Supplementary Material: Cross-Modal Event Encoder: Bridging Image–Text Knowledge to Event Streams

1. Proposition Details

This section compares the query-key alignment mechanisms of three learning methods: MoCo [4], CLIP [11], and ZSCL [15], focusing on the similarities between momentum-based updates and fixed target space alignment. MoCo uses momentum to progressively align the query and key encoders, while CLIP employs a fixed key encoder to align image and text embeddings. ZSCL targets fixed key and text embeddings to align query embeddings, offering a gradual alignment approach similar to MoCo's momentum updates. The goal is to illustrate how each method effectively addresses the query-key alignment challenge.

Momentum Contrast (MoCo). In MoCo, the query and key encoders are defined as $q = h_q(E)$ and $k = h_k(E)$, respectively. The contrastive loss L_{ct} aims to maximize the similarity between positive query-key pairs while minimizing it for negative samples. This is mathematically represented as:

$$L_{\text{ct}} = -\log\left(\frac{\exp(q \cdot k_+/\tau)}{\sum_{k_i \in K}\exp(q \cdot k_i/\tau)}\right)$$

where k_+ is the positive key corresponding to q, and τ is a temperature parameter controlling the concentration of the distribution. The query encoder parameters θ_q are updated by minimizing this loss:

$$\theta_q \leftarrow \theta_q - \eta \cdot \nabla_{\theta_q} L_{\text{ct}}$$

The key encoder θ_k is updated using a momentum-based approach:

$$\theta_k \leftarrow m\theta_k + (1-m)(\theta_q - \eta \cdot \nabla_{\theta_q} L_{\text{ct}})$$

This update ensures gradual alignment of the key encoder θ_k with the query encoder θ_q.

Contrastive Language–Image Pretraining (CLIP). CLIP employs query and key encoders, $q = f_E(E)$ for text embeddings and $k = f_I(I)$ for image embeddings, to align text and image representations in a shared feature space. The contrastive loss for CLIP is:

$$L_{\text{ct}} = -\log\left(\frac{\exp(f_E(E) \cdot f_I(I_+)/\tau)}{\sum_{f_I(I_i) \in K}\exp(f_E(E) \cdot f_I(I_i)/\tau)}\right)$$

Similar to MoCo, the parameters for the query encoder θ_q are updated via gradient descent:

$$\theta_q \leftarrow \theta_q - \eta \cdot \nabla_{\theta_q} L_{\text{ct}}$$

However, CLIP maintains a fixed key encoder, ensuring a stable target for the query encoder's updates.

Zero-Shot Contrastive Learning (ZSCL). ZSCL integrates query encoder $f_E(E)$, a fixed key encoder $f_I(I)$, and a text embedding module $f_T(T)$. The similarity $s_{i,j}$ between embeddings is given by:

$$s_{i,j} = f_E(E_i) \cdot f_T(T_j)$$

The ZSCL loss is composed of two terms: L_q for aligning query embeddings with text embeddings, and L_k for aligning key embeddings with text embeddings:

1. **Query Alignment Loss** L_q:

$$L_q = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(f_E(E_i) \cdot f_T(T_i)/\tau)}{\sum_{j=1}^{N}\exp(f_E(E_i) \cdot f_T(T_j)/\tau)}\right)$$

2. **Key Alignment Loss** L_k:

$$L_k = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp(f_I(I_i) \cdot f_T(T_i)/\tau)}{\sum_{j=1}^{N}\exp(f_I(I_i) \cdot f_T(T_j)/\tau)}\right)$$

The total ZSCL loss is:

$$L_{\text{ZSCL}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log\left(\frac{\exp(f_E(E_i) \cdot f_T(T_i)/\tau)}{\sum_{j=1}^{N}\exp(f_E(E_i) \cdot f_T(T_j)/\tau)}\right) + \log\left(\frac{\exp(f_I(I_i) \cdot f_T(T_i)/\tau)}{\sum_{j=1}^{N}\exp(f_I(I_i) \cdot f_T(T_j)/\tau)}\right)\right)$$

The query encoder f_E is updated as:

$$\theta_E \leftarrow \theta_E - \eta \cdot \nabla_{\theta_E} L_{\text{ZSCL}}$$

Momentum-Like Behavior in ZSCL The ZSCL model aligns f_E with a fixed target space (from f_I and f_T) through:

$$\theta_E \leftarrow \theta_E - \eta \cdot \nabla_{\theta_E} L_q$$

This behavior can be approximated with a momentum-based formula, similar to MoCo:

$$\theta_E \leftarrow m\theta_E + (1-m)(\theta_{\text{target}} - \eta \cdot \nabla_{\theta_E} L_q)$$

This approximation demonstrates the progressive alignment of f_E with a stable target space, akin to the momentum mechanism in MoCo.

