Entropy-driven Unsupervised Keypoint Representation Learning in Videos

Ali Younes [1] **Simone Schaub-Meyer** [1 2] **Georgia Chalvatzaki** [1 2 3]

Abstract

Extracting informative representations from videos is fundamental for effectively learning various downstream tasks. We present a novel approach for unsupervised learning of meaningful representations from videos, leveraging the concept of image spatial entropy (ISE) that quantifies the per-pixel information in an image. We argue that *local entropy* of pixel neighborhoods and their temporal evolution create valuable intrinsic supervisory signals for learning prominent features. Building on this idea, we abstract visual features into a concise representation of keypoints that act as *dynamic information transmitters*, and design a deep learning model that learns, purely unsupervised, spatially *and* temporally consistent representations *directly* from video frames. Two original information-theoretic losses, computed from local entropy, guide our model to discover consistent keypoint representations; a loss that maximizes the spatial information covered by the keypoints and a loss that optimizes the keypoints' information transportation over time. We compare our keypoint representation to strong baselines for various downstream tasks, *e.g.*, learning object dynamics. Our empirical results show superior performance for our information-driven keypoints that resolve challenges like attendance to static and dynamic objects or objects abruptly entering and leaving the scene. [1]

1. Introduction

Humans are remarkable for their ability to form representations of essential visual entities and store information to effectively learn downstream tasks from experience (Cooper, 1990; Radulescu et al., 2021). Research evidence shows that the human visual system processes visual information in two stages; first, it extracts sparse features of salient objects (Bruce & Tsotsos, 2005); second, it discovers the interrelations of local features for grouping them to find correspondences (Marr, 2010; Kadir & Brady, 2001). For videos with dynamic entities, humans not only focus on dynamic objects, but also on the structure of the background scene if it plays a key role in the information flow (Riche et al., 2012; Borji et al., 2012). Ideally, we want a learning algorithm to extract similar sparse representations that can be useful for various downstream tasks. Notable research works in Computer Vision (CV) and machine learning have proposed different feature representations from pixels (Szeliski, 2010; Harris et al., 1988; Lowe, 2004; Rublee et al., 2011; Mur-Artal et al., 2015). In the deep learning era, convolutional neural network architectures have proven superior to handcrafted features, leading to new approaches for learning representations of Points of Interest (PoI) for tasks like localization and pose estimation (DeTone et al., 2018; Ono et al., 2018; Sarlin et al., 2019; Dusmanu et al., 2019; Sarlin et al., 2020).

Keypoints stand out as sparse PoI (Jiang et al., 2009; Alexe et al., 2010) representing, *e.g.*, objects (Xiongwei et al., 2020), human joints (Kreiss et al., 2019), or structure useful for learning control (Xiong et al., 2021). Many keypoint detectors are trained in a supervised way (Cao et al., 2017). Unsupervised and self-supervised learning can compensate the need for expensive human annotations (Wang et al., 2020; Kim et al., 2019; Yang et al., 2020; Gopalakrishnan et al., 2021; Chen et al., 2019). Current state-of-the-art methods for unsupervised keypoint discovery focus mainly on dynamic entities in videos (Kulkarni et al., 2019; Minderer et al., 2019). Namely, these methods are trained to reconstruct differences between frames, not effectively representing the scene's structure, while not easily disambiguating occlusions or consistently representing abruptly appearing/disappearing objects in a video.

We introduce Maximum Information keypoiNTs (MINT), an information-theoretic approach for unsupervised keypoint representation learning, treating keypoints as "transmitters" of prominent information in a video. Our proposed method relies on *spatial information* computed in local neighborhoods (patches) around potential keypoints. We argue that the image spatial entropy (ISE) (Brink, 1996),

[1]Department of Computer Science, Technische Universität Darmstadt, Germany [2]Hessian.AI [3]Center for Mind, Brain and Behavior (CMBB), Uni. Marburg and JLU Giessen. Correspondence to: Ali Younes <ali.younes@tu-darmstadt.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]https://sites.google.com/view/mint-kp

which quantifies the amount of local information of pixels in an image, and its evolution in a video, provide a strong *inductive bias* for learning keypoint representations related to objects. Early works in CV pointed out the relation of image entropy and object discovery (Kadir & Brady, 2001; Bruce & Tsotsos, 2005; Li et al., 2010), but suffered from the need of filtering and tuning for every new setting to compute an accurate ISE (Razlighi et al., 2009). Contrarily, our deep learning approach benefits from the approximation power of deep convolutional networks that learn nonlinear relations *directly from image frames of a video* leading to spatially *and* temporally consistent representations, that further generalize well. MINT guides the spatio-temporal entropy coverage by the keypoints in a video, relying on an original formulation of unsupervised keypoint discovery with loss functions that *maximize the represented image information entropy* and the *information transportation across frames* by the keypoints, relying on a simple spatial entropy model and regularizers. Imposing spatio-temporal consistency of the represented entities enables MINT to effectively recover scene structure, allowing the subsequent *simultaneous* detection and tracking of objects.

We provide qualitative and quantitative empirical results on four different video-datasets against strong baselines for unsupervised temporal keypoint discovery, unveiling the superior representation power of MINT. To address the challenge of quantitative evaluation of unsupervised keypoint discovery due to the absence of designated datasets, we provide a set of new metrics and a benchmark based on videos from CLEVRER (Yi et al., 2019). Moreover, we provide results on two challenging datasets, MIME (Sharma et al., 2018) and SIMITATE (Memmesheimer et al., 2019), that contain realistic scenes of various difficulties (close-up frames with dynamic interactions vs. high-res wide frames with clutter). We show that MINT economizes the use of keypoints, deactivating excessive ones when the information is well contained, and dynamically activating them to represent new entities entering the scene temporarily. Finally, to demonstrate the suitability of MINT as a representation for control, we devise an imitation learning downstream task on environments from MAGICAL (Toyer et al., 2020).

Contributions. In summary, we introduce: *(1)* an original unsupervised keypoint representation learning approach using information-theoretic measures, via the classical concept of ISE that inspired us to postulate keypoints as information transmitters; *(2)* an entropy layer for computing spatial image entropy efficiently; *(3)* an unsupervised way for representing variable number of entities in videos by switching on/off keypoints for covering spatio-temporal information; and *(4)* a new set of evaluation metrics for an intuitive downstream task for benchmarking the performance of unsupervised temporal keypoint discovery methods.

2. Maximum Information Keypoints

We propose an unsupervised method for keypoint discovery in videos based on information-theoretic principles. Keypoints should adequately represent the scene and the dynamic changes in it. Starting from our original assumption that a keypoint represents the spatial information of a patch of an image frame, we leverage the classical concept of ISE (Brink, 1996; Razlighi & Kehtarnavaz, 2009) to measure the amount of information represented by a keypoint. We argue that keypoints should cover areas in the image that are rich in information, while the number of keypoints should dynamically adapt to represent new information. Finally, keypoints should consistently represent the same information pattern spatio-temporally in a video. With this motivation, we propose to maximize the information covered by the keypoint representation in a video by introducing original losses for unsupervised temporal keypoint discovery. We mainly introduce two losses based on information-theoretic measures: *(1)* An *information maximization loss* that encourages the keypoints to cover areas with high spatial entropy in a single frame. *(2)* An *information transportation loss* that enables the keypoints to represent the same entity over subsequent frames. We present these losses and theoretical analyses supporting their design in the following.

2.1. Image Spatial Entropy (ISE)

Our information-theoretic approach for unsupervised keypoint discovery requires quantifying the amount of information each pixel location in a single frame carries. We leverage the idea of computing the information of patches in an image (local neighborhoods around a keypoint), using the classical concept of ISE (Razlighi & Kehtarnavaz, 2009). ISE provides the pixel-wise information in the spatial domain of the image, and it has been greatly explored in computer vision, *e.g.*, in Markov Random Fields (Razlighi et al., 2009). Images can be considered as lattices where pixels are random variables (Li, 2009). We compute the discrete probability density of a pixel using the statistics of the color intensities in its neighborhood, represented by a normalized histogram of the neighboring pixel values (Sabuncu, 2006). This way of computing ISE (Razlighi & Kehtarnavaz, 2009) assumes that pixels in the image lattice are *i. i. d.*and their entropy is computed using Shannon's definition (Shannon, 2001) based on the probability of each pixel. To compute these histograms efficiently and to derive the final ISE, we developed a computationally optimized entropy layer as detailed in Appendix B.

Our entropy layer estimates the pixel-wise image spatial entropy ISE $\mathcal{H}(I)$ for an RGB input image $I \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3}$, with H being the height and W the width of an image frame with 3 color channels. $\mathcal{H}(I)$ consists of the local entropies $\mathcal{H}(I(x,y))$ computed at each pixel location (x,y) by es-

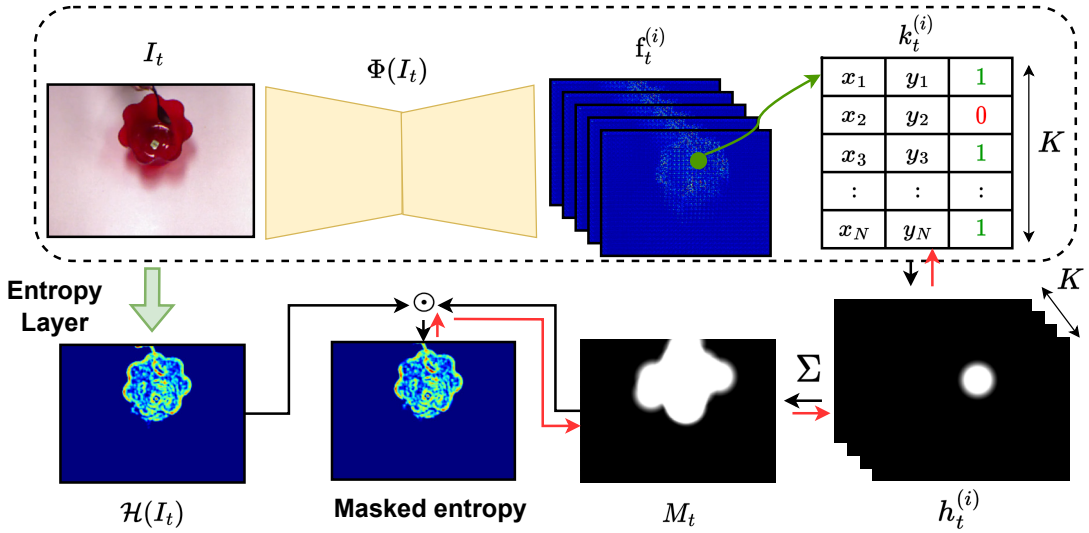

Figure 1. The architecture of our keypoint model $\Phi(I_t)$ (Section 2.2) and the masked entropy (Section 2.2.1). For an input image I_t our model $\Phi(I_t)$ outputs K feature maps $\mathrm{f}_t^{(i)}$ for each keypoint $\mathrm{k}_t^{(i)}$, $i \in \{1, \ldots, K\}$. A heatmap $h_t^{(i)}$ is generated for each keypoint, while the active keypoints are aggregated to form the mask M_t. The entropy layer computes the entropy of the image $\mathcal{H}(I_t)$. Our ME loss maximizes the percentage of the entropy in the masked entropy image. Red arrows show the backward gradient flow. Only the part encircled by the dashed line is used during inference.

timating the entropy of the neighborhood region $R(x, y)$ centered at (x, y), using a normalized histogram-based discrete probability function $p(b, R(x, y))$ for each color value b in the region $R(x, y)$ summed and normalized over the color channels (details in Appendix B). The final per-pixel local entropy is

$$\mathcal{H}(I(x, y)) = -\sum_{b=0}^{255} p(b, R(x, y)) \log(p(b, R(x, y))). \quad (1)$$

2.2. Entropy-driven Keypoint Discovery

We consider keypoints as a compact sparse representation of images, which attend to prominent entities in a scene (Szeliski, 2010). Keypoints should represent distinctive information patterns overlaid on a set of neighboring pixels (patches) in an image frame. We explicitly treat the keypoint (at the center of a patch) as the information transmitter of its neighborhood. Based on ISE (Razlighi & Kehtarnavaz, 2009), we compute the spatial entropy of each keypoint, which allows for developing an end-to-end unsupervised keypoint discovery approach using information-theoretic measures. Maximizing the keypoint information acts as an *intrinsic inductive bias* for learning to represent areas of high entropy. Although a simple model to compute ISE can lead to local entropy overestimation (Brink, 1996), we show empirically (cf. Section 3) that when we regularize the proposed losses effectively, we get useful, well-behaved keypoint representations.

We define a keypoint discovery model $\Phi(I_t)$ (cf. Figure 1), which is a deep neural network that discovers K keypoints $\mathrm{k}_t^{(i)}$, $i \in \{1, ..., K\}$, in an input color image I_t at time t. It outputs K feature maps $\mathrm{f}_t^{(i)}$, each corresponding to one keypoint. The coordinates $(x_i, y_i)_t$ of the respective keypoint $k_t^{(i)}$ are obtained with a spatial soft-argmax (Levine et al., 2016). Besides predicting the coordinates, the model also assigns an activation status $s_t^{(i)} = \{0, 1\}$ per keypoint. The activation status determines whether a keypoint is active ($s_t^{(i)} = 1$) or not ($s_t^{(i)} = 0$) in a specific frame t, allowing the network to decide on the ideal number of active keypoints. Overall, a keypoint is defined by its coordinates and the activation score $\mathrm{k}_t^{(i)} = (x_i, y_i, s^{(i)})_t$. To get the information coverage, we define a differentiable heatmap $h_t^{(i)} \in \mathbb{R}^{H \times W}$ for each i^{th} keypoint by thresholding a distance-based Gaussian $G_t^{(i)}$ centered at the coordinates of the keypoint (details in Appendix A.3). As we want to maximize information coverage by the keypoints spatio-temporally, we need to ensure that both the inter-frame and intra-frame information is sufficiently transmitted. Inspired by information theory, we derive novel losses that allow us to learn information-driven keypoint representations while providing error bounds that theoretically justify the design of those losses (Sabuncu, 2006; Yu et al., 2021).

