

Masked Event Modeling: Self-Supervised Pretraining for Event Cameras

Simon Klenk^{1,2*} David Bonello^{1*} Lukas Koestler^{1,2*} Nikita Araslanov^{1,2} Daniel Cremers^{1,2}
¹ Technical University of Munich ² Munich Center for Machine Learning

{simon.klenk, david.bonello, lukas.koestler, nikita.araslanov, cremers}@tum.de

* Equal contribution

Abstract

Event cameras asynchronously capture brightness changes with low latency, high temporal resolution, and high dynamic range. However, annotation of event data is a costly and laborious process, which limits the use of deep learning methods for classification and other semantic tasks with the event modality. To reduce the dependency on labeled event data, we introduce Masked Event Modeling (MEM), a self-supervised framework for events. Our method pretrains a neural network on unlabeled events, which can originate from any event camera recording. Subsequently, the pretrained model is finetuned on a downstream task, leading to a consistent improvement of the task accuracy. For example, our method reaches state-of-the-art classification accuracy across three datasets, N-ImageNet, N-Cars, and N-Caltech101, increasing the top-1 accuracy of previous work by significant margins. When tested on real-world event data, MEM is even superior to supervised RGB-based pretraining. The models pretrained with MEM are also label-efficient and generalize well to the dense task of semantic image segmentation.

1. Introduction

Event cameras are promising imaging sensors for robotics and virtual reality (VR). They have the potential to enable applications previously inaccessible for frame-based (RGB) cameras, such as object classification in high-speed autonomous driving or eye and hand tracking in low-power VR systems. This is because event cameras track pixels independently, triggering them asynchronously only if their brightness change exceeds a threshold [17]. Such property allows them to have high temporal resolution, high dynamic range (HDR), low latency, and low power consumption.

Addressing high-level vision tasks from event data, such as object classification, is significantly more challenging than using RGB images. RGB-based approaches profited

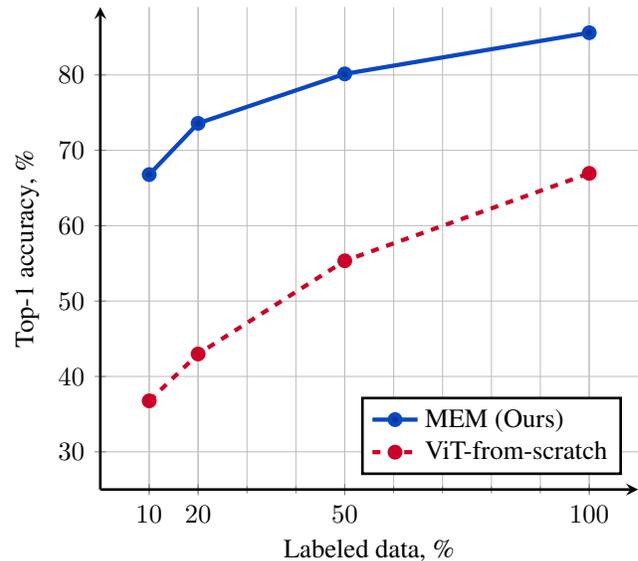


Figure 1. Classification accuracy on N-Caltech101 [41] with limited labeled data. Our self-supervised pretraining method MEM (blue) greatly improves the baseline (red) trained from random initialization. The benefits are increasingly pronounced with a scarce amount of labeled data.

significantly from large labeled training datasets [54]. Indeed, the remarkable progress in this field can be largely attributed to the abundance of labeled datasets. However, the event-based modality is temporal and lacks color and texture, which impedes the creation of large annotated event-based datasets. Despite the recently released event-based datasets like N-ImageNet [29] and N-EPIC-Kitchens [43], labeled event data is still in short supply [17, 38, 51].

One solution to counteract the dependency on large-scale labeled datasets is self-supervised learning (SSL). SSL techniques have already precipitated a significant advance in the natural language processing (NLP) and vision communities [3, 5, 8, 9, 12, 21, 24, 28]. SSL consists of two consecutive stages. First, a neural network is pre-trained without labels by solving a pretext task. For example, BERT [12] performs pretraining on a large corpus of

unlabeled text by predicting the words masked out in the input sequence. The second stage finetunes the network on a downstream task. Compared to a randomly initialized network, the pretraining is expected to improve network accuracy and label efficiency on the downstream task, as well as to reduce the number of iterations required for finetuning.

In contrast to previous work leveraging RGB images [8, 24], we aim at pretraining in the event domain. To this end, we develop Masked Event Modeling (MEM). MEM performs self-supervised pretraining on arbitrary event recordings to alleviate the need for labeled events at a large scale. Our approach is close to the recently proposed frame-based method BEIT [3] and is inspired by a multitude of proposed extensions to other data modalities [2, 56, 59, 64].

Fig. 1 illustrates the benefit of our pretraining. Using the parameter initialization provided by MEM, we consistently outperform all event-based baselines from the literature on object classification. We also show the benefit of MEM for semantic segmentation. In summary, our contributions are:

- We present a framework for self-supervised learning on the event modality using a simple event masking approach. During pretraining, our method does not require any labels or access to image data, which makes it applicable to any event recording.
- We set a new state-of-the-art top-1 accuracy for event-based object classification on all our datasets. On N-ImageNet, MEM surpasses the previous state-of-the-art by +7.96%. On N-Cars, it improves by +1.49% and attains up to +14% improvement on N-Caltech101.
- On the downstream task of semantic segmentation, MEM yields an improvement over training from scratch, especially on rare semantic categories.
- We empirically show that the common practice of transferring pretrained weights from RGB data to the event domain [22, 45] is not always optimal. For example, on the N-Cars dataset, where the data originates from real-world recordings, we demonstrate that MEM achieves better accuracy than a *supervised* RGB-based pretraining on ImageNet [11].

2. Related Work

Event-only self-supervised learning is a nascent research avenue. Therefore, we discuss related work on self-supervised frame-based (RGB) learning and review the methods that tackle the lack of annotated event data.

Self-supervised learning (SSL) The idea of SSL is to first train a network on an unlabeled dataset by solving a pretext task [28] and then to finetune the network on a

downstream task. The pretext task is defined such that the network learns feature representations highly relevant to the downstream task. After pretraining, the network is finetuned on a downstream task with a small labeled dataset in a supervised fashion. Examples of pretext tasks in vision are image reconstruction from masked or transformed input patches [1, 24], re-ordering of image patches [40], or predicting the degree of image rotations [20].

One notable SSL framework is BEIT [3], which was largely inspired by the recent success of BERT [12] in the NLP domain. Like BERT, the pretext task of BEIT is to mask out sections of an input image with the intention of reconstructing the original content. However, instead of directly predicting the pixels of the masked patches, BEIT predicts visual tokens, which encode semantic information of each patch in a single vector.

While BEIT and related methods [13, 16, 24, 57, 61, 67] enhance BERT-style pretraining with the focus on images, several extensions to different data modalities have been recently proposed. Point-BERT [64] extends the idea to point clouds; BEVT [59] and VIMPAC [56] apply the reconstruction task for videos; MultiMAE [2] jointly reconstructs RGB images with depth and semantic maps. While Barchid *et al.* [4] take initial steps toward adapting pretraining for events, the accuracy of their method remains limited.

Overcoming the lack of labeled event data Although SSL is not established for event cameras, numerous other works have proposed solutions against the lack of labeled events. For example, Rebecq *et al.* [46] show that it is possible to reconstruct grayscale frames from events, which can then be used with conventional frame-based networks. Conversely, events can be derived from existing frame-based datasets with simulation. The drawback of these approaches is the increased computational demand and reconstruction artifacts, which can alter the semantic meaning or neglect event-specific characteristics.

SSL has been used to solve low-level event vision tasks, like optical flow [68], intensity reconstruction using a generative event model [42], and, more recently, object classification [62]. However, these models either work under strong assumptions, such as the availability of temporally synchronized and pixel-aligned grayscale frames or require a substantial amount of RGB data. Semi-supervised approaches also exist [25, 65], but also rely on synchronous recordings of the events and frames.

Recently, a few models have emerged, which leverage unpaired datasets of labeled frames and unlabeled events [38, 55, 58]. These approaches transfer knowledge from a powerful network in the RGB domain to the event domain. However, these methods depend on labeled images and require datasets of both modalities to be captured under similar conditions, which is not feasible in many applications.

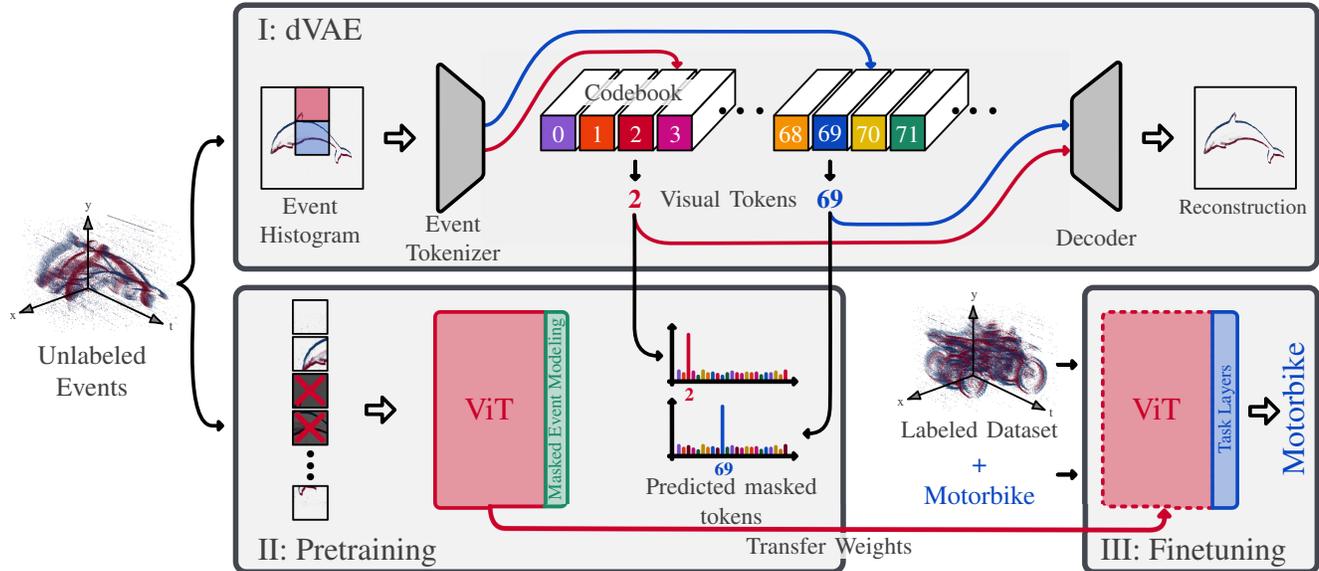


Figure 2. An overview of Masked Event Modeling (MEM). The proposed method is a three-stage pipeline. (I) In the first stage, we train a discrete variational autoencoder (dVAE) [44, 49] to compress the input event histograms to a list of discrete visual tokens. Each token – described by a fixed vector in the codebook – represents one input patch. The training objective in this stage is event histogram reconstruction. (II) In the second stage, we perform self-supervised pretraining of a ViT. The event histogram is divided into patches. We mask 50% of these patches, and the ViT predicts their corresponding (masked) visual tokens, similar to BERT [12] and BEIT [12]. The masked patches are replaced by a learnable embedding. Since the event tokenizer generates the targets for the ViT, no actual labels are needed. (III) In the final stage, we finetune the pretrained ViT on a downstream task. This is the only stage that requires labeled data.