2. Conceptual Analysis

Our primary objective is to adapt CLIP for event representation by mapping the image embedding, I' , and the event embedding, E', to the same position within the embedding space $\mathbb{R}^z$. Despite the simplicity of this concept, the learning process encounters challenges due to the information gap between E and I, leading to difficulties in convergence and a tendency to overfit E. This overfitting can undermine the capabilities of CLIP, which are derived from training on large-scale datasets. Consequently, it is crucial to design an architecture and learning mechanism that effectively accommodates the characteristics of E while preserving the model's overall performance.

To address these challenges, we draw inspiration from the foundational architecture proposed [14]. This model employs a momentum training strategy, featuring two parallel encoders and a reference image encoder to facilitate learning. By leveraging this methodology, the model achieved state-of-the-art performance in supervised learning tasks. However, this approach lacks image-text alignment within the parallel encoders, as alignment is restricted to the reference image encoder, resulting in the absence of both zero-shot capabilities and event-text alignment. To overcome these limitations, we begin by replacing the conventional parallel MoCoV3 [2] encoders with CLIP, enabling a more comprehensive alignment framework.

Through this architecture, we observed a significant improvement in the performance of E. However, the zero-shot performance remained insufficient, posing substantial challenges to expanding the applications of E via zero-shot scenarios. We attribute this issue to the model's forgetting of image understanding capabilities and undertake efforts to mitigate this phenomenon.

As anticipated, achieving zero-shot capabilities unlocked significant advantages, enabling the expansion of E into previously inaccessible tasks such as event-based video anomaly detection. Additionally, zero-shot capabilities facilitated applications like event-based retrieval without requiring additional training, thereby providing concrete examples of E's utility. These results underscore the potential for broader adoption of the event modality in future research, highlighting its versatility and applicability across diverse domains.

3. Implmentation Details

We use NVIDIA RTX 4090×2 for training. The pre-training process required approximately 3 days for the ViT-B/32 model and a week for the ViT-L/14 model. For fine-tuning on the N-Caltech and N-MNIST datasets, the training duration was 5 hours and 12 hours, respectively.

3.1. Object Recognition

Dataset. We preprocess event data by converting it into a 2 channel image format across object recognition datasets. First, we separate the positive (pos) and negative (neg) polarity events from the event, then count the number of events for each polarity to construct a 2D histogram image with height H (224) and width W (224). Finally, each histogram is merged into a single tensor, resulting in an image format of shape (2, H, W). We apply three data augmentation techniques to enhance the variability of event-based data. First, a temporal flip randomly reverses the time axis of events with probability p and inverts their polarity. Second, a spatial shift shifts events in the x and y directions within a defined range, retaining only events that remain within the spatial bounds. Lastly, correlated event addition generates new events around existing ones by introducing Gaussian noise in spatial and temporal dimensions, ensuring correlated patterns.

In the N-Caltech [9] and N-MNIST [9] datasets, the range of time steps (t) differs from that in N-ImageNet [6], which poses challenges when representing data using a stacked histogram approach. To address this, we applied clamping to limit the number of stacks, which prevents the distribution from becoming overly sharp. This adjustment was found to impact performance in both zero-shot and few-shot settings. Notably, the effect was more pronounced in N-Caltech, attributed to the larger time-step intervals in this dataset.

Pre-training. We utilized the ImageNet mini dataset for pre-training, with 80% of the classes, and the remaining 20% in [n01795545, n01855672]. The AdamW optimizer was employed, with a learning rate of 1×10^{-6}, β parameters set to (0.9, 0.999), and a weight decay of 0.03. The model was trained for 200 epochs, of which 40 epochs were dedicated to a warmup with batch size 16. Additionally, cosine annealing [7] was applied to adjust the learning rate during training.

Fine-tuning. In the fine-tuning process, we employed a learning rate of 5×10^{-6} for the N-ImageNet, N-Caltech, and N-MNIST datasets, training each model over a total of 100 epochs. The warmup period spanned the initial 40 epochs, and the optimizer configuration remained consistent with that used during pre-training.

Few-Shot. For few-shot experiments, we followed the same protocol as EventCLIP and EventBind. Specifically, we selected k samples per class ($k = 1, 2, 5$) with a fixed random s ed, and used the identical subsets across all compared methods to ensure fairness and reproducibility. The remaining samples were used for evaluation.

3.2. Event Video Anomaly Detection

Dataset. To extract events from the UCFCrime [12], XD-Violence [12], and Shanghaitech [8] video datasets, three key processes are undertaken. First, a threshold is applied to activate pixels based on the frame-to-frame gap differences. Second, the activated event frames are stacked in groups. Finally, clamping is used to restrict the data distribution. The generated event representation, E, is then assigned a label based on the majority rule.