2.2.1. Maximizing Keypoint Information

With information maximization, we encourage keypoints to represent image regions rich in information (high spatial entropy). We want to enforce maximum collective spatial information coverage by the keypoints for representing all entities in a frame. For that, we introduce two losses: the masked entropy (ME) loss and the masked conditional entropy (MCE) loss.

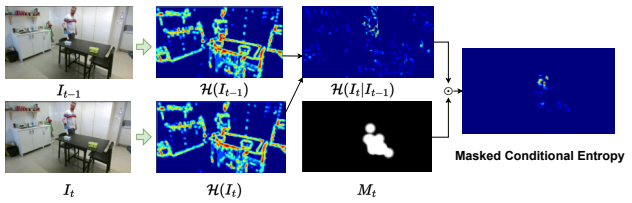

Figure 2. Masked conditional entropy (MCE) computation. Given two consecutive images I_{t-1} and I_t, we extract their spatial entropies $\mathcal{H}(I_{t-1})$ and $\mathcal{H}(I_t)$. The conditional spatial entropy $\mathcal{H}(I_t|I_{t-1})$ depends on the spatial entropy of both images. Multiplying the conditional spatial entropy by the aggregated mask M_t gives the masked conditional entropy image. The MCE loss maximizes the percentage of the masked conditional spatial entropy.

The **ME loss** encourages maximum information coverage by the keypoints in a single frame. We use the heatmap $h_t^{(i)}$ of each keypoint $k_t^{(i)}$ to retrieve the local image information at time t. We filter out inactive keypoints by multiplying the heatmap with the activation status $s_t^{(i)}$. Aggregating all heatmaps gives the aggregated mask $M_t = \min(\sum_i^K h_t^{(i)} \odot s_i^{(t)}, 1)$ (cf. Figure 1). With this masking approach, we can consider keypoints as channels of local information, and thus, we arrive at the following proposition that bounds the spatial information loss by the keypoints' masking of the original image. The bound follows Fano's inequality (Sabuncu, 2006; Scarlett & Cevher, 2019), and proves that maximizing the keypoints' masked spatial entropy indeed lowers the probability of error of the information loss by this keypoint representation.

Proposition 2.1. *Let I_t^M be the masked image at time t, obtained by the operation $I_t^M = I_t \odot M_t$, where $\odot$ denotes the Hadamard (*i.e.*, element-wise) product. Let $\mathcal{B}$ be the "vocabulary" of pixel intensities, and we assume that every pixel in location (x, y) is uniform on $\mathcal{B}$. The average error probability $\bar{P}_\varepsilon$ over all pixels $N = \mathrm{H} \times \mathrm{W}$ of the spatial information approximated by I_t^M* w.r.t. *to the original image I_t can be lower bounded by*

$$\bar{P}_\varepsilon \geq 1 - \frac{\sum_{x,y}(\mathcal{H}(I_t^M(x,y)))}{N \log |\mathcal{B}|} - \frac{\log 2}{\log |\mathcal{B}|} \,. \tag{2}$$

Proof in Appendix C.1. We can assume that the upper bound for the error probability remains 1, because of the activation s of the keypoints, there is a probability that the masked image is "empty", *i.e.*, all keypoints inactive. From Equation (2) we can see that the ME maximization lowers the probability of error. This motivates the practical implementation of the ME loss $\mathcal{L}_{ME}(I_t)$ that optimizes the percentage of the masked entropy over all pixel locations (x, y), $\sum_{x,y} \mathcal{H}(I_t) \odot M_t$ *w.r.t.* the total image entropy $\sum_{x,y} \mathcal{H}(I_t)$

$$\begin{aligned} \mathcal{L}_{ME}(I_t) &= 1 - \frac{\sum_{x,y} \mathcal{H}(I_t) \odot M_t}{\sum_{x,y} \mathcal{H}(I_t)} \\ &= 1 - \frac{\sum_{x,y} \mathcal{H}(I_t) \odot \min(\sum_{i=1}^K h_t^{(i)} \odot s_t^{(i)}, 1)}{\sum_{x,y} \mathcal{H}(I_t)} \,. \end{aligned} \tag{3}$$

The **MCE loss** encourages the keypoints to pay special attention to dynamic entities when the available number of keypoints is insufficient for covering the information in a sequence of frames. The conditional entropy of an image I_t at time t given a reference image I_{t-1} at time $t-1$ measures the information change of pixels, indicating moved objects. Optimizing the conditional entropy $\mathcal{H}(I_t|I_{t-1})$ in a sequence of images encourages the keypoint detector to attend to moving objects (cf. Figure 2). The spatial conditional entropy can be computed by subtracting the reference image entropy from the joint entropy of two images $\mathcal{H}(I_t|I_{t-1}) = \mathcal{H}(I_t, I_{t-1}) - \mathcal{H}(I_{t-1})$, where $\mathcal{H}(I_t, I_{t-1}) \approx \max(\mathcal{H}(I_t), \mathcal{H}(I_{t-1}))$ following

Lemma 2.2. *The joint spatial entropy of two images I_1 and I_2 can be approximated by $\mathcal{H}(I_1(x,y), I_2(x,y)) \approx max(\mathcal{H}(I_1(x,y)), \mathcal{H}(I_2(x,y))), \forall(x,y)$, since the per pixel maximum of the marginal entropies is a lower bound of the joint entropy.*

Proof in Appendix C.2. Accordingly, we can bound the information loss by the keypoints in a sequence of frames.

Corollary 2.3. *Following Proposition 2.1, we can bound the average probability of error $\bar{P}_\varepsilon^{cond}$ of the conditioned masked images between timestep $t-1$ and t as*

$$\bar{P}_\varepsilon^{cond} \geq 1 - \frac{\sum_{x,y} \mathcal{H}(I_t^M(x,y)|I_{t-1}^M(x,y))}{N \log |\mathcal{B}|} - \frac{\log 2}{\log |\mathcal{B}|} \,. \tag{4}$$

Following Equation (4), we observe that the MCE maximization lowers the probability of error of the conditional spatial information loss between frames, leading to the practical implementation of the MCE loss, similarly to the ME loss. The MCE loss $\mathcal{L}_{MCE}(I_t, I_{t-1})$ maximizes the percentage of total masked conditional entropy $\sum_{x,y} \mathcal{H}(I_t|I_{t-1}) \odot M_t$ to the total conditional entropy $\sum_{x,y} \mathcal{H}(I_t|I_{t-1})$

$$\mathcal{L}_{MCE}(I_t, I_{t-1}) = 1 - \frac{\sum_{x,y} \mathcal{H}(I_t|I_{t-1}) \odot M_t}{\sum_{x,y} \mathcal{H}(I_t|I_{t-1})} \,. \tag{5}$$

2.2.2. Maximizing Keypoint Information Transportation

Keypoints should transmit information about the same entity over time. Temporal consistency means aligning each keypoint to the same information pattern across its occurrences. Thus, we propose the operation of information

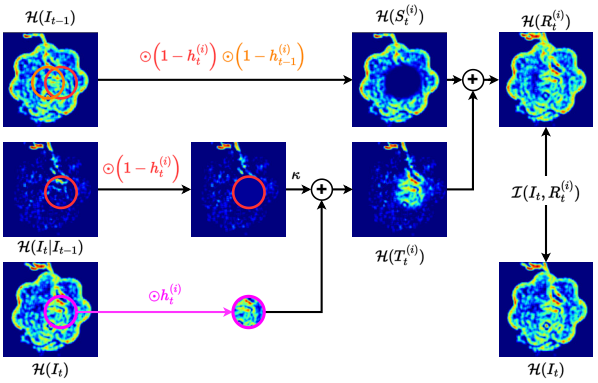

Figure 3. IT for keypoint $k^{(i)}$. Removing the heatmap masks of the i^{th} keypoint at times $t-1$ (orange circle) and t (red cricle) from the spatial entropy $\mathcal{H}(I_{t-1})$ of the image I_{t-1} gives the source entropy $\mathcal{H}(S_t^{(i)})$. Implanting the local entropy of the keypoint at time t from the current frame $\mathcal{H}(I_t)$ (magenta circle) into the conditional entropy $\mathcal{H}(I_t|I_{t-1})$ (weighted by κ) after removing the heatmap mask of the keypoint at time t (red cricles) gives the target spatial entropy $\mathcal{H}(T_t^{(i)})$. The reconstructed information after transportation $\mathcal{H}(R_t^{(i)})$ is the sum of target and source entropy. The objective of IT is to maximize the mutual information between the reconstructed entropy and the entropy of the current frame $\mathcal{I}(I_t, R_t^{(i)})$.

transportation (IT) based on ISE, contrarily to methods that rely on image reconstruction performing feature transportation (Kulkarni et al., 2019).

In a temporal sequence of frames, we can perform keypoint IT by *reconstructing the image spatial entropy* of the current frame $\mathcal{H}(I_t)$ using the image entropy of the previous frame $\mathcal{H}(I_{t-1})$ (cf. Figure 3). Let's consider the i^{th} keypoint at time step t (coordinates are omitted for avoiding verbosity). Its associated heatmap $h_t^{(i)}$ is a mask on the entropy image that allows localizing the spatial information conveyed by the i^{th} keypoint. We can construct a *source entropy image* $\mathcal{H}(S_t^{(i)})$ by subtracting the local entropy of the i^{th} keypoint in frames $t-1$ and t from the entropy image $\mathcal{H}(I_{t-1})$, *i.e.*, $\mathcal{H}(S_t^{(i)}) = \mathcal{H}(I_{t-1}) \odot (1-h_{t-1}^{(i)}) \odot (1-h_t^{(i)})$. The conditional spatial entropy of the two frames $\mathcal{H}(I_t|I_{t-1})$ represents the amount of pixel-wise information needed to quantify the information of $\mathcal{H}(I_t)$ given $\mathcal{H}(I_{t-1})$. Implanting the keypoint's spatial entropy covered by $h_t^{(i)}$ onto the conditional image entropy $\mathcal{H}(I_t|I_{t-1})$, that contains all conditional information except for the information transmitted by the i^{th} keypoint, forms the target image entropy $\mathcal{H}(T_t^{(i)}) = \mathcal{H}(I_t) \odot (h_t^{(i)}) + \kappa\mathcal{H}(I_t|I_{t-1}) \odot (1-h_t^{(i)})$.[2] The reconstruction of the image entropy $\mathcal{H}(I_t)$ results from the pixel-wise sum of the source and target image entropies $\mathcal{H}(R_t^{(i)}) = \mathcal{H}(S_t^{(i)}) + \mathcal{H}(T_t^{(i)})$. The transportation loss is computed independently per keypoint, and enforces each keypoint to consistently represent the same information pattern spatio-temporally. The reconstruction process of our IT leads us to the following proposition, showing that maximizing the mutual information (MI) between the per keypoint reconstructed information and the original image entropy lowers the probability of error due to information loss.

Proposition 2.4. *Following Fano's inequality (Sabuncu, 2006; Scarlett & Cevher, 2019), we prove that the average error probability of the transportation of the i^{th} keypoint $P_e^{IT(i)}$, assuming each keypoint transportation independently, is lower bounded by*

$$\bar{P}_e^{IT(i)} \geq 1 - \frac{\sum_{x,y} \mathcal{I}(I_t(x,y), R_t^{(i)}(x,y))}{N \log |\mathcal{B}|} - \frac{\log 2}{\log \mathcal{B}} \,. \quad (6)$$

Proof in Appendix C.3. From Equation (6), we deduce that for optimizing the i^{th} keypoint's IT, we should maximize the MI $\mathcal{I}(I_t, R_t^{(i)})$. This motivates our practical implementation of the IT loss for all keypoints, and we construct the IT loss through the difference $\mathcal{H}(I_t) - \mathcal{I}(I_t, R_t^{(i)})$ normalized by the area of the heatmap A_h (equal for all keypoints). Minimizing $\mathcal{H}(I_t) - \mathcal{I}(I_t, R_t^{(i)})$ maximizes MI, as dictated by Proposition 2.4. We found that normalizing with A_h helps having a better loss scale. We also regularize the excessive keypoint movement by minimizing the norm of the distance traveled by each keypoint $d_t^{(i)} = ||(x_i, y_i)_t - (x_i, y_i)_{t-1}||_2^2$ (scaled by a weight m_d). The practical implementation of the IT loss for all keypoints becomes

$$\mathcal{L}_{\text{IT}}(I_t, I_{t-1}) = \sum_{i=1}^{K} \frac{\sum_{x,y} \mathcal{H}(I_t) - \mathcal{I}(I_t, R_t^{(i)})}{A_h} + m_d \cdot d_t^{(i)} \,. \quad (7)$$

2.2.3. The MINT loss & Auxiliary losses

The **overlapping loss** provides an auxiliary supervisory signal that spreads the keypoints over the image, encouraging them to cover distinctive regions. The sum of the Gaussians $G_t^{(i)}$ (cf. Appendix A.3) around the keypoints $k_t^{(i)}$ helps to estimate their overlap. The overlapping loss,

$$\mathcal{L}_o = \frac{1}{K} \min(\max(\sum_i^K G_t^{(i)}) - \beta, 0) \,, \quad (8)$$

minimizes the maximum of the aggregated Gaussians normalized by the number of keypoints K with a lower bound β to allow some occlusions and avoid over-penalization.