In contrast to the above methods, we propose a general framework that performs SSL directly on the event data, requiring neither additional labels nor corresponding image data. In most applications, such unlabeled event data is readily available, whereas generating labels or accessing labeled image datasets from the same domain can be very costly or impractical.

3. Masked Event Modeling

We propose Masked Event Modeling (MEM), an adaptation of BERT-style pretraining to events [3]. Fig. 2 provides an overview. Leveraging the properties of event-based sensors, our method has a conceptual advantage over standard RGB-based models, such as in low-light conditions or applications requiring high temporal resolution. The events are preprocessed and passed through the MEM pipeline, consisting of three stages. In the first stage, we train a dVAE to represent the event data in a discrete latent space. The second pretraining stage employs a self-supervised proxy task of reconstructing partially removed event-based input. This reconstruction takes place in the latent space learned by the first stage. In the finetuning stage, we use the weights from pretraining as initialization and finetune the network on a target dataset with labels.

3.1. Event Processing

The raw event data is preprocessed to an event histogram before entering the MEM pipeline. Fig. 3 illustrates a few examples of event histograms. Since an event camera asynchronously reports brightness changes at a pixel, the sensor’s output is a stream of individual events. Each event includes a polarity that indicates an increase or decrease in the observed brightness. For a moving camera under constant illumination, such brightness changes tend to coincide with edges in the conventional image. To obtain an image-like data structure with visible edges, we accumulate the events separated by polarities into a two-channel image $\mathbf{H} \in \mathbb{R}^{H \times W \times 2}$ using up to N_{\max} events. We additionally perform data augmentation on the event histograms. In contrast to related RGB-based works [3, 24], we find that RandAugment [10] is necessary for the best performance with event data (*cf.* Tab. 5). Note that the input event histogram representation is dense, which enables simple storage, loading, preprocessing, and the use of deeper networks (graph neural networks on sparse representations tend to be relatively shallow [32, 50]).

3.2. Discrete Variational Autoencoder

To reconstruct the masked event histograms during the pretraining phase, we must first reduce the model’s output space. Directly predicting raw histogram values of input

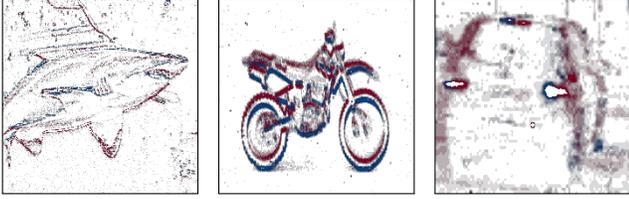


Figure 3. Left to right: Example histograms from N-ImageNet [29], N-Caltech101 [41], and N-Cars [51]. N-Cars is the only real event dataset. It features different event statistics and noise distribution than the two other semi-synthetic dataset (*cf.* Sec. 4).

patches would lead to a higher computational cost and can cause overfitting on low-level visual details [56]. Instead, we use visual tokens that summarize high-level semantic information in a single vector per patch. Additionally, the tokens are discrete, which allows for formulating the pre-training stage as a classification task.

Following Ramesh *et al.* [44], we employ a discrete variational autoencoder (dVAE) to learn the compact representation. The idea is to compress each input patch of size $P_x \times P_y$ to a codebook vector $\mathbf{z} \in \mathbb{R}^d$, which summarizes the visual features of the patch. Each codebook vector has a unique index, called a visual token, which later serves as the prediction target in the pretraining phase. Since the codebook is fixed during pretraining, the index of a visual token is sufficient to represent the semantic content of the patch.

The dVAE consists of three parts: the event tokenizer (encoder), the codebook bottleneck $\mathbf{Z} \in \mathbb{R}^{N \times d}$, and the decoder (*cf.* phase I in Fig. 2). The event tokenizer $q_\theta(\mathbf{z} | \mathbf{h})$ takes the full event histogram as the input. Each input patch $\mathbf{h} \in \mathbb{R}^{P_x \times P_y \times 2}$ is mapped to a latent codebook vector \mathbf{z} . The decoder $p_\Phi(\mathbf{h} | \mathbf{z})$ learns to recover the event patch given the visual tokens. We place a uniform prior on the token distribution. It can be shown that maximizing the marginal likelihood $p(\mathbf{h})$ corresponds to maximizing the evidence lower bound (ELBO) [31, 47]:

$$\mathcal{L}_{\text{ELBO}}(\theta, \Phi; \mathbf{h}) := \mathbb{E}_{q_\theta(\mathbf{z} | \mathbf{h})} [\log(p_\Phi(\mathbf{h} | \mathbf{z}))] - D_{\text{KL}}(q_\theta(\mathbf{z} | \mathbf{h}) \| p_\Phi(\mathbf{z})). \quad (1)$$

Due to the tokens being discrete, we employ the Gumbel-Softmax relaxation technique [27, 36] for obtaining the gradient, which was previously found useful in related tasks [3, 44, 64]. We visualize the decoded tokens in Fig. 5. We observe that the tokens capture visual primitives, such as lines and smooth contours. We also found that gradient clipping is required for stable training of the dVAE on event histograms, which is not required in the RGB setting [3, 24].

3.3. Pretraining

Our pretraining encourages the network – in our case, a ViT [14] – to embed semantic-level content of the event data

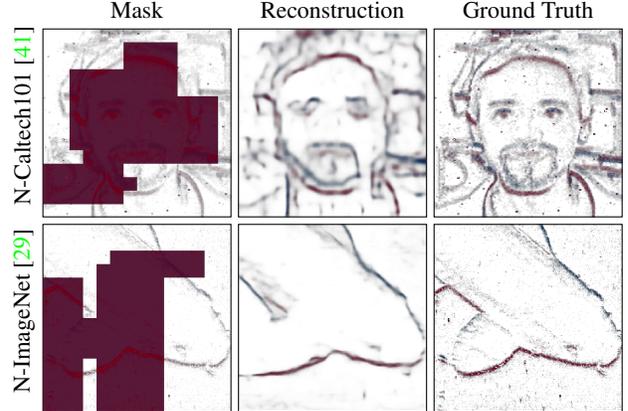


Figure 4. Left to right: Masked input histograms, reconstructions during pretraining, and the ground truth. The ground truth can be recovered despite large parts of the input being masked. The ViT predicts tokens for all patches, which are then decoded into the predicted event histogram by the decoder of the dVAE. Visualizations are rendered from the test set.

into the corresponding feature representations. First, we divide the event histogram into patches and replace 50%¹ of the patches with learnable mask embeddings, \mathbf{e} . The partially masked input is then used to train the pretraining network, which consists of the ViT and a linear projection (discarded after pretraining). We denote the set of masked patch indices as M . Using the remaining non-masked patches, $\mathbf{h}_{k \notin M}$, and learnable mask embeddings, $\mathbf{e}_{k \in M}$, MEM learns to predict the visual tokens $\mathbf{z}_{k \in M}$ of the masked patches. The training objective is

$$\max \mathbb{E}_{\mathbf{h} \sim \mathcal{D}, M \sim \mathcal{M}} [p_{\text{MEM}}(\mathbf{z}_{k \in M} | \mathbf{h}_{k \notin M}, \mathbf{e}_{k \in M})], \quad (2)$$

where \mathcal{D} is our data distribution; \mathcal{M} models random sampling of patch indices, and p_{MEM} is the posterior of visual tokens. Following this pretraining process, the ViT can reconstruct masked event histogram, as we illustrate in Fig. 4. To succeed on this reconstruction task, the ViT must learn semantically-aware feature representations embedding context information, as we confirm experimentally in Sec. 4.

3.4. Finetuning

In the last stage, we transfer the weights of the pretrained network to initialize the ViT for finetuning on a labeled dataset. A task-specific layer replaces the final linear output layer. The dVAE is no longer used. Since the ViT has learned to complete a partially masked event histogram during pretraining, we expect its weights to effectively process event data for high-level vision tasks. Note that our method requires labeled data only for finetuning.

¹We study alternative choices of the masking ratio in Sec. 4.4.

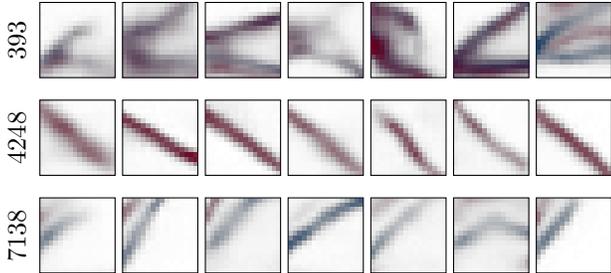


Figure 5. Examples of decoded codebook vectors with indices 393, 4248, and 7138 on a test set. Each codebook index corresponds to a specific visual feature, e.g. a wedge of positive and negative polarity (393), or a diagonal red line (4248). The codebook uses the polarity information. The codebook size is 8092.

4. Experiments

We evaluate the proposed Masked Event Modeling on object classification and semantic segmentation – the two high-level vision tasks with established benchmarks and well-defined metrics. Recall that event sensors bring advantages over RGB cameras, such as HDR, low power consumption, and high temporal resolution. However, since the events capture only changes in brightness, inferring semantic information from the events only is very challenging.