Prompt. To designate abnormal and normal classes, we use text prompts as shown in Tab. 1. For each dataset, the class names are used directly as prompts for abnormal events. Similarly, an approximately equal number of descriptive prompts are designated for the normal frames category. This approach aims to maintain a balance between abnormal and normal by adding specific descriptions for normal events, rather than relying solely on an understanding of normality itself. The distinction between normal and abnormal events is then determined by comparing the prediction probabilities: for abnormal,$\sum_{CLS_{\text{abnormal}}} E' \cdot T'_{CLS}$ and for normal, $\sum_{CLS_{\text{normal}}} E' \cdot T'_{CLS}$.

UCFCrime & XD-Violence		Shanghaitech	
Abnormal	**Normal**	**Abnormal**	**Normal**
Abuse	Peace	chasing	street
Arson	Calm	push	walking
Burglary	Quiet	monocycle	waiting
Fighting	Normalcy	throwing object	standing
Road Accidents	Routine	vaudeville	crossing
Shooting	Stability	fighting	sitting
Stealing	Tranquility	car	gathering
Arrest	Serenity	running	office
Assault	Nothing	stoop	shopping
Explosion	Order	robbery	family
Robbery	Normal	vehicle	commuting
Shoplifting	Safe	skateboard	-
Vandalism	Silence	jumping	-
-	-	fall	-
-	-	circuit	-

Table 1. UCFCrime, XD-Violence, and Shanghaitech abnormal and normal categories for text prompt

3.3. Event Retrieval

Dataset. Our model has already aligned Event-Image-Text representations through pre-training, enabling Event-Image-Text retrieval using the N-Caltech dataset. Without additional training, we use the pre-trained model to perform retrieval. In each batch (set to 20), we extract and compare the sample with the highest similarity between the query E' and I', T'. This process is conducted on a sample-wise basis.

Additionally, for the sound modality, we extract corresponding classes from the ESC-50 [10] based on the N-Caltech dataset, as shown in Tab. 2. Here, the query E' calculates its similarity with the sound embedding S'. This similarity computation is performed on a class-wise basis within each given batch.

ESC-50	**Caltech101**
airplane	airplanes
helicopter	helicopter
clock_alarm	watch
cat	wild_cat
car_horn	car_side
insects	ant, butterfly, dragonfly
engine	ferry
keyboard_typing	laptop
sheep	llama
dog	dalmatian

Table 2. Mapping of Sound categories to Caltech101 categories

The Dense [5] is a road-view dataset, which is significantly different from the N-ImageNet used for pre-training. Due to the difficulty in extracting precise information from a single event frame E at a given time t, we construct E by stacking 32 event frames. However, given the high similarity between frames, we extract the depth embedding D' on a video-wise basis to calculate the similarity between E' embeddings.

Architecture. Our ViT-L/14 model has an embedding dimensionality of $\mathbb{R}^z$=768, which differs from the 1024-dimensional embeddings used by ImageBind [3], making direct comparison infeasible. To address this, we incorporate a post-processing adapter as utilized in ImageBind. This adapter includes a linear layer and a scaling factor to project E' into a compatible embedding space. Notably, the backbone network remains frozen, while the adapter is fine-tuned specifically on Event-Image pairs. This fine-tuning process spans 5 epochs and is applied exclusively to the sound and depth modalities

```
Adapter Module(x):
    x = nn.Linear(768, 1024)(x)
    x = Normalize(x)
    x = x * LearnableLogitScaling
    return x
```

Metrics. We utilize Recall@K, Mean Reciprocal Rank (MRR), and Mean Average Precision@K (mAP@K) metrics for the evaluation of retrieval tasks. Recall@K measures the fraction of all relevant items successfully retrieved within the top K, defined as:

$$\text{Recall@K} = \frac{\text{Number of relevant items in top } K}{\text{Total number of relevant items}}$$

This metric emphasizes the system's ability to cover relevant results. MRR evaluates the ranking quality by calculating the reciprocal rank of the first relevant item for each query and averaging over all queries:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}$$

where rank_i is the rank position of the first relevant item for the i-th query. Lastly, mAP@K assesses the overall ranking quality by averaging precision scores at each relevant item's position in the top K, defined as:

$$\text{mAP@K} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{R}_i|} \sum_{j \in \mathcal{R}_i} \frac{\text{Rank}_j}{j}$$

where $\mathcal{R}_i$ is the set of relevant items for the i-th query, and Rank_j is the precision at position j. These metrics collectively ensure a comprehensive assessment of retrieval performance, focusing on relevance ranking, early retrieval, and completeness.