The **active status loss** encourages the model to deactivate unnecessary keypoints, *i.e.*, setting the status s_t to 0, by minimizing the normalized sum of active keypoints while maximizing ME. The interplay of the losses allows the method to eventually reach a trade-off between the number of active keypoints and covered spatial entropy.

[2] The factor $\kappa \leq 1$ encourages the network to concentrate more on transportation than reconstruction.

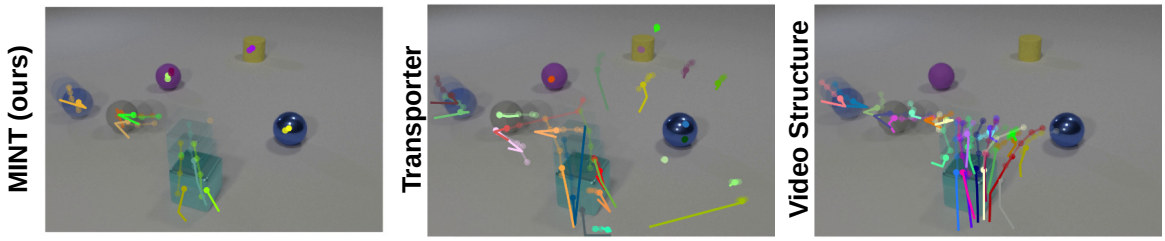

Figure 4. Qualitative results on CLEVRER for Task I (object detection and tracking) and Task II (learning dynamics). Our method is able to assign keypoints to all objects, independently of whether they move or not, and follows their trajectory. The number of keypoints is dynamically adjusted to the number of objects. Future states, the predicted keypoint and trajectories, are transparent.

Table 1. Quantitative evaluation of keypoint detection and tracking on CLEVRER (Yi et al., 2019).

Method	**DOP** ⇑	**TOP** ⇑	**UAK** ⇓	**RAK** ⇓
MINT w/o Reg. (ours)	**0.918 ± 0.073**	**0.897 ± 0.078**	6.793 ± 1.956	2.478 ± 0.865
MINT (ours)	*0.855 ± 0.118*	*0.838 ± 0.121*	**0.889 ± 0.639**	**1.123 ± 0.448**
Transporter	0.787 ± 0.113	0.745 ± 0.119	18.417 ± 1.639	1.157 ± 0.323
Transporter-modified	0.832 ± 0.107	0.794 ± 0.114	16.267 ± 2.349	1.764 ± 0.671
Video Structure	0.567 ± 0.256	0.543 ± 0.253	18.104 ± 3.538	1.922 ± 0.652

Table 2. Prediction success rate on CLEVRER (Yi et al., 2019).

Method	1-step prediction	2-steps prediction	3-steps prediction
MINT (ours)	**0.844 ± 0.116**	**0.827 ±0.126**	**0.811±0.132**
Transporter	0.746 ± 0.116	0.716 ± 0.120	0.692 ± 0.122
Transporter-modified	0.814 ± 0.099	0.791 ± 0.106	0.769 ± 0.110
Video Structure	0.734 ± 0.124	0.719 ± 0.125	0.699 ± 0.127

The active status loss optimizes

$$\mathcal{L}_s = \frac{1}{K}\sum_{i}^{K} s_t^{(i)} \ . \tag{9}$$

The overall **MINT loss** $\mathcal{L}_{MINT}$ is a weighted combination of all losses (with a dedicated weight λ per loss), with the weight of the status loss reversed to schedule it according to the percentage of ME,

$$\begin{aligned}\mathcal{L}_{MINT} =& \lambda_{ME}\mathcal{L}_{ME} + \lambda_{MCE}\mathcal{L}_{MCE} + \lambda_{IT}\mathcal{L}_{IT} \\ &+ \lambda_o\mathcal{L}_o + (1-\mathcal{L}_{ME})\lambda_s\mathcal{L}_s \ .\end{aligned} \tag{10}$$

Further information about the hyperparameters are available in Appendix E.2.

3. Experiments

We evaluate MINT on four datasets ranging from videos of synthetic objects – CLEVRER (Yi et al., 2019) and MAGICAL (Toyer et al., 2020) – to realistic human video demonstrations – MIME (Sharma et al., 2018) and SIMITATE (Memmesheimer et al., 2019). Our experiments show the efficacy of our method as a representation for different tasks, and we provide quantitative results *w.r.t.* evaluation metrics (for object detection and tracking on CLEVRER) for several downstream tasks (learning dynamics on CLEVRER, imitation learning on MAGICAL), and qualitative results on the challenging datasets of MIME and SIMITATE.

We compare against baselines for unsupervised end-to-end keypoint representation learning from *videos*. To the best of our knowledge, the only baselines in this context (cf. Section 4) are **Transporter** (Kulkarni et al., 2019) and **Video Structure** (Minderer et al., 2019). Additionally, we include **Transporter-modified**, a modified version with a smaller receptive field that we designed for comparison. Further, we compare to MINT without the regularization terms (**MINT w/o Reg.**), and an end-to-end CNN-based feature extraction. We report statistics for all quantitative results over 5 seeds. An extensive ablation study of MINT is provided in Appendix E.1 and baselines are discussed in Appendix E.3.

Downstream task I: *Object detection and tracking*. Capturing scene structure requires detecting all objects in an image, while object tracking is essential for representing the scene's dynamics. MINT can successfully train a spatio-temporally consistent keypoint representation on videos, leading to its natural application for object (static/dynamic, appearing/disappearing) detection and tracking.
We use CLEVRER (Yi et al., 2019), a dataset for visual reasoning with complete object annotations, containing videos with static and dynamic objects, with good variability in scenes, as a testbed. To quantitatively assess the performance of MINT, we developed evaluation metrics for CLEVRER. We propose the **percentage of the detected object (DOP)** and the **percentage of tracked objects (TOP)** as two metrics, with higher values corresponding to better keypoint detection and tracking. A keypoint detects an object if it lies on its mask, and tracks it, if it detects the same

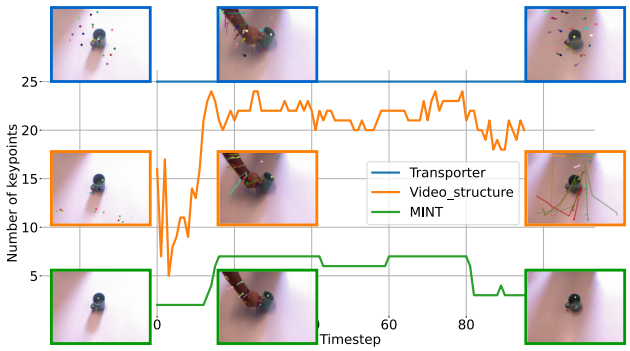

Figure 5. Come-and-go scenario in MIME. The hand enters after the start of the video and departs before the end. We plot the number of active keypoints *w.r.t.* time. Transporter (Kulkarni et al., 2019) has a fixed number of keypoints. Video structure (Minderer et al., 2019) increases the number of active keypoints when the hand appears, but struggles when it disappears. MINT uses a suitable number of keypoints.

object in two consecutive frame. Assigning keypoints to areas already represented by other keypoints or empty spaces signals bad keypoint detection. To evaluate these cases, we define two additional metrics for the **redundant keypoint assignment (RAK)** and **unsuccessful keypoint assignment (UAK)**, with lower values corresponding to better detection. The metrics are described in detail in Appendix D.

We train all keypoint detectors on a subset of 20 videos from CLEVRER and test them on 100. The train-test split emulates a low-data regime and tests the methods' generalization abilities. As seen in Table 1, MINT w/o Reg. detects more objects (DOP) and tracks them better (TOP), showing the benefit of our information-theoretic losses. The proposed MINT model exhibits the best trade-off between superior performance against the baselines on all metrics, and better handling of keypoint assignment (UAK and RAK) than MINT w/o Reg. This is due to the computation of the supervisory entropy signal being overestimated, but the regularizers balance this effect. See visual comparisons in Figure 4 or in the video results,[3] and more discussion about the ablations in Appendix E.1.

Downstream task II: *Learning dynamics*. Proper object detection allows us to learn the underlying dynamics that evolve in a scene. We test the representation power of the discovered keypoints by training a prediction model (*i.e.*, a model predicting the next state of the objects) using the pre-trained keypoint detectors from Task I (using the best seed for each method). The prediction model treats the keypoints as graph nodes in an Interaction Network (IN) (Battaglia et al., 2016) to model the relational dynamics (cf. Appendix E.4). We train the prediction model to forecast the future positions of the keypoints given a history

[3]Videos on the website https://sites.google.com/view/mint-kp.

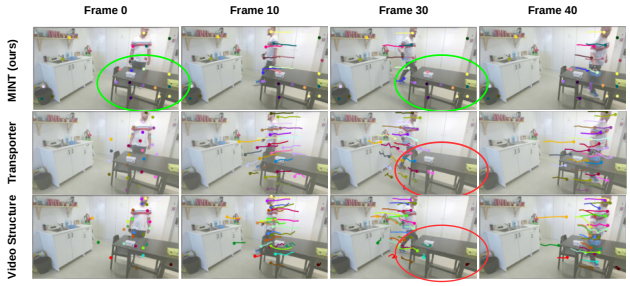

Figure 6. Crowded scene from SIMITATE with a human moving in a room. All methods can track the human successfully, but only MINT can keep keypoints on the static objects consistently (green ellipses), while the baselines lose track of them (red ellipses).

of four-time steps. We compare the prediction against the ground truth position of the object in the predicted frame using CLEVRER (Yi et al., 2019). We report in Table 2 the ratio of successfully predicted objects (*i.e.*, a predicted keypoint lying on the same object in the next frame) to the ground truth number of objects in the next time step. The comparison demonstrates that keypoints detected by our method represent the scene better than the baselines and help to predict the next state. Figure 4 shows the prediction performance using different keypoint detectors.

Downstream task III: *Object discovery in realistic scenes*. Our method addresses challenging aspects beyond synthetic datasets. We evaluate the keypoint detectors on two additional datasets: (1) MIME (Sharma et al., 2018): a collection of close-up videos of human hands manipulating objects, and (2) SIMITATE (Memmesheimer et al., 2019): a video dataset of humans performing manipulation tasks in wide-view cluttered scenes. Since no annotations are provided in these datasets, we perform only qualitative analysis.

In MIME, the human hand enters and leaves the scene abruptly, allowing to evaluate MINT in come-and-go scenarios as shown in Figure 5. MINT only activates the necessary number of keypoints, while Transporter uses a static number, and Video structure fails to deactivate the excessive keypoints when the hand disappears. Figure 5 shows the number of active keypoints over time, revealing our method's superior performance for the number of keypoints and the qualitative representation of objects in the scene.

The qualitative results for SIMITATE in Figure 6, on the other hand, show that only MINT can disambiguate between static and dynamic objects, tracking human movement, while maintaining the structure of the keypoints relatively constant over the static objects. The baselines rely on reconstructing the movement, failing to represent the scene's structure. The qualitative results reveal the need for the conditional entropy loss (forcing attention on moving objects when the number of available keypoints is restricted) and the information transportation loss (ensuring the spatio-temporal consistency). We further include ablation study

Table 3. Average score for imitation learning on MAGICAL (Toyer et al., 2020). Higher values are better.

Method	MoveToRegion		MoveToCorner		MakeLine	
	Demo	TestJitter	Demo	TestJitter	Demo	TestJitter
MINT (ours)	**1.00 ± 0.00**	**0.86 ± 0.31**	**1.00 ± 0.00**	**0.80 ± 0.34**	**0.2 ± 0.22**	0.06 ± 0.14
CNN	**1.00 ± 0.00**	0.84 ± 0.32	0.74 ± 0.35	0.30 ± 0.38	0.00 ± 0.00	0.01 ± 0.06

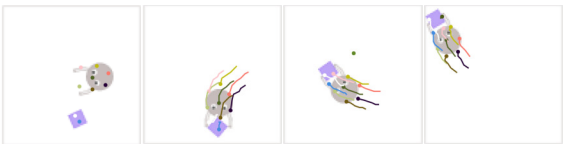

Figure 7. Keypoint-based imitation learning in MAGICAL. The figure showcases the MoveToCorner environment, where our agent's objective is to move the purple object to the top-left corner. Our approach, MINT, enables the agent to observe keypoints that describe the environment and predict the next action accurately. We demonstrate the effectiveness of our method by training the agent to imitate expert trajectories. The visualization overlays MINT keypoints on sample frames from a successful rollout.

results on both realistic datasets in Appendix E.1.

Downstream task IV: *Imitation learning*. Imitation learning from video frames is a long-standing challenge for control. Keypoints can define a low-dimensional representation that could reduce the computational burden considerably. In this experiment, we investigate the suitability of our keypoint representation for control tasks, like imitation learning in MAGICAL (Toyer et al., 2020) (cf. Figure 7). We first pretrain MINT on 24 demonstration videos from different tasks. Then, we fix the keypoint detector and train an agent to mimic the demonstrated actions, using an IN (Battaglia et al., 2016), followed by a fully-connected layer that decodes the actions (cf. Appendix E.4). The agent uses as input the observed keypoints from four frames. We also found it useful to predict the next state as an auxiliary task. We compare the MINT-based agent against an agent that uses a CNN to extract features directly from pixels. The CNN agent is trained from scratch for each environment (cf. Appendix E.5). We consider three environments with different levels of difficulty; **MoveToRegion**: move an agent to a specific region, only the agent is involved (easy). **MoveToCorner**: move an object to the top-left corner, one object and the agent are involved (medium). **MakeLine**: place multiple objects in a line, four objects and the agent are involved (hard). We evaluate the learned policy on environment instances from demonstrations (Demo) and randomly initialized (TestJitter). The results in Table 3 reveal that a pretrained keypoint model with MINT is suitable for control, achieving comparable or even superior performance to a task-specific CNN-based agent (cf. Appendix E.5 for more details).