We report top-1 classification accuracy on N-ImageNet [29], N-Caltech101 [41], and N-Cars [51]. We compare MEM on each dataset to three baseline categories: (i) baselines from the literature; (ii) random weight initialization (ViT-from-scratch), and (iii) supervised networks pretrained on RGB data: ImageNet-1k (ViT-1k) and ImageNet-21k (ViT-21k) [11, 48]. MEM and our baseline methods – ViT-from-scratch, ViT-1k, and ViT-21k – use the same implementation, detailed in Sec. 12.

4.1. Object Classification

N-ImageNet N-ImageNet [29] contains 1.78 million event streams recorded by a DVS-Gen3 with resolution 480×640 [52]. The event camera is moved in front of an LCD monitor, which displays ImageNet-1k [11] images containing 1000 classes. It is the largest event-based object classification dataset. The second largest dataset, ASL-DVS [6], is substantially smaller and features only 24 classes. N-ImageNet is very challenging, as the best-published result so far has only achieved 48.94% top-1 accuracy, which is significantly below the 90% mark currently achieved on ImageNet-1k [63, 66]. Pretraining for the N-ImageNet dataset took about 4 days on 4 NVIDIA A100 GPUs and about 2 days on 2 NVIDIA A40 GPUs for the other datasets.

In Tab. 1, we show that our proposed Masked Event Modeling outperforms the baseline N-ImNet-EST [18] on N-ImageNet by +7.96%. By contrast, our baseline ViT-from-scratch cannot reach on-par performance with the state of the art. Presumably, this is because training a ViT on

Method	Pretraining		Top-1
	Aux. Data	Labels	
ViT-1k	ImNet-1k	✓	61.90
ViT-21k	ImNet-21k	✓	65.00
Yang et al. [62]	ImNet-21k [†]	✗	64.83
N-ImNet-Hist [37]	✗	✗	47.73
N-ImNet-DiST [29]	✗	✗	48.43
N-ImNet-EST [18]	✗	✗	<u>48.93</u>
ViT-from-scratch	✗	✗	43.13
MEM (ours)	✗	✗	57.89

Table 1. Top-1 classification accuracies on N-ImageNet-1k [29]. MEM (ours) outperforms all baseline methods which use only event-only. Supervised pretraining on RGB-ImageNet achieves an even higher accuracy. Baseline numbers are from [29]. [†] Yang et al. [62] also require paired RGB-event training data, and use a backbone pretrained on ImNet-21k with self-supervision.

ImageNet is challenging due to overfitting [53]. The results demonstrate that employing MEM pretraining is an effective way to boost performance on this complex event-based classification task.

The shaded top part of Tab. 1 shows that our supervised baselines ViT-1k and ViT-21k achieve higher accuracy than our self-supervised MEM. Note that the employed checkpoints for ViT-1k and ViT-21k perform *supervised* pretraining on RGB ImageNet-1k and ImageNet-21k for 300 epochs and require extensive hyperparameter search [53]. As a derivative of ImageNet-1k, N-ImageNet shares many similarities with the original dataset. Due to computational constraints, we only perform pretraining on N-ImageNet for 75 epochs. We believe longer training and hyperparameter optimization could further reduce the gap between RGB-based and event-only pretraining.

The work by Yang *et al.* [62] is a recent self-supervised pretraining scheme requiring temporally synchronized and aligned pairs of event and RGB frames. Their approach performs comparably to our ViT-21k baseline on N-ImageNet. However, their approach is more restrictive and requires a self-supervised pretrained backbone on ImageNet-21k for extracting a latent representation from the RGB frames.

N-Caltech101 Similar to N-ImageNet, N-Caltech101 [41] originates from an event camera moving in front of a monitor displaying images of the original RGB-based Caltech101 [15]. It contains 101 classes spread across 8246 samples, each with a timespan of 300 milliseconds.

In Tab. 2, we show that our method outperforms all baselines that have access only to the labeled events of N-Caltech101 [41] and improves over the previous state of the art by +9.5%. Remarkably, we can further improve this result by leveraging *unlabeled* event data from N-ImageNet

Method	Pretraining		Top-1
	Aux. Data	Labels	
HATS-Resnet [51]	ImNet-1k	✓	70.00
EST [18]	ImNet-1k	✓	81.70
DVS-ViT [60]	ImNet-21k	✓	83.00
E2VID [45]	ImNet-1k	✓	86.60
EventDrop [22]	ImNet-1k	✓	87.14
ACE-BET [33]	ImNet-1k	✓	89.95
ViT-1k	ImNet-1k	✓	92.06
ViT-21k	ImNet-21k	✓	91.10
HATS [51]	✗	✗	64.20
AEGNN [50]	✗	✗	66.80
SSRL [4]	✗	✗	72.79
AsynNet [39]	✗	✗	74.50
EvS-S [32]	✗	✗	76.10
ViT-from-scratch	✗	✗	66.94
MEM (ours)	✗	✗	85.60
MEM-NImNet (ours)	NImNet-1k	✗	90.10

Table 2. Top-1 classification accuracy on N-Caltech101 [41]. MEM (ours) outperforms all baseline methods, which only have access to event information. However, MEM (ours) does not fully close the gap between event-only and RGB-based supervised methods. However, using MEM pretraining with N-ImageNet-1k (MEM-NImNet) outperforms them as well.

(instead of N-Caltech101) in the pretraining stage, which yields top-1 accuracy of 90.1% (MEM-NImNet). This nearly closes the gap to fully supervised methods pretrained on the RGB data (*cf.* shaded rows in Tab. 2). MEM-NImNet stems from pretraining MEM on the N-ImageNet [29] dataset for 75 epochs and finetuning MEM using the labeled event data from the N-Caltech101 training set. The fact that we can leverage event data from a different dataset suggests a strong cross-domain generalization of MEM. Note that MEM-NImNet requires labels only from N-Caltech101 and relies solely on the event modality.

MEM-NImNet is only marginally outperformed by our supervised baseline methods, ViT-1k and ViT-21k, which perform slightly better than other methods supervised on the RGB-based ImageNet. This could be explained by our use of advanced data augmentation in the supervised baselines, such as RandAugment [10] (playing an important role, see Sec. 4.4), a cosine learning rate scheduler, and regularization, inspired by state-of-the-art image-based ViT training practice [37, 53], see Sec. 12 for further details.

N-Cars We also benchmark MEM on the non-synthetic N-Cars dataset [51], which captures the real-world event dynamics. It was recorded by an event camera mounted behind the windshield of a car in an urban environment. The N-Cars dataset contains 12,336 samples of the class "Car"

Method	Pretraining		Top-1
	Aux. Data	Labels	
HATS-Resnet [51]	ImNet-1k	✓	90.40
E2VID [45]	ImNet-1k	✓	91.00
EST [18]	ImNet-1k	✓	92.50
EventDrop [22]	ImNet-1k	✓	95.50
ACE-BET [33]	ImNet-1k	✓	97.06
ViT-1k	ImNet-1k	✓	98.00
ViT-21k	ImNet-21k	✓	96.24
SSRL [4]	ASL-DVS [7]	✗	<u>95.64</u>
HATS [51]	✗	✗	90.20
AsynNet [39]	✗	✗	94.40
EvS-S [32]	✗	✗	93.10
AEGNN [50]	✗	✗	94.50
ViT-from-scratch	✗	✗	92.71
MEM (ours)	✗	✗	98.55
MEM-NImNet (ours)	NImNet-1k	✗	93.27

Table 3. Top-1 classification accuracy on N-Cars [51]. MEM (ours) achieves a new state-of-the-art on this benchmark compared to all baselines. Since N-Cars contains real event data and only two classes, pretraining on ImageNet [11] or on N-ImageNet [29] does not perform as well as for this dataset. Our method requires much less data and compute since it only uses events from N-Cars.

and 11,693 samples of the class "Background", where each sample is 100 milliseconds long. One fundamental difference between N-Cars and the other datasets is its different spatio-temporal event distributions since the events now originate from a dynamic 3D scene. By contrast, the event data in N-ImageNet and N-Caltech101 comes from a homography w.r.t. a flat LCD monitor under constant light [23]. In contrast, events in N-Cars are also triggered due to brightness changes in the physical world, such as those caused by the blinking of a car's rear lights and traffic lights. As visualized in Fig. 3, such fluctuations are distinctive marks of the real-world capturing conditions of N-Cars.

In Tab. 3, we show that MEM pretrained only on N-Cars outperforms all presented baselines, including the methods with access to RGB-based pretraining on ImageNet-1k and ImageNet-21k. Compared to ACE-BET [33], we raise the state-of-the-art from 97.06% to 98.55% top-1 accuracy, which halves the classification error. While supervised RGB-based pretraining on ImageNet does boost performance on semi-synthetic N-ImageNet (*cf.* Tab. 1) and N-Caltech101 (*cf.* Tab. 2), this effect is much weaker on the real-world event data in N-Cars, as can be seen by the similar accuracy of the upper and lower part of Tab. 3. We hypothesize the reason to be the *domain gap*, *i.e.* the difference in spatio-temporal event statistics of N-Cars compared to the semi-artificial N-ImageNet and N-Caltech101 datasets. This hypothesis can be further sup-

ported by our experiment “MEM-NImNet” in the last row of Tab. 3, where instead of N-Cars, we use N-ImageNet-1k for pretraining. MEM-NImNet on N-Cars does not improve significantly over ViT-from-scratch, even though it can access much more event data during pretraining. Recall that, by contrast, pretraining on N-ImageNet – MEM-NImNet in Tab. 2 – achieves a remarkable finetuning performance on N-Caltech101. This discrepancy strongly suggests that RGB-based pretraining may be limited when the ultimate domain of operation is the event data. Since MEM operates directly on the event modality, there is no domain gap w.r.t. the test-time setting, which explains its advantage over RGB-based approaches.

Overall, the experiments on N-Cars suggest that the common practice in the community of simply transferring RGB-pretrained weights to the event domain is not always the best option. This is especially true if the target task exhibits a specific event distribution distinct from the RGB data used during pretraining.

4.2. Semantic Segmentation

As a second downstream task, we evaluate MEM on semantic segmentation – a high-level vision task of assigning a class label to each image pixel. We use the DSEC-Semantic [19] dataset. It contains 10891 event streams annotated with semantic masks. When providing the event histograms to ViT as the input, we resize the original spatial resolution of 640×480 to 512×512 to preserve the aspect ratio used by the ViT during pretraining. Tab. 4 reports quantitative results. They strongly support our conclusions from the image classification task: the models pretrained with MEM provide a clear advantage over random initialization. From qualitative results, exemplified by Fig. 6, we further observe that MEM pretraining improves segmentation accuracy of less dominant semantic categories, such as pedestrians, which can be critical in practice. Sec. 6 provides further examples.

