EVENT CAMERA MOTION–APPEARANCE DISENTANGLEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Event cameras respond to local brightness changes caused by moving edges in the scene. Therefore, motion and appearance are entangled in the event stream, and it is expected that transformations of the data (e.g., representations) contain information about both. A representation that separated them would enable recognizing an object across different motions, comparing motions across different objects, and combining the appearance of one event stream with the motion of another. To this end, we ask how much of this separation is *already present* in latent event representations, and how far it can be *exposed* without retraining their encoders. To our knowledge, this question has not been studied. We frame it around two complementary questions: whether each factor (motion or apperance) can be *extracted* from a frozen representation, and whether the two can be *used* independently, for example, in the above-mentioned combination. We introduce EventPaths, a synthetic benchmark in which every object (appearance) is recorded across many trajectories (motion), and we complement it with real event-camera data. We find that existing event encoders already implicitly separate appearance from motion, and that this separation can be extracted, even without training, using pairs of event streams that share one factor. We also introduce PRISMA (Predictive Inference of Separable Motion and Appearance), a procedure in which the appearance code of one event stream and the motion code of another are used to predict the observed representation of the recording they describe. This technique raises recombination accuracy from 56.8% to 79.2% through an unchanged frozen decoder. Finally, we show that the recombined event streams are useful as training data for recognising objects under unseen motion.

1 INTRODUCTION

Event cameras record per-pixel brightness changes asynchronously, called "events". Under constant illumination, this means that events are produced mainly by moving edges in the scene (Gallego et al., 2022). As a consequence, motion and appearance are entangled in the event signal: changing the scene and/or its relative motion changes the space-time structure of the events. When an artificial neural network processes such input, we therefore expect its latent representation to contain information about both appearance and motion. However, the fact that a representation is useful for recognizing an object or estimating its motion does not necessarily imply that these two factors are represented separately.

Existing approaches that separate appearance and motion (in video (Denton & Birodkar, 2017; Tulyakov et al., 2018), and more recently for hybrid frame-event data (Chen et al., 2025)) train two dedicated models for that purpose. To our knowledge, the complementary question has not been asked for event representations: in this work we ask how much of this separation is *already present* in latent event representations, and how far it can be *exposed* without retraining their encoders. More specifically, given a frozen event encoder, we aim to extract an appearance code that remains stable across different motion trajectories while preserving object information, and a motion code that remains stable across different appearances while preserving trajectory information. We consider *factor separation* the main objective, while the tasks of (object and/or motion) *retrieval* and *recombination* provide two complementary ways to evaluate how successful the separation is.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our starting point is intuitive: paired samples sharing one factor reveal which feature variations the corresponding code should learn to ignore. Same-object samples under different motions capture motion variation, while same-motion samples with different objects capture appearance variation. We use these pairs to derive factor-selective subspaces or train small projection modules on frozen features. Rather than training an encoder to explicitly predict appearance or motion, we probe what information about these factors is already encoded in representations learned for other purposes.

We study this question using several frozen event representations. EventPretrain (Klenk et al., 2024), E2FAI (Guo et al., 2025), and GEP (Cao et al., 2026) are independently pretrained encoders that remain fixed throughout our analysis. For experiments that require the generation of event data, we additionally use an autoencoder whose encoder and decoder are frozen before factor extraction. This allows us to study not only whether motion and appearance can be extracted from the representation, but also whether the corresponding codes can be used independently through the frozen decoder.

We find evidence of separation, and the reason it is difficult to obtain is physical rather than architectural. Object edges parallel to the direction of motion do not produce events, which means that event streams and their corresponding frozen features are only partial observations of the object. A balanced decomposition of the features confirms this: the dominant term is neither the object nor the motion alone, but their *interaction*, that is, the part of the feature that depends on the specific object–motion combination. Recognizing the two factors, however, does not make them usable. In our experiments, we successfully extract disentangled codes that successfully allow retrieval, but fail to recombine correctly half of the times. Hence, the factors are accessible, but not yet independently usable.

This gap motivates us to propose PRISMA (Predictive Inference of Separable Motion and Appearance), which replaces invariance with *prediction*. Given the appearance code of one event-data sample and the motion code of another sample of the same object, our model must predict the frozen feature of the second sample. Because the two samples share one factor (either appearance or motion) the requested combination corresponds to an observed sample. This avoids both counterfactual supervision and a decoder during training. The resulting objective resembles joint-embedding prediction (Assran et al., 2023), but differs in three important ways: (i) the target encoder is frozen and pretrained, eliminating the need for collapse-prevention mechanisms; (ii) the context and target are different data samples rather than masked regions of the same input, (iii) the transformation between them is not given as metadata; it is represented by one of the two learned codes.