We hypothesize that there is still room for improvement to unleash the potential of MINT sparse keypoint representation for control. One viable option is to use a more expressive network architecture (e.g., graph attention network) that may provide a better representation of the keypoint-induced graph. Another promising direction is to boost a reinforcement learning agent with the imitation learning policy. However, this is out of the scope of the current work.

Limitations. Our method relies on ISE after filtering high-frequency color changes. As a result, the method has difficulties in recognizing transparent objects and objects with the same color as the background. We plan to investigate the integration of implicit representation learning to counteract this issue. Another limitation is the interpretation of the keypoints in the three-dimensional space. The current method operates on images and does not provide 3D information. Adding depth information or extending to a multi-view setting are options for future improvements. Our method can use high-level features from a pretrained encoder to estimate the entropy, which may solve the current limitations. However, while we treated pixels as discrete random variables with RGB values, high-level features lie on a continuous latent space and, therefore, would require variational inference techniques. This direction would require a new treatment compared to our current analysis that relies on discrete probability theory, which goes beyond the scope of the current paper that lays the ground for such future work.

4. Related Work

Representation learning. The idea of extracting sparse feature representations of high-dimensional visual data is dominant in computer vision and machine learning research (Harris et al., 1988; Lowe, 2004), and connects to the functioning of the human visual system (Marr, 2010). Such sparse representations are generally known as PoI, which are 2D locations that are stable and repeatable under various lighting conditions and viewpoints (DeTone et al., 2018). Traditional geometric computer vision methods relied on the extraction of hand-crafted feature descriptors (Lowe, 2004; Rublee et al., 2011) for tasks like localization (Schmid et al., 2000; Mur-Artal et al., 2015). In the deep learning era, CNN architectures have proven superior to handcrafted features (Yi et al., 2016; DeTone et al., 2018; Song et al., 2020; Zheng et al., 2017). Deep approaches extract clouds of PoI that are useful for correspondence searching in visual place recognition from different viewpoints (Hausler et al., 2021), or pose-estimation for control (Florence et al., 2019). Related to our method are object-centric approaches (Singh et al., 2021; Locatello et al., 2020; Dittadi et al., 2022),

which aim to learn abstract representations for objects in a scene. Our approach to keypoint discovery, alongside our metrics, are directed at learning and evaluating keypoints as an object-centric representation.

Image information entropy. Our work draws inspiration from classical approaches in saliency detection in images that use local information to detect salient entities (Kadir & Brady, 2001; Bruce & Tsotsos, 2005; Fritz et al., 2004; Renninger et al., 2004; Borji & Itti, 2012). Bruce & Tsotsos (2005) proposes that regions with high self-information typically correspond to salient objects, and Alexe et al. (2010) quantified objectiveness by self-information approximated via center-surround feature differences. Extracting sparse feature representations of high-dimensional visual data is also dominant in CV (Harris et al., 1988; Lowe, 2004; DeTone et al., 2018). Traditional CV methods relied on the extraction of hand-crafted feature descriptors (Lowe, 2004; Rublee et al., 2011; Schmid et al., 2000; Mur-Artal et al., 2015). Notably, image information entropy has additional applications in various CV problems, like image registration (Sabuncu, 2006), active vision (Ferraro et al., 2002), medical image analysis (Hržić et al., 2019), nuclear detection (Hamahashi et al., 2005), image compression (Minnen et al., 2018), and image randomness (Wu et al., 2013). Our method proposes information-theoretic losses based on the ISE (Brink, 1996; Razlighi & Kehtarnavaz, 2009). The use of ISE was prevalent in CV applications for image reconstruction (Gull & Daniell, 1978) and in Markov Random Fields (Li, 2009; Razlighi et al., 2009).

Keypoint learning. Keypoints represent a class of PoI that have a semantic entity, *e.g.*, representing objects (Duan et al., 2019), or human joints (Cao et al., 2017; McNally et al., 2022), but most methods rely on explicit annotations of keypoint locations. Related to our work are unsupervised methods for keypoint detection. Jakab et al. (2018) use an autoencoder architecture with a differentiable keypoint bottleneck trained on the difference between a source and a target image, trying to restrict the information flow. MINT also uses a differentiable keypoint representation, but it operates on the output of an hourglass architecture. Our results suggest that learning to redistribute the information after compression is beneficial for keypoint discovery (Newell et al., 2016). Minderer et al. (2019) use a similar architecture as Jakab et al. (2018) but operate on video sequences for detecting keypoints, using the intensity of the bottleneck heatmap as an indicator of the importance of a keypoint. Setting up a threshold on the intensity is challenging and domain-specific. Contrarily, we learn a binary classification of active/inactive keypoints and optimize the number of keypoints used in every frame. Kulkarni et al. (2019) propose feature transportation in the keypoint bottleneck of Jakab et al. (2018) before reconstruction. MINT performs information transportation and waives the need for image reconstruction, which would require an additional appearance encoder and a reconstruction decoder from keypoints. Gopalakrishnan et al. (2021) devised a three-stage architecture that first learns a spatial feature embedding, then solves a local spatial prediction task related to object permanence, and finally converts error maps into keypoints. However, this method operates on single images and does not consider the temporal consistency of the extracted keypoints. MINT, on the other hand, is a parameter-efficient single model that provides spatio-temporally consistent keypoints in videos. To the best of our knowledge, (Kulkarni et al., 2019; Minderer et al., 2019) are the only comparable methods that set the state-of-the-art for unsupervised temporal keypoint discovery, also proven by their successful adoption for behavior recognition (Sun et al., 2022), causal discovery (Li et al., 2020), and control (Bechtle et al., 2023).

In Appendix G, we also discuss the use of information-theoretic measures in deep learning.

5. Conclusion

We presented MINT, a novel unsupervised keypoint representation learning method from videos using entropy-based intrinsic supervisory signals. We treat keypoints as *transmitters* of information, and defined a deep model that learns consistent keypoint representations from video frames, thanks to two original losses; an information maximization loss and an information transportation loss. These losses drive the keypoints to cover areas of high spatial entropy, while ensuring spatio-temporal keypoint consistency. Auxiliary losses enable MINT to learn to switch on/off keypoints when required to preserve the information flow. Our experimental evaluation showcased the superior performance of our method on various downstream tasks, ranging from object detection to dynamics prediction and imitation learning. Moreover, we showed qualitatively that MINT tackles key challenges in realistic scenarios, such as attending to static and dynamic objects and handling appearing/disappearing entities. Overall, we proposed a method for learning reasonable keypoint representations from videos purely unsupervised, with promising results for future applications.

Acknowledgements

This work is funded by the DFG Emmy Noether Programme No. CH 2676/1-1, and the Hessian.AI through the Connectom Fund on Lifelong Explainable Robot Learning. The project has also been supported in part by the State of Hesse through the cluster project "The Third Wave of Artificial Intelligence (3AI). We would like to thank Stefan Roth, Carlo D'Eramo, and An Thai Le for the discussions and comments. We would also like to thank the anonymous ICML reviewers for their valuable feedback on the manuscript.

References

Alexe, B., Deselaers, T., and Ferrari, V. What is an object? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 73–80, 2010. 1, 9

Avi-Aharon, M., Arbelle, A., and Raviv, T. R. DeepHist: Differentiable joint and color histogram layers for image-to-image translation. *arXiv preprint arXiv:2005.03995*, 2020. 18, 29

Barroso-Laguna, A., Riba, E., Ponsa, D., and Mikolajczyk, K. Key.Net: Keypoint detection by handcrafted and learned cnn filters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5836–5844, 2019. 26

Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al. Interaction networks for learning about objects, relations and physics. *Advances in Neural Information Processing Systems*, 29, 2016. 7, 8, 27

Bechtle, S., Das, N., and Meier, F. Multimodal learning of keypoint predictive models for visual object manipulation. *IEEE Transactions on Robotics*, 2023. 9

Borji, A. and Itti, L. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2012. 9

Borji, A., Sihite, D. N., and Itti, L. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 22(1):55–69, 2012. 1

Brink, A. Using spatial information as an aid to maximum entropy image threshold selection. *Pattern Recognition Letters*, 17(1):29–36, 1996. 1, 2, 3, 9, 17

Bruce, N. and Tsotsos, J. Saliency based on information maximization. *Advances in Neural Information Processing Systems*, 18, 2005. 1, 2, 9

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017. 1, 9

Chen, P.-Y., Liu, A. H., Liu, Y.-C., and Wang, Y.-C. F. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2624–2632, 2019. 1

Cooper, L. A. Mental representation of three-dimensional objects in visual problem solving and recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(6):1097, 1990. 1

DeTone, D., Malisiewicz, T., and Rabinovich, A. SuperPoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition workshops*, pp. 224–236, 2018. 1, 8, 9

Dittadi, A., Papa, S. S., De Vita, M., Schölkopf, B., Winther, O., and Locatello, F. Generalization and robustness implications in object-centric learning. In *International Conference on Machine Learning*, pp. 5221–5285. PMLR, 2022. 8

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. CenterNet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569–6578, 2019. 9

Dusmanu, M., Rocco, I., Pajdla, T., Pollefeys, M., Sivic, J., Torii, A., and Sattler, T. D2-Net: A trainable cnn for joint description and detection of local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8092–8101, 2019. 1

Ewerton, M., Martínez-González, A., and Odobez, J.-M. An efficient image-to-image translation hourglass-based architecture for object pushing policy learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3478–3484. IEEE, 2021. 15

Ferraro, M., Boccignone, G., and Caelli, T. Entropy-based representation of image information. *Pattern Recognition Letters*, 23(12):1391–1398, 2002. 9

Florence, P., Manuelli, L., and Tedrake, R. Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2):492–499, 2019. 8

Fritz, G., Seifert, C., Paletta, L., and Bischof, H. Attentive object detection using an information theoretic saliency measure. In *International Workshop on Attention and Performance in Computational Vision*, pp. 29–41. Springer, 2004. 9

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010. 15

Good, I. J. Rational decisions. In *Breakthroughs in statistics*, pp. 365–377. Springer, 1992. 29

Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press, 2016. 15, 29

Gopalakrishnan, A., van Steenkiste, S., and Schmidhuber, J. Unsupervised object keypoint learning using local spatial predictability. In *International Conference on Learning*

Representations, 2021. URL https://openreview.net/forum?id=GJwMHetHc73. 1, 9

Gull, S. F. and Daniell, G. J. Image reconstruction from incomplete and noisy data. *Nature*, 272(5655):686–690, 1978. 9

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018. 29

Hamahashi, S., Onami, S., and Kitano, H. Detection of nuclei in 4d nomarski dic microscope images of early caenorhabditis elegans embryos using local image entropy and object tracking. *BMC bioinformatics*, 6(1): 1–15, 2005. 9

Harris, C., Stephens, M., et al. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, volume 15, pp. 10–5244. Alvety Vision Club, 1988. 1, 8, 9

Hausler, S., Garg, S., Xu, M., Milford, M., and Fischer, T. Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14141–14152, 2021. 8

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 29

Hržić, F., Štajduhar, I., Tschauner, S., Sorantin, E., and Lerga, J. Local-entropy based approach for x-ray image segmentation and fracture detection. *Entropy*, 21(4):338, 2019. 9

Jakab, T., Gupta, A., Bilen, H., and Vedaldi, A. Unsupervised learning of object landmarks through conditional image generation. *Advances in Neural Information Processing Systems*, 31, 2018. 9

Jiang, Y.-G., Yang, J., Ngo, C.-W., and Hauptmann, A. G. Representations of keypoint-based semantic concept detection: A comprehensive study. *IEEE Transactions on Multimedia*, 12(1):42–53, 2009. 1

Kadir, T. and Brady, M. Saliency, scale and image description. *International Journal of Computer Vision*, 45(2): 83–105, 2001. 1, 2, 9

Kim, Y., Nam, S., Cho, I., and Kim, S. J. Unsupervised keypoint learning for guiding class-conditional video prediction. *Advances in Neural Information Processing Systems*, 32, 2019. 1

Kong, L., de Masson d'Autume, C., Yu, L., Ling, W., Dai, Z., and Yogatama, D. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2019. 29

Kreiss, S., Bertoni, L., and Alahi, A. PifPaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11977–11986, 2019. 1

Kulkarni, T. D., Gupta, A., Ionescu, C., Borgeaud, S., Reynolds, M., Zisserman, A., and Mnih, V. Unsupervised learning of object keypoints for perception and control. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 5, 6, 7, 9, 15, 26

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 3, 15

Li, J., Li, B., and Lu, Y. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 1503–1511, 2022. 29

Li, S. Z. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009. 2, 9

Li, Y., Zhou, Y., Yan, J., Niu, Z., and Yang, J. Visual saliency based on conditional entropy. In *Computer Vision–ACCV 2009: 9th Asian Conference on Computer Vision, Xi'an, September 23-27, 2009, Revised Selected Papers, Part I 9*, pp. 246–257. Springer, 2010. 2

Li, Y., Torralba, A., Anandkumar, A., Fox, D., and Garg, A. Causal discovery in physical systems from videos. *Advances in Neural Information Processing Systems*, 33: 9180–9192, 2020. 9, 29

Linsker, R. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988. 18, 29

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 8

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60 (2):91–110, 2004. 1, 8, 9

Marr, D. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010. 1, 8

McNally, W., Vats, K., Wong, A., and McPhee, J. Rethinking keypoint representations: Modeling keypoints and poses as objects for multi-person human pose estimation. In *European Conference on Computer Vision*, pp. 37–54. Springer, 2022. 9

Memmesheimer, R., Kramer, I., Seib, V., and Paulus, D. Simitate: A hybrid imitation learning benchmark. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5243–5249. IEEE, 2019. 2, 6, 7, 24