4.3. Masked Event Modeling with Few Labels

To demonstrate the benefits of our method in a few-shot scenario, we train MEM with increasingly smaller subsets of the dataset. We split the original train set into mutually exclusive smaller subsets: 100%, 50%, 20%, and 10%.

MEM is pretrained on the full N-Caltech101 [41] dataset *without labels*. Afterward, we finetuned the pretrained model on the smaller subsets. MEM is compared with the ViT-from-scratch baseline. Both models have access to the same data and labels during finetuning. In Fig. 1, we show that MEM consistently outperforms the baseline ViT-from-scratch. The benefit of MEM becomes increasingly pronounced as the amount of labeled data is reduced. For example, using only 10% of labeled data – 650 samples over 101 classes – MEM achieves 66.78% top-1 accuracy,

Ablation Setup	N-Caltech101	N-Cars
Full method	85.60	98.55
8x8 patches	80.81	96.13
32x32 patches	78.58	97.14
25% mask ratio	81.16	98.47
75% mask ratio	82.31	97.55
33% pretrain steps	81.17	95.16
No randAug [10]	80.67	98.14

Table 5. Ablation study of our method on N-Caltech101 [41] and N-Cars [51]. Default values for the full method are: patch size 16, masking ratio 50% and using RandAugment [10]. Event histograms are 224×224 .

whereas ViT-from-scratch reaches only 36.67%.

4.4. Ablation Study MEM components

We study our method by modifying individual components and report the resulting top-1 accuracy on N-Cars [51] and N-Caltech101 [41]. All other hyperparameters are fixed. In the first row of Tab. 5, we show the base accuracy of MEM on N-Cars or N-Caltech101. We ablate the patch size by changing it to 8×8 and 32×32 , respectively. The patch size influences the number of visual tokens and the number of patches. We find that the patch size of 16×16 yields the overall best result. We suspect that if the patch size is small, the information stored inside a patch is not enough to be semantically meaningful. If the patch size is too large, too many concepts must be represented by one patch, negatively influencing the pretraining result.

We next experiment with the masking ratio. Our default choice of 50% is superior to 25% or 75%. By comparison, 40% is the default in BEIT [3]; MAE [24] masks out 75% of the input patches. We suspect that a small masking ratio makes the pretraining task too easy, and the ViT tends to learn only a few semantic concepts as a result. Conversely, a high masking ratio makes it hard for the learning to extract semantically relevant features.

We test a shorter schedule of MEM pretraining by only performing 33% of the pretraining steps. The significant drop in classification accuracy on both datasets suggests that a longer pretraining is crucial.

Disabling RandAugment [10] results in a performance drop. Our empirical observations on N-ImageNet confirm that RandAugment is effective for event histograms, too. Note that RandAugment is not used in related RGB methods [3, 24] (*cf.* Sec. 3.1). This finding and the high downstream accuracy of our baselines ViT-1k and ViT-21k show that modern augmentation techniques are highly beneficial for event data as well and tend to outperform almost all event-specific methods from the literature (*cf.* Tabs. 1 to 3).

Method	aAcc	mAcc	mIoU
ViT-from-scratch	85.07	49.90	42.79
MEM (ours)	86.62	51.39	44.62

Table 4. Finetuning on DSEC-Semantic [19] validation: mean IoU (mIoU), all pixel accuracy (aAcc), mean accuracy of each class (mAcc), after 6k steps with batch size 32.

Method	N-Caltech	N-Cars
(i) MEM (ours)	85.60	98.55
(ii) eMAE-naive-only-mask	58.11	95.70
(iii) eMAE-entire-hist	80.61	97.50

Table 6. Top-1 accuracy. (i) MEM, (ii) naive eMAE with loss applied only on masked patches (as proposed by MAE [24] paper), (iii) eMAE-entire-hist loss on the entire histogram (our proposed modification for event-MAE).

4.5. Ablation of Pretraining Task

To validate the choice of our pretraining objective, we also investigate the alternative formulation based on the Masked Autoencoder (MAE) [24]. MAE directly reconstructs the masked input instead of indirectly through visual tokens, as we do. Tab. 6 reports the results. We found our MEM reconstruction and the top-1 accuracy to be superior. In order to adapt MAE to events, dubbed as “eMAE”, we found that the MAE loss needs to be applied to the entire histogram (eMAE-entire-hist) and not just on the masked patches (eMAE-naive-only-mask) as proposed in the original MAE formulation. Nevertheless, it is an interesting insight that MAE can be applied to events with our modification. Since MAE does not rely on the visual tokens produced by dVAE, the overall framework becomes simpler and hence deserves further investigation.

4.6. Ablation of Input Representation

Event histograms are our input representation of choice due to their simplicity and practical considerations, such as easy storage and preprocessing. To justify this design choice, we tested an 8-time-slice input, similar to voxel grids [45]. We did not observe any significant difference in terms of the top-1 accuracy of object classification compared to our representation, as detailed in Sec. 10. Furthermore, our preliminary experiments did not suggest any benefit for object classification if the event histograms encode the event timestamps in a third input channel. A 2-channel event histogram appears to be sufficient.

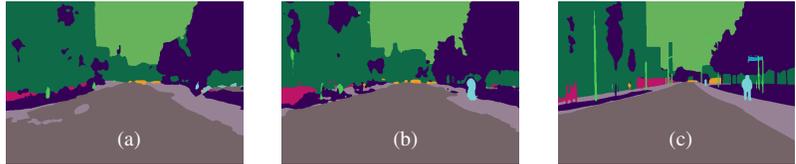


Figure 6. Semantic segmentation on DSEC-Semantic [19]. (a) ViT-from-scratch fails to detect the pedestrian and produces some artifacts on the road. (b) Our MEM rectifies both issues. (c) Ground truth. See Sec. 6 for more examples.

5. Conclusion

In practice, large amounts of unlabeled events from the target domain are easily accessible, whereas labeling event data is a laborious and costly process. In fact, labeling events is even more difficult than images because of their temporal nature and spatial sparsity (*cf.* Sec. 4.1). We introduced Masked Event Modeling, MEM, which brings self-supervised learning to event data. MEM consistently improves the state-of-the-art top-1 classification accuracy across all benchmarks we have tested: by +7.96% on N-ImageNet, by +1.49% on N-Cars, and by up to +14% N-Caltech101. Furthermore, MEM can also be employed for semantic segmentation with clear accuracy benefits.

Our work also revealed that the common practice of applying an RGB-pretrained network to an event vision task is not always optimal due to the domain gap, *i.e.* when the finetuning dataset has different characteristics from the training data. MEM can address this problem by pretraining on the same type of data modality as the target dataset.

Limitations While the computational expense of SSL may be amortized over a number of downstream tasks, SSL pretraining generally remains computationally demanding. Furthermore, despite empirical justification of our input representation, our current framework does not fully leverage the sparsity and high temporal resolution of event cameras. We have also shown promising results only on two downstream tasks, object classification, and semantic segmentation, but other downstream tasks, such as object detection, may be an interesting testbed as well. In future work, we plan to explore pretraining formulations for dynamic tasks, such as optical flow estimation. We will consider using event-specific network architectures, such as asynchronous, sparse (graph) neural networks. It may also be worthwhile to explore pretraining on neuromorphic hardware, where network training is still challenging.

Acknowledgements This work was supported by the ERC Advanced Grant SIMULACRON.

Masked Event Modeling: Self-Supervised Pretraining for Event Cameras

– Supplementary Material –

Simon Klenk^{1,2,*}, David Bonello^{1*}, Lukas Koestler^{1,2*}, Nikita Araslanov^{1,2}, Daniel Cremers^{1,2}

¹ Technical University of Munich ² Munich Center for Machine Learning

{simon.klenk, david.bonello, lukas.koestler, nikita.araslanov, cremers}@tum.de

* Equal contribution

6. Semantic Segmentation: More Examples

In Figs. 7 to 9, we show additional qualitative examples for the downstream task of semantic segmentation. As discussed in Sec. 4.2 of the main paper, MEM-pretrained models tend to exhibit improved segmentation of fine-grained scene structures, such as pedestrians and lamp poles.

7. Convergence Curve

In Fig. 10, we show that the convergence speed of finetuning is considerably higher for MEM than from ViT-from-scratch. Recall also that the top-1 accuracy of the finetuned model is substantially higher than that of the ViT-from-scratch baseline, *e.g.* +18.66% on N-Caltech101 in Tab. 2.

8. Self-Supervised Baselines

We experiment with another baseline by reconstructing frames from events using E2VID [45] to further verify our design choices. We analyze the finetuning of both MAE and MEM on top of E2VID reconstructions in Tab. 7. Both methods are strong baselines, but directly using MEM on the raw event histograms is more effective. Moreover, the problem with a two-stage E2VID pipeline is that it uses significantly more compute and storage. Another serious disadvantage is that the classifier uses *reconstructed* images – a lossy transformation of the raw event data, which limits the accuracy and requires domain-specific labeled training data. Analogous to Tab. 6 in the main paper, we investigate two versions of MAE for completeness: E2VID+MAE-entire-hist, where the MAE loss is applied to the entire reconstructed histogram (our proposed modification for event histograms); and E2VID+MAE-only-mask, where the MAE loss is only applied to the masked patches (*cf.* [24]). While our modified version yields a significant improvement when pretraining on raw event histograms (*cf.* Tab. 6), it seems to be slightly beneficial to adopt the original formulation for reconstructed frames.

9. Linear probing (LP)

We also evaluate linear probing results with MEM pretraining. The conclusions from the main text still hold for LP: MEM-pretrained models achieve higher top-1 accuracy

Method	N-Caltech101		N-Cars	
	FT	LP	FT	LP
E2VID+MEM	76.86	56.53	94.53	90.88
E2VID+MAE-entire-hist	77.45	58.70	93.09	88.33
E2VID+MAE-only-mask	78.56	60.22	94.39	89.51
MEM (ours)	85.60	71.20	98.55	97.58

Table 7. E2VID baselines. Top-1 accuracy for finetuning (FT) and linear probing (LP) on N-Caltech101 and N-Cars.