4. Additional Experiments.

4.1. Model Architectures Dependency

The backbones used in the experiments were ViT-B/32 and ViT-L/14, with 83M and 307M training parameters, respectively. Consequently, a performance difference was observed after pre-training in both zero-shot and few-shot evaluations, with an approximate 7.2% gap in zero-shot and a 14% difference in 5-shot performance Tab. 3.

Shots	ViT-B/32	ViT-L/14
0-shot	30.7	37.9
1-shot	32.3	44.3
2-shot	35.6	45.91
5-shot	36.4	50.4

Table 3. Performance of zero & few shot, ViT-B/32 and ViT-L/14 on N-ImageNet

4.2. Impact of Event distribution

When modifying the event distribution of the N-Caltech dataset through clamping, we observed that a minimal clamping (with the lowest possible value) yielded a zero-shot accuracy of 66.34%, the highest result in this setting. This outcome is likely due to the distribution closely resembling that of N-ImageNet. Additionally, for the unique distribution of N-Caltech, setting the clamp value to 20 produced the highest fine-tuning accuracy of 95.41%, with results showing little variation when the clamp was increased to 30. Furthermore, in the ViT-B/32 model, an increase in distribution complexity made the representation less interpretable relative to the model structure, achieving a peak accuracy of 89.26% at a clamp value of 10 Tab. 4.

Model	Clamp	5	10	20	30
ViT-L	0-shot	66.34	63.47	63.3	63.99
	1-shot	77.66	76.68	79.43	75.88
	2-shot	80.01	80.04	81.91	78.17
	5-shot	83.8	83.66	85.01	82.08
	all	94.43	94.89	95.41	95.4
ViT-B	0-shot	44.46	44.06	43.02	40.84
	1-shot	61.06	61.17	59.56	58.36
	2-shot	67.49	66.4	65.37	64.96
	5-shot	73.75	73.18	72.95	72.77
	all	89.14	89.26	89.2	88.69

Table 4. Performance of ViT-L and ViT-B on Caltech/CLAMP Dataset

4.3. Impact of Prompt

The N-MNIST dataset exhibits a markedly different structure compared to datasets like N-ImageNet and N-Caltech. This divergence arises due to variations in data distribution, information density, and complexity, which, in turn, lead to distinct performance outcomes for our CLIP model when applied with specific prompts. Notably, we found that using the prompt "The digit CLS", where CLS represents "0" instead of "zero," yielded the highest accuracy, reaching 44.66%. Additionally, by limiting the data distribution to a specific subset (i.e., 2,000 samples) rather than utilizing all time-step samples, we achieved an even higher zero-shot performance, peaking at 46.71% Tab. 5.

4.4. Anomaly Detection Details

Reconstruction impact. In this section, we explore the results of event construction for the UCFCrime, XD-Violence, and Shanghaitech datasets. Specifically, we examine the Area Under Curve (AUC) in relation to the pixel threshold (range 0–255) and the stack size of event frames, with the clamping fixed at 10. Lower thresholds activate more event pixels, and larger stack sizes incorporate more information. While lower thresholds increase complexity, they yield higher average performance across the three datasets. Furthermore, we observe that an excessively large stack size does not guarantee better performance, indicating the need for careful selection of both parameters, as shown in Tab. 6

Prompt	**Accuracy (%)**	**Samples**
The digit zero	26.08	all
a number of the zero	28.46	all
a photo of number zero	30.5	all
a photo of 0	32.92	all
a photo of number the zero	33.44	all
a photo of zero	38.13	all
zero	39.31	all
a number of 0	39.82	all
This is the number of zero	40.98	all
A handwritten digit 0	41.31	all
A handwritten digit zero	42.06	all
0	43.33	all
This is the number of 0	43.54	all
The digit 0	44.66	all
The digit 0	44.81	1000
The digit 0	45.27	5000
The digit 0	46.2	3000
The digit 0	46.46	1050
The digit 0	46.71	2000

Table 5. Accuracy of different prompts for recognizing the N-MNIST zero-shot performance

Threshold	Stack Size	AUC		
		UCFCrime	XD-Violence	ShanghaiTech
40	5	57.11	63.99	53.34
25	5	59.05	60.36	55.84
10	5	62.66	55.66	48.77
5	5	63.29	64.28	58.43
40	10	57.85	60.34	54.12
25	10	60.01	61.73	56.81
10	10	63.83	49.07	50.22
5	10	62.44	63.53	53.90
40	16	58.88	55.80	51.20
25	16	64.14	57.65	54.41
10	16	60.03	44.94	49.87
5	16	55.46	55.78	49.92
40	32	54.46	48.35	47.34
25	32	55.22	50.37	45.37
10	32	62.43	40.50	44.77
5	32	56.18	56.92	48.96

Table 6. AUC Results for Different Event Constructions with clamp 10.

