

# NOT JUST A FLASH IN TIME: INTERPRETING LONG EVENT STREAMS THROUGH LANGUAGE – APPENDIX

**Anonymous authors**  
Paper under double-blind review

## 1 CODE AND DATA APPENDIX.

In the pre-training fine-tuning dataset EIQA-1M, we aligned the open-source, large-scale N-ImageNet with question–answer pairs. The aligned text was refined through a combination of Chat-GPT generation and human editing, and we crafted multiple instruction and QA formats to boost the model’s ability to handle event-stream QA tasks. Figure 1 illustrates examples from our EIQA-1M.

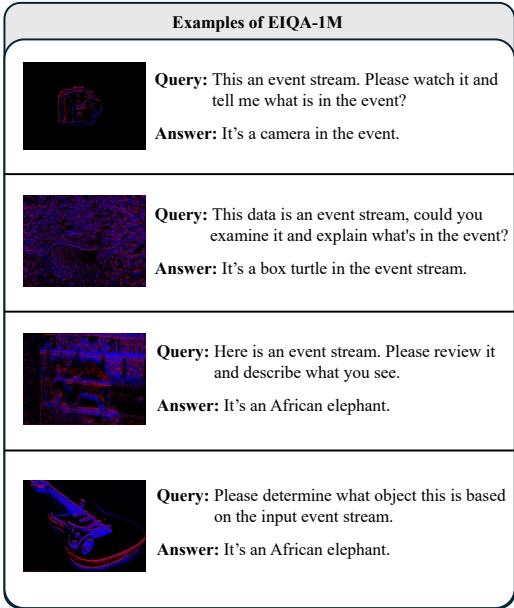


Figure 1: Examples of EIQA-1M.

For the benchmark EVQA-Bench, we used an event simulator to produce event streams in  $(t, x, y, p)$  tuples, applied our pipeline to annotate QA pairs, and further adjusted the instructions and answers wherever they conflicted with the actual event content. We also annotated QA pairs for the open-source N-Caltech101 and incorporated it into the benchmark as a short-temporal classification task. Across the entire EVQA-bench we cover multiple scenarios, including driving roads, human activities, and object motion, collisions, and recognition. Figure 2 presents dataset examples for each scenario.

It is worth noting that, while this work was under way the EventGPT team had not yet released the event-text fine-tuning data they used (despite stating in their paper that it would be public, it was released on 25 June). Hence all our training and fine-tuning relied solely on our self-annotated EIQA-1M. In experiments—whether on the public N-Caltech101 or on our own datasets—our model achieved performance higher than EventGPT, as shown in the main-text comparisons. In Figure 3 we further present LET-US’s performance on long-sequence event streams.

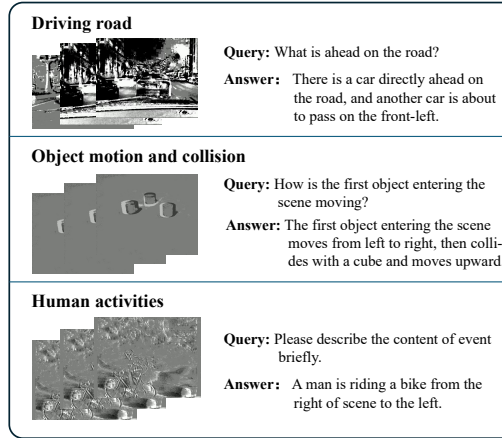


Figure 2: Dataset presentation under different scenarios.

## 2 LIMITATION AND FUTURE WORK.

**Limitation.** Although we have achieved performance that surpasses all state-of-the-art methods in understanding long-sequence event streams and demonstrated LET-US’s ability to produce coherent predictions over such data, it is important to acknowledge a set of persistent limitations that stem from the native characteristics of event cameras themselves. First, in complex outdoor scenes or low-light indoor environments, sudden illumination changes, background flicker, and sensor jitter generate large volumes of spurious ON/OFF events that obscure the truly informative dynamics of the scene, forcing the model to sift through a high noise-to-signal ratio before meaningful reasoning can even begin. Second, the low spatial resolution of most commercial event sensors means that small or distant objects may be represented by only a handful of activated pixels; subtle features such as a pedestrian’s hand gesture or a vehicle’s indicator light are therefore frequently lost, reducing both recognition accuracy and safety-critical situational awareness. Third, the event representation itself is fundamentally sparse and edge-centric: each pixel reports only binary polarity flips without color, absolute intensity, or texture, depriving the model of rich visual cues that conventional frame-based systems exploit for fine-grained tasks such as material classification or attribute recognition. Fourth, events arrive asynchronously at microsecond precision yet in highly bursty patterns; batching them into fixed-length windows or voxel grids introduces unavoidable trade-offs between temporal fidelity and computational tractability, and misalignment artifacts can propagate through the network’s spatiotemporal hierarchy. Finally, the annotation of raw event clouds remains labor-intensive and error-prone, limiting the availability of large-scale, high-quality supervised datasets; this scarcity hinders progress on rare-event understanding, long-tail classes, and safety-critical edge cases. These issues show that event-based perception still needs key advances. We need stronger denoising methods, adaptive spatiotemporal representations, self-supervised pre-training on unlabeled data, and domain adaptation that accounts for hardware differences. With these improvements, event-based vision can rival or even exceed frame-based vision in real-world applications.

