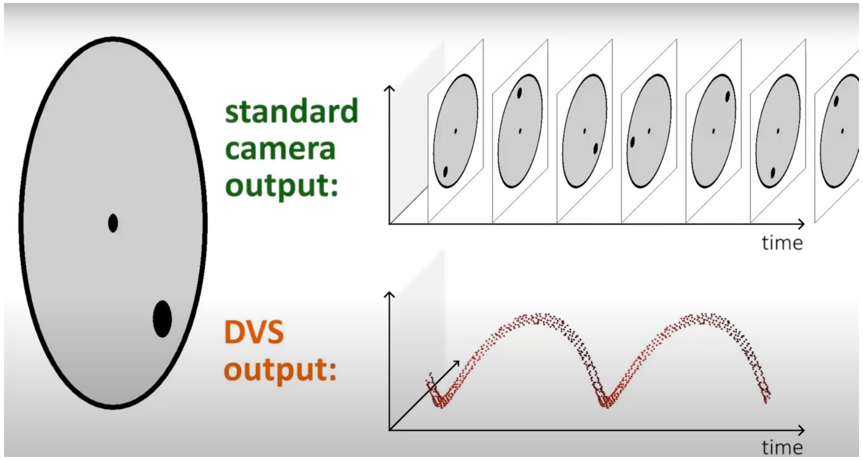
Associative Memory Augmented Asynchronous Spatiotemporal Representation Learning for Event-based Perception

Uday Kamal*, Saurabh Dash*, Saibal Mukhopadhyay

School of Electrical and Computer Engineering
Georgia Institute of Technology

ICLR 2023
Eleventh International Conference on Learning Representations

Event-based Camera: An Overview



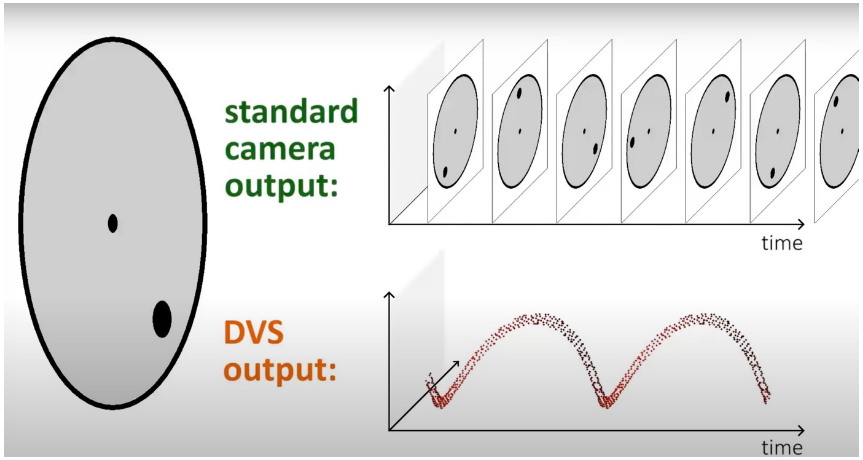
Video source: https://youtu.be/LauQ6LWTkxM?t=30

Event camera records per-pixel brightness change asynchronously

Advantage over frame-based camera:

- High dynamic range (>120db)
- Ultra-low latency
- High temporal resolution (~15 μs)
- Low power consumption (~10 mW)

Event-based Camera: An Overview



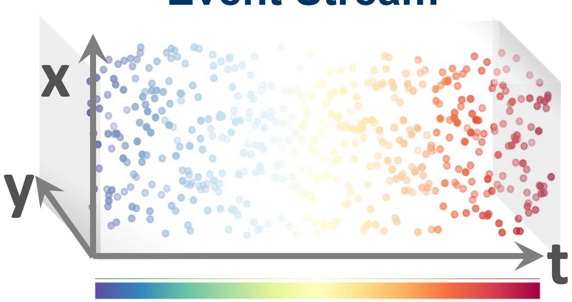
Video source: https://youtu.be/LauQ6LWTkxM?t=30

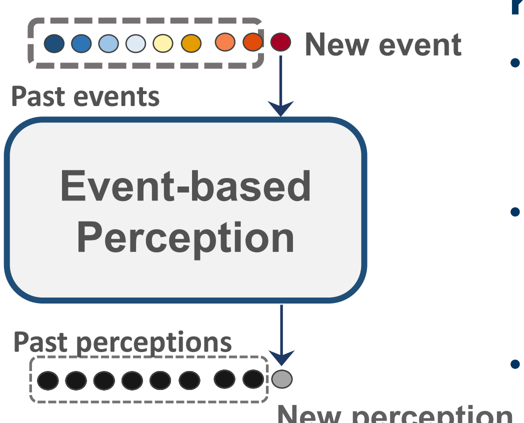
Event camera records per-pixel brightness change asynchronously

Advantage over frame-based camera:

- High dynamic range (>120db)
- Ultra-low latency
- High temporal resolution (~15 μs)
- Low power consumption (~10 mW)

Event Stream

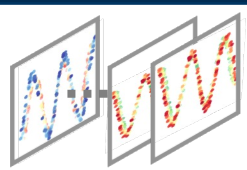
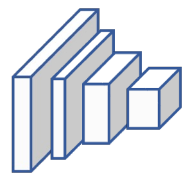
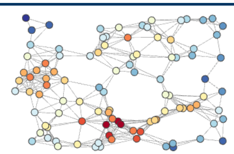
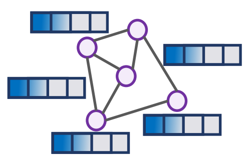
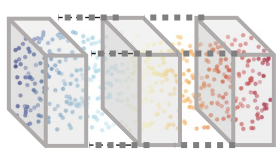
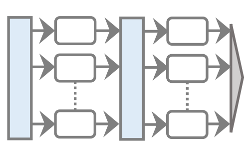




Key Algorithm Challenges:

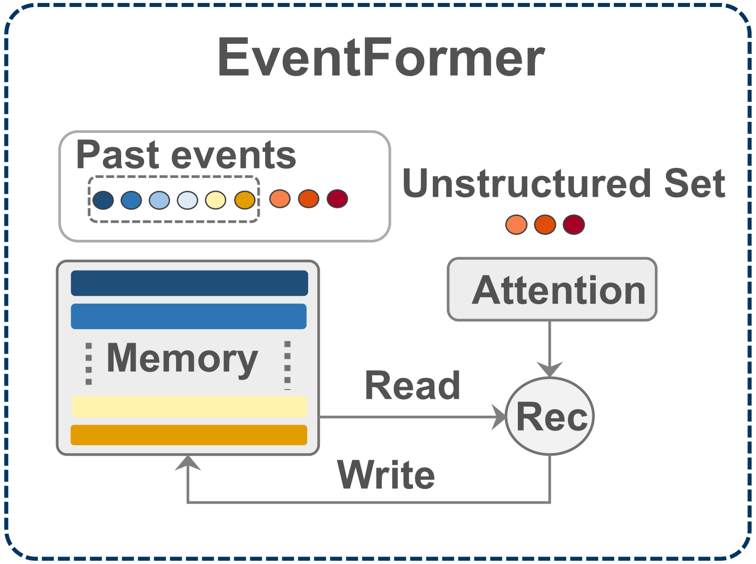
- **Spatiotemporal encoding:** Encoding of events to a latent-representation that can correlate across both space and time
- **Efficient update:** Update latent representation only '*when*' and '*where*' there is an event
- **Task specific decoding:** Decode the latent representation to target task output.