Method	N-Caltech101	N-Cars
MEM + LP	71.20	97.58
MEM-NImNet + LP	68.19	89.82
Random Init + LP	25.34	75.94
ViT-from-scratch	66.94	92.71

Table 8. Top-1 accuracy of linear probing (LP) for MEM pretraining on N-Caltech101 and N-Cars. Random Init + LP is the LP accuracy of a randomly initialized ViT.

than ViT-from-scratch. We report top-1 LP accuracy on N-Caltech101 and N-Cars in Tab. 8. Compared to finetuning, LP benefits more from longer pretraining (*cf.* [24, Fig. 7]). We trained our model for 75 epochs on N-ImageNet, which is sufficient for standard finetuning, but proves to be too short for LP. Additionally, LP requires careful hyperparameter tuning (*cf.* [24, A.1 and Tab. 10]). While we used here the same hyperparameters for MEM as in finetuning, we could substantially improve upon the initial LP results by using new hyperparameters and removing regularization, following the insights of MAEs [24].

10. Ablation of Input Representation

Event histograms already encode temporal information implicitly since they consider up to N_{\max} events per histogram and accumulate the polarities into separate channels. As discussed in Sec. 4.6, we did not observe a significant benefit by explicitly including temporal information in the input, such as the event timestamps. In fact, as Tab. 9 shows, including this information into the event histograms as a third channel leads to slightly worse top-1 classification

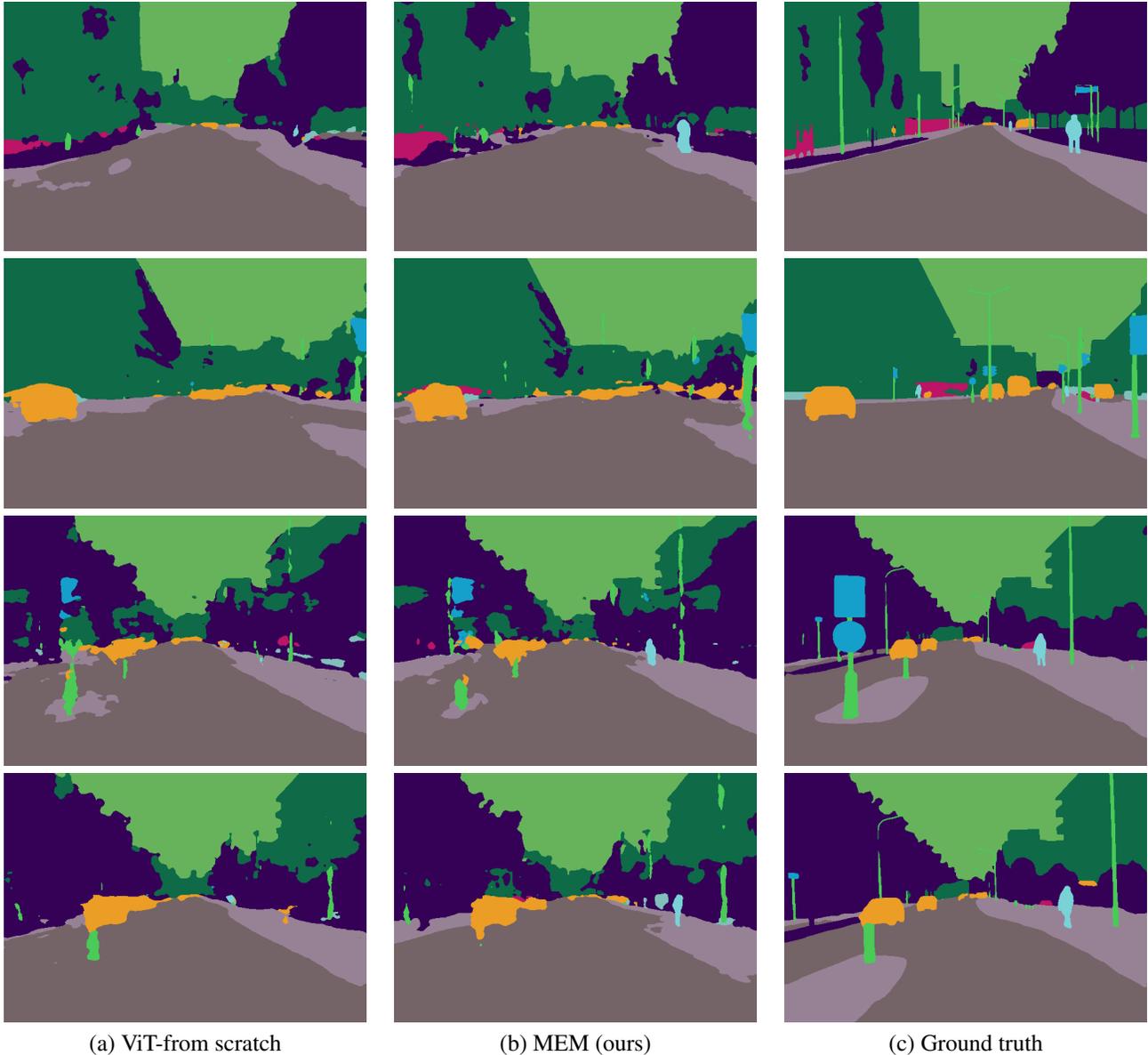


Figure 7. Semantic segmentation examples: (a) ViT-from-scratch, (b) MEM (ours), (c) the ground truth. MEM recovers the pedestrians (the right image half, 1st, 3rd, and 4th row), as well as the lamp pole (the right image half, 2nd, 3rd, and 4th row) more reliably.

accuracy – compare lines (i) and (ii).

To further investigate the importance of the temporal dimension for classification, we employ an 8-channel input histogram as the input. In contrast to our default 2-channel histograms, which collapse all recent events into two channels, this 8-channel histogram distributes the stream of N_{\max} events into four equally spaced chunks of time and computes the histogram per chunk (similar to voxel grids [45]). Compared to simply using the latest timestamp in the third channel, this representation encodes the temporal information in more fine-grained manner. However, it does not yield a consistent advantage over our 2-channel baseline, as

shown in lines (iii) and (iv) of Tab. 5. While the 8-channel representation somewhat improves the top-1 accuracy on N-Cars, the result is considerably worse on N-Caltech101.

The marginal benefit of the more explicit temporal encoding, as demonstrated in these experiments, has an intuitive explanation if we consider the underlying task – object classification. Semantic meaning is easily accessible from the *spatial* context, such as object shape, rather than the temporal distribution of brightness changes, which the more explicit temporal representations provide.

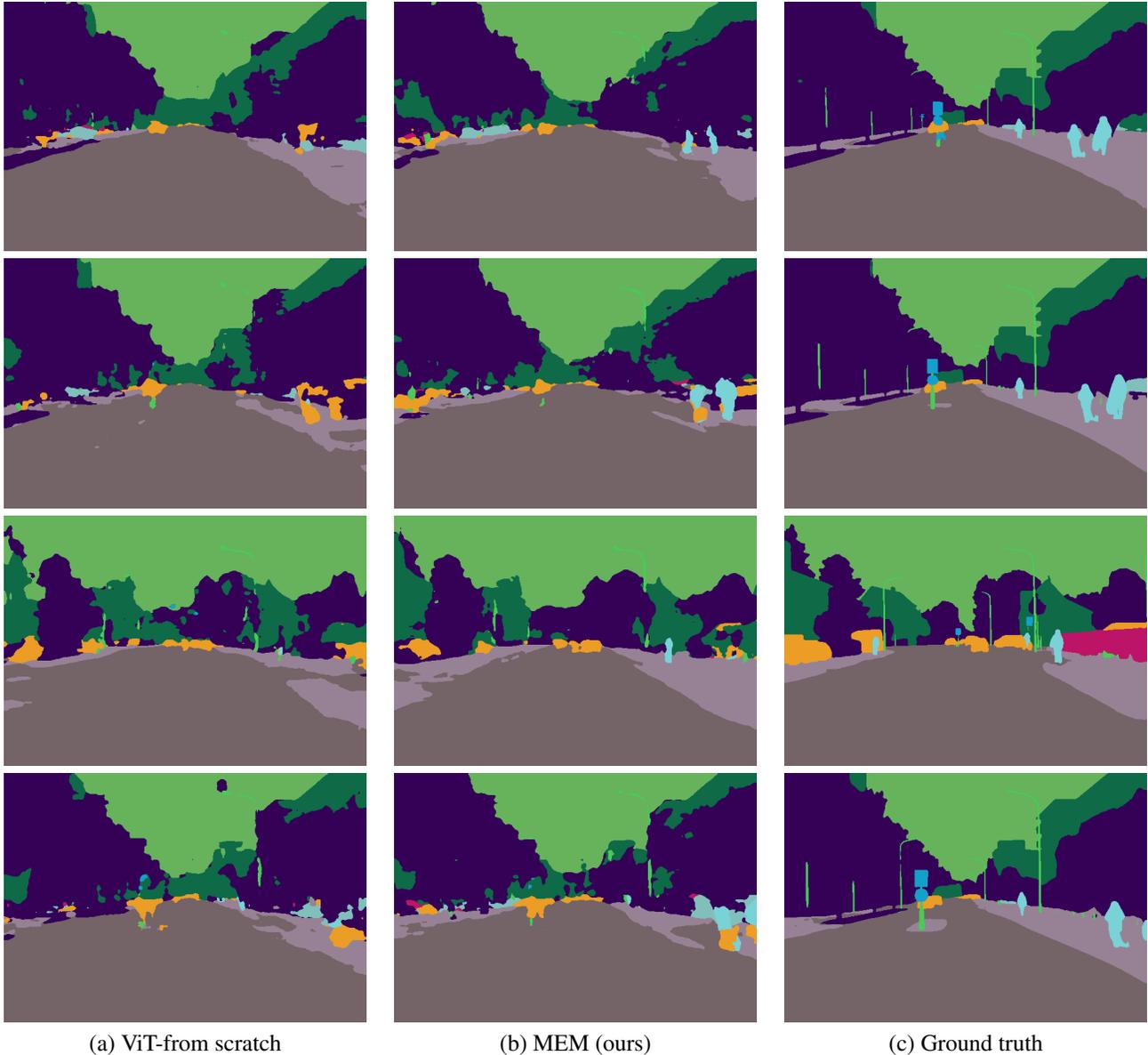


Figure 8. Semantic segmentation examples: (a) ViT-from-scratch, (b) MEM (ours), (c) the ground truth. MEM recovers the pedestrians (the right image half, 1st - 4th row), as well as the lamp pole (the right image half, 2nd and 4th row) more reliably.