Thus, our contributions are fourfold:

- We introduce the problem of motion–appearance factorisation in *frozen* event representations. Rather than training an encoder to separate the two factors, we ask whether encoders trained for other objectives already contain them in separable form, and whether this separation supports recombination;
- We show that the two factors can be almost perfectly *readable* yet only partially *usable*. Across three released encoders and real event-camera recordings, appearance and motion can be recovered from frozen features (even without training) despite most feature variance arising from object–motion interactions. Yet codes that identify the intended event-data sample almost perfectly recombine correctly in only 56.8% of cases under the conventional invariance objective.
- We introduce PRISMA, a predictive procedure that targets this gap by predicting the frozen representation of a real recorded combination. With an unchanged frozen decoder, this raises recombination accuracy from 56.8% to 79.2%;
- We release EventPaths, a dataset with a comprehensive coverage of combinations of object–motion trajectories, enabling recombinations to be evaluated against their ground truth. We show that recombined event samples are useful for training, recovering 61% of the gain from recording the missing samples at $214\times$ less compute.

We release both the code and dataset to support a fair and reproducible benchmark.

Table 1: **Comparison of event-based motion–appearance separation methods.**

Method	Input	Space	# Encoders	Frozen	Flow-free	Edge-prior-free
MAD (Chen et al., 2025)	E	Input tensors	2	✗	✗	✓
Zhu et al. (2025)	E+F	Learned feature maps	2	✗	✓	✗
EvInsMOS (Wan et al., 2025)	E+F	Learned feature maps	2	✗	✗	✓/✗†
DIMOS (Huang et al., 2026)	E+F	Learned feature maps	2	✗	✗	✗†
PRISMA (ours)	E	Pretrained latent space	1	✓	✓	✓

E: events; F: frames; Input: inference modalities; Frozen: encoder is not retrained for separation. Flow-free means no explicit flow estimation. Edge-prior-free means no edge guidance or edge-weighted regularization. † ✓/✗ denotes supervised/self-supervised flow training scenarios.

2 Related Works

Motion–appearance disentanglement for event data. Event-based reconstruction, optical flow, and self-supervised pretraining learn spatiotemporal features from asynchronous events (Gehrig et al., 2021; Liu et al., 2023; Klenk et al., 2024; Cao et al., 2026). Event synthesis methods learn generators for event data (Zhu et al., 2021; Hu et al., 2026). Table 1 compares our method with four prior approaches that aim at separating motion from appearance or edge cues. MAD (Chen et al., 2025) performs separation at the input level: estimated optical flow forms a motion tensor representation, while an image of flow-warped events with clear contours and edges forms an appearance tensor; dedicated branches process these tensors for downstream tasks. Zhu et al. (2025) separate edge and motion cues into learned feature maps for deblurring, using short-window events as an edge prior without explicitly estimating optical flow. EvInsMOS (Wan et al., 2025) trains separate image and event encoders to learn complementary texture and motion features that enable moving object segmentation; it uses contrastive learning and an auxiliary optical-flow objective during training. DIMOS (Huang et al., 2026) also operates on learned feature maps, using dedicated appearance and motion encoders for both modalities, intra-modal contrastive learning, and an auxiliary optical-flow objective during training. In released implementations, DIMOS and the self-supervised variant of EvInsMOS additionally use edge-aware flow smoothness terms weighted by image gradients. PRISMA separates factors in a pretrained latent space while keeping the encoder frozen; its splitter uses no optical-flow estimation, flow losses, explicit edge priors, or edge-weighted regularization.

Content–motion separation. Content–motion separation has also been studied in RGB video modeling. DRNet, MCNet, MoCoGAN, and DDPAE learn to separate content from temporal dynamics for video representation learning, prediction and generation (Denton & Birodkar, 2017; Villegas et al., 2017; Tulyakov et al., 2018; Hsieh et al., 2018). Image animation methods transfer motion from a driving sequence to a source image through learned motion representations (Siarohin et al., 2019; Wang et al., 2022). These approaches optimize their representations or generative models explicitly for content–motion separation.

Paired supervision and feature geometry. Grouped observations provide supervision for disentanglement that is unavailable in fully unsupervised settings (Locatello et al., 2020b;a; von Kügelgen et al., 2021; Bouchacourt et al., 2018). Contrastive PCA and generalized eigenproblems (Abid et al., 2018), VICReg-style regularization (Bardes et al., 2022), contrastive learning (Chen et al., 2020; Khosla et al., 2020), and concept erasure (Ravfogel et al., 2020; Belrose et al., 2023) provide tools for characterizing and constraining feature geometry. However, factor representations do not need to support correct recombination through a decoder to allow accurate linear prediction (Montero et al., 2021; Schott et al., 2022; Lachapelle et al., 2023; Wiedemer et al., 2023).

Frozen visual representations. Recent work examines factor information in frozen RGB and video encoders. Wang et al. (2026) analyze compositional factor structure in frozen image features using factorial data grids with held-out combinations. Musa et al. (2026) probe frozen video models for camera motion and use the geometry of latent trajectories to steer video generation.

In summary, previous event-based methods compared in Table 1 train task-specific feature extractors and use estimated optical flow, auxiliary flow objectives, or event-derived edge priors to guide separation. In contrast, our approach disentangles frozen pretrained latent representations from a single encoder without additional training or the aboe-mentioned cues, and evaluates whether the resulting motion and appearance factors can be recombined to generate new event observations.