Minderer, M., Sun, C., Villegas, R., Cole, F., Murphy, K. P., and Lee, H. Unsupervised learning of object structure and dynamics from videos. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 6, 7, 9, 15, 24, 29

Minnen, D., Toderici, G., Singh, S., Hwang, S. J., and Covell, M. Image-dependent local entropy models for learned image compression. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 430–434. IEEE, 2018. 9

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015. 1, 8, 9

Murphy, K. P. *Probabilistic machine learning: an introduction*. MIT press, 2022. 19

Newell, A., Yang, K., and Deng, J. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pp. 483–499. Springer, 2016. 9, 15

Ono, Y., Trulls, E., Fua, P., and Yi, K. M. LF-Net: Learning local features from images. *Advances in Neural Information Processing Systems*, 31, 2018. 1

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 29

Pardo, L. *Statistical inference based on divergence measures*. Chapman and Hall/CRC, 2018. 29

Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3): 1065–1076, 1962. 29

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019. 16, 17

Peters, J., Mulling, K., and Altun, Y. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010. 29

Radulescu, A., Shin, Y. S., and Niv, Y. Human representation learning. *Annual Review of Neuroscience*, 44(1): 253–273, 2021. 1

Rakelly, K., Gupta, A., Florensa, C., and Levine, S. Which mutual-information representation learning objectives are sufficient for control? *Advances in Neural Information Processing Systems*, 34:26345–26357, 2021. 29

Razlighi, Q. and Kehtarnavaz, N. A comparison study of image spatial entropy. In *Visual Communications and Image Processing 2009*, volume 7257, pp. 615–624. SPIE, 2009. 2, 3, 9, 17

Razlighi, Q. R., Kehtarnavaz, N., and Nosratinia, A. Computation of image spatial entropy using quadrilateral markov random field. *IEEE Transactions on Image Processing*, 18 (12):2629–2639, 2009. doi: 10.1109/TIP.2009.2029988. 2, 9

Renninger, L., Coughlan, J., Verghese, P., and Malik, J. An information maximization model of eye movements. *Advances in Neural Information Processing Systems*, 17, 2004. 9

Riche, N., Mancas, M., Culibrk, D., Crnojevic, V., Gosselin, B., and Dutoit, T. Dynamic saliency models and human attention: A comparative study on videos. In *Asian Conference on Computer Vision*, pp. 586–598. Springer, 2012. 1

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. ORB: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571. IEEE, 2011. 1, 8, 9

Sabuncu, M. R. *Entropy-based image registration*. Princeton University, 2006. 2, 3, 4, 5, 9, 19

Sarlin, P.-E., Cadena, C., Siegwart, R., and Dymczyk, M. From coarse to fine: Robust hierarchical localization at large scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12716–12725, 2019. 1

Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. SuperGlue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4938–4947, 2020. 1

Scarlett, J. and Cevher, V. An introductory guide to fano's inequality with applications in statistical estimation. Technical report, Cambridge University Press, 2019. 4, 5, 19, 20

Schmid, C., Mohr, R., and Bauckhage, C. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000. 8, 9

Shannon, C. E. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001. 2, 29

Sharma, P., Mohan, L., Pinto, L., and Gupta, A. Multiple interactions made easy (mime): Large scale demonstrations data for imitation. In *Conference on Robot Learning*, pp. 906–915. PMLR, 2018. 2, 6, 7, 24

Singh, G., Deng, F., and Ahn, S. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2021. 8

Song, Y., Cai, L., Li, J., Tian, Y., and Li, M. SEKD: Self-evolving keypoint detection and description. *arXiv preprint arXiv:2006.05077*, 2020. 8

Sun, J. J., Ryou, S., Goldshmid, R. H., Weissbourd, B., Dabiri, J. O., Anderson, D. J., Kennedy, A., Yue, Y., and Perona, P. Self-supervised keypoint discovery in behavioral videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2171–2180, 2022. 9

Szeliski, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 1, 3

Tandon, R., Shanmugam, K., Ravikumar, P. K., and Dimakis, A. G. On the information theoretic limits of learning ising models. *Advances in Neural Information Processing Systems*, 27, 2014. 19

Toyer, S., Shah, R., Critch, A., and Russell, S. The MAGICAL benchmark for robust imitation. *Advances in Neural Information Processing Systems*, 33:18284–18295, 2020. 2, 6, 8, 24, 28

Wang, Z., Qinami, K., Karakozis, I. C., Genova, K., Nair, P., Hata, K., and Russakovsky, O. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8919–8928, 2020. 1

Wu, Y., Zhou, Y., Saveriades, G., Agaian, S., Noonan, J. P., and Natarajan, P. Local shannon entropy measure with statistical tests for image randomness. *Information Sciences*, 222:323–342, 2013. 9

Xiong, H., Li, Q., Chen, Y.-C., Bharadhwaj, H., Sinha, S., and Garg, A. Learning by watching: Physical imitation of manipulation skills from human videos. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7827–7834. IEEE, 2021. 1

Xiongwei, W., Sahoo, D., and Steven, H. PolarNet: Learning to optimize polar keypoints for keypoint based object detection. In *International Conference on Learning Representations*, 2020. 1

Xu, T. and Takano, W. Graph stacked hourglass networks for 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16105–16114, 2021. 15

Yang, G., Zhang, A., Morcos, A., Pineau, J., Abbeel, P., and Calandra, R. Plan2Vec: Unsupervised representation learning by latent plans. In *Learning for Dynamics and Control*, pp. 935–946. PMLR, 2020. 1

Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., and Tenenbaum, J. B. Clevrer: Collision events for video representation and reasoning. In *International Conference on Learning Representations*, 2019. 2, 6, 7, 20, 22, 24, 27

Yi, K. M., Trulls, E., Lepetit, V., and Fua, P. LIFT: Learned invariant feature transform. In *European Conference on Computer Vision*, pp. 467–483. Springer, 2016. 8

Yu, L., Song, Y., Song, J., and Ermon, S. Training deep energy-based models with f-divergence minimization. In *International Conference on Machine Learning*, pp. 10957–10967. PMLR, 2020. 29

Yu, S., Giraldo, L. G. S., and Príncipe, J. C. Information-theoretic methods in deep neural networks: Recent advances and emerging opportunities. In *IJCAI*, pp. 4669–4678, 2021. 3, 29

Zhang, Q., Wu, Y. N., and Zhu, S.-C. Interpretable convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8827–8836, 2018. 29

Zheng, L., Yang, Y., and Tian, Q. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(5):1224–1244, 2017. 8

Appendix

The appendix provides additional details on the architecture, the entropy layer, proofs, the description of evaluation metrics for object detection and tracking task, additional experimental analysis with implementation details and discussions, and an extended related work section. Moreover, we provide video results on the project website[4] and the code[5].

[4] https://sites.google.com/view/mint-icml

[5] https://github.com/iROSA-lab/MINT

A. Architecture

This section contains additional information about the model architecture in Section 2.2, implementation details on how to get the keypoint coordinates from the feature map, and the heatmaps for keypoints to ensure reproducibility.

A.1. Keypoint Model

The backbone of our model is an hourglass architecture (Newell et al., 2016), followed by a soft-argmax operator to receive the coordinates of the keypoints. Keypoints provide a low-dimensional representation of high-dimensional RGB images. Therefore, many keypoint detection techniques are inspired by the autoencoder architecture (Goodfellow et al., 2016) which uses the bottleneck to consolidate the information into a reduced dimensionality for keypoint extraction (Minderer et al., 2019; Kulkarni et al., 2019). Instead, we suggest taking an hourglass architecture (Newell et al., 2016) which upscales the compressed information again and outputs several feature maps with high activation in places with eminent information (Ewerton et al., 2021; Xu & Takano, 2021; Newell et al., 2016). This allows the network to predict the information at the original image size yielding finer resolution of the coordinates and correspondence to the original pixels.

Our keypoint detector consists of an hourglass convolutional neural network with three convolutional layers, with kernel sizes of $5, 3, 3$ and strides of $3, 2, 2$, respectively. The upsampling part of the model consists of three transposed convolutional layers, with kernel sizes of $3, 3, 3$ and strides of $1, 2, 2$. The number of input and output channels for each layer depends on the number of keypoints K and the number of input image channels C, see Table 4. The result is passed through a softplus layer to ensure the positivity of the feature maps. Lastly, we append a spatial soft-argmax layer (see Appendix A.2) to get the coordinates of the keypoints from the feature maps f_i. We initialize all the convolutional layers with Xavier's normal initialization (Glorot & Bengio, 2010) and add a leaky ReLU activation and a batch normalization layer after each of them. We normalize the input to the range $[-0.5, 0.5]$. The total number of the parameters is $58,725$ for the input of size 320×420.

Table 4. Architecture details for an RGB image of 320×480 and $K = 25$ keypoints. There is a leaky ReLU layer and a BatchNorm2d layer (50 parameters) after each convolutional layer.

Layer (type)	Input channels	Output channels	Kernel size	Stride	Output shape	# params
Normalize-1	C	C	-	-	[3, 320, 480]	0
Conv2d-1	C	K	5	3	[25, 79, 106]	1,900
Conv2d-2	K	K	3	2	[50, 26, 35]	5,650
Conv2d-3	K	2K	3	2	[50, 26, 35]	11,300
ConvTranspose2d-1	2K	2K	3	1	[50, 53, 71]	22,550
ConvTranspose2d-2	2K	K	3	2	[25, 107, 143]	11,275
ConvTranspose2d-3	K	K	3	2	[25, 107, 143]	5,650
Softplus-14	K	K	-	-	=	0
SpatialSoftargmaxLayer-15	K	K	-	-	[25, 2]	0
Final output					[25,3]	total: 58,725

A.2. Feature Map to Keypoint

The coordinates of the keypoints are determined by the location of the maximum value in its corresponding feature map. The argmax operator is not differentiable, so we opted to use a differentiable spatial soft-argmax as an alternative to extract the keypoints coordinates from the feature maps. The spatial soft-argmax (Levine et al., 2016) takes K 2D feature maps $f^{(i)}$, one for each keypoint k_i, flattens the feature maps and computes the weights ω_i for each pixel

$$w_i = \frac{e^{f^{(i)}-max(f^{(i)})}}{\sum e^{f^{(i)}-max(f^{(i)})}} \,. \tag{11}$$

Before applying the softmax, we subtract the maximum value from the input, which does not change the output of the softmax but helps for numerical stability. In order to map the weights to coordinates, we generate a mesh grid (x_{grid}, y_{grid}) of x and y coordinates, with the same size as the input image. We flatten the mesh grid and compute the expected keypoint coordinates $[\hat{x}_i, \hat{y}_i]$ as the weighted sum of the coordinate grid with ω_i. This process is visualized in Figure 8.

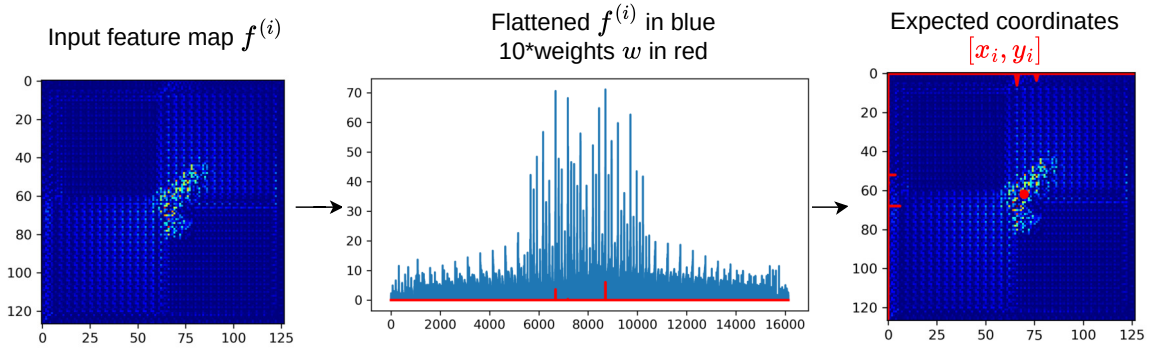

Figure 8. Feature map to keypoint (spatial soft-argmax). The keypoint detector outputs a feature map $f^{(i)}$ for each keypoint $k^{(i)}$. The spatial soft argmax operator polls the coordinates of the keypoint $[x_i, , y_i]$ in a differentiable way by flattening a mesh grid of coordinates and using the feature-map values as weights to vote for the coordinates of the maximum value.

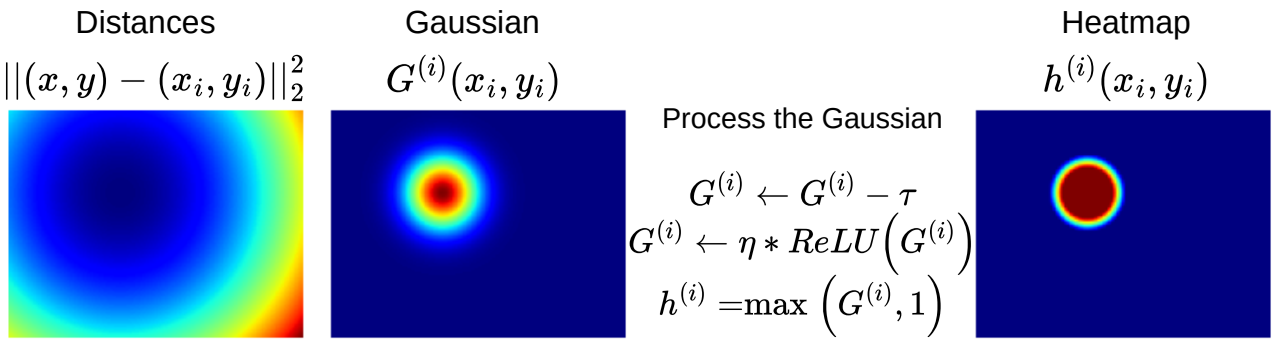

Figure 9. Keypoints to heatmap. We build the heatmap $h^{(i)}$ centered at the coordinates of a keypoint $k^{(i)}$ in a differentiable fashion. The process starts by computing the distances between the center and all the pixels forming a 2D distances image. Inducing the distances into Gaussian forms a multivariate Gaussian distribution $G^{(i)}$ over the image, whose mean is at the keypoint coordinates. Thresholding and clamping the Gaussian gives the final heatmap, which represents a keypoint's information coverage.

