$. Model size scales with the VLM backbone: 54.8M parameters for RS-SigLIP (1152-dim) and 29.7M for RemoteCLIP (768-dim). All models are trained on a single NVIDIA A100 40GB GPU with 170GB system memory. Training the PS RS-SigLIP model (177K samples, 200 epochs) takes approximately 14 hours; S2 variants (163K samples) train in approximately 12 hours. We select the best checkpoint by seen-class mAP@10 on the validation split. All experiments use the same hyperparameters with no per-backbone tuning.

I Logging Permit Dataset

In Section 6 we discuss one of the limitations of our work which is the ambiguity of time-to-completion. We created an independent, unbiased test set by labeling a small set of logging and deforestation events recorded in publicly available Washington State DNR logging-permit records Washington State Department of Natural Resources [2025], so that no sample was selected using embeddings. Each site has hand-verified dates for when harvesting first becomes visible and when it is complete. We run two experiments on this data.

(1) How accurate is our embedding-based completion detector on events not collected via embeddings? On 14 hand-labeled sites we found that our embeddings driven completion detection agreed with the human labels with a median of 8 days (79% within 30 days), with two outliers at sites where the clearing happened in several stages.

(2) How well do we detect and retrieve these events? We ranked every chip in an 8,878 km² region of Washington by predicted logging, 2–4 months ahead. On visual inspection, 7 of the model's top 20 picks were confirmed future clear-cuts (static baseline: 3), and only 1 pick showed no logging at all.

J Time-to-completion Ambiguity

Event completion is ambiguous, especially when observed remotely. We define completion as the point at which a location becomes retrievable as its "after" state, but this can disagree with a human assessment of *completion*. For example, when a data center is constructed, the main building may appear finished from above months before the parking lots and auxiliary facilities around it are complete, so different observers may reasonably place completion at different times.

To assess this disagreement, we compare human-labeled completion dates with our embedding-based completion detection. Agreement varies by event type: where the end state is visually clear (housing, highways, bridges), the two match closely (median 8–24 days); where the change is gradual, cyclical or reversible, and humans themselves find completion hard to pin down, they disagree more.

Completion is therefore not a single well-defined moment for every event type. This supports our definition, which applies consistently across event types and agrees closely with human judgement where the end state is clear.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, we claim to present a new global event retrieval method, and a new dataset with this pipeline, followed by an event prediction model. Which we present results for in the results section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a limitations section in our discussion in Section. 6.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer:[N/A]

   Justification: We present capabilities demonstrated by embeddings from VLMs, and mathematically describe them, however we do not have any theoretical proofs.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All code and data will be publicly released and we include implementation details in Section 5 and Appendices F, G and H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All code and data will be publicly released upon publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We include train and test details in Section. 5 and model implementation details in Appendix. H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: To test robustness and statistical significance of our results we perform a bootstrapping experiment by subsampling within our test set 1,000 times, and a permutation test on the per-class AP between the static baseline and our model using 10,000 permutations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes we include compute resources used in our implementation details in Appendix H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We conform to the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We include a paragraph on the broader impacts of our work in Section. 6, paragraph 6.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: We do not include any sensitive information in our dataset.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators and original owners are cited and credited. We use two commercially available products, i.e. the Rs-sigLIP model and Planetscope imagery both of which are credited and with legal permission to publish. We include the MCLA license of Planetscope with the CC BY-SA 4.0 license applying to our dataset and code on release.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include details of the dataset and plan to release datasets and code upon publication, and have received permission to release Planetscope imagery for research purposes under CC BY-SA 4.0 licensing.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Screenshots of our labelling application are including in Appendix G, no compensation was provided and no personal data was collected.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:The human verification was conducted on a voluntary basis, no personal data was collected, and no compensation was provided. The activity did not constitute human subjects research as defined by IRB guidelines, and therefore no IRB approval was required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

Justification: We describe the usage of MLLMs in the ARF and Event description generation components of our work in Sections 3 and 5.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.