Class-wise results. In this section, we report the AUC for each class in the UCFCrime and ShanghaiTech datasets, as presented in the detailed results in Table3 in main context, while noting that the absence of a video label for XD-Violence precludes it from being reported. Note that the analysis excludes normal video cases. For UCFCrime, the Vandalism class achieved the highest AUC score of 79.38, while the Arrest class recorded the lowest at 48.3. Similarly, for ShanghaiTech, the Vaudeville class yielded the highest AUC score of 83.61, whereas the Monocycle class showed the lowest at 46.71 Tab. 7. These findings highlight the disparity in anomaly detection performance across different event categories. By focusing on the classes with the lowest AUC scores, such as Arrest and Monocycle, careful prompt engineering and additional targeted training could improve anomaly detection based solely on event characteristics. We leave this as a direction for future research.

UCFCrime		**Shanghai**	
Abuse	0.6221	skateboard	0.5118
Fighting	0.7047	vehicle	0.4741
Shooting	0.6736	running	0.5418
Arrest	0.4830	robbery	0.5784
Stealing	0.7233	car	0.8057
Assault	0.6964	fall	0.5199
Explosion	0.6132	fighting	0.7162
Vandalism	0.7938	throwing_object	0.6127
Arson	0.6471	vaudeville	0.8361
Shoplifting	0.5249	chasing	0.8157
Robbery	0.6780	monocycle	0.4671
Road Accidents	0.6850	-	-
Burglary	0.6528	-	-

Table 7. Class-wise AUC (%) in UCFCrime and Shanghaitech

4.4.1. Class-wise Event vs. Image Comparison

To further validate the uniqueness of our event-based pipeline, we extended the image-based VADCLIP architecture to our event modality using our event construction procedure (frame differencing, thresholding, stacking, and majority voting). Table 8 reports class-wise results across UCFCrime, XD-Violence, and ShanghaiTech.

Overall, image inputs tend to provide stronger average performance because of their richer spatial and appearance information. However, event signals capture highly discriminative, action-centric patterns that become key for particular anomaly categories. For example, in UCFCrime, *Assault* and *Shoplifting* show notable gains from events (72.03 vs. 56.44, 73.29 vs. 64.27). In ShanghaiTech, *Monocycle* and *Robbery* benefit similarly (75.32 vs. 67.23, 87.26 vs. 76.74). These cases highlight that while images dominate in terms of information richness, the action-centric representation of events provides complementary advantages and can be decisive in recognizing motion-driven anomalies.

4.5. Inference Efficiency

To complement the main results, we provide a comparison of inference efficiency in terms of latency, memory footprint, parameter count, and GFLOPs (Tab. 9). Latency was measured as the average over 100 runs with batch size = 1 on a single NVIDIA RTX 4090 GPU.

UCFCrime			XD-Violence			ShanghaiTech		
Class	Image	Event	Class	Image	Event	Class	Image	Event
Abuse	68.02	70.09	Fighting	79.59	67.81	Car	70.07	74.76
Arrest	72.21	47.09	Shooting	54.59	42.94	Chasing	94.49	84.36
Arson	65.49	66.75	Riot	97.62	86.07	Fall	72.96	65.30
Assault	56.44	72.03	Abuse	59.42	54.49	Fighting	76.91	63.48
Burglary	68.02	65.88	CarAcc.	50.83	32.53	Monocycle	67.23	75.32
Explosion	56.33	57.64	Explosion	64.32	39.22	Robbery	76.74	87.26
Fighting	58.14	79.27				Running	37.95	60.95
RoadAcc.	57.41	59.11				Skateboard	76.04	78.29
Robbery	76.03	62.39				ThrowObj.	89.63	83.14
Stealing	74.91	61.51				Vehicle	79.39	67.38
Shooting	60.95	38.42				Vaudeville	44.04	53.66
Shoplifting	64.27	73.29						
Vandalism	66.89	63.05						
AUC	86.77	78.67	AP	84.22	55.96	AUC	97.58	93.69

Table 8. Class-wise comparison of image-based and event-based VADCLIP [13] across UCFCrime, XD-Violence, and ShanghaiTech datasets. Results show that events provide complementary advantages on motion-centric classes.