Existing Works

Representation	Processing Method	Limitation
 Event-aggregated Frames	 CNN — Temporal aggregation of the events maps them into discrete, dense frames and use CNN[1]	Dense processing, discards the sparse nature of the events and wasteful of computations
 Spatiotemporal Event Graph	 GNN — Maps events into a spatiotemporal graph and apply graph neural networks (GNN) [2]	Requires storing the past events on the space-time graph and re-processing them
 Events as Point Clouds	 PointNet — Treats events as space-time point-cloud and applies point-based processing[3]	Requires storing and processing the past events to correlate with new events

1. Gehrig et al., End-to-end Representation Learning for Event-based Camera, ICCV'19
2. Li et al., Graph-based Asynchronous Event Processing for Rapid Object Recognition. ICCV'21
3. Wang et al., Space-time Event Cloud for Hand Gesture Recognition, WACV'19

Proposed Method: Overview

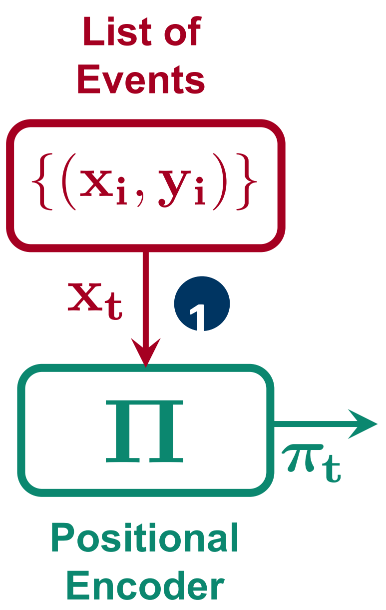


Given a list of events at time *t*, EventFormer:

- **Refine :** Models higher-order interaction among the events (space).
- **Read:** Retrieves past representations at current event locations (time).
- **Recurrence:** Combines spatial and temporal information using an event-based recurrent mechanism to compute refined spatio-temporal representation.
- **Write:** Update the associative memory with the refined representation.

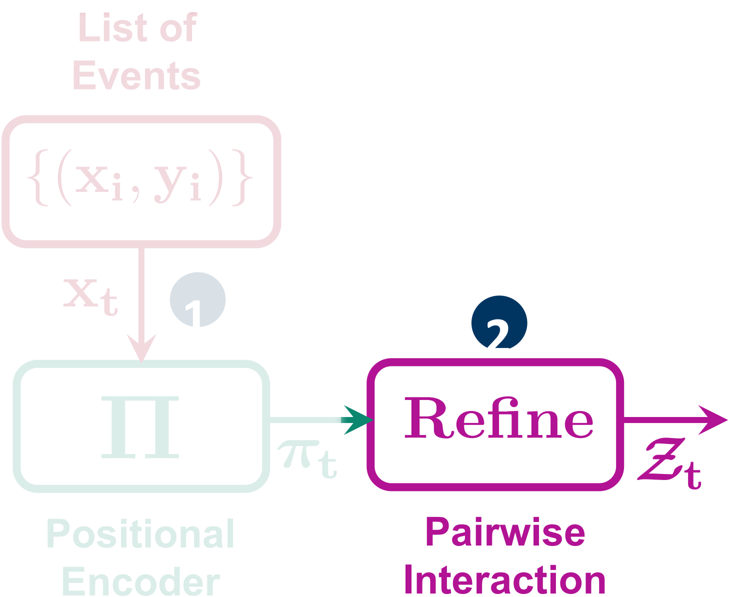
Proposed Method: Details

Proposed Method: Details



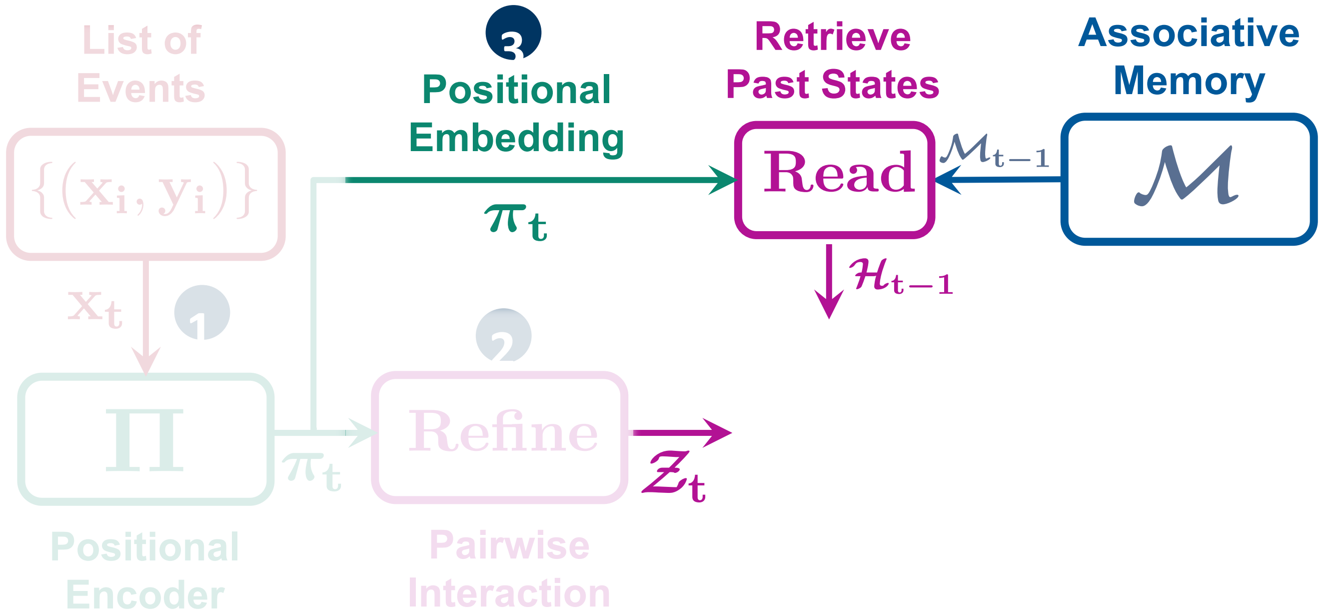
1. Generate positional embedding, π_t for list of event locations

Proposed Method: Details



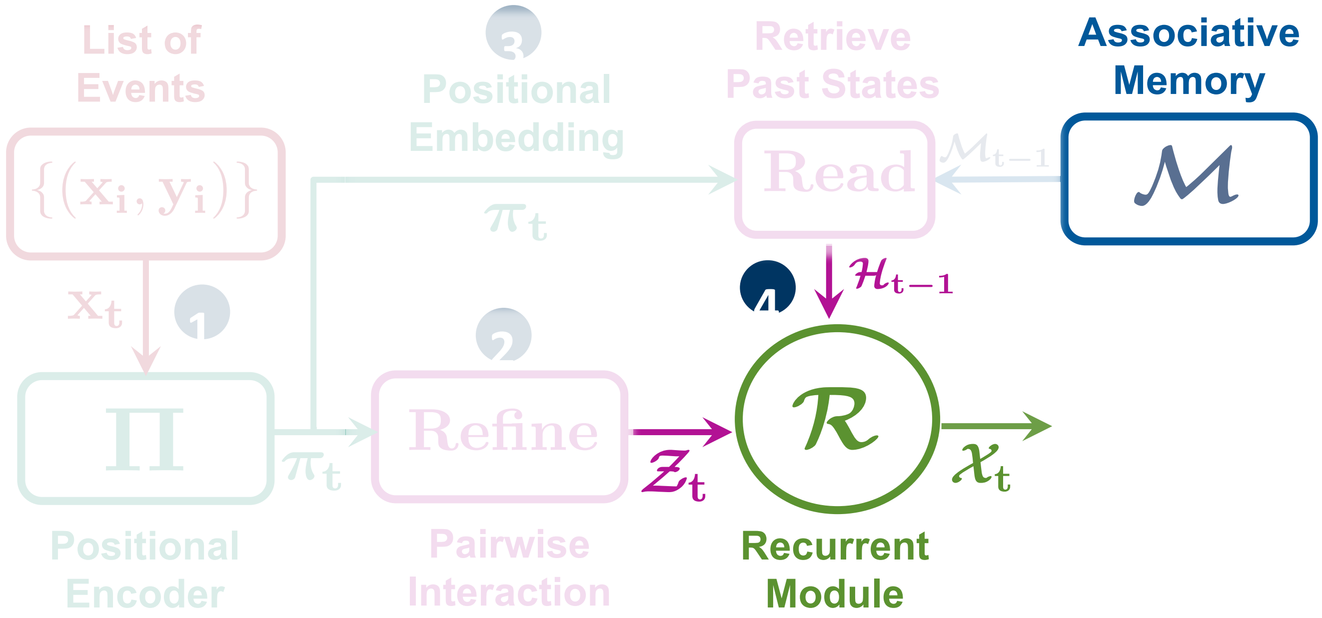
1. Generate positional embedding, π_t for list of event locations
2. Self-attention to capture their pair-wise interactions, $\mathcal{Z}_t$