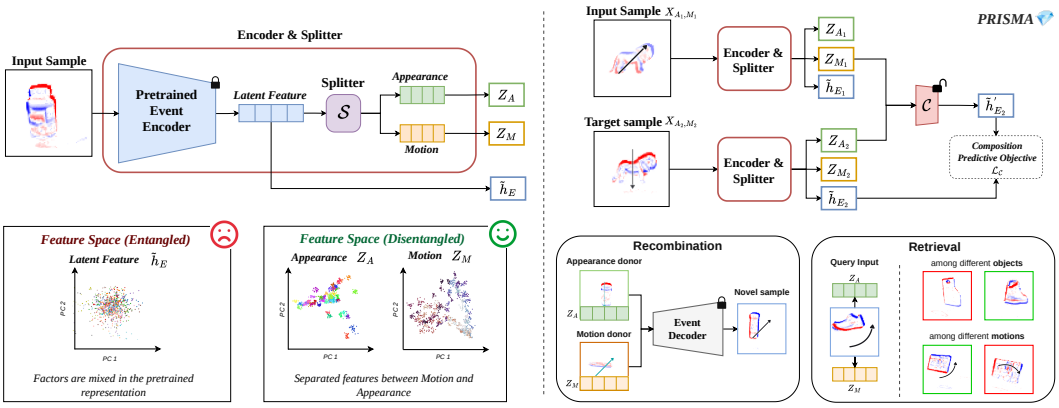


Figure 1: **PRISMA.** *Left*: an "encoder & splitter" converts events E into a latent feature $\tilde{h}(E)$ and separates it into appearance and motion factors. *Right*: PRISMA composes these factors from paired input event-data samples (using two "encoder & splitter" blocks) to predict a target representation, enhancing factor retrieval and recombination through the frozen decoder.

3 Methodology

3.1 Setup

Motion-Appearance Disentanglement. Let $x_{a,m}$ be an event-data sample (i.e., window of events) of object a under motion m, accumulated into B signed temporal bins, and let E be a frozen encoder with representation $h = E(x_{a,m})$. We look for two maps (an appearance neural network head A and a motion head M) producing codes/representations z_A and z_M such that z_A retains appearance while suppressing motion, and z_M retains motion while suppressing appearance. We evaluate the resulting codes through factor retrieval (i.e., ability to identify motion and appearance) and through recombination of appearance and motion across data samples.

Whitening. The two maps are applied to whitened representations. Writing the training covariance of the representations as $U\Sigma^2U^\top$, we use $\tilde{h} = \Sigma^{-1}U^\top(h - \bar{h})$, truncated to R components, with $\bar{h}$, U and Σ estimated on training samples and representing the average feature vector and the whitening rotation and scaling matrices, respectively. Whitening matters here since the directions that distinguish objects and trajectories are not the directions that carry the most variance. In our experiments, we found that out of 512 feature dimensions, 99% of the training variance lies in 147 dimensions, and truncating there costs several points of recombination accuracy. We therefore set $R = 384$. The two heads operate on this whitened representation, with $z_A = A(\tilde{h})$ and $z_M = M(\tilde{h})$.

3.2 Measuring what frozen features keep apart

Data sample pairs as experiments. By encoding two samples of the same object under different motions, we can identify which features are due to motion. Let $\mathcal{P}_A$ be such pairs. Vice-versa, pairs $\mathcal{P}_M$ of different objects with the same motion will exhibit only appearance differences. We then identify the feature difference within a pair as $\Delta\tilde{h}$. The covariances $C_M = \frac{1}{2}\mathbb{E}_{\mathcal{P}_A}[\Delta\tilde{h}\Delta\tilde{h}^\top]$ and $C_A = \frac{1}{2}\mathbb{E}_{\mathcal{P}_M}[\Delta\tilde{h}\Delta\tilde{h}^\top]$ say how the feature varies when only motion, or only appearance, changes.

Learned neural network heads use the same data-sample pairs: an invariance term pulls the two representations of a pair together in the head that should not change, and variance, covariance and cross-covariance terms keep the representations informative. We propose the loss:

$$\mathcal{L}_{\text{inv}} = \omega_1\left(\|\Delta z_A\|^2 + \|\Delta z_M\|^2\right) + \omega_2\,(v_A + v_M) + \omega_3\,(c_A + c_M) + \omega_4\,c_{AM}, \tag{1}$$

where Δz_A is the difference of appearance codes over a pair in $\mathcal{P}_A$ (similarly for Δz_M motion code difference over a pair in $\mathcal{P}_M$), v is a hinged standard-deviation term that prevents a code from collapsing, c penalises the off-diagonal covariance within a code, and c_{AM} the cross-covariance between the two codes. The weights are given in Section B. As a control, we also train the same heads with NT-Xent on the same pairs (Section D.1).

Equation 1 is a VICReg-style objective (Bardes et al., 2022) applied to each code, with samples sharing a factor playing the role of augmented views, plus the cross-code decorrelation term c_{AM}.