11. Reconstruction and Token Visualization

We visualize additional codebook vectors in Fig. 11. We observe that codebook vectors, which have a fixed index, tend to exhibit recurring shape characteristics and a consistent preference for polarity (*e.g.*, either positive or negative or an equal share of both). The most common codebook vectors are completely blank since the event histogram is sparsely populated with event count values. Due to redundancy, we do not visualize these blank codebook vectors. Although all codebook vectors are fixed, the decoder adapts each patch to its surroundings to form a coherent im-

age. Hence, the visualized examples of each decoded codebook vector show some appearance variation (column-wise in Fig. 11). All visualized codebook vectors are rendered using the test set of N-Cars [51].

Complementing Fig. 4, Fig. 12 illustrates the reconstruction of masked patches during pretraining on all datasets used in this work. As we discuss in Sec. 4.5, the MAE pre-training task can also be employed on event histograms (in contrast to the original MAE paper [24]). However, it requires the loss to be formulated on the entire histogram. We found the reconstructions from eMAE, which are visualized in Fig. 13, to be not as sharp as for MEM.

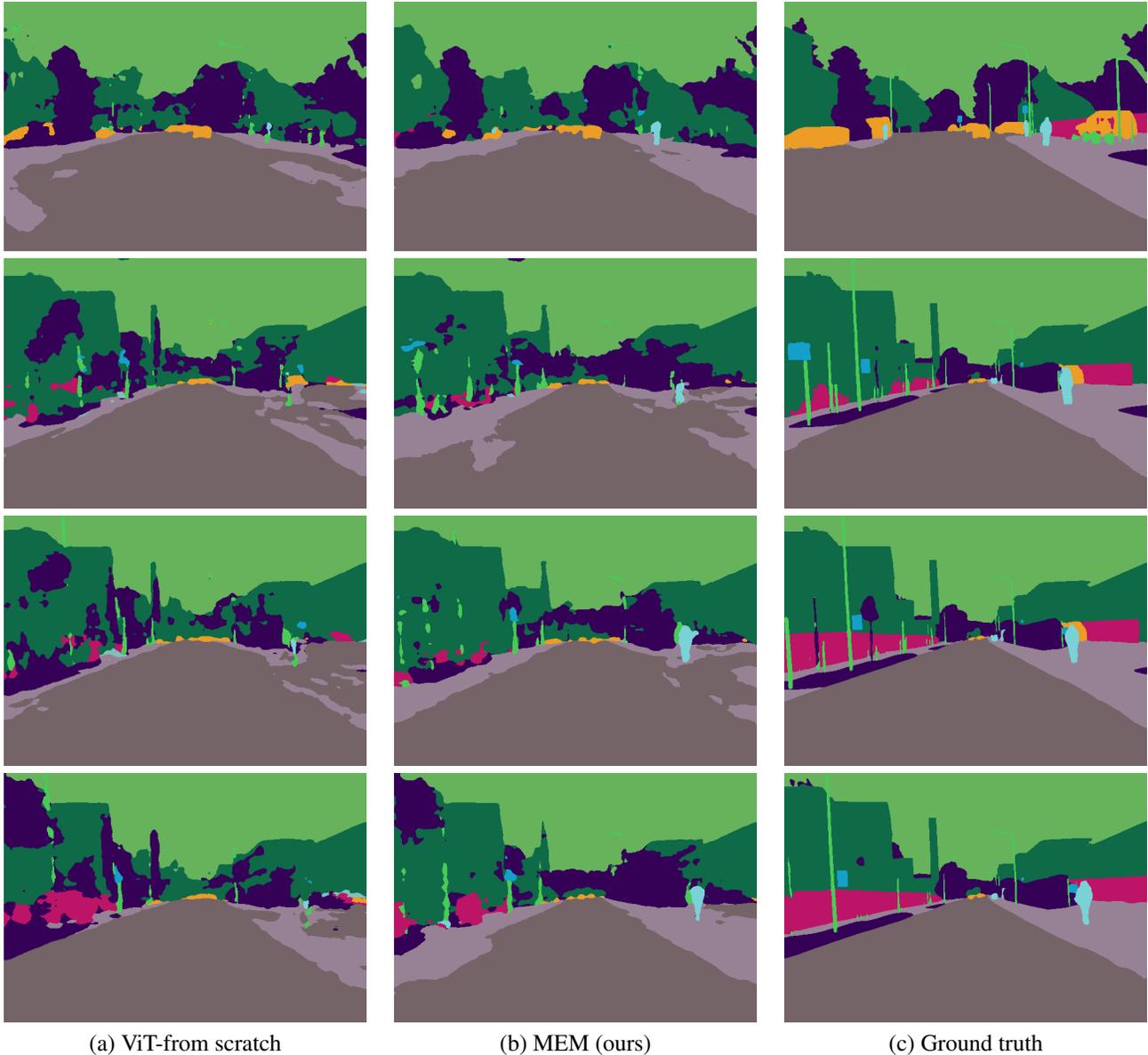


Figure 9. Semantic segmentation examples: (a) ViT-from-scratch, (b) MEM (ours), (c) ground-truth annotation map. MEM pretraining recovers the pedestrians (the right image half, 1st - 4th row), as well as the lamp pole (the left image half, 2nd - 4th row) more reliably.

12. Implementation Details

As discussed in Sec. 4, our MEM pretraining, as well as the baselines ViT-from-scratch, ViT-1k, and ViT-21k, share the same implementation. As detailed next, we make a significant effort to ensure strong baseline performance by using best-practice training techniques.

ViT Architecture We use ViT-Base described in [14] with patch size 16×16 . It consists of 12 layers and has 12 heads in the self-attention layers. The feature size is 768, maintained by MLPs with 3072 hidden units. We add a lin-

ear projection on the ViT features during pretraining, which outputs the visual tokens. We discard this linear projection during finetuning and train a new linear layer for classification. We employ relative positional encoding [14].

Hyperparameters We report hyperparameters for dVAE in Tab. 11, pretraining in Tab. 12, and finetuning in Tab. 13. As discussed in Sec. 3.2, gradient clipping is a vital hyperparameter. In Tab. 10, it can be seen that without gradient clipping, the accuracy on N-Caltech101 and N-Cars is worse than the baseline ViT-from-scratch. Hence, gradient

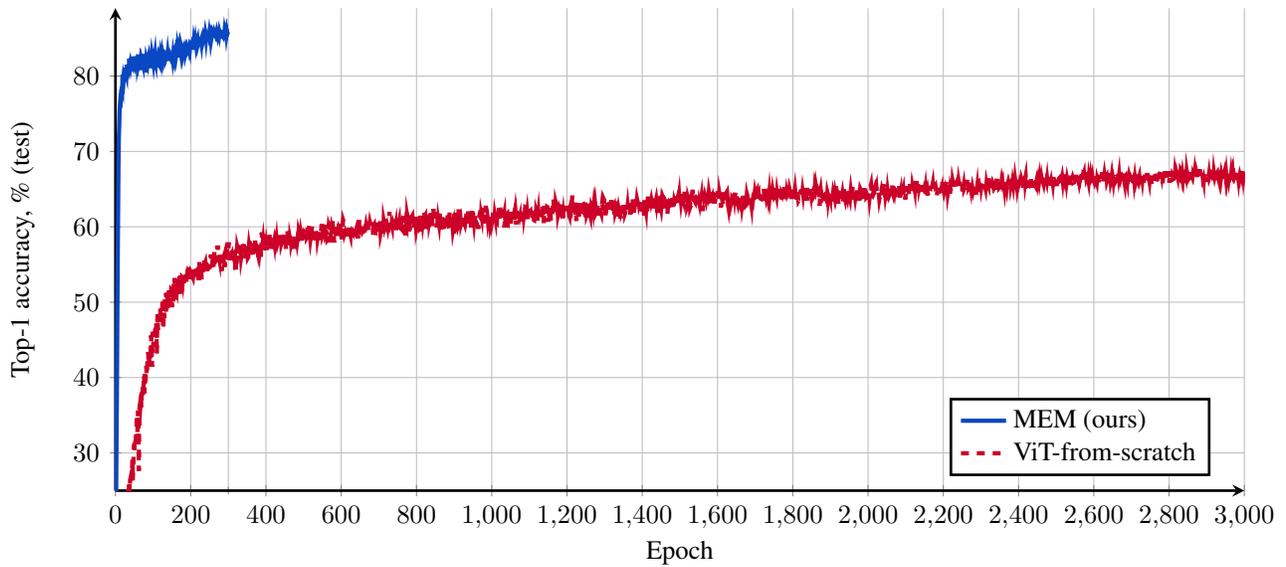


Figure 10. Finetuning accuracy vs. epochs on N-Caltech101 [51]. With our proposed pretraining (MEM), the accuracy increases much faster. It reaches a higher final accuracy of 85.60% compared to finetuning without pretraining (ViT-from-scratch), where the final accuracy is only 66.94%. Both the pretraining and the finetuning tasks use the entire N-Caltech101 train dataset.



Figure 11. Additional examples of decoded codebook vectors with the codebook indexes of 490, 804, 1041, 1964, 3128, 4228, 5910, and 8058. Although all codebook vectors are fixed, the decoder adapts each patch to its surroundings to form a coherent image. Observe that each codebook index corresponds to a specific visual feature, *e.g.*, a red horizontal line at the bottom of the patch (4228) or a round mixture of red and blue polarity (1964). The codebook size is 8092. These visualizations are rendered from the test set of N-Cars [51].