Proposed Method: Details



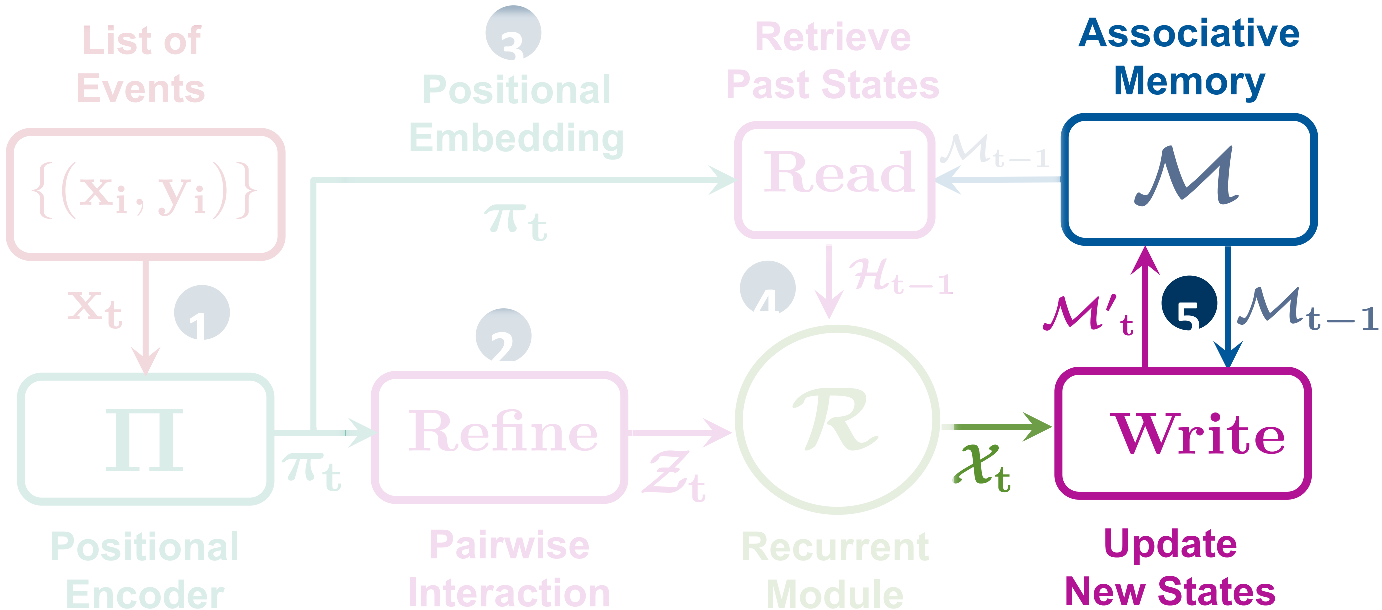
1. Generate positional embedding, π_t for list of event locations
2. Self-attention to capture their pair-wise interactions, $\mathcal{Z}_t$
3. Retrieve the past hidden states, $\mathcal{H}_{t-1}$ at current positional embedding, π_t

Proposed Method: Details



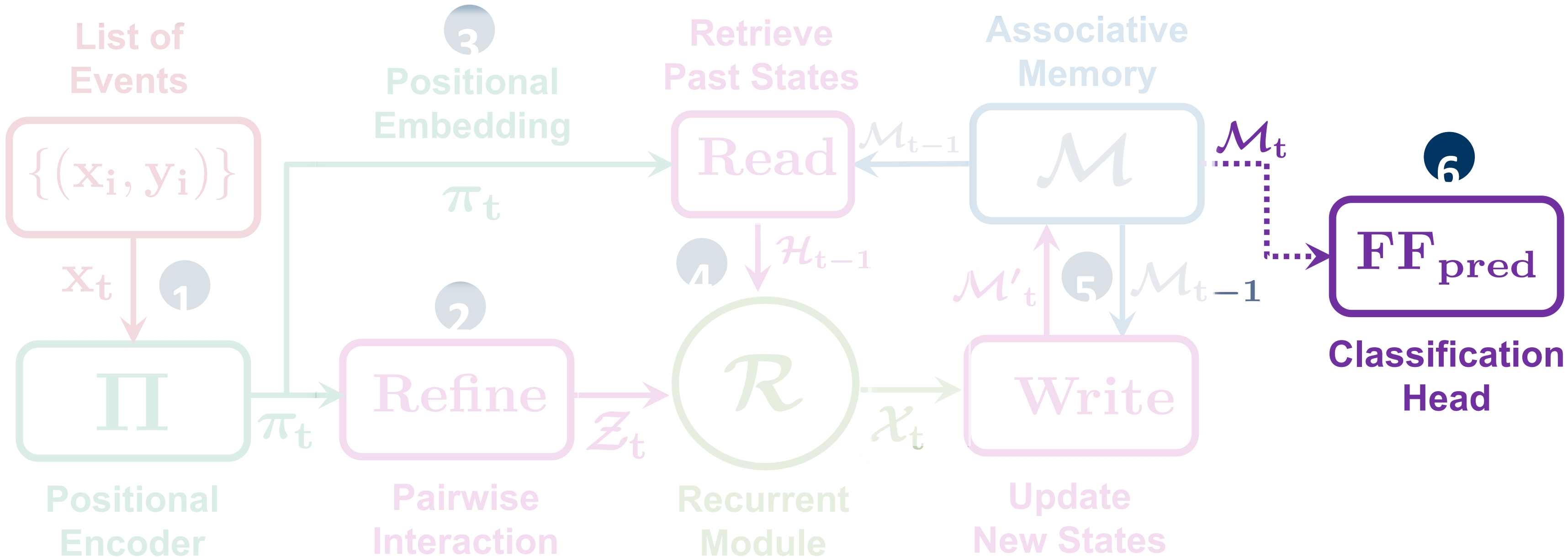
1. Generate positional embedding, π_t for list of event locations
2. Self-attention to capture their pair-wise interactions, $\mathcal{Z}_t$
3. Retrieve the past hidden states, $\mathcal{H}_{t-1}$ at current positional embedding, π_t
4. Updated representation, $\mathcal{X}_t$ through a recurrent module using $\mathcal{H}_{t-1}$ and $\mathcal{Z}_t$

Proposed Method: Details



1. Generate positional embedding, π_t for list of event locations
2. Self-attention to capture their pair-wise interactions, $\mathcal{Z}_t$
3. Retrieve the past hidden states, $\mathcal{H}_{t-1}$ at current positional embedding, π_t
4. Updated representation, $\mathcal{X}_t$ through a recurrent module using $\mathcal{H}_{t-1}$ and $\mathcal{Z}_t$
5. Update the associative memory with new representation, $\mathcal{X}_t$

Proposed Method: Details



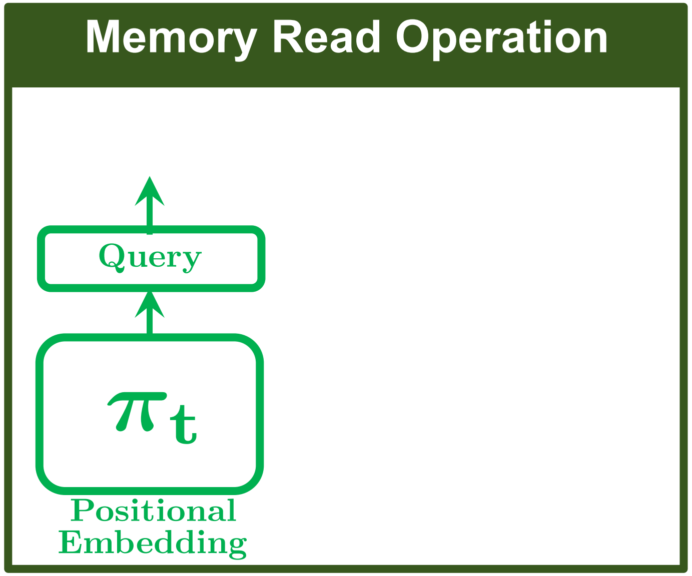
1. Generate positional embedding, π_t for list of event locations
2. Self-attention to capture their pair-wise interactions, $\mathcal{Z}_t$
3. Retrieve the past hidden states, $\mathcal{H}_{t-1}$ at current positional embedding, π_t
4. Updated representation, $\mathcal{X}_t$ through a recurrent module using $\mathcal{H}_{t-1}$ and $\mathcal{Z}_t$
5. Update the associative memory with new representation, $\mathcal{X}_t$
6. Leverage the updated memory representation, $\mathcal{M}_t$ to optimize the task.