Training-free codes. The same pair covariances also define a splitter with no learned parameters. Directions along which C_M is large and C_A small are motion-selective, and vice versa. We solve

$$C_M\, w = \lambda\, (C_A + \epsilon \mathbb{1})\, w, \tag{2}$$

a generalized eigenproblem Fukunaga (2013), which ranks directions by their relative sensitivity to motion and appearance. Its leading eigenvectors span a motion subspace. With the matrices exchanged, the same procedure leads to an appearance subspace. Projecting $\tilde{h}$ onto these subspaces yields the corresponding training-free codes (Section A).

3.3 PRISMA

The predictive objective. The appearance of an object is captured differently depending on its motion, so the invariance term of Eq. (1) alone may discard differences that carry appearance information. We therefore augment the invariant splitter with a predictive objective, yielding Predictive Inference of Separable Motion and Appearance (PRISMA). Using the pair sets defined above, for $(x_1, x_2) \sim \mathcal{P}_A$ a predictor $\mathcal{C}$ receives the appearance code of x_1 and the motion code of x_2, and predicts the frozen representation of x_2:

$$\mathcal{L}_{\text{pred}} = \left\| \mathcal{C}\big(z_A(x_1), z_M(x_2)\big) - \tilde{h}(x_2) \right\|^2, \qquad (x_1, x_2) \sim \mathcal{P}_A. \tag{3}$$

The symmetric construction is used for pairs in $\mathcal{P}_M$. Collecting terms, the full objective is

$$\mathcal{L} = \mathcal{L}_{\text{inv}} + \lambda\, \mathcal{L}_{\text{pred}}, \tag{4}$$

and neural network heads and predictor are trained together.

Two things distinguish this from an autoencoding objective. The target is a *latent representation*, thus nothing in the loss rewards rendering. Secondly, the target belongs to a *different sample* from the appearance code, so the predictor must model how the two factors (re-)combine rather than learn an identity map. Setting the target to $\tilde{h}(x_1)$ instead (i.e., predicting a sample from its own codes) removes the transformation to be modelled, and we report it as a control check.

4 Experiments

4.1 Experimental Setup

Datasets We use N-ImageNet (Kim et al., 2021), consisting of ImageNet images viewed by a real DVS under six camera trajectories. In our experiments, we held out one trajectory at a time. Its motion is a camera saccade over a still image, narrower than object motion. To the best of our knowledge, this is the only captured data available in the literature where the same appearance is observed under repeated physical trajectories.

We also propose EventPaths, a novel dataset rendered by combining Kubric Greff et al. (2022) and ESIM Rebecq et al. (2018). EventPaths is built by rendering 512 objects along 96 trajectories, with a total of 49,152 data samples of 1.2s each. We render every possible motion-appearance combination, thus enabling us to have a ground truth for each motion-appearance swap. The main split withholds eight whole trajectories (96 test samples) from training, including the autoencoder's pretraining; a *strict curved* split does the same for curved trajectories, the harder case.

Encoders. We evaluate four encoders. EventPretrain (Klenk et al., 2024), E2FAI (Guo et al., 2025), and GEP (Cao et al., 2026) are released models never trained on EventPaths: EventPretrain is pretrained on event data, E2FAI is trained on unsupervised flow and intensity estimation tasks, and GEP is trained through RGB–event alignment. Voxel-AE is a reconstruction autoencoder trained from scratch on EventPaths and frozen with its decoder; its 64-dimensional latent is the only representation with a decoder producing event samples, so recombination is evaluated there (Section B). Together, these encoders cover event-only self-supervision, task-oriented pretraining, RGB–event alignment, and signal (e.g., voxelgrid) reconstruction. For all encoders, $h = E(x)$ is whitened as described in Section 3.1, and all neural network heads operate on the whitened feature $\tilde{h}$.

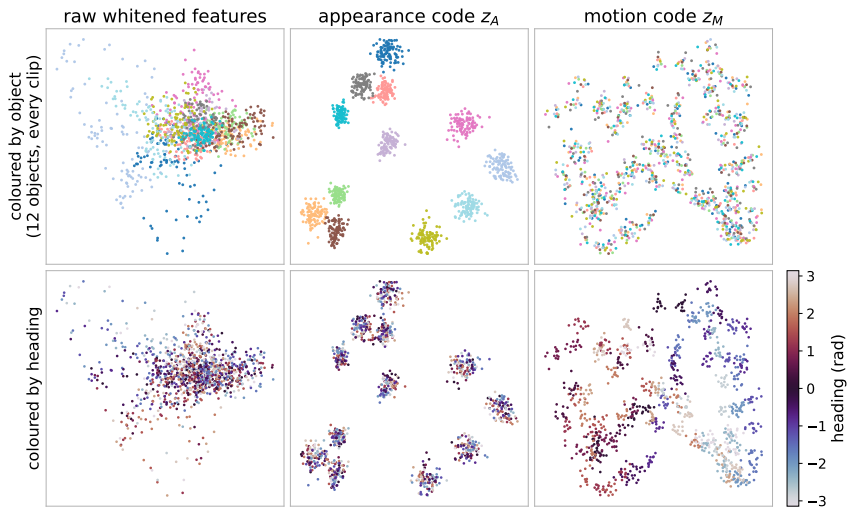


Figure 2: **What each representation organises.** Twelve objects, with all their 96 samples drawn and coloured by object (1152 points, no subsampling). The projection is linear (PCA), so the structure is not produced by a nonlinear embedding. In the raw features (left) the objects are mixed into one cloud, in the appearance code (centre) they fall into twelve distinguishable clusters, and in the motion code (right) the colours are instead interleaved everywhere. The bottom row colours the same three maps by motion direction and shows the opposite pattern.