A.3. Keypoints to Heatmaps

The heatmaps mask out the information coverage areas of keypoints, and are essential to define our losses. We developed a differentiable way to generate heatmaps from keypoint coordinates. The heatmap $h^{(i)}$ generation for a keypoint takes coordinates as a pair of real numbers $(x_i, y_i) \in \mathbb{R}^2$. We start by generating a pixel-coordinates array with the same width and height as the original image $H \times W \times 2$, where 2 denotes the coordinates of each pixel $(x, y) \in \mathbb{N}^{H \times W \times 2}$. Then we compute the squared distance between the input and all the pixels $||(x, y) - (x_i, y_i)||_2^2$. We use the squared distance to generate a Gaussian distribution around the input coordinates $G^{(i)}(x_i, y_i)$ with a standard deviation $\sigma_{G^{(i)}}$.

The heatmap defines the area and weighting of information belonging to each keypoint. The heatmap should be 1 around the center of the keypoint, as the keypoint covers the information in this point completely, and descend gradually to 0 representing information out of reach of the keypoint. We achieve that by thresholding and clamping the Gaussian. We use a threshold τ and a scaling factor η for the thresholded Gaussian to get the final heatmap h_i. Table 6 provides more details on the scale of these hyperparameters. The process is visualized in Figure 9.

B. Entropy Layer

The entropy layer (Section 2.1) is one of the main modules of our method. For an input image, the entropy layer outputs the image spatial entropy that is the basis for computing our intrinsic supervisory signals for our representation learning method. In this section, we provide additional implementation details. We split the explanation into two subsections: *(1)* the entropy module definition in PyTorch (Paszke et al., 2019), and *(2)* the CUDA extension for the parallel execution.

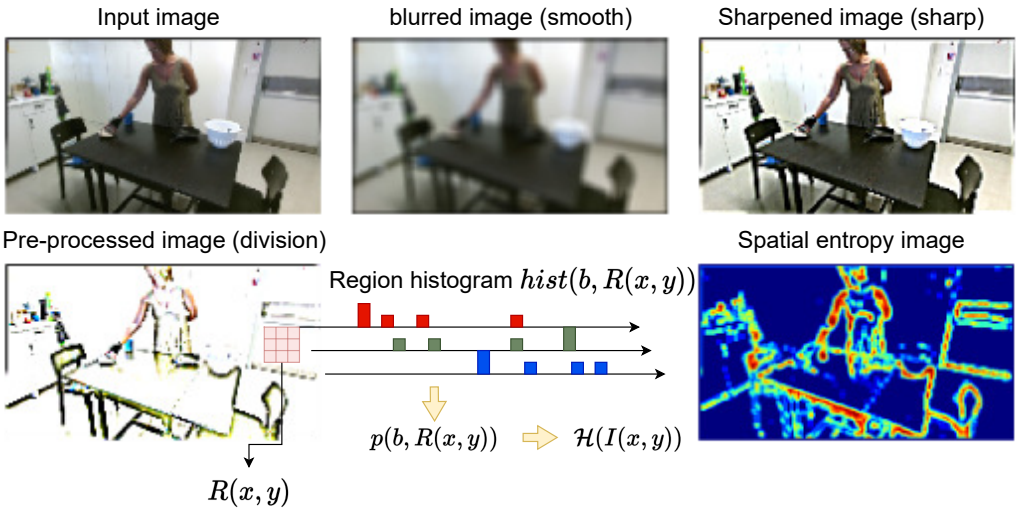

Figure 10. Entropy computation in the entropy layer (Appendix B), which consists of the entropy module and the CUDA extension. **The entropy module** takes as input an RGB image (*input image*), blurs it to get a smoothed image (*smooth* image), and uses the result to sharpen the input image (we get the *sharp* image). The final preprocessed image (*division* image) is the result of dividing the sharpened image (*sharp*) by the blurred image (*smooth*). Further, the entropy module extracts non-overlapping patches and forwards them to the CUDA extension. **CUDA extension** computes the region histograms for each region (patch) $hist(b, R(x, y))$, and uses the histogram to compute the probability of each bin $p(b, R(x, y))$. The entropy of a pixel at location (x, y) is Shannon's entropy of the region around it $\mathcal{H}(I(x, y))$ depending on the probabilities of the color values inside that region. The final *spatial entropy image* is formed by the individual entropies of pixels. Our CUDA extension provides a highly efficient and parallelizable implementation for the process.

B.1. Entropy Module

The entropy module is a PyTorch (Paszke et al., 2019) module, which preprocesses an input image and forwards it to the CUDA extension for efficient entropy computation. The input of the entropy module is an RGB image $I \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3}$. The input image is first processed to remove high-frequency color changes. The processing is completely vectorized to allow efficient execution using PyTorch during training. The processing of the input image starts by blurring the image using an average blur layer, followed by sharpening the result, and finally dividing the sharp image by the smooth image. Figure 10 shows an example of the intermediate steps of the entropy layer.

Before forwarding the processed image to the entropy function, we generate the neighborhood region $R(x, y)$ for each pixel location (x, y). These regions have square shape with the corresponding pixel (x, y) being the center. Instead of iterating over all the pixels with for loops, we use the stridden operation to factorize the extraction of the regions. Using strides to extract the regions aligns with how the image is stored in the memory and does not create overhead. These tricks are essential for the efficiency of our entropy layer.

B.2. CUDA Extension

The CUDA extension is a high-efficient program for parallel image spatial entropy ISE computation, according to the monkey model entropy (MME) (Brink, 1996; Razlighi & Kehtarnavaz, 2009), for which we need the histogram of the color values. The naive histogram computation via vectorizing the code requires computing a pairwise distance matrix between each pixel with every histogram bin, corresponding to multiplying the number of possible regions by 256. This causes exploding GPU memory requirements (more than 50GB). Motivated by this observation, we present an efficient entropy layer based on kernel density estimation in this work.

To estimate the value for each histogram bin b inside a region $R(x, y)$ (patch) centered on pixel at location (x, y), we use the kernel density estimator

$$\hat{f}(b, x, y) = \sum_{(x_n, y_n)\in R(x,y)} \mathcal{K}\left(\frac{I(x_n, y_n) - b}{B}\right), \tag{12}$$

where $I(x_n, y_n)$ is the pixel value at location (x_n, y_n) inside the region $R(x, y)$, and B is the bandwidth, used as a smoothing parameter.

We follow (Avi-Aharon et al., 2020) and use the derivative of the logistic regression function, the Sigmoid function $\sigma(.)$, as a kernel $\mathcal{K}(.)$, that is for a variable v

$$\mathcal{K}(v) = \frac{d}{dv}\sigma(v) = \sigma(v)\sigma(-v) \ . \tag{13}$$

The integral of the function $\hat{f}(b, x, y)$ defined in Equation (12) over the region gives the histogram value of the bin b in a color channel c:

$$hist_c(b, R(x, y)) = \sum_{(x_n, y_n) \in R(x,y)} \left[\sigma(\frac{I_c(x_n, y_n) - b - L/2}{B}) - \sigma(\frac{I_c(x_n, y_n) - b + L/2}{B}) \right] \ , \tag{14}$$

where $L = 1/256$ is the bin size, so that each bin represents a color value. We get the probability of each color value by dividing the sum of the histogram values by the size of the region $|R|$ and the number of channels C

$$p(I(x, y)) = p(b, R(x, y)) = \frac{1}{C \cdot |R|} \sum_{c \in \{r,g,b\}} hist_c(b, R(x, y)) \ . \tag{15}$$

The entropy of the pixel in the center of the patch is

$$\mathcal{H}(I(x, y)) = - \sum_{(x_n, y_n) \in R(x,y)} p(I(x_n, y_n)) \log(p(I(x_n, y_n))) \tag{16}$$

$$= - \sum_{b \in [0,255]} p(b, R(x, y)) log(p(b, R(x, y))) \ . \tag{17}$$

The entropy module uses our entropy function, implemented as an autograd function in PyTorch, to realize the CUDA extension of the entropy computation. The input to the entropy function are the regions of the preprocessed images $R(x, y)$. The CUDA extension allocates a GPU block for each region, hence, the grid size equals to the number of all possible regions for all images in the batch. The block size is 256 threads, *i.e.*, a thread for each bin b. Each thread iterates over the whole region and computes the histogram of its corresponding bin value b according to Equation (14). Then, it normalizes the result by the region size and the number of channels to get the probability according to Equation (15). Finally, each thread computes the entropy of the pixel (Equation (16)), which is equivalent to the sum of the entropy of the histogram bins (Equation (17)).[6]

C. Proofs

C.1. Proof of Proposition 2.1

We can consider the network as an information channel similar to the information maximization principle (InfoMax) (Linsker, 1988). The input of the network is the actual image I_t, and the output is the masked image I_t^M by the keypoints. We want to minimize the average probability of error of how well the output I_t^M represents the information of the input I_t. First, we will work on a pixel level to bound the probability of error in the intensity of the n^{th} pixel at location (x, y) in images I_t^M and I_t, denoted as $P_\varepsilon^{(n)} = \mathbb{P}(I_t(x, y) \neq I_t^M(x, y))$. Images can in general be considered as lattices, with pixels being the random variables over intensities $\mathcal{B}$ (in our case these are the number of bins in the histogram as described in Appendix B).

Since the error event for the n^{th} pixel is a binary event, it follows that $P_\varepsilon^{(n)}$ is a binary probability. Therefore, the average error probability over all pixels N of the image can be computed as

$$\bar{P}_\epsilon = \frac{1}{N} \sum_{n}^{N} P_\varepsilon^{(n)} \ , \tag{18}$$

where $N = H \times W$ is the total number of pixels, computed as the product of the height H and width W of the image.

[6]The entropy layer is opensourced in https://github.com/iROSA-lab/MINT

On a pixel-level, following Fano's inequality[7] (Sabuncu, 2006; Scarlett & Cevher, 2019; Tandon et al., 2014) and assuming that the n^{th} pixel in position (x, y) in an image can take a value uniformly on $\mathcal{B}$, we get

$$\mathcal{H}(I_t(x,y)|I_t^M(x,y)) \leq \mathcal{H}_2(P_\varepsilon^{(n)}) + P_\varepsilon^{(n)} \log(|\mathcal{B}| - 1) \,, \tag{19}$$

where $\mathcal{H}_2(\alpha) = \alpha \log \frac{1}{\alpha} + (1-\alpha) \log \frac{1}{1-\alpha}$ is the binary entropy function (with maximum entropy corresponding to $\alpha = \frac{1}{2}$), and $\mathcal{B}$ is the support of the pixel value.

Equation (19) can be further bounded as $\mathcal{H}_2(P_\varepsilon^{(n)}) \leq \log 2$, and $|\mathcal{B}| - 1 \leq |\mathcal{B}|$. Moreover, since a pixel is uniform on $|\mathcal{B}|$ its entropy can be considered $\mathcal{H}(I_t(x,y)) = \log |\mathcal{B}|$. Therefore, we can further bound Equation (19) as

$$\mathcal{H}(I_t(x,y)|I_t^M(x,y)) \leq \log 2 + P_\varepsilon^{(n)} \log(|\mathcal{B}|) \tag{20}$$

$$\Leftrightarrow \mathcal{H}(I_t(x,y)|I_t^M(x,y)) - \mathcal{H}(I_t(x,y)) \leq \log 2 + P_\varepsilon^{(n)} \log(|\mathcal{B}|) - \mathcal{H}(I_t(x,y)) \tag{21}$$

$$\Rightarrow \mathcal{I}(I_t(x,y), I_t^M(x,y)) \geq (1 - P_\varepsilon^{(n)}) \log(|\mathcal{B}|) - \log 2 \tag{22}$$

$$\Leftrightarrow P_\varepsilon^{(n)} \geq 1 - \frac{\mathcal{I}(I_t(x,y), I_t^M(x,y)) + \log 2}{\log |\mathcal{B}|} \,. \tag{23}$$

The mutual information of two random variables is upper bounded by the minimum entropy of the marginals, therefore, $\mathcal{I}(I_t(x,y), I_t^M(x,y)) \leq \min(\mathcal{H}(I_t(x,y)), \mathcal{H}(I_t^M(x,y)))$. But the masked image will by definition represent less information than the original image, therefore, we have $\mathcal{I}(I_t(x,y), I_t^M(x,y)) \leq \mathcal{H}(I_t^M(x,y))$. We can take the *worse case scenario* and assume $\mathcal{I}(I_t(x,y), I_t^M(x,y)) \approx \mathcal{H}(I_t^M(x,y))$ (Murphy, 2022). Therefore, Equation (23) becomes

$$P_\varepsilon^{(n)} \geq 1 - \frac{\mathcal{H}(I_t^M(x,y))) + \log 2}{\log |\mathcal{B}|} \,, \tag{24}$$

which bounds the error on the information carried by a pixel in the masked image.