EventFormer: Memory Read Operation

Gigascale Reliable Energy Efficient Nanosystems LAB

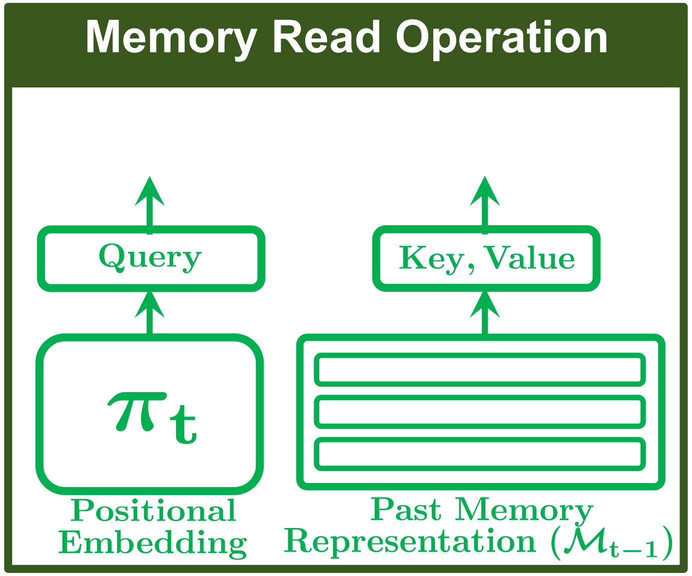
Memory Read Operation

EventFormer: Memory Read Operation



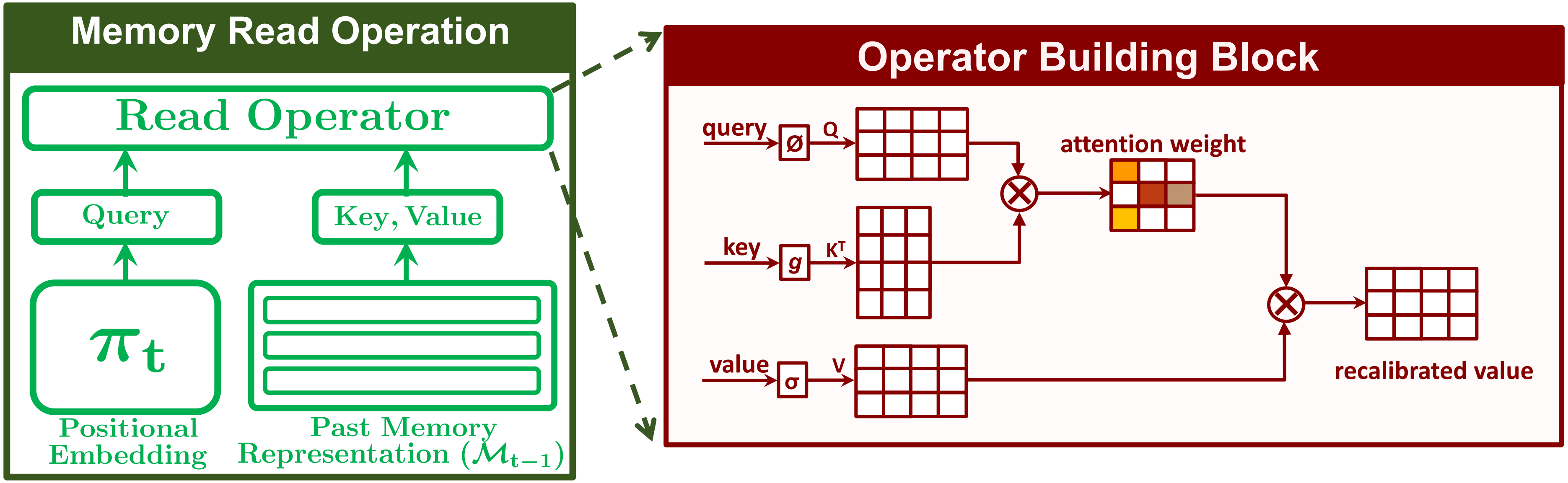
- ***Query*** the associative memory with the positional embedding of the current event locations, π_t

EventFormer: Memory Read Operation



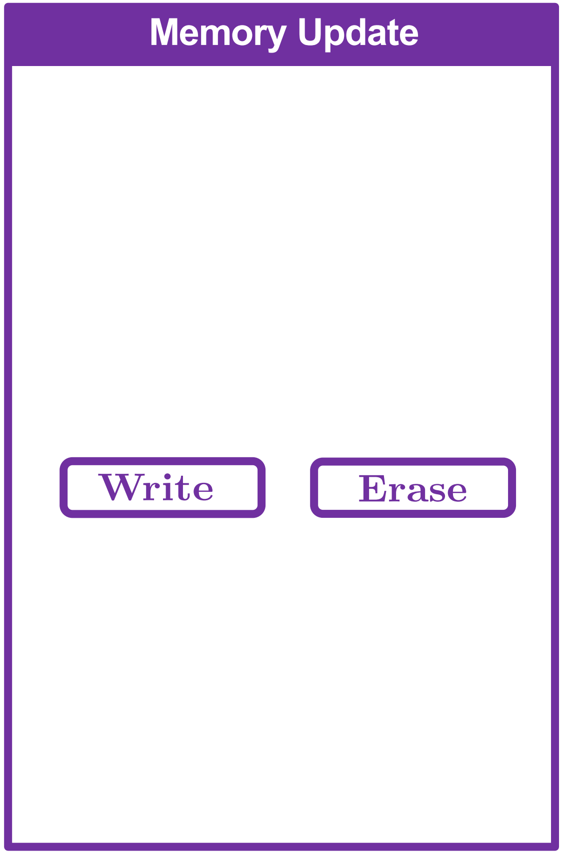
- ***Query*** the associative memory with the positional embedding of the current event locations, π_t
- Project the past memory representation, $\mathcal{M}_{t-1}$ into ***Key*** and ***Value*** space

EventFormer: Memory Read Operation

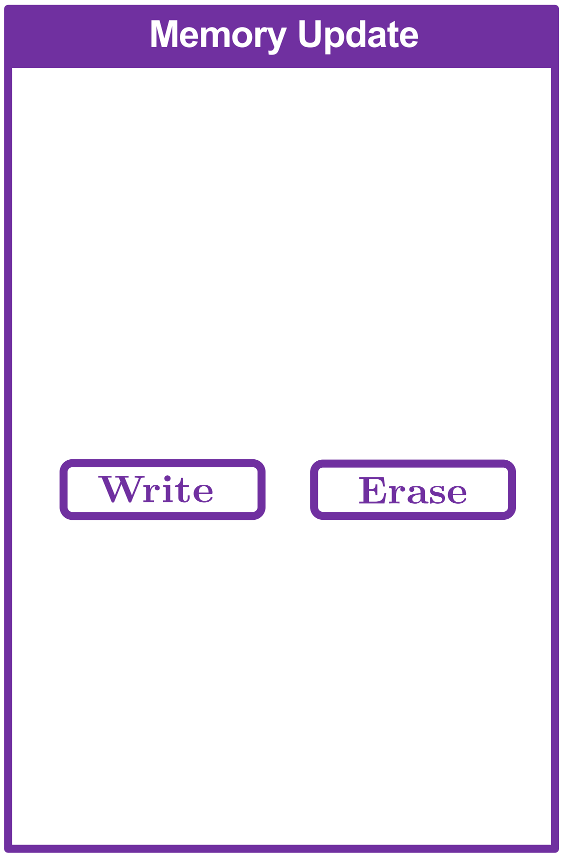


- ***Query*** the associative memory with the positional embedding of the current event locations, π_t
- Project the past memory representation, $\mathcal{M}_{t-1}$ into ***Key*** and ***Value*** space
- The retrieved state is a weighted sum of past memory representation where the weights are computed through cross-attention from the projected ***Query, Key, Value*** space.