Table 2: **Factor probes on frozen features.** 5-NN accuracy on the 4096 held-out samples (%). The motion code contains no object information, while the appearance code retains some trajectory information.

Objective	**Using**	**Samples identified** (↑)	
		Object	Trajectory
Chance		0.2	1.0
Invariance	appear. z_A	93.0	2.5
	motion z_M	0.0	**100.0**
PRISMA (Ours)	appear. z_A	**95.0**	7.8
	motion z_M	0.0	**100.0**

Table 3: **Generated samples** (Voxel-AE, the one encoder with a decoder). The composed feature is decoded and judged. The column *Both* is the recombination accuracy (chance 1.6%). Indented rows: the controls of Section 4.3.

Splitter	**Object**	**Trajectory**	**Both**
Covariance contrast	69.3	92.2	68.0
Generalised eigenproblem	70.1	93.0	68.4
Invariance head	59.0	84.9	56.8
w/ pred. target, heads frozen	60.7	85.2	58.2
w/ own-sample target, λ=100	27.8	47.8	20.2
PRISMA (ours)	**79.6**	**97.2**	**79.2**

Evaluation Tasks and Metrics. A recombination query takes the appearance code of one held-out sample and the motion code of another; its answer is the recording of that object on that motion trajectory (which exists in EventPaths). *Exact sample R@1* asks whether the composed feature is closer to that recording than to any of the 49152 samples (chance 0.002%). *Object* and *trajectory* admit eight choices each: the answer must be closer than seven recordings of the same object on other trajectories, and than seven recordings of other objects on the same trajectory; *both* requires the two choices to be correct at once (chance 1.6%). As distance between recordings we use the L1 distance over the event-converted voxel grids (Zhu et al., 2019). The same choices are asked of the codes: of the composed feature and of the sample generated from it by the frozen decoder.

4.2 Frozen representations already contain a factorised geometry

We first ask whether the two factors can be read separately from a frozen representation. Table 2 probes the representations with a 5-NN classifier on the 4096 held-out samples, whose trajectories are unseen during training. With PRISMA, the appearance code identifies the object in 95.0% of samples (0.2% chance), while the motion code identifies the motion trajectory in 100.0%. The separation is also visible in Fig. 2: samples of the same object cluster tightly in the appearance code but remain dispersed in the raw features and motion code.

Table 4: **Splitters on EventPaths across frozen encoders.** Held-out trajectories; means in %. Training-free rows use covariance-based splitters; trained rows use the same predictor budget without per-encoder retuning. *Both*: correct object and trajectory; *exact sample R@1*: target ranked first among all 49152 samples. Bold: best; underline: second best.

encoder	splitter	codes	composed feature	
		both	both	exact sample R@1
Voxel-AE	Covariance contrast (no training)	63.3	97.9	63.9
	Generalised eigenproblem	97.9	<u>98.6</u>	<u>64.1</u>
	Invariance head (trained)	<u>99.6</u>	89.6	29.7
	PRISMA (ours)	**99.9**	**99.3**	**79.8**
EventPretrain	Covariance contrast (no training)	50.0	<u>30.9</u>	<u>1.2</u>
	Generalised eigenproblem	<u>75.4</u>	**31.4**	**2.0**
	Invariance head (trained)	**83.1**	27.1	0.3
	PRISMA (ours)	65.4	28.5	1.0
E2FAI	Covariance contrast (no training)	44.3	**68.4**	**10.7**
	Generalised eigenproblem	62.9	65.4	<u>10.5</u>
	Invariance head (trained)	<u>87.6</u>	44.8	2.9
	PRISMA (ours)	**95.2**	<u>67.4</u>	8.1
GEP-small	Covariance contrast (no training)	60.0	73.6	10.9
	Generalised eigenproblem	90.2	<u>74.6</u>	<u>12.3</u>
	Invariance head (trained)	<u>92.4</u>	69.1	5.5
	PRISMA (ours)	**95.2**	**79.2**	**13.3**

The codes are not redundant. Object identity is unrecoverable from the motion code (0.0%), consistently across all 10 code widths (Section C). Conversely, some motion trajectory information remains in the appearance code (7.8% vs. 1.04% chance; 2.5% under invariance). This residual coupling is expected from the event-generation process: which parts of an object are observed depends on its motion. It is therefore also larger for PRISMA, which preserves motion-dependent evidence that invariance tends to suppress.