	Latency (ms)		Memory (MB)		Parameters		GFLOPs	
Model	Encoder	Ext.	Encoder	Ext.	Encoder	Ext.	Encoder	Ext.
EventCLIP (ViT-L/14)	47.1	13.35	2637.05	145.38	729,397,760	38,110,561	347.02	503.59
EventCLIP (ViT-B/32)	29.52	13.46	952.10	96.80	249,590,016	25,374,273	30.77	63.25
EventBind (ViT-L/14)	47.51	13.67	2637.05	145.38	729,397,760	38,110,561	347.02	503.59
EventBind (ViT-B/32)	29.71	13.77	952.10	96.80	249,590,016	25,374,273	30.77	63.25
Ours (ViT-L/14)	20.75	0.091	1631.23	–	427,616,513	–	56.26	0
Ours (ViT-B/32)	13.79	0.090	577.08	–	151,277,313	–	4.88	0

Table 9. Comparison of latency, memory usage, parameters, and GFLOPs. Latency is measured as the average over 100 runs with batch size = 1. Here, **Encoder** refers to the vision backbone (e.g., ViT), and **Ext.** indicates the additional external modules outside the encoder (e.g., projection layers, fusion heads).

Our method achieves significantly lower inference latency than both EventCLIP and EventBind across backbones. For example, with ViT-L/14, our encoder runs in **20.75 ms** compared to 47.1–47.5 ms for EventCLIP and EventBind. With ViT-B/32, we observe **13.79 ms** versus $\sim$29.5 ms for the baselines. This corresponds to more than a 2× speed-up, demonstrating the suitability of our approach for real-time applications.

The efficiency gains can be attributed to architectural simplicity. EventCLIP and EventBind introduce additional external modules such as temporal reconstruction layers, projection heads, or fusion components, which incur substantial overhead outside the backbone encoder. In contrast, our design remains structurally identical to CLIP: the event representation is directly processed by the vision backbone, and only a similarity computation with text or image embeddings is performed afterwards. This design eliminates heavy external computation, reduces memory usage and GFLOPs, and ensures scalability without sacrificing alignment quality.

These results confirm that the proposed encoder not only improves recognition and retrieval performance but also maintains practical efficiency, a critical property for deploying event-based systems in real-world, low-latency scenarios.

5. Visualization

5.1. Visual Understanding of model.

We visualize the model's internal representations using attention maps and text relevance scores in Fig. 4. The top row highlights relevance between textual descriptions and RGB inputs, while the bottom focuses on class label association. Interestingly, our model attends to semanti-

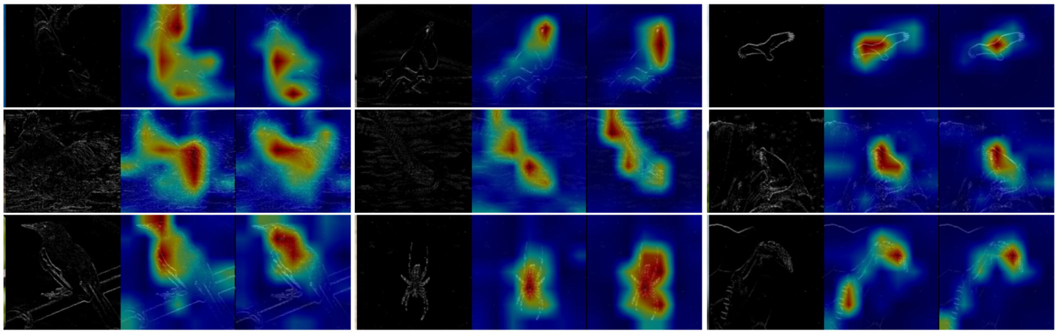


Figure 1. N-ImageNet pre-training model attention map vs fine-tuning model attention map. **Left** is E, **Middle** is pre-training attention and **Right** is fine-tuned attention.

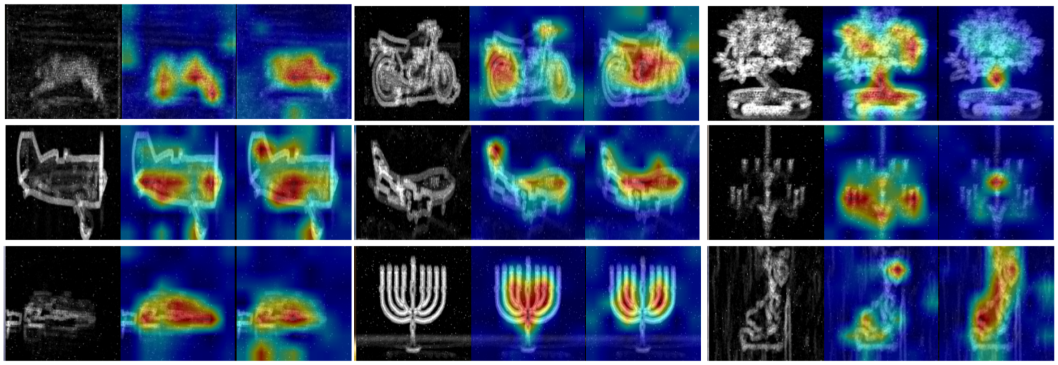


Figure 2. N-Caltech pre-training model attention map vs fine-tuning model attention map. **Left** is E, **Middle** is pre-training attention and **Right** is fine-tuned attention.

cally meaningful regions and de-emphasizes irrelevant visual cues (e.g., color), even for untrained background elements—showing evidence of robust alignment between event/image inputs and textual representations.