EventFormer: Memory Update Operation

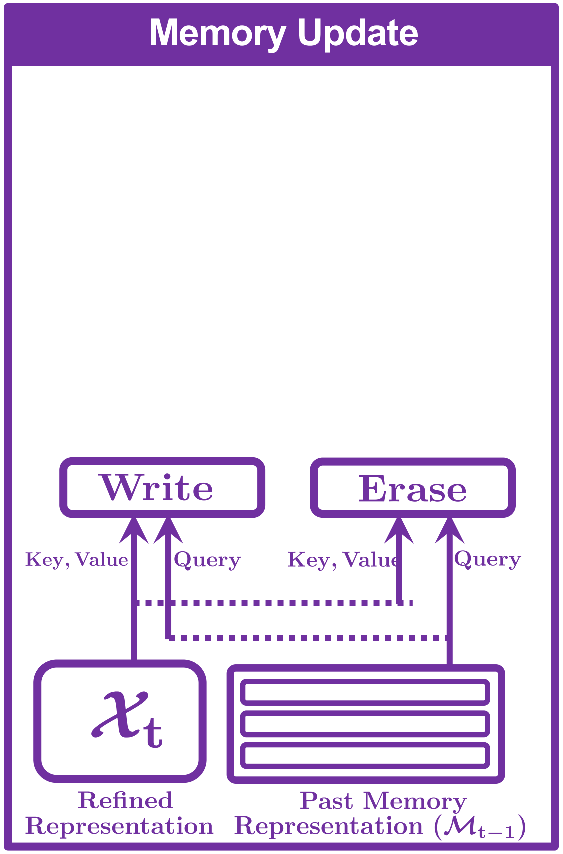


EventFormer: Memory Update Operation



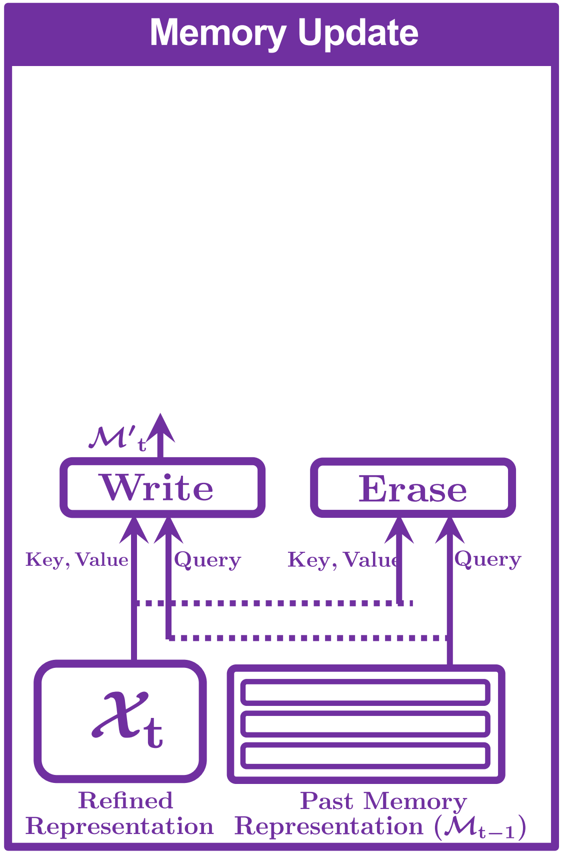
- Memory update occurs through a combination of ‘***Write***’ and ‘***Erase***’ operator.

EventFormer: Memory Update Operation



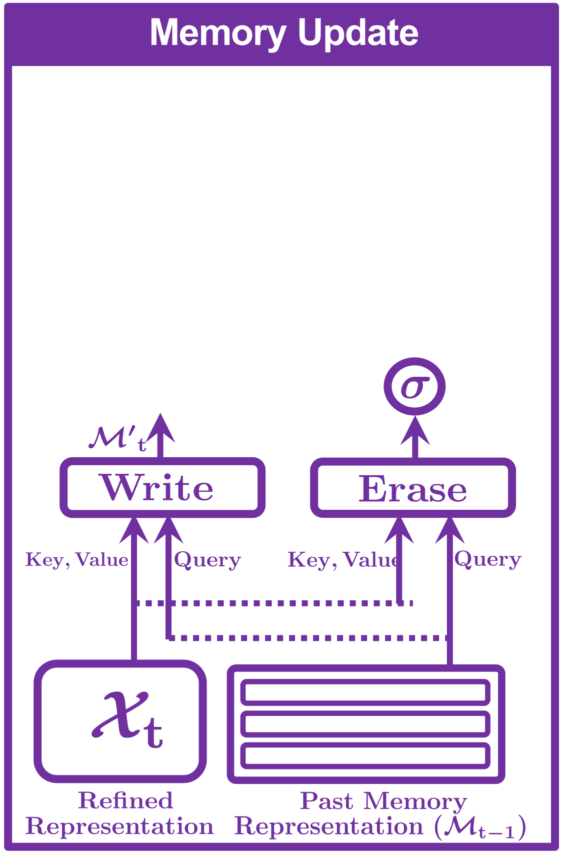
- Memory update occurs through a combination of '***Write***' and '***Erase***' operator.
- Both '***Write***' and '***Erase***' operator takes past memory representation, M_{t-1} as **query** and refined representation, X_t as **key-value** pair.

EventFormer: Memory Update Operation



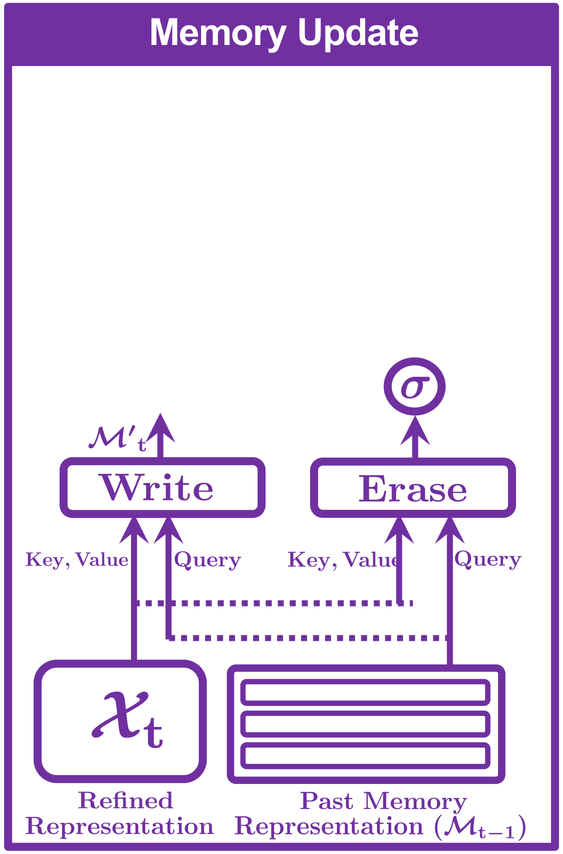
- Memory update occurs through a combination of '***Write***' and '***Erase***' operator.
- Both '***Write***' and '***Erase***' operator takes past memory representation, M_{t-1} as **query** and refined representation, X_t as **key-value** pair.
- '***Write***' operator computes the new memory representation, M'_t