Training-free codes. We next ask whether the factor subspaces can be recovered without training. The generalised eigenproblem in Eq. (2) yields such codes directly from the within-factor covariances. With no fitted parameters, a recombination query is answered correctly from these codes in 97.9% of cases and from the composed feature in 98.6% (Table 4), and the decoded sample reaches 68.4% recombination accuracy (Table 3), above the trained invariance splitter (56.8%) and below PRISMA (79.2%). The simpler covariance contrast reaches a similar accuracy after decoding (68.0%) but answers the query from its codes far less often (63.3% vs. 97.9%), indicating that the relative scaling of the two covariances matters. Thus, the factorisation is already present in the frozen representation: training improves the splitter, but does not create the structure. On the three public encoders, the generalised eigenproblem codes likewise answer the recombination query more often than the contrast (75.4, 62.9 and 90.2%, against 50.0, 44.3 and 60.0%).

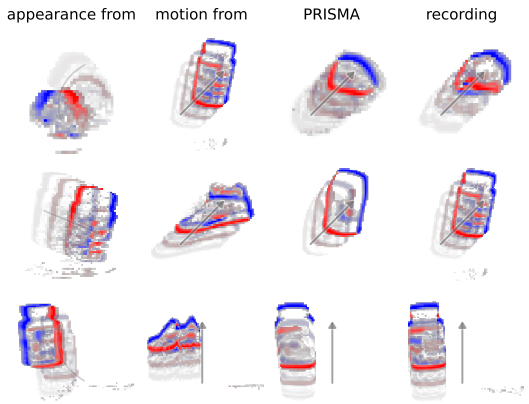


Figure 3: **Recombination examples.** Each row combines the appearance from the first column with the motion from the second using PRISMA, then decodes the composed factors with the frozen decoder (third column). The last column shows the corresponding real recording. For readability, we show three temporal slices rather than all events; arrows indicate the recorded motion trajectories.

Real data. All results so far use synthetic data, so we repeat the comparison on N-ImageNet with three publicly released encoders (Table 5). The pattern of the rendered grid holds on a real sensor: every splitter makes the codes readable, yet composing them is far harder, and combination R@1 after the same predictor stays below 18% for every splitter. PRISMA improves it over invariance

Table 5: **Readable is not composable on a real sensor.** N-ImageNet with one of six recording conditions held out at a time. We report condition retrieval from the codes, combination R@1 after the predictor, and 100-way class accuracy under seen conditions. Bold: best per encoder; underline: second best.

Frozen encoder	Splitter	same condition R@1 (codes)	combination R@1 (after predictor)	class acc. (seen conditions)
EventPretrain	contrastive (NT-Xent)	52.9 ±12.4	0.5 ±0.2	9.2 ±0.6
	invariance	**92.6 ±5.3**	0.3 ±0.3	8.1 ±0.6
	generalised eigenproblem (ours, training-free)	47.9 ±7.4	0.5 ±0.2	7.1 ±0.2
	PRISMA (ours)	69.5 ±4.9	**0.5 ±0.2**	**17.7 ±1.5**
E2FAI	contrastive (NT-Xent)	59.0 ±10.8	2.5 ±0.9	11.2 ±0.6
	invariance	**90.9 ±8.2**	1.7 ±0.5	9.3 ±0.5
	generalised eigenproblem (ours, training-free)	53.1 ±10.3	2.4 ±1.5	7.5 ±0.3
	PRISMA (ours)	63.5 ±7.6	**2.9 ±1.3**	**16.7 ±1.2**
GEP-small	contrastive (NT-Xent)	51.8 ±16.9	17.3 ±4.2	17.5 ±0.3
	invariance	**89.0 ±10.4**	9.3 ±5.0	14.3 ±0.1
	generalised eigenproblem (ours, training-free)	56.8 ±5.0	**17.5 ±7.3**	13.2 ±0.3
	PRISMA(ours)	71.4 ±7.3	13.1 ±7.0	**19.3 ±0.4**

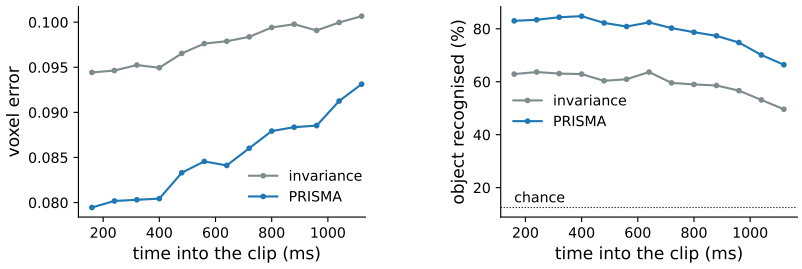


Figure 4: **Temporal decay of a recombination.** Left: voxelgrid error of the generated sample against the recording it should match, per temporal bin. Right: eight-way object verdict computed from one bin at a time, so that every point uses the same amount of evidence. PRISMA is better at every bin (68.0 against 45.3 at the first, 44.5 against 35.0 at the last), and both remain well above the 12.5% chance rate, but both decay as the motion accumulates.

on all three encoders (from 9.3 to 13.1 on GEP-small), and its appearance code keeps the most object information of the four splitters under the training conditions, with 17.7, 16.7 and 19.3% class accuracy against at most 17.5% for the others. Its motion code also retrieves the recording condition better than NT-Xent and the training-free codes on every encoder.