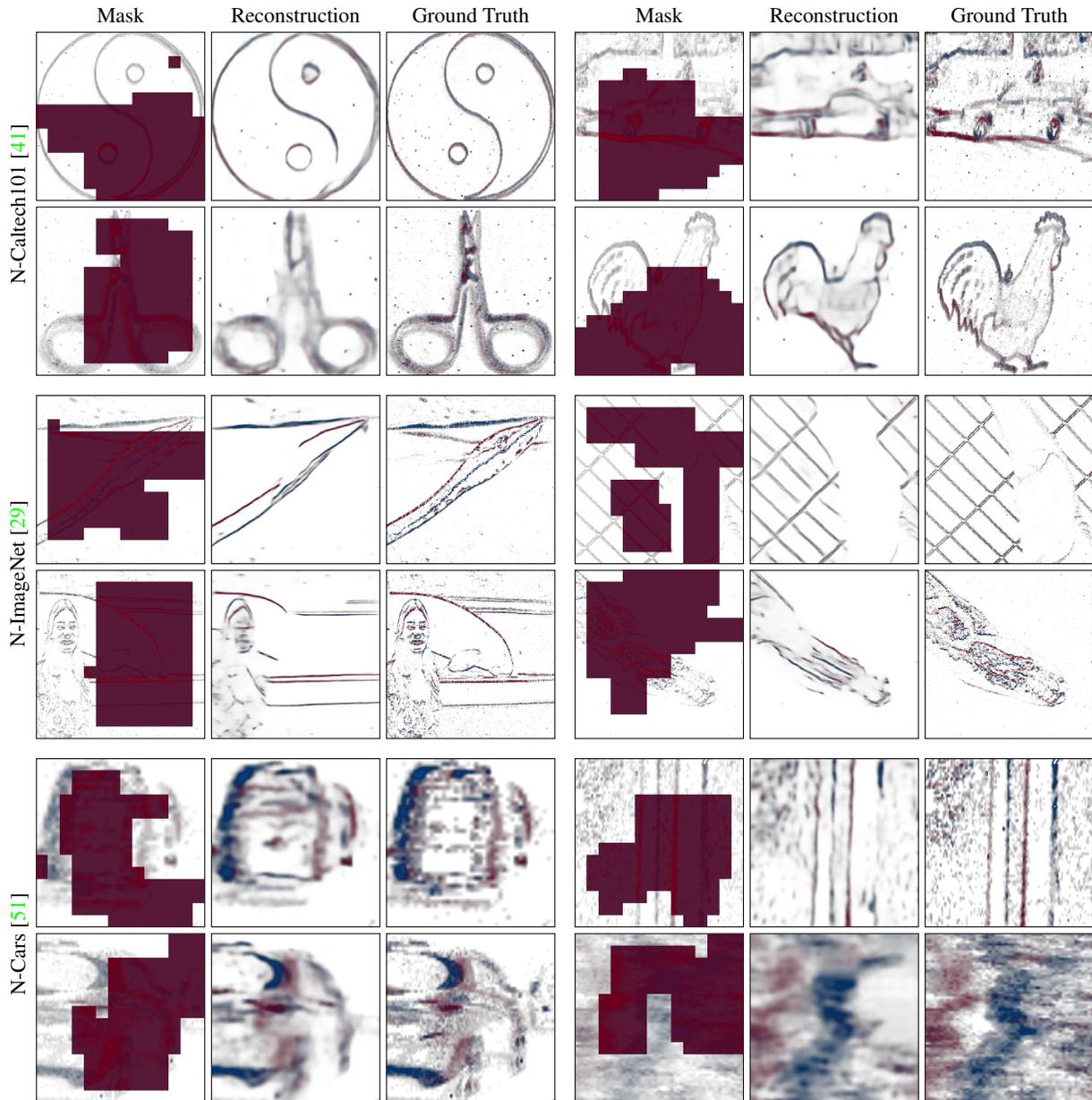


Figure 12. Additional masked patch predictions. From left to right: We visualize the masked input histograms, the reconstructions during pretraining, and the ground truth for N-Caltech101 [41] (top), for N-ImageNet [29] (middle); and for N-Cars [51] (bottom). We observe that the ground truth can be recovered even if large parts of the input to the ViT are masked. The ViT predicts the tokens for all masked patches, which are then decoded into the predicted event histogram by the decoder of the dVAE. These visualizations are rendered from the test set.

clipping is essential to employ MEM pretraining on sparse event histograms successfully. Note that gradient clipping is not used in the RGB setting [3, 24].

12.1. Details on Event Preprocessing

After loading all events for a given sample (*e.g.*, accumulated over 300ms in N-Caltech101), we slice the events in time by randomly selecting one contiguous batch com-

prising up to $N_{\max} = 30,000$ events. During training, we perform (i) a random flip of all event polarities (with probability $p = 0.5$); (ii) a random horizontal flip (with probability $p = 0.5$); and (iii) random shifts (per event) by Δx and Δy of x -coordinates and y -coordinates using uniform sampling, *i.e.* $\Delta x \sim \mathcal{U}(-15, 15)$ and $\Delta y \sim \mathcal{U}(-15, 15)$.

We accumulate the augmented events into a two-channel histogram. For N-Caltech101 and N-Cars, we resize the his-

Ablation	N-Caltech101	N-Cars
(i) MEM	85.60	98.55
(ii) w/ timestamps (3rd chan.)	84.90	98.10
<i>with 33% pretrain steps:</i>		
(iii) MEM (2 channels)	81.17	95.16
(iv) 8-channels	79.70	95.84

Table 9. A study of alternative input representations on N-Caltech101 [41] and N-Cars [51]. As the baseline, here we use 2-channel histograms (see Sec. 4.6) of size 224×224 , patch size 16, masking ratio 50%, RandAugment [10] and gradient clipping (see Tabs. 11 and 12 for other hyperparameters).

Ablation	N-Caltech101	N-Cars
(i) MEM	85.60	98.55
(ii) ViT-from-scratch	66.94	92.71
(iii) MEM w/o grad. clip.	22.87	90.73

Table 10. Ablation of gradient clipping for dVAE and pretraining stage on N-Caltech101 and N-Cars. Gradient clip values are in Tab. 11 and Tab. 12. Gradient clipping is essential to employ MEM pretraining on sparse event histograms. Note that gradient clipping is not employed in the RGB setting [3, 24].

tograms to spatial resolution 224×224 . For N-ImageNet, we resize the histogram to 256×341 and randomly crop the image to 224×224 . Next, we remove “hot pixels”, a variant of noise specific to event cameras, which manifests as a continuously triggering event [26]. We define a pixel as a “hot pixel” if its event count is ten standard deviations above the mean value in the event batch. We normalize the histogram values to $[0, 1]$. Lastly, during training, we perform RandAugment [10] with two operations and a magnitude of 20. The three stages of MEM (dVAE, pretraining, and finetuning) share the same input preprocessing (*i.e.*, the event histogram).

12.2. Datasets

We use the official train and test splits of N-ImageNet [29] and N-Cars [51] and DSEC-Semantic [19]. For N-Caltech101 [41], we randomly split the data into 80% training data and 20% test data. We ran 5-fold cross-validation to confirm that all random splits yield approximately the same result on N-Caltech101. The top-1 accuracy on the test sets in these experiments were 84.6%, 84.7%, 85.3%, 85.6% (reported in the main text), and 85.8%.

References

- [1] Sara Atito, Muhammad Awais, and Josef Kittler. SiT: Self-supervised vision transformer. *arXiv:2104.03602 [cs.CV]*, 2021. 2
- [2] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. MultiMAE: Multi-modal multi-task masked autoencoders. In *ECCV*, volume 13697, pages 348–367, 2022. 2
- [3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *ICLR*, 2022. 1, 2, 3, 4, 7, 14, 15
- [4] Sami Barchid, José Mennesson, and Chaabane Djéraba. Exploring joint embedding architectures and data augmentations for self-supervised representation learning in event-based vision. In *CVPR*, pages 3902–3911, 2023. 2, 6
- [5] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance regularization for self-supervised learning. In *ICLR*, 2022. 1
- [6] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatzé, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *CVPR*, pages 491–501, 2019. 5
- [7] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatzé, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. In *IEEE TIP*, page 29:9084–9098, 2020. 6
- [8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *CVPR*, pages 9650–9660, 2021. 1, 2
- [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 1
- [10] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc Le. RandAugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 3, 6, 7, 15
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2, 5, 6
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 1, 2, 3
- [13] Xiaoyi Dong, Jianmin Bao, Ting Zhang, Dongdong Chen, Weiming Zhang, Lu Yuan, Dong Chen, Fang Wen, and Nenghai Yu. PeCo: Perceptual codebook for BERT pre-training of vision transformers. *arXiv:2111.12710 [cs.CV]*, 2021. 2
- [14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4, 12
- [15] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incre-

Hyperparameter	N-ImageNet [29]	N-Caltech101 [41]	N-Cars [51]
Optimizer	Adam [30]	Adam [30]	Adam [30]
Optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.999)$	$\beta_1, \beta_2 = (0.9, 0.999)$	$\beta_1, \beta_2 = (0.9, 0.999)$
Learning rate	1e-3	2e-4	2e-4
Learning rate schedule	exponential (0.99)	exponential (0.99)	exponential (0.99)
Learning rate layer decay	0.98	0.98	0.98
KL weight	1e-10	1e-10	1e-10
Batch size	512	192	192
Grad clip	1e-2	1e-2	1e-2
Epochs	50	300	300

Table 11. Hyperparameters for the dVAE.

Hyperparameter	N-ImageNet [29]	N-Caltech101 [41]	N-Cars [51]
Optimizer	AdamW [35]	AdamW [35]	AdamW [35]
Optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$
Learning rate	1e-4	5e-4	3e-4
Learning rate schedule	cosine decay [34]	cosine decay [34]	cosine decay [34]
Warmup steps	1000	1000	1000
Weight decay	0.05	0.05	0.05
Batch size	512	512	384
Grad clip	30	30	30
Epochs	75 [†]	3000	1000 [‡]

Table 12. Hyperparameters for pretraining. [†] Cosine scheduler set for 300 epochs, but for computational reasons, only training for 75 epochs. [‡] Cosine scheduler set for 3000 epochs, but only training for 1000 epochs.