EventFormer: Memory Update Operation



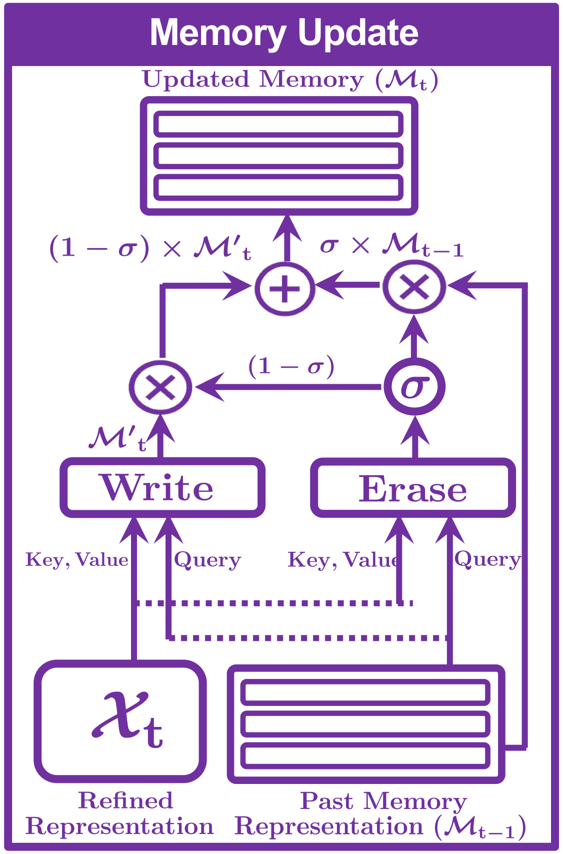
- Memory update occurs through a combination of '***Write***' and '***Erase***' operator.
- Both '***Write***' and '***Erase***' operator takes past memory representation, M_{t-1} as **query** and refined representation, X_t as **key-value** pair.
- '***Write***' operator computes the new memory representation, M'_t
- 'Erase' operator computes the relative update strength, $\sigma_t \in [0,1]$

EventFormer: Memory Update Operation



- Memory update occurs through a combination of '***Write***' and '***Erase***' operator.
- Both '***Write***' and '***Erase***' operator takes past memory representation, M_{t-1} as **query** and refined representation, X_t as **key-value** pair.
- '***Write***' operator computes the new memory representation, M'_t
- 'Erase' operator computes the relative update strength, $\sigma_t \in [0,1]$
- We use σ_t to partially forget the past information and $(1 - \sigma_t)$ to modulate the new information.

EventFormer: Memory Update Operation



- Memory update occurs through a combination of ‘***Write***’ and ‘***Erase***’ operator.
- Both ‘***Write***’ and ‘***Erase***’ operator takes past memory representation, M_{t-1} as **query** and refined representation, X_t as **key-value** pair.
- ‘***Write***’ operator computes the new memory representation, M'_t
- ‘Erase’ operator computes the relative update strength, $\sigma_t \in [0,1]$
- We use σ_t to partially forget the past information and $(1-\sigma_t)$ to modulate the new information.
- Finally, we update the memory using a linear combination of past and new information: $M_t = \sigma_t * M_{t-1} + (1-\sigma_t) * M'_t$

EventFormer: Quantitative Performance

GIGASCALE RELIABLE ENERGY EFFICIENT NANOSYSTEMS LAB

We evaluate EventFormer on event-based classification task on N-Caltech101 and N-Cars dataset

Methods	Representation	Event-based?	N-Caltech101		N-Cars	
			Accuracy	MFLOPs/ev	Accuracy	MFLOPs/ev
H-First	Spike	✓	0.054	-	0.561	-
Gabor-SNN	Spike	✓	0.284	-	0.789	-
HOTS	Time-Surface	✓	0.21	54	0.624	14
HATS	Time-Surface	✓	0.642	4.3	0.902	0.03
DART	Time-Surface	✓	0.664	-	-	-
EST	Event-Histogram	X	0.817	4150	0.925	1050
Matrix-LSTM	Event-Histogram	X	0.843	1580	0.926	1250
YOLE	Voxel-Grid	✓	0.702	3659	0.927	328.16
AsyNet	Voxel-Grid	✓	0.745	202	0.944	21.5
EvS-S	Graph	✓	0.761	11.5	0.931	6.1
AEGNN	Graph	✓	0.668	0.369	0.945	0.03

MFLOPs/ev: total number of FLOPs to process one event on average

EventFormer: Quantitative Performance

GIGASCALE RELIABLE ENERGY EFFICIENT NANOSYSTEMS LAB

We evaluate EventFormer on event-based classification task on N-Caltech101 and N-Cars dataset

Methods	Representation	Event-based?	N-Caltech101		N-Cars	
			Accuracy	MFLOPs/ev	Accuracy	MFLOPs/ev
H-First	Spike	✓	0.054	-	0.561	-
Gabor-SNN	Spike	✓	0.284	-	0.789	-
HOTS	Time-Surface	✓	0.21	54	0.624	14
HATS	Time-Surface	✓	0.642	4.3	0.902	0.03
DART	Time-Surface	✓	0.664	-	-	-
EST	Event-Histogram	X	0.817	4150	0.925	1050
Matrix-LSTM	Event-Histogram	X	0.843	1580	0.926	1250
YOLE	Voxel-Grid	✓	0.702	3659	0.927	328.16
AsyNet	Voxel-Grid	✓	0.745	202	0.944	21.5
EvS-S	Graph	✓	0.761	11.5	0.931	6.1
AEGNN	Graph	✓	0.668	0.369	0.945	0.03

MFLOPs/ev: total number of FLOPs to process one event on average

- Frame based methods achieve higher accuracy at the cost of high compute cost, due to their dense processing

EventFormer: Quantitative Performance

GIGASCALE RELIABLE ENERGY EFFICIENT NANOSYSTEMS LAB

We evaluate EventFormer on event-based classification task on N-Caltech101 and N-Cars dataset

Methods	Representation	Event-based?	N-Caltech101		N-Cars	
			Accuracy	MFLOPs/ev	Accuracy	MFLOPs/ev
H-First	Spike	✓	0.054	-	0.561	-
Gabor-SNN	Spike	✓	0.284	-	0.789	-
HOTS	Time-Surface	✓	0.21	54	0.624	14
HATS	Time-Surface	✓	0.642	4.3	0.902	0.03
DART	Time-Surface	✓	0.664	-	-	-
EST	Event-Histogram	X	0.817	4150	0.925	1050
Matrix-LSTM	Event-Histogram	X	0.843	1580	0.926	1250
YOLE	Voxel-Grid	✓	0.702	3659	0.927	328.16
AsyNet	Voxel-Grid	✓	0.745	202	0.944	21.5
EvS-S	Graph	✓	0.761	11.5	0.931	6.1
AEGNN	Graph	✓	0.668	0.369	0.945	0.03

MFLOPs/ev: total number of FLOPs to process one event on average

- Frame based methods achieve higher accuracy at the cost of high compute cost, due to their dense processing

EventFormer: Quantitative Performance

GIGASCALE RELIABLE ENERGY EFFICIENT NANOSYSTEMS LAB

We evaluate EventFormer on event-based classification task on N-Caltech101 and N-Cars dataset

Methods	Representation	Event-based?	N-Caltech101		N-Cars	
			Accuracy	MFLOPs/ev	Accuracy	MFLOPs/ev
H-First	Spike	✓	0.054	-	0.561	-
Gabor-SNN	Spike	✓	0.284	-	0.789	-
HOTS	Time-Surface	✓	0.21	54	0.624	14
HATS	Time-Surface	✓	0.642	4.3	0.902	0.03
DART	Time-Surface	✓	0.664	-	-	-
EST	Event-Histogram	X	0.817	4150	0.925	1050
Matrix-LSTM	Event-Histogram	X	0.843	1580	0.926	1250
YOLE	Voxel-Grid	✓	0.702	3659	0.927	328.16
AsyNet	Voxel-Grid	✓	0.745	202	0.944	21.5
EvS-S	Graph	✓	0.761	11.5	0.931	6.1
AEGNN	Graph	✓	0.668	0.369	0.945	0.03

MFLOPs/ev: total number of FLOPs to process one event on average

- Frame based methods achieve higher accuracy at the cost of high compute cost, due to their dense processing
- Asynchronous graph-based methods have the high compute efficiency but their performance is limited by their capacity of storing and processing past events on the graph