To acquire the bound of the average error probability in Equation (18), we sum Equation (24) for all pixels and divide by N to get

$$\frac{1}{N} \sum_n^N P_\varepsilon^{(n)} \geq \frac{1}{N} \sum_n^N 1 - \frac{\sum_{x,y} \mathcal{H}(I_t^M(x,y)) + \sum_n^N \log 2}{N \log |\mathcal{B}|} \tag{25}$$

$$\bar{P}_\varepsilon \geq 1 - \frac{\sum_n^N \mathcal{H}(I_t^M(x_n, y_n))}{N \log |\mathcal{B}|} - \frac{\log 2}{\log |\mathcal{B}|} \,. \tag{26}$$

□

C.2. Proof of Lemma 2.2

We first derive the complete proof for the relation $\mathcal{H}(X,Y) \geq \max(\mathcal{H}(X), \mathcal{H}(Y)) \geq 0$ that holds for any two discrete random variables, as also stated in (Murphy, 2022)-Equation(6.10). Then, we extend this proof for the case of joint ISE.

Let X, Y be two discrete random variables, and the respective entropies are lower-bounded $\mathcal{H}(X) \geq 0$, $\mathcal{H}(Y) \geq 0$. The joint entropy between the two random variables can be expressed as

$$\mathcal{H}(X,Y) = \mathcal{H}(X) + \mathcal{H}(Y|X) = \mathcal{H}(Y) + \mathcal{H}(X|Y) \,. \tag{27}$$

As any conditional entropy is greater or equal to zero, we get $\mathcal{H}(X,Y) \geq \mathcal{H}(X)$, and similarly $\mathcal{H}(X,Y) \geq \mathcal{H}(Y)$. If $\mathcal{H}(X) \geq \mathcal{H}(Y)$ then

$$\mathcal{H}(X,Y) \geq \mathcal{H}(X) \geq \mathcal{H}(Y) \,. \tag{28}$$

If $\mathcal{H}(Y) \geq \mathcal{H}(X)$ then

$$\mathcal{H}(X,Y) \geq \mathcal{H}(Y) \geq \mathcal{H}(X) \,. \tag{29}$$

[7] Fano's inequality uses information-theoretic measures to provide the relation between the average information loss from a noisy channel and the probability of categorization error.

Therefore, for discrete random variables, we get the lower bound of the joint entropy as

$$\mathcal{H}(X,Y) \geq \max(\mathcal{H}(X), \mathcal{H}(Y)) \geq 0 \,. \tag{30}$$

Following the previous derivation, when considering two images I_1, I_2 whose pixels are discrete random variables over intensities $\mathcal{B}$, we can lower bound the joint image spatial entropy by the pixel-wise maximum of the two marginal ISE

$$\mathcal{H}(I_1, I_2)(x,y) \geq \max(\mathcal{H}(I_1(x,y)), \mathcal{H}(I_2(x,y))) \,. \tag{31}$$

□

Given this lower bound we can approximate the joint ISE as

$$\mathcal{H}(I_1, I_2)(x,y) \approx \max(\mathcal{H}(I_1(x,y)), \mathcal{H}(I_2(x,y))) \,. \tag{32}$$

□

Remark. Approximating the joint entropy by the lower bound corresponds to the worst-case scenario for the MCE loss. Hence, the approximation ensures better information reconstruction according to Corollary 2.3. Meanwhile, given that the spatial mutual information of two images is upper bounded by the joint entropy, and since the maximization of the mutual information optimizes the keypoint transportation over frames according to Proposition 2.4, approximating the joint entropy by the lower bound corresponds to a higher probability of reducing the IT loss.

C.3. Proof of Proposition 2.4

Due to the process of information transportation of the keypoints, we try to reconstruct the information each keypoint carries. Therefore, we can again leverage Fano's inequality (Scarlett & Cevher, 2019), to provide a lower bound for the average error probability of information transportation per keypoint.

We formalize our error probability of information transportation of the K keypoints as the per-pixel error event $P_\varepsilon^{\text{IT}} = \mathbb{P}(I_t(x,y) \neq R_t^{(i)}(x,y))$, *i.e.*, aggregating the reconstruction from all keypoints. Therefore, from Fano's inequality similar to Equation (19), we have the per-keypoint inequality for each keypoint $k^{(i)}$

$$H(I_t(x,y)|R_t^{(i)}(x,y)) \leq H_2(P_\varepsilon^{\text{IT}(i)}) + P_\varepsilon^{\text{IT}(i)} \log(|\mathcal{B}| - 1) \,. \tag{33}$$

Following similar derivation steps as in Appendix C.1, we end up in the equivalent version of Equation (23)

$$\mathcal{I}(I_t(x,y), R_t^{(i)}(x,y)) \geq (1 - P_\varepsilon^{\text{IT}(i)}) \log(|\mathcal{B}|) - \log 2 \Leftrightarrow \tag{34}$$

$$P_\varepsilon^{\text{IT}(i)} \geq 1 - \frac{\mathcal{I}(I_t(x,y), R_t^{(i)}(x,y)) + \log 2}{\log |\mathcal{B}|} \,. \tag{35}$$

Note that we assume the IT operation per keypoint, independently, and assuming that it is an exclusive event. Therefore, every single transportation is bounded by Equation (35). □

D. Evaluation Metrics for Object Detection and Tracking

Each keypoint should provide a representation of a feature in an object, and keypoints should be distinctive and distributed over the scene. Keypoints assigned to empty spaces are considered unsuccessfully assigned. To judge the performance of our method, we propose metrics that use the object masks provided by CLEVRER (Yi et al., 2019) over a set of test videos V, each of which is of length T.

D.1. Percentage of the Detected Objects (DOP)

We consider an object detected if there is at least one keypoint on its mask M_{obj}. At each time frame, we count the percentage of detected objects with respect to the ground truth (GT) number of objects and average these values over the whole video. We get the final result by averaging the value over all the videos in the test dataset. Better detection corresponds to a higher percentage of detected objects.

$$M_{DOP} = \frac{1}{V \cdot T} \sum_{v=1}^{V} \sum_{t=1}^{T} \frac{N_{detected}}{N_{GT}} \,, \tag{36}$$

where $N_{detected}$ is the number of detected objects (at least one keypoint lies in the object mask)

$$N_{detected} = \sum_{obj \in O} \left[\sum_{i=1}^{K} \mathbb{I}((x_i, y_i) \in M_{obj}) \right] > 0, \tag{37}$$

with N_{GT} being the ground truth number of objects and O is the set of all objects in the scene.

D.2. Percentage of Tracked Objects (TOP)

We consider an object tracked if there is at least one keypoint on its mask in the current and the previous timeframe. At each time frame, we count the percentage of tracked objects with respect to the ground truth (GT) number of objects and average these values over the whole video. We get the final result by averaging the value over all the videos in the test dataset. Better detection corresponds to a higher percentage of tracked objects

$$M_{TOP} = \frac{1}{V \cdot T} \sum_{v=1}^{V} \sum_{t=1}^{T} \frac{N_{tracked}}{N_{GT}} \,, \tag{38}$$

where $N_{tracked}$ is the number of tracked objects (at least one keypoint lies in the object mask in time frames t and t-1)

$$N_{tracked} = \sum_{obj \in O} \left[\sum_{i=1}^{K} [\mathbb{I}((x_i, y_i)_t \in M_{obj}^{(t)}] \cdot [\mathbb{I}((x_i, y_i)_{t-1} \in M_{obj}^{(t-1)})] \right] > 0. \tag{39}$$

D.3. Unsuccessful Keypoint Assignment (UAK)

A keypoint is unsuccessfully assigned in a time frame if it does not belong to any object. We average the number of unsuccessful keypoint over the whole video, and then over test videos to get a global value over the testset

$$M_{UKA} = \frac{1}{V \cdot T} \sum_{v=1}^{V} \sum_{t=1}^{T} N_{uk} \,, \tag{40}$$

where N_{uk} is the number of unsuccessful keypoints (does not belong to the sum of the masks)

$$N_{uk} = \sum_{i=1}^{K} \sim \mathbb{I}((x_i, y_i) \notin \sum_{obj \in O} M_{obj}) \,. \tag{41}$$

A lower unsuccessful keypoint assignment metric M_{UKA} corresponds to better keypoints activation.

D.4. Redundant Keypoint Assignment (RAK)

Assigning keypoints to areas already represented by other keypoints signals bad keypoint detection. The RAK metric accounts for the number of keypoints over the area of the object. The number of keypoints on an object mask should be proportional to its area A_{obj}. We assume a keypoint can represent some area of pixels A_k. If the keypoints cover the object, the RAK metric will have a value of 0, with higher values if more or fewer keypoints were assigned to that object.

$$M_{RAK} = \frac{1}{V \cdot T \cdot O} \sum_{v=1}^{V} \sum_{t=1}^{T} \sum_{obj \in O} \frac{|A_{obj} - A_k n_{obj}|}{A_{obj}} \,, \tag{42}$$

where A_k is the representation area of a keypoint (e.g. average object areas in the dataset) A_{obj} is the area of the object's mask and n_{obj} is the number of keypoints assigned to the object

$$n_{obj} = \sum_{i=1}^{K} \mathbb{I}((x_i, y_i) \in M_{obj}) \qquad \text{for each } \; obj \in O \; . \tag{43}$$

The lower the value of M_{RAK}, the better, because more efficient, is the distribution of the keypoints.

The metrics collectively judge the efficacy of keypoint detection and tracking methods, where only detected objects can be tracked, so the DOP metric is an upper bound for the TOP metric. The value of the metric RAK will go to one in the case of not detecting any object, but can go higher in case of assigning redundandant keypoints to the same object. Following this observation, we recommend judging the value of the RAK metric jointly with the value of the DOP metric.

E. Additional Experimental Analysis

E.1. Ablation Study

Our method for unsupervised keypoint discovery in video streams uses a collection of information-theoretic losses and some regularizers. In the ablation study, we investigate the role of each component and discuss our design choices. In the following, we analyze different design choices like the entropy region size, the conditional entropy in the information transportation loss, and the regularizers.

Ablation analysis. Using the proposed evaluation metrics, we analyzed several aspects of MINT on CLEVRER (Yi et al., 2019). We report the results in Table 5.
Since we compute the local entropy using the probability of the pixel value in its neighborhood region, we investigated the effect of the region size on the performance by varying the region size while using the information maximization (IM) (*i.e.*, masked entropy (ME) and masked conditional entropy (MCE) from Section 2.2.1) loss alone. The results show that a region of size 5×5 gives the highest values for the percentage of the detected object (DOP) and percentage of tracked objects (TOP) metrics. We observe also that increasing the region size led to an increase in unsuccessful keypoint assignment (UAK), with a decrease in redundant keypoint assignment (RAK); we hypothesize this is due to an over-smoothing effect of the bigger region, which leaks some information outside the objects. We noticed, on the other hand, an increase in the order of 30 minutes in the training time (50% of the training time) of one seed when increasing the region size by 2. Given the marginal improvement and the need for more resources, we adopted a region size of 3×3 for all of our experiments.

We examined the information-theoretic losses without regularization to ablate the additional hyperparameter κ which sets the contribution of the conditional entropy in the information transportation (IT) loss. The results prove that adding conditional information improves the keypoint detection, with $\kappa = 0.5$ giving the best results for DOP and TOP followed by $\kappa = 0.9$. The value of RAK increases with lower κ, because keypoints seek the same areas of high information to reconstruct as much information as possible, leading to the redundant assignment. The introduction of conditional entropy in the IT loss, as describe in Section 2.2.2, helps mitigate this behavior by lowering the reconstruction error outside the transportation regions, *i.e.*, the keypoint position in the current and the previous time frame. We highlight two values from this part; with $\kappa = 0.5$ we get the best scores for DOP and TOP, while $\kappa = 0.9$ trades off well all of the metrics (we call this model MINT w/o Reg. - highlighted in light blue, that is also referenced in Table 1).

Next, we investigate the regularization terms proposed in our method: (1) the movement loss controlled by the weight m_d in the information transportation loss, (2) the overlapping loss (O), and (3) the active status loss (S). We experimented with all possible combinations of those regularizers. We can observe that the movement regularizer helps decrease the UAK metric, as this regularizer stabilizes the keypoint movement and constraints the keypoints from jumping into the background. The overlapping loss reduces the RAK value by almost half (from 3.982 to 2.079), but this comes with a higher UAK. The status loss reduces the UAK but comes at the cost of lower DOP and TOP. Introducing the overlapping and the status loss together allows better overall performance, where the overlapping loss increases the DOP. We achieved the best trade-off across all metrics by setting $\kappa = 0.9$ while using all the regularizers (highlighted in light green). We adopt this option for our method MINT, and it proved to outperform the baselines both in the synthetic dataset (quantitatively proved in Table 1, and qualitatively shown in Figure 4) and for realistic scenarios (Figures 5 and 6).

Finally, we investigated the performance of the losses that work for single images, mainly the ME with the regularizers: the active status loss (S) and the overlapping loss (O). This combination of losses does not use any temporal information, hence, it can operate on static images. We train this combination of losses on CLEVRER, operating on single images. We can observe that the model learns to track objects despite being trained on single images only. However, we argue that

Table 5. Ablation study on MINT losses. We report the statistics of the metric values over 5 seeds. IM stands for the information maximization losses (ME + MCE), IT for information transportation, κ decides the contribution of the conditional entropy in the IT loss, m_d is the movement regularizer weight in the IT loss, O is the overlapping loss and S is the active status loss. The ablations picked for MINT w/o reg. , MINT w/o Temp. and MINT are highlighted with light blue, light red, and light green consequently. The weight scales used for all the ablations are $\lambda_{ME} = \lambda_{MCE} = 100, \lambda_{IT} = 20, \lambda_s = 10, \lambda_o = 30$, and $K = 25$. The * near the method's name indicates a longer training time.