- mental Bayesian approach tested on 101 object categories. In *CVPRW*, pages 178–178, 2004. 5
- [16] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-end video-language transformers with masked visual-token modeling. *arXiv:2111.12681 [cs.CV]*, 2021. 2
- [17] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conrath, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE TPAMI*, 44(1):154–180, 2022. 1
- [18] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *CVPR*, pages 5633–5643, 2019. 5, 6
- [19] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robotics Autom. Lett.*, 6(3):4947–4954, 2021. 7, 8, 15
- [20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Un-supervised representation learning by predicting image rotations. In *ICLR*, 2018. 2
- [21] Jean-Bastien Grill, Florian Strub, and Florent Althé et al. Bootstrap your own latent – A new approach to self-supervised learning. In *NeurIPS*, 2020. 1
- [22] Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. EventDrop: Data augmentation for event-based learning. In *IJCAI*, pages 700–707, 2021. 2, 6
- [23] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 6
- [24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 1, 2, 3, 4, 7, 8, 9, 11, 14, 15, 17
- [25] Yuhuang Hu, Tobi Delbruck, and Shih-Chii Liu. Learning to exploit multiple vision modalities by using grafted networks. In *ECCV*, volume 12361, pages 85–101, 2020. 2
- [26] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic DVS events. In *CVPR*, pages 1312–1321, 2021. 15
- [27] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017. 4
- [28] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE TPAMI*, 43(11):4037–4058, 2020. 1, 2
- [29] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-ImageNet: Towards robust, fine-grained object recognition with event cameras. In *CVPR*, pages 2146–2156, 2021. 1, 4, 5, 6, 14, 15, 16, 17

Hyperparameter	N-ImageNet [29]	N-Caltech101 [41]	N-Cars [51]
Optimizer	AdamW [35]	AdamW [35]	AdamW [35]
Optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$	$\beta_1, \beta_2 = (0.9, 0.95)$
Learning rate	1e-3	4e-3	5e-4
Learning rate schedule	cosine decay [34]	cosine decay [34]	cosine decay [34]
Learning rate layer decay	0.65	0.65	0.65
Warmup epochs	20	20	20
Weight decay	0.3	0.05	0.05
Drop path	0.1	0.1	0.1
Dropout	0.0	0.1	0.1
Batch size	1024	1024	1024
Epochs	200 [†]	300	300

Table 13. Hyperparameters for finetuning. [†]Cosine scheduler set for 300 epochs, but for computational reasons, only finetuning for 200 epochs. We report the exponential moving average accuracy on N-ImageNet with a decay factor of 0.9999.

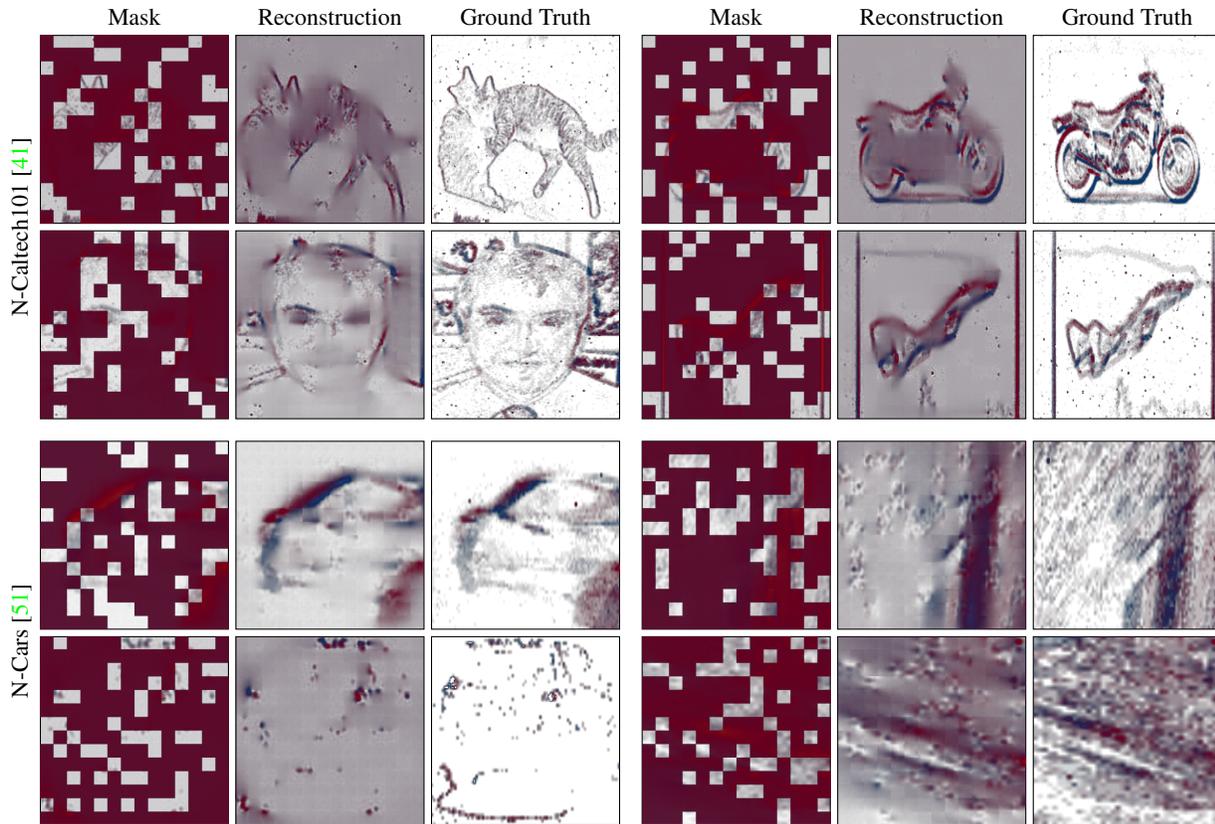


Figure 13. Additional masked patch predictions with the MAE loss (eMAE-entire-hist, see Tab. 6). From left to right: We visualize the masked input histograms, the reconstructions during eMAE-entire-hist pretraining, and the ground truth for N-Caltech101 [41] (top) and for N-Cars [51] (bottom). The MAE pretraining tasks struggles to recover sharp reconstructions, compared to MEM (*cf.* Fig. 12). We employ the default values of the MAE paper [24], *i.e.* a masking ratio of 75%, and random masking.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 16

[31] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 4

[32] Yijin Li, Han Zhou, Bangbang Yang, Ye Zhang, Zhaopeng Cui, Hujun Bao, and Guofeng Zhang. Graph-based asynchronous event processing for rapid object recognition. In *CVPR*, pages 934–943, 2021. 3, 6

[33] Chang Liu, Xiaojuan Qi, Edmund Y. Lam, and Ngai Wong.

- Fast classification and action recognition with event-based imaging. *IEEE Access*, 10:55638–55649, 2022. [6](#)
- [34] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. [16](#), [17](#)
- [35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. [16](#), [17](#)
- [36] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017. [4](#)
- [37] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *CVPR*, pages 5419–5427, 2018. [5](#), [6](#)
- [38] Nico Messikommer, Daniel Gehrig, Mathias Gehrig, and Davide Scaramuzza. Bridging the gap between events and frames through unsupervised domain adaptation. *IEEE Robotics Autom. Lett.*, 7(2):3515–3522, 2022. [1](#), [2](#)
- [39] Nico Messikommer, Daniel Gehrig, Antonio Loquercio, and Davide Scaramuzza. Event-based asynchronous sparse convolutional networks. In *ECCV*, volume 12353, pages 415–431, 2020. [6](#)
- [40] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, volume 9910, pages 69–84, 2016. [2](#)
- [41] 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. [1](#), [4](#), [5](#), [6](#), [7](#), [14](#), [15](#), [16](#), [17](#)
- [42] Federico Paredes-Vallés and Guido C. H. E. de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *CVPR*, pages 3446–3455, 2021. [2](#)
- [43] Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, and Barbara Caputo. E²(GO)MOTION: Motion augmented event stream for egocentric action recognition. In *CVPR*, pages 19903–19915, 2022. [1](#)
- [44] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, pages 8821–8831, 2021. [3](#), [4](#)
- [45] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *CVPR*, pages 3857–3866, 2019. [2](#), [6](#), [8](#), [9](#), [10](#)
- [46] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE TPAMI*, 43(6):1964–1980, 2019. [2](#)
- [47] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014. [4](#)
- [48] Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik. ImageNet-21K pretraining for the masses. In *NeurIPS*, 2021. [5](#)
- [49] Jason Tyler Rolfe. Discrete variational autoencoders. In *ICLR*, 2017. [3](#)
- [50] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. AEGNN: Asynchronous event-based graph neural networks. In *CVPR*, pages 12371–12381, 2022. [3](#), [6](#)
- [51] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. HATS: Histograms of averaged time surfaces for robust event-based object classification. In *CVPR*, pages 1731–1740, 2018. [1](#), [4](#), [5](#), [6](#), [7](#), [11](#), [13](#), [14](#), [15](#), [16](#), [17](#)
- [52] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, et al. A 640×480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In *ISSCC*, pages 66–67. IEEE, 2017. [5](#)
- [53] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, augmentation, and regularization in Vision Transformers. *Trans. Mach. Learn. Res.*, 2022. [5](#), [6](#)
- [54] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *CVPR*, pages 843–852, 2017. [1](#)
- [55] Zhaoning Sun, Nico Messikommer, Daniel Gehrig, and Davide Scaramuzza. ESS: Learning event-based semantic segmentation from still images. In *ECCV*, volume 13694, pages 341–357, 2022. [2](#)
- [56] Hao Tan, Jie Lei, Thomas Wolf, and Mohit Bansal. VIM-PAC: Video pre-training via masked token prediction and contrastive learning. *arXiv:2106.11250 [cs.CV]*, 2021. [2](#), [4](#)
- [57] Ziyu Wan, Jingbo Zhang, Dongdong Chen, and Jing Liao. High-fidelity pluralistic image completion with transformers. In *CVPR*, pages 4692–4701, 2021. [2](#)
- [58] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. EvDistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *CVPR*, pages 608–619, 2021. [2](#)
- [59] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luwei Zhou, and Lu Yuan. BEVT: BERT pretraining of video transformers. In *CVPR*, pages 14713–14723, 2022. [2](#)
- [60] Zuowen Wang, Yuhuang Hu, and Shih-Chii Liu. Exploiting spatial sparsity for event cameras with visual transformers. In *ICIP*, pages 411–415, 2022. [6](#)
- [61] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *CVPR*, pages 9643–9653, 2022. [2](#)
- [62] Yan Yang, Liyuan Pan, and Liu Liu. Event camera data pre-training. In *ICCV*, pages 10699–10709, 2023. [2](#), [5](#)
- [63] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *arXiv:2205.01917 [cs.CV]*, 2022. [5](#)
- [64] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3D point

- cloud transformers with masked point modeling. In *CVPR*, pages 19291–19300, 2022. [2](#), [4](#)
- [65] Alessandro Zanardi, Andreas Aumiller, Julian G. Zilly, Andrea Censi, and Emilio Frazzoli. Cross-modal learning filters for rgb-neuromorphic wormhole learning. In *Robotics: Science and Systems*, 2019. [2](#)
- [66] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, pages 12104–12113, 2022. [5](#)
- [67] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. *arXiv:2111.07832 [cs.CV]*, 2021. [2](#)
- [68] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *RSS*, 2018. [2](#)