EventFormer: Quantitative Performance

GIGASCALE RELIABLE ENERGY EFFICIENT NANOSYSTEMS LAB

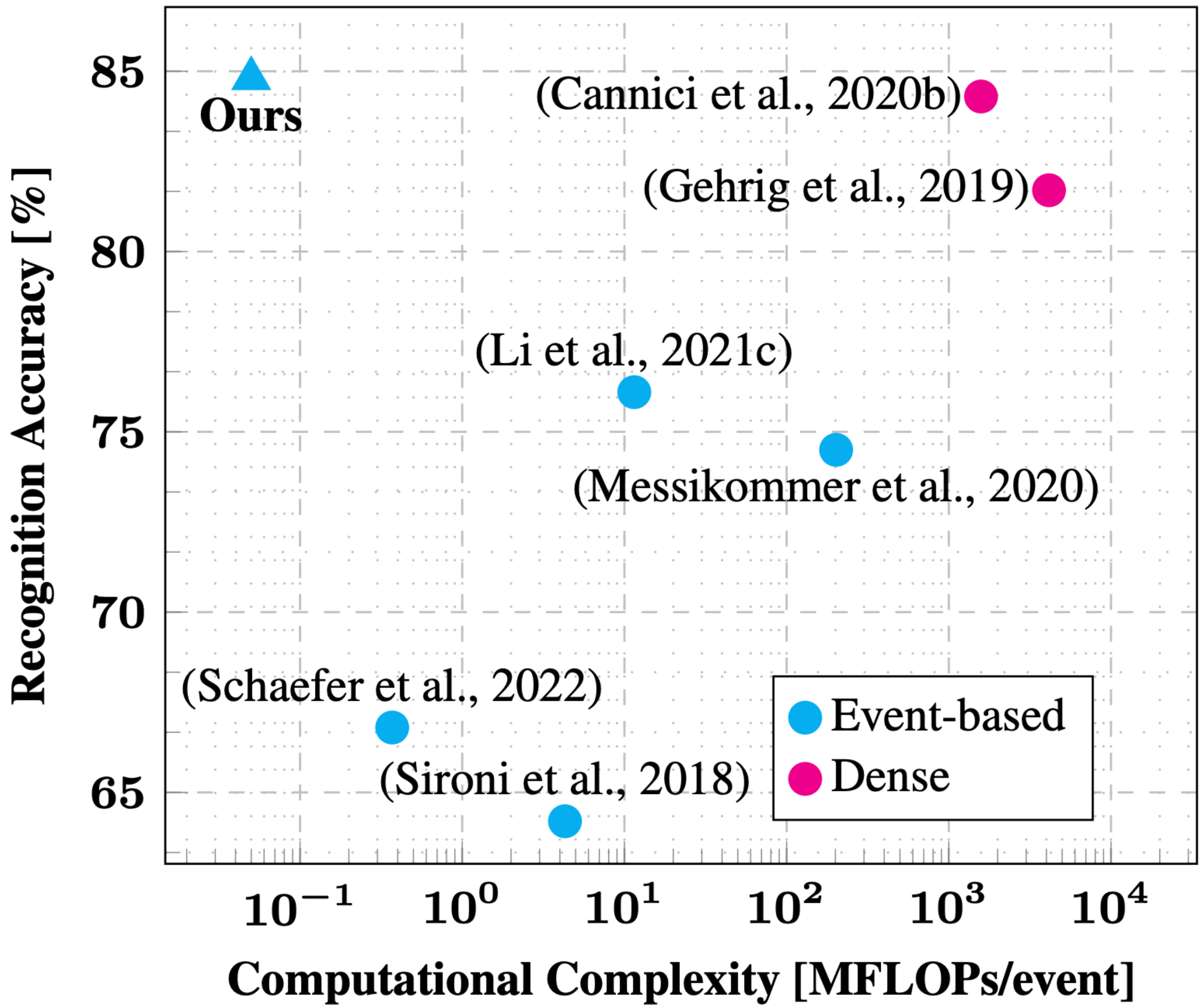
We evaluate EventFormer on event-based classification task on N-Caltech101 and N-Cars dataset

Methods	Representation	Event-based?	N-Caltech101		N-Cars	
			Accuracy	MFLOPs/ev	Accuracy	MFLOPs/ev
H-First	Spike	✓	0.054	-	0.561	-
Gabor-SNN	Spike	✓	0.284	-	0.789	-
HOTS	Time-Surface	✓	0.21	54	0.624	14
HATS	Time-Surface	✓	0.642	4.3	0.902	0.03
DART	Time-Surface	✓	0.664	-	-	-
EST	Event-Histogram	X	0.817	4150	0.925	1050
Matrix-LSTM	Event-Histogram	X	0.843	1580	0.926	1250
YOLE	Voxel-Grid	✓	0.702	3659	0.927	328.16
AsyNet	Voxel-Grid	✓	0.745	202	0.944	21.5
EvS-S	Graph	✓	0.761	11.5	0.931	6.1
AEGNN	Graph	✓	0.668	0.369	0.945	0.03
EventFormer	Unstructured Set	✓	0.848	0.048	0.943	0.013

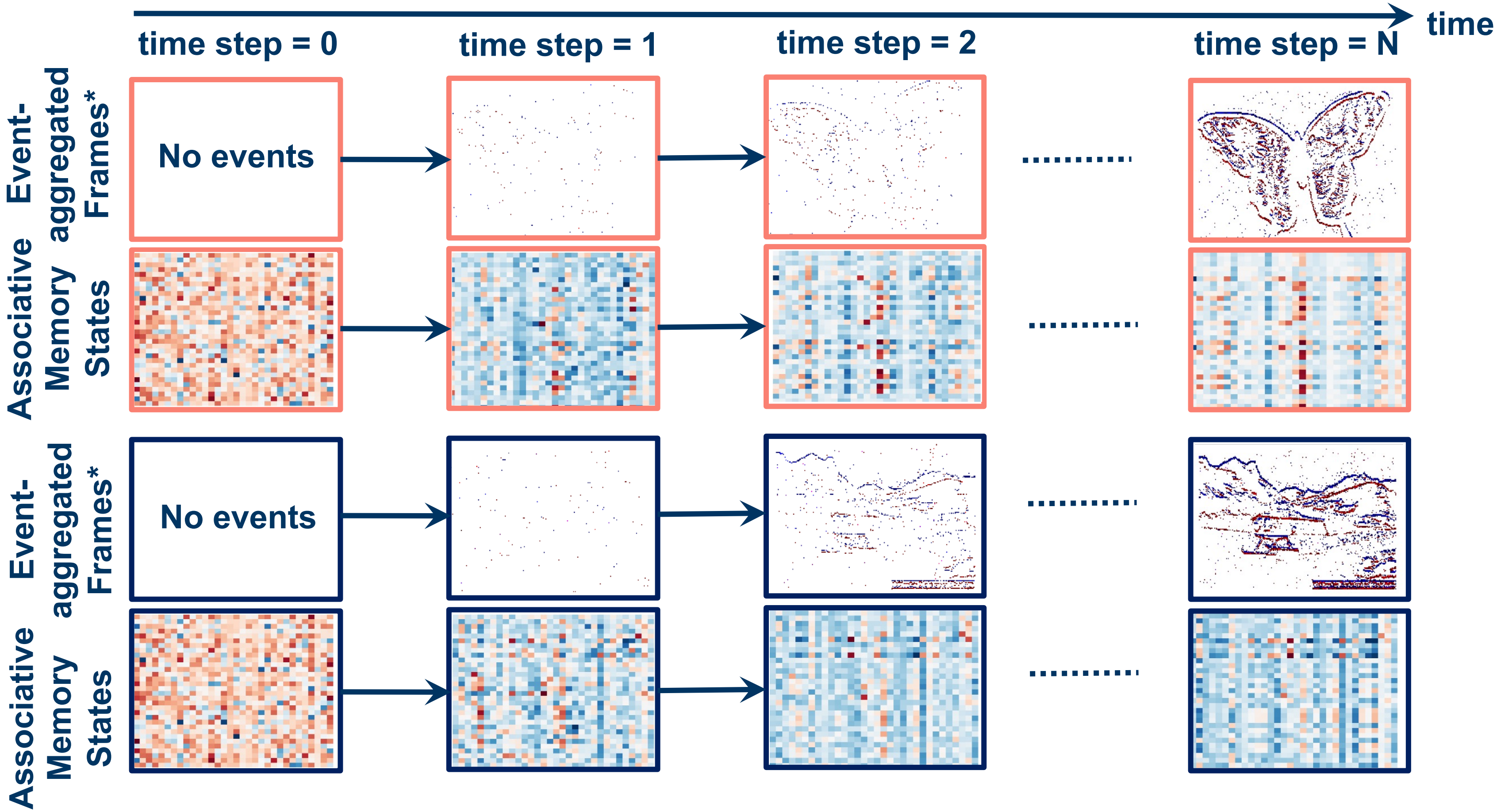
MFLOPs/ev: total number of FLOPs to process one event on average