4.3 Prediction turns readable factors into compositional ones

Retrieval tests whether the factors can be read from a frozen representation; recombination asks whether they can also be used independently. The trained rows of Table 4 compare invariance (Eq. (1)) with PRISMA (Eq. (4)) using the same heads, predictor, and training budget. From the codes alone, both objectives answer the recombination query almost perfectly (99.6 and 99.9%), so the difference lies in composition: exact-sample R@1 rises from 29.7% to 79.8%, and decoded recombination accuracy from 56.8% to 79.2% (Table 3). The object remains identifiable in 79.6% of generated samples, against 59.0% with invariance (Fig. 3). The effect also transfers to the three public encoders, with larger gains where the frozen representation retains more information for prediction; EventPretrain remains near floor on exact-sample R@1.

Controls. Two controls isolate the source of the improvement. Training only the predictor while keeping the invariance heads frozen changes recombination only from 56.8 to 58.2%, showing that the predictive objective must shape the codes themselves. Conversely, training the same architecture to predict each sample from its own codes reduces recombination to 20.2%, far below PRISMA at 79.2%, ruling out capacity or optimization budget as the explanation. A wrong donor remains below 2%. Additional PCA and NT-Xent splitters are reported in Section D.1.

Table 6: **Recombination as data augmentation.** Each object is collected on six of its 96 trajectories; the 12288 missing combinations are then supplied by each splitter's generator, or taken from their recordings (upper bound). An object classifier is trained on the frozen representation and tested on held-out trajectories. The two controls add the same number of samples without new combinations (jittered copies) or with wrong labels (shuffled). Three seeds, mean±s.e.; costs measured on one GPU, relative to generation.

Training data	samples	object recognised ↑	cost of the added samples	relative cost ↓
sparse collection only	3072	46.8 ± 0.3	—	—
+ jittered copies of the real samples (control)	+12288	47.2 ± 0.1	—	—
+ generated, labels shuffled (control)	+12288	17.1 ± 0.4	10 s	1×
+ generated by invariance	+12288	64.9 ± 1.0	10 s	1×
+ generated by **PRISMA (ours)**	+12288	$\mathbf{77.8 \pm 0.3}$	**10 s**	**1**×
+ recorded (upper bound)	+12288	98.5 ± 0.1	36 min	214×

Temporal decay. Recombination quality decreases along the sample (Fig. 4). Using one temporal bin at a time, object recognition with PRISMA falls from 68.0% to 44.5%, while the invariance baseline falls from 45.3% to 35.0%; voxel error increases over the same interval. This is consistent with the sensor model: as motion accumulates, more appearance information is lost. PRISMA remains better throughout, but does not remove this decay.

Localising the remaining error. Following one swap through the pipeline shows where the remaining error lies (Section D.3). Both factors are identified from the codes in 99.9% of queries, from the composed feature in 99.3%, and the frozen decoder, when given a true representation, in 100.0%; given the composed representation, however, recombination accuracy is 79.2%. The loss is therefore neither in the factorisation nor in the decoder: the composed representation is close enough to the target for retrieval, but not for synthesis.

4.4 Recombinations improve recognition under unseen motion

Finally, we test whether recombined samples improve recognition under unseen motion. Each object is observed on only six of its 96 trajectories (3072 real samples), and the heads and predictor are trained on this sparse set. We then generate the 12288 missing object–trajectory combinations and use them to augment training of an object classifier. Evaluation is performed on held-out trajectories, unseen by the entire pipeline. As shown in Table 6, accuracy rises from 46.8 to 77.8%. Because no new object labels are introduced, the gain comes from the additional motion diversity provided by recombination.

Comparison with recording the missing samples. Because EventPaths contains every object–trajectory combination, we can compare generated samples with the corresponding recorded upper bound. Recorded samples reach 98.5% accuracy, while recombination reaches 77.8%, recovering about 60% of the gain from sparse to complete coverage. Generating the 12288 samples takes about 10 s, versus 36 minutes for rendering and simulation ($214\times$ slower). Thus, recombination recovers much of the benefit of additional data at a fraction of the cost.

5 Conclusion

In this paper, we asked whether frozen event encoders already separate appearance from motion, and whether these two factors are merely readable or can also be used compositionally. Our results show that substantial factor structure is already present before any disentanglement model is trained: a training-free generalised eigenspace reaches 68.4% recombination accuracy, above a learned invariance splitter at 56.8%. This suggests that disentanglement need not always be created by retraining an encoder; part of it may already be latent in representations learned for other objectives. At the same time, readability does not imply compositionality. PRISMA targets this gap by learning how appearance and motion combine in the frozen representation, raising recombination accuracy to 79.2% and exact-sample R@1 to 79.8%. This opens a broader use of frozen event representations: factor-specific retrieval, controlled recombination, and data generation without retraining an underlying encoder. In our case, generated object–motion combinations improve recognition under unseen trajectories from 46.8 to 77.8% without adding new object labels.