5.2. Clamp Distribution

We observe from Tab. 4 that the performance varies depending on the distribution of E, motivating further investigation into the visual differences introduced by different clamp values. Visual analysis reveals that smaller clamp values result in a higher number of activated pixels with large values after normalization, whereas larger clamp values reduce the number of pixels with high intensity (white regions). While clamp 5 may appear visually clearer in the input dimension, the value range is constrained between 0 and 5, which does not necessarily ensure superior performance, as also indicated in Tab. 4. Therefore, exploring an appropriate representation becomes a critical consideration when processing event modality data.

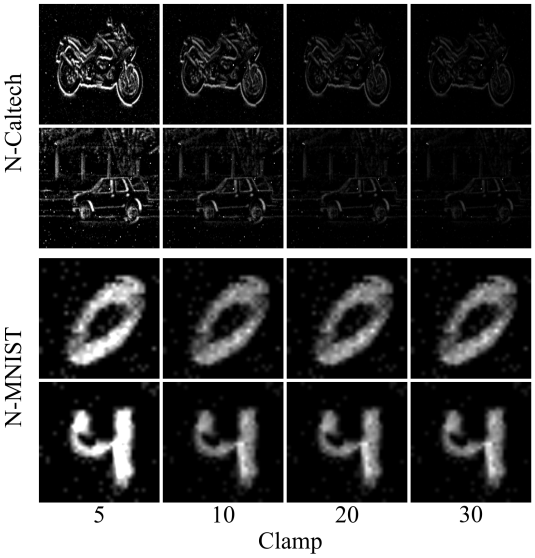
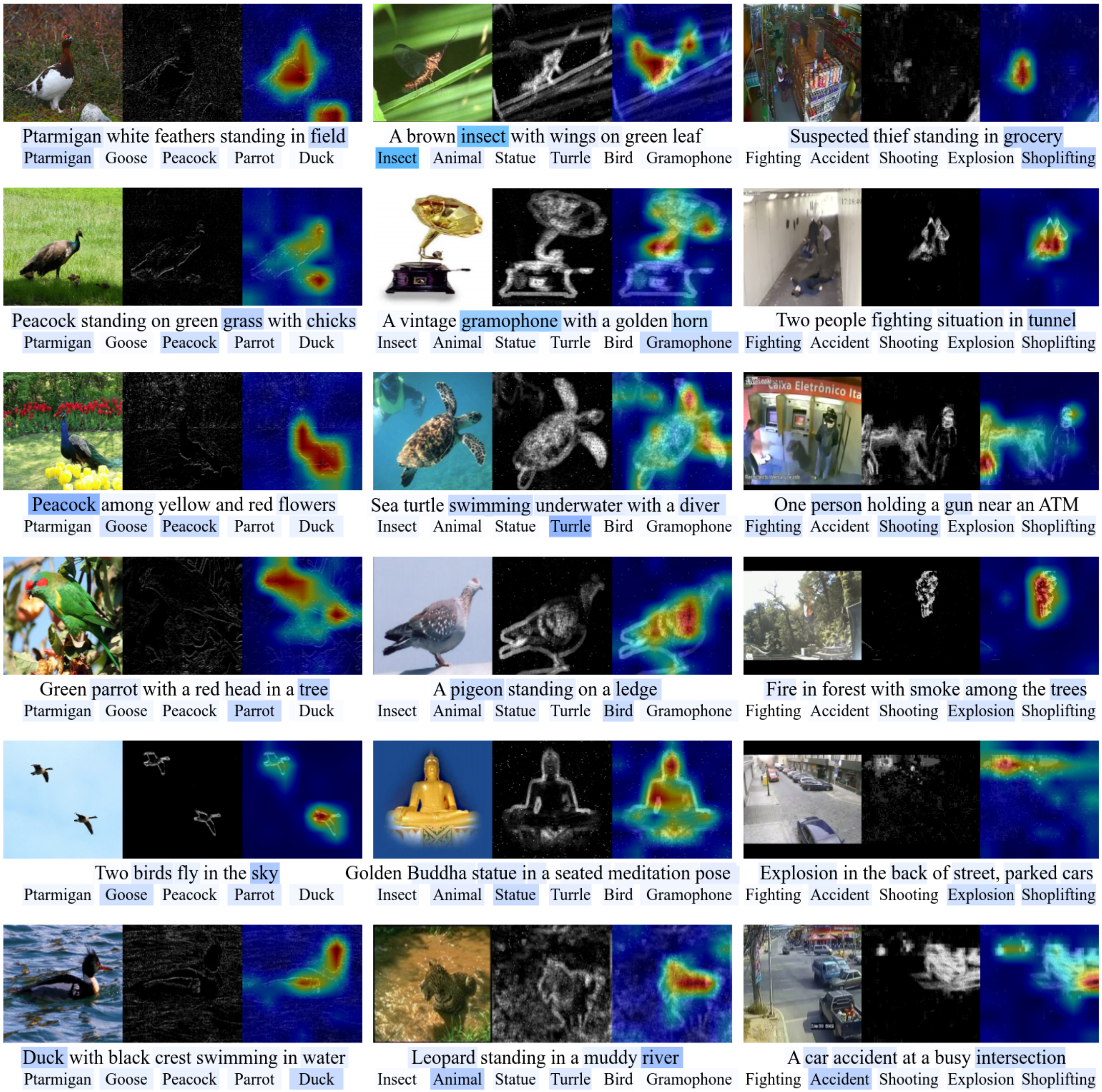