- Frame based methods achieve higher accuracy at the cost of high compute cost, due to their dense processing
- Asynchronous graph-based methods have the high compute efficiency but their performance is limited by their capacity of storing and processing past events on the graph
- EventFormer enjoys both the high compute efficiency (storing and processing past events in a compressed latent space) and high accuracy (rich spatiotemporal features)

EventFormer: Quantitative Performance



EventFormer: Qualitative Evaluation

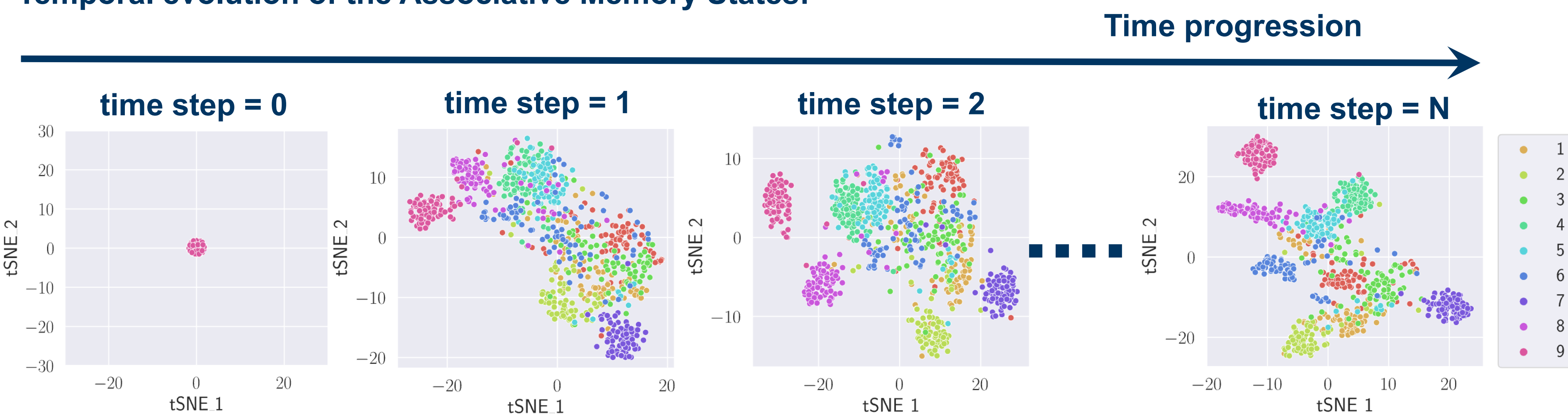


- We visualize the associative memory states for two different class samples from N-Caltech101 dataset.
- Although the memory states at same at the beginning (initial states), it quickly evolves into different patterns as we get more and more events.

*Only for visualization purpose

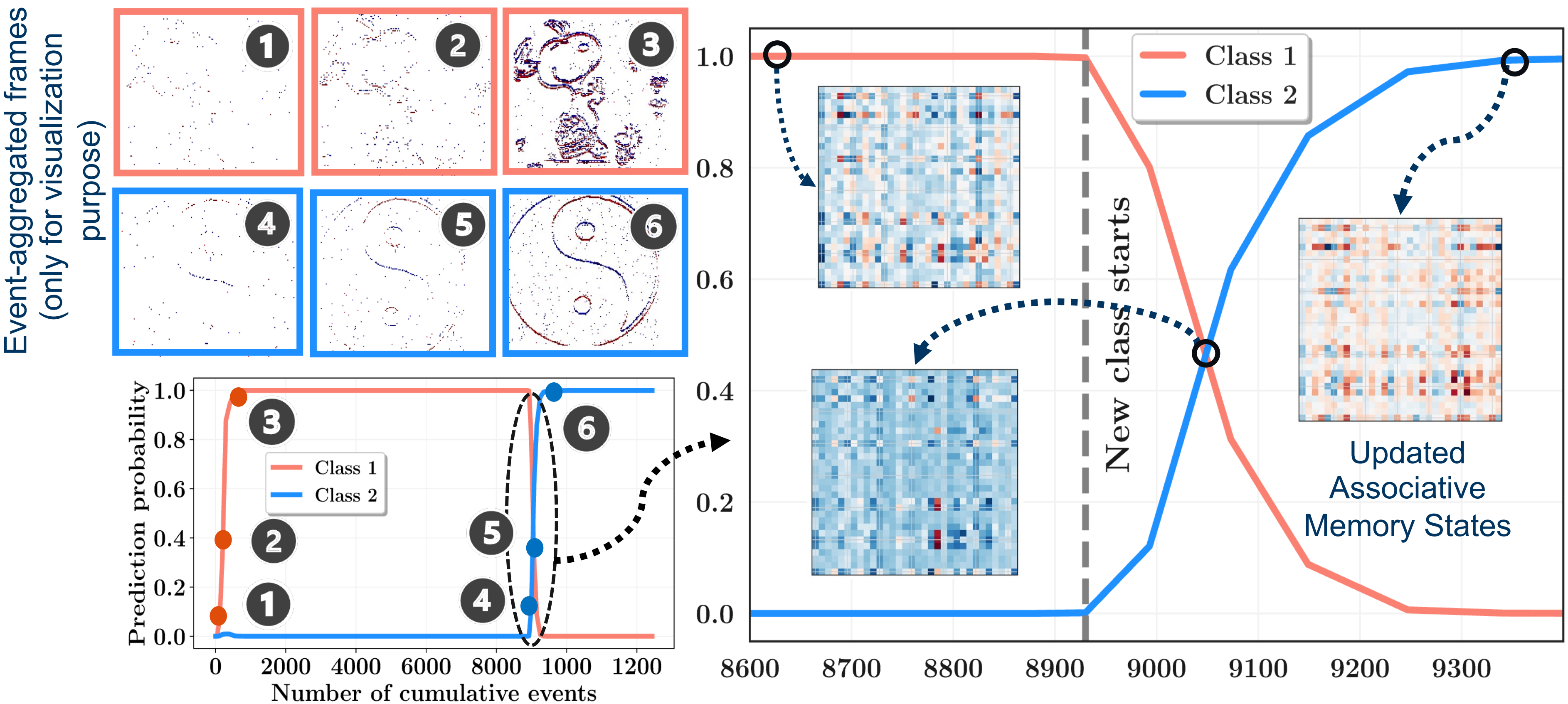
EventFormer: Qualitative Evaluation

Temporal evolution of the Associative Memory States:



- 2D tSNE visualization of the memory states (32x32) for 10-different class samples from N-Caltech101 dataset
- Memory states clusters towards the same location in the absence of any events (time step = 0) due to the same initial states for all the classes
- With the evolution of time, the associative memory states start to form more distinguishable clusters
- More sparable class boundaries shows how our memory representation capture more discriminative features as it process more events

EventFormer: Quantitative Performance



- EventFormer is capable of continuous class prediction thanks to its recurrent memory processing
- It takes ~1000 events only to reach reasonable class-confidence probability score
- Associative memory states update accordingly when it observes sample from difference class
- The memory learns to preserve useful information from its previous input, resulting in faster update

Summary

GIGASCALE RELIABLE ENERGY EFFICIENT NANOSYSTEMS LAB

We present EventFormer:

- A memory-augmented spatiotemporal representation learning framework for event-based perception
- Processes set-structured data and learns to perform spatio-temporal correlation in the latent memory space
- It achieves superior performance on the existing event-camera object classification benchmarks with massive computational efficiency gains

Summary

We present EventFormer:

- A memory-augmented spatiotemporal representation learning framework for event-based perception
- Processes set-structured data and learns to perform spatio-temporal correlation in the latent memory space
- It achieves superior performance on the existing event-camera object classification benchmarks with massive computational efficiency gains

Future works:

- Possible adaption of EventFormer on more challenging spatiotemporal tasks including event-based object detection, motion segmentation, optical flow estimation, etc.