AI DISCLOSURE

In this work, we used generative AI tools to (*tasks with required disclosure*): assist with translation clean and reformat dataset, improve readability of some sentences in the text after an initial writing.

We have not used generative AI tools to (*other tasks with required disclosure*): generate synthetic data sets, design or provide feedback on research methodology or experiments, help develop theoretical models or conceptual frameworks, support qualitative and thematic data analysis, interpret results, implement methods.

The rest of the *required disclosure tasks*: formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, are not applicable to this work.

Additionally, we used generative AI tools to (*tasks with recommended disclosure*): edit a research paper to improve readability, create or edit software code, sourcing/searching for information.

We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Abubakar Abid, Martin J. Zhang, Vivek K. Bagaria, and James Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature Communications*, 9:2134, 2018.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2105.04906.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.03819.

Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

Jianwen Cao, Jiaxu Xing, Nico Messikommer, and Davide Scaramuzza. Generative event pretraining with foundation model alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Nuo Chen, Boyang Li, Yingqian Wang, Xinyi Ying, Longguang Wang, Chushu Zhang, Yulan Guo, Miao Li, and Wei An. Motion and appearance decoupling representation for event cameras. *IEEE Transactions on Image Processing*, 34:5964–5977, 2025. doi: 10.1109/TIP.2025.3607632.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, 2020.

Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(1):154–180, 2022. doi: 10.1109/TPAMI.2020.3008413.

Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *2021 International Conference on 3D Vision (3DV)*, pp. 197–206. IEEE, 2021.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J. Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti Derek Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: A scalable dataset generator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Shuang Guo, Friedhelm Hamann, and Guillermo Gallego. Unsupervised joint learning of optical flow and intensity with event cameras. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7980–7989. IEEE, 2025.

Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Learning to decompose and disentangle representations for video prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Yixuan Hu, Yuxuan Xue, Simon Klenk, Daniel Cremers, and Gerard Pons-Moll. ControlEvents: Controllable synthesis of event camera data with foundational prior from image diffusion models. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026.

Hongxiang Huang, Hongwei Ren, Xiaopeng Lin, Yulong Huang, Zeke Xie, and Bojun Cheng. DIMOS: Disentangling instance-level moving object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39806–39816, 2026.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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 (ICCV)*, 2021.

Simon Klenk, David Bonello, Lukas Koestler, Nikita Araslanov, and Daniel Cremers. Masked event modeling: Self-supervised pretraining for event cameras. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024. doi: 10.1109/WACV57701.2024.00237. arXiv:2212.10368.

Sébastien Lachapelle, Divyat Mahajan, Ioannis Mitliagkas, and Simon Lacoste-Julien. Additive decoders for latent variables identification and cartesian-product extrapolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Haotian Liu, Guang Chen, Sanqing Qu, Yanping Zhang, Zhijun Li, Alois Knoll, and Changjun Jiang. TMA: Temporal motion aggregation for event-based optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning (ICML)*, 2020a.

Francesco Locatello et al. A commentary on the unsupervised learning of disentangled representations. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020b.

Milton Llera Montero, Casimir J. H. Ludwig, Rui Ponte Costa, Gaurav Malhotra, and Jeffrey S. Bowers. The role of disentanglement in generalisation. In *International Conference on Learning Representations (ICLR)*, 2021.

Sharon S. Musa, Fereshteh Forghani, Harrish Thasarathan, Sonia Joseph, Matthew Kowal, and Konstantinos G. Derpanis. What, where, and how: Probing spatiotemporal representations in video foundation models. *arXiv preprint arXiv:2609.01551*, 2026.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th annual meeting of the association for computational linguistics (ACL)*, 2020. arXiv:2004.07667.

Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. ESIM: an open event camera simulator. In *Conference on Robot Learning (CoRL)*, 2018.

Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. In *International Conference on Learning Representations (ICLR)*, 2022.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN: Decomposing motion and content for video generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. In *International Conference on Learning Representations (ICLR)*, 2017.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Zhexiong Wan, Bin Fan, Le Hui, Yuchao Dai, and Gim Hee Lee. Instance-level moving object segmentation from a single image with events. *International Journal of Computer Vision*, 133: 4042–4063, 2025. doi: 10.1007/s11263-025-02380-z.

Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator (LIA): Learning to animate images via latent space navigation. In *International Conference on Learning Representations (ICLR)*, 2022.

Zhongyao Wang, Wanli Ouyang, Taoyong Cui, and Pheng Ann Heng. Support operation factorization: Compositional readout of frozen vision encoders under controlled interventions. *arXiv preprint arXiv:2608.06174*, 2026.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 989–997, 2019. doi: 10.1109/CVPR.2019.00108.

Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. EventGAN: Leveraging large scale image datasets for event cameras. In *IEEE International Conference on Computational Photography (ICCP)*, 2021.

Yufei Zhu, Hao Chen, Yongjian Deng, and Wei You. Separation for better integration: Disentangling edge and motion in event-based deblurring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14732–14742, 2025.