Method	**DOP** ⇑	**TOP** ⇑	**UAK** ⇓	**RAK** ⇓
IM (3x3)	0.951 ± 0.042	0.929 ± 0.048	**6.777 ± 1.369**	3.885 ± 1.090
IM (5x5)*	**0.956 ± 0.036**	**0.932 ± 0.043**	8.276 ± 1.428	3.660 ± 1.083
IM (7x7)*	0.951 ± 0.041	0.926 ± 0.048	9.946 ± 1.593	**3.098 ± 0.959**
IM+IT ($m_d = 0$,κ=0)	0.917 ± 0.072	0.897 ± 0.077	**3.543 ± 1.529**	5.096 ± 1.587
IM+IT ($m_d = 0$,κ=0.5)	**0.935 ± 0.058**	**0.916 ± 0.063**	4.754 ± 1.463	3.982 ± 1.226
IM+IT ($m_d = 0$,κ=0.9)	0.918 ± 0.073	0.897 ± 0.078	6.793 ± 1.956	2.478 ± 0.865
IM+IT ($m_d = 0$,κ=1)	0.916 ± 0.073	0.895 ± 0.078	5.645 ± 1.873	**2.336 ± 0.768**
IM+IT ($m_d = 1$)	0.883 ± 0.097	0.865 ± 0.102	1.665 ± 0.954	1.896 ± 0.706
IM+IT ($m_d = 0$)+O	**0.921 ± 0.066**	**0.898 ± 0.073**	7.769 ± 1.880	2.079 ± 0.604
IM+IT ($m_d = 1$)+O	0.879 ± 0.102	0.861 ± 0.105	2.196 ± 1.228	1.705 ± 0.582
IM+IT ($m_d = 0$)+S	0.851 ± 0.114	0.830 ± 0.118	1.057 ± 0.666	1.159 ± 0.455
IM+IT ($m_d = 1$)+S	0.842 ± 0.116	0.823 ± 0.119	1.060 ± 0.735	1.180 ± 0.475
IM+IT ($m_d = 0$)+S+O	0.859 ± 0.112	0.840 ± 0.116	1.130 ± 0.710	1.324 ± 0.508
IM+IT ($m_d = 1$,κ=0.5)+S+O	0.844 ± 0.120	0.826 ± 0.123	1.100 ± 0.716	**1.121** ± 0.451
IM+IT ($m_d = 1$,κ=0.9)+S+O	0.855 ± 0.118	0.838 ± 0.121	**0.889 ± 0.639**	1.123 ± 0.448
ME+S+O	0.849 ± 0.115	0.826 ± 0.119	0.958 ± 0.615	1.142 ± 0.446
MINT w/o Reg.	**0.918 ± 0.073**	**0.897 ± 0.078**	6.793 ± 1.956	2.478 ± 0.865
MINT w/o Temp.	0.849 ± 0.115	0.826 ± 0.119	0.958 ± 0.615	1.142 ± 0.446
MINT (ours)	0.855 ± 0.118	0.838 ± 0.121	**0.889 ± 0.639**	**1.123 ± 0.448**
K=10				
IM (3x3)	0.879 ± 0.085	0.847 ± 0.095	1.662 ± 0.781	1.536 ± 0.554

the structure of our training process, which uses samples from a sequence of images, biases the model towards reducing the movement of the keypoints while attending to features, leading to good tracking performance. We call this ablation MINT w/o Temp. , and we discuss it further later.

We show that if we have enough knowledge about the environment and we can decide on the suitable number of keypoint (*e.g.*, K=10 keypoints for CLEVRER), then the information maximization (IM) loss alone is enough to get good performance (last row in Table 5), with low UAK and RAK, as the keypoint assignment is easier. We further discuss the two major ablations MINT w/o Reg. and MINT w/o Temp. in detail regarding their performance qualitatively on the realistic datasets.[8]

MINT w/o Reg. With MINT w/o Reg. we refer to our method MINT without the regularization terms, *i.e.*, (1) removing the regularization for the keypoints' movement $m_d = 0$ in the information transportation loss, (2) removing the overlapping loss $\lambda_o = 0$, and (3) removing the active status loss $\lambda_s = 0$. Besides the quantitative results in Table 5, which show that the information-theoretic losses can detect and track objects better than the baselines (outperforming all of the baselines in the DOP and TOP metrics), we provide qualitative evidence of the performance of the proposed information-theoretic losses, where MINT w/o Reg. can detect and track the object in synthetic (Figure 13) and realistic scenes (Figures 11 and 12).

On the other hand, the experiments justify the role of regularization in stabilizing keypoint detection and removing excessive keypoints. Figure 13 from CLEVRER and Figure 11 from SIMITATE show a better distribution of keypoint when using MINT over MINT w/o Reg. We refer also to the zoomed-in regions in Figure 13 where we show MINT w/o Reg. assign keypoints around the object's edges due to the entropy overestimation; MINT regularizes the keypoint towards the center of the object. Figure 12 depicts the contribution of the regularization losses for economizing the number of used keypoints in the come-and-go situation, allowing MINT to outperform the other models.

MINT w/o Temp. refers to our method MINT operating on single images without the losses that operate temporally over two images. MINT w/o Temp. requires (1) removing the masked conditional entropy loss $\lambda_{\text{MCE}} = 0$, and (2) the information

[8]Video results for the ablation study: https://sites.google.com/view/mint-kp/ablations

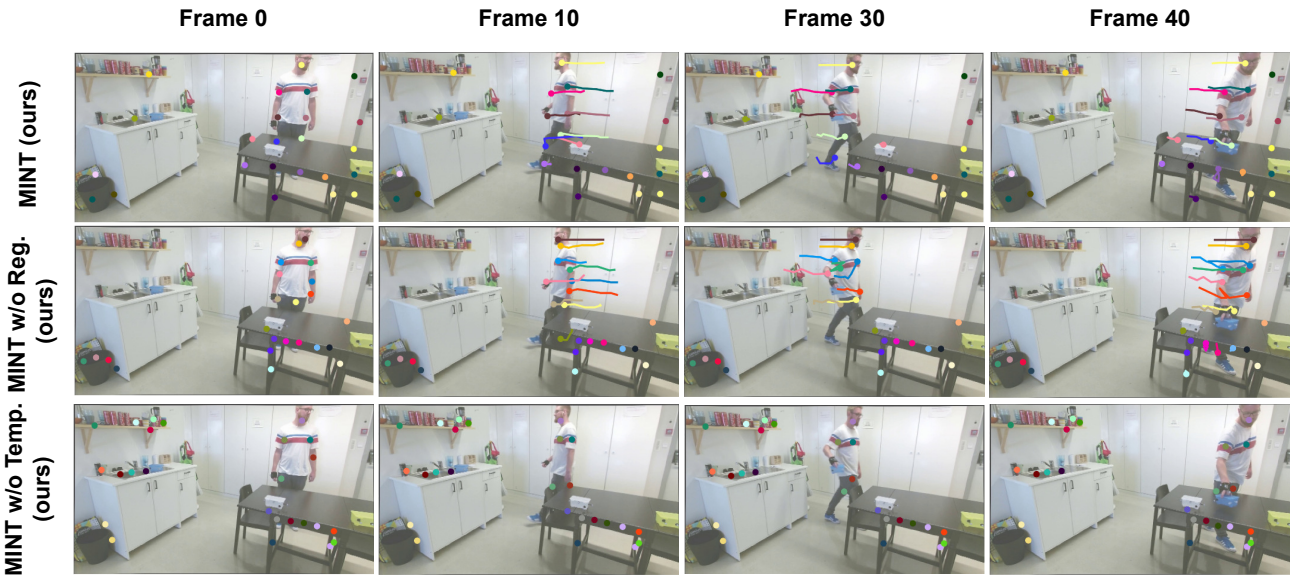

Figure 11. Crowded scenes. A video from SIMITATE dataset with a human moving in a room. We compare MINT with its ablations MINT w/o Reg. and MINT w/o Temp. from Appendix E.1

transportation loss $\lambda_{\text{IT}} = 0$.
This ablation proposes good performance for keypoint detection on static images when using the losses that operate on a single image. Table 5 shows that MINT w/o Temp. can detect 85% of the objects in the scene while distributing the keypoint reasonably. Figure 13 shows that the MINT w/o Temp. assigns keypoints to objects in the scene successfully. Figure 11 shows that MINT w/o Temp. can detect the human and objects in the background, but due to the lack of temporal information, it does not concentrate on the moving objects (*e.g.*, the hand of the human).

Overall, the full MINT model (*cf.* Table 5) trades off the need for good detection and tracking performance, but with a reasonable distribution of keypoints, to adequately represent the information in the video when minimizing our information theoretic losses (*i.e.*, maximizing the covered information entropy spatio-temporally), as dictated by Propositions 2.1 and 2.4.

E.2. Hyperparameters

Table 6 provides the hyperparameters used for CLEVRER (Yi et al., 2019) in our experiments. We use the same values for all other datasets, *i.e.*, also for MIME (Sharma et al., 2018), SIMITATE (Memmesheimer et al., 2019), and MAGICAL (Toyer et al., 2020). The only exceptions are the activation threshold γ, the std for heatmap σ_{G_i} and the threshold of the heatmap τ, where these values depend on the size of the input image (*i.e.*, $\gamma = 15$, $\sigma_{G_i} = 9.0$, $\tau = 0.1$ for MIME, $\gamma = 10$, $\sigma_{G_i} = 9.0$, $\tau = 0.5$ for SIMITATE, $\gamma = 10$, $\sigma_{G_i} = 7.0$, $\tau = 0.3$ for MAGICAL). Our method requires a sequence of 2 frames for the loss computation, and we found that the batch size does not affect the training and can be chosen based on the available GPU resources. In our experiments, we used a PC with a GPU NVIDIA Tesla V100-DGXS-32GB. MINT consumes around 5GB of GPU memory for a batch size of 32 and trains the model in around 1 hour and 5 minutes (for each seed). We use the same weights for the losses in all experiments over different datasets, which suggests that the model is robust against the hyperparameters. Moreover, we ran additional experiments on our benchmark with different sets of hyperparameters (*cf.* Table 7), and the results were always close, which provides further proof of the robustness of our method.

E.3. Baselines

Video structure (Minderer et al., 2019) is an unsupervised method for learning keypoint-based representation from videos. Video structure learns a keypoint detector $\phi^{det}(v_t) = x_t$ for a video sequence v_t that captures the spatial structure of the objects in each frame in a set of keypoints x_i. It learns a reconstruction model ϕ^{rec} that reconstructs frame v_t from its keypoint representation x_t and the first frame of the sequence v_1. An additional skip connection from the first frame to the

Table 6. Hyperparameters

Parameter name	Value	Parameter name	Value	Parameter name	Value
learning rate	0.001	clip value	10.0	weight decay	0.00001
epochs	100	num keypoints K	25	number of stacked frames	3
activation threshold γ	15	entropy region size $\sqrt{\lvert R\rvert}$	3	std for heatmap σ_{G_i}	9.0
Threshold for heatmap τ	0.1	Thresholded heatmap scale η	3.5	CE contribution (IT) κ	0.5
movement weight (IT) m_d	1.0	ME weight λ_{ME}	100	MCE weight λ_{MCE}	100
IT weight λ_{IT}	20	active status weight λ_{s}	10	overlapping weight λ_{o}	30

Table 7. Hyperparameters experiments.

Hyperparams	**DOP** ⇑	**TOP** ⇑	**UAK** ⇓	**RAK** ⇓
$\kappa = 0.9$ $\lambda_{IM} = 100$				
$\lambda_{IT} = 10, \lambda_s = 0, \lambda_o = 10, \beta = 4$, m=0	0.917 ± 0.070	0.894 ± 0.076	7.606 ± 1.509	1.978 ± 0.593
$\lambda_{IT} = 10, \lambda_s = 0.1, \lambda_o = 10, \beta = 4$, m=0	0.913 ± 0.073	0.889 ± 0.078	5.813 ± 1.325	1.887 ± 0.587
$\lambda_{IT} = 10, \lambda_s = 0.1, \lambda_o = 10, \beta = 4$, m=1	0.879 ± 0.097	0.859 ± 0.102	2.468 ± 1.087	1.710 ± 0.585
$\kappa = 0.5$ $\lambda_{IM} = 100$				
$\lambda_{IT} = 10, \lambda_s = 1, \lambda_o = 10, \beta = 4$, m=1	0.870 ± 0.107	0.852 ± 0.110	2.418 ± 1.121	1.606 ± 0.574
$\lambda_{IT} = 10, \lambda_s = 0, \lambda_o = 1, \beta = 4$, m=0	0.929 ± 0.062	0.907 ± 0.068	8.611 ± 1.686	2.228 ± 0.696
$\lambda_{IT} = 10, \lambda_s = 5, \lambda_o = 10, \beta = 2$, m=1	0.857 ± 0.115	0.838 ± 0.118	1.287 ± 0.689	1.090 ± 0.438
$\lambda_{IT} = 10, \lambda_s = 5, \lambda_o = 30, \beta = 2$, m=1	0.851 ± 0.117	0.832 ± 0.121	1.646 ± 0.706	1.050 ± 0.390
$\lambda_{IT} = 10, \lambda_s = 5, \lambda_o = 30, \beta = 4$, m=1	0.857 ± 0.115	0.838 ± 0.118	1.102 ± 0.703	1.269 ± 0.469
$\lambda_{IT} = 20, \lambda_s = 5, \lambda_o = 30, \beta = 4$, m=1	0.856 ± 0.117	0.838 ± 0.121	1.697 ± 1.232	1.391 ± 0.544
$\lambda_{IT} = 20, \lambda_s = 5, \lambda_o = 30, \beta = 4$, m=0	0.859 ± 0.113	0.839 ± 0.117	1.021 ± 0.599	