Figure 3. Event visualization across different clamp.

5.3. Attention map

This section provides a visual analysis of the changes in the model's attention maps before and after fine-tuning. During pre-training, the attention scope is relatively narrow, whereas fine-tuning often results in broader attention spans. However, there are also cases where the attention range becomes significantly more constrained follow-


Ptarmigan white feathers standing in field
Ptarmigan Goose Peacock Parrot Duck
A brown insect with wings on green leaf
Insect Animal Statue Turrle Bird Gramophone
Suspected thief standing in grocery
Fighting Accident Shooting Explosion Shoplifting
Peacock standing on green grass with chicks
Ptarmigan Goose Peacock Parrot Duck
A vintage gramophone with a golden horn
Insect Animal Statue Turrle Bird Gramophone
Two people fighting situation in tunnel
Fighting Accident Shooting Explosion Shoplifting
Peacock among yellow and red flowers
Ptarmigan Goose Peacock Parrot Duck
Sea turtle swimming underwater with a diver
Insect Animal Statue Turrle Bird Gramophone
One person holding a gun near an ATM
Fighting Accident Shooting Explosion Shoplifting
Green parrot with a red head in a tree
Ptarmigan Goose Peacock Parrot Duck
A pigeon standing on a ledge
Insect Animal Statue Turrle Bird Gramophone
Fire in forest with smoke among the trees
Fighting Accident Shooting Explosion Shoplifting
Two birds fly in the sky
Ptarmigan Goose Peacock Parrot Duck
Golden Buddha statue in a seated meditation pose
Insect Animal Statue Turrle Bird Gramophone
Explosion in the back of street, parked cars
Fighting Accident Shooting Explosion Shoplifting
Duck with black crest swimming in water
Ptarmigan Goose Peacock Parrot Duck
Leopard standing in a muddy river
Insect Animal Statue Turrle Bird Gramophone
A car accident at a busy intersection
Fighting Accident Shooting Explosion Shoplifting

(a) N-ImageNet (b) N-Caltech (c) UCFCrime

Figure 4. Visualize grad-based attention maps and text relevance scores with our pre-trained model (a) N-ImageNet unseen classes, (b) N-Caltech, (c) UCFCrime following the method in [1].

ing fine-tuning. Pre-trained models are typically regarded as extracting more generalized features. Notably, most attention regions remain consistent after fine-tuning, with observed changes primarily involving expansion or contraction within existing regions. These findings suggest that pre-training effectively establishes a robust foundation, enabling the model to adapt efficiently through minimal fine-tuning adjustments Fig. 1 and Fig. 2.

References

[1] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021. 8

[2] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9640–9649, 2021. 2

[3] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190, 2023. 3

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1

[5] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*, pages 534–542. IEEE, 2020. 3

[6] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2146–2156, 2021. 2

[7] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[8] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE international conference on computer vision*, pages 341–349, 2017. 3

[9] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015. 2

[10] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015. 3

[11] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[12] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018. 3

[13] Peng Wu, Xuerong Zhou, Guansong Pang, Lingru Zhou, Qingsen Yan, Peng Wang, and Yanning Zhang. Vadclip: Adapting vision-language models for weakly supervised video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. 6

[14] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10699–10709, 2023. 2

[15] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19125–19136, 2023. 1