**Future work.** How to leverage high temporal resolution to more accurately interpret the scenes described by an event stream when spatial information is limited remains a key challenge. We envision future work proceeding along several fronts: firstly, we will add adaptive denoising and stronger filters to handle noisy, low-resolution scenes. Secondly, we will test multi-scale and hierarchical Transformer blocks to capture both quick motions and long-term trends. Thirdly, we will build much larger event-to-text datasets, including more real-world streams, to support stronger cross-modal pretraining. Fourthly, we will fuse event data with RGB images, depth, and audio through cross-modal attention to improve fine-grained recognition and scene understanding. Fifthly, we will apply self-supervised and contrastive learning on unlabeled streams to lower annotation costs and improve transferability. Sixthly, we will explore pruning, quantization, and knowledge distillation so the system can run in real time on low-power edge devices. Seventhly, we will design tougher benchmarks that demand multi-step causal reasoning and interactions among multiple objects. Eighthly, we will build visualization tools that show which events and moments the model attends to, making

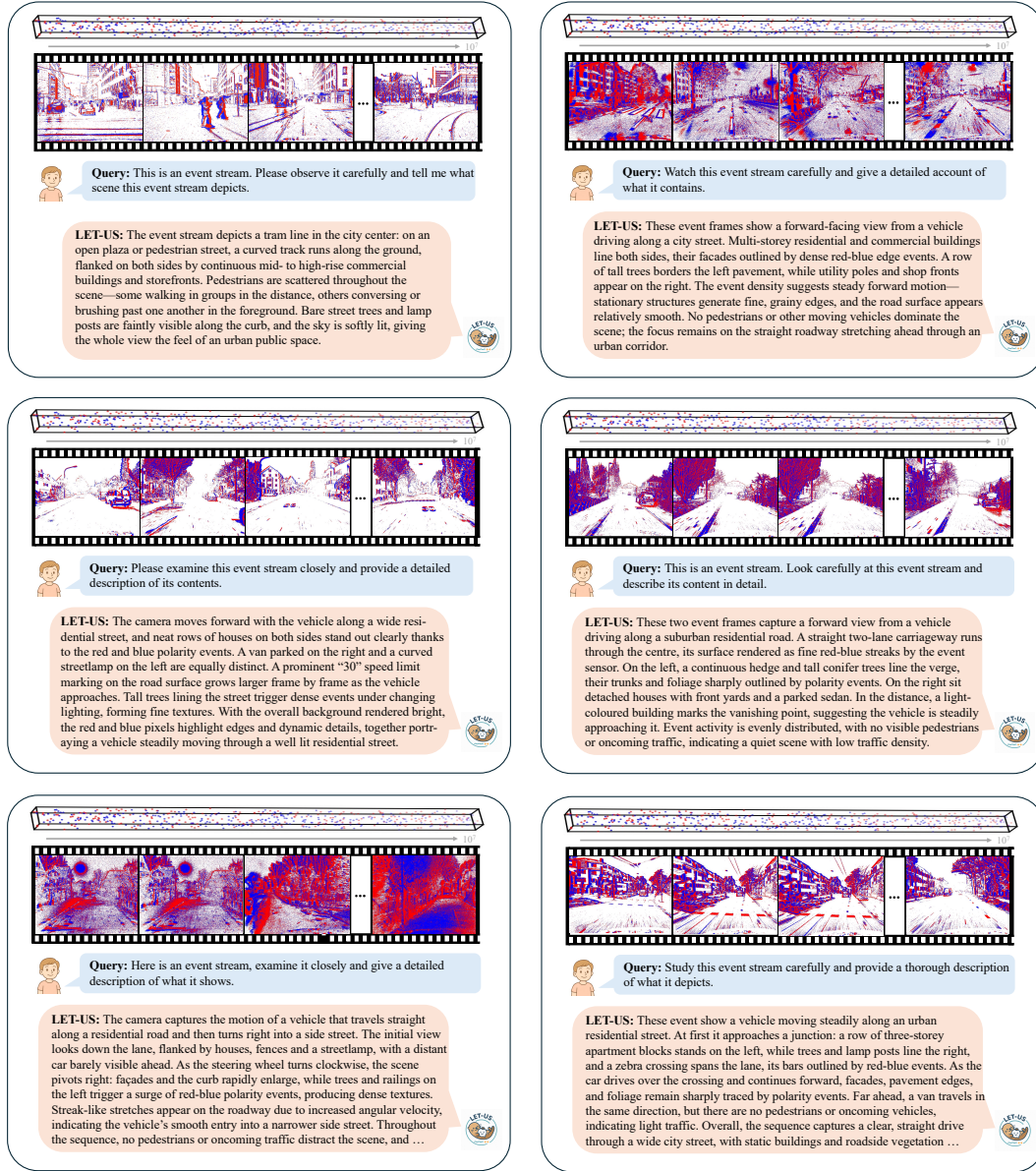


Figure 3: Dataset presentation under different scenarios.

its decisions easier to interpret. Finally, we will deploy the model in applications such as robotics, autonomous driving, and industrial monitoring to test its robustness under dynamic lighting and complex backgrounds, and we will integrate it with large vision-language models for fluent temporal language understanding.

### 3 DEMO

To better illustrate our results, we used the real-world DSEC dataset to showcase the model’s scene understanding capabilities, which includes a variety of complex driving road scenarios. We present LET-US’s performance on some examples of DSEC dataset in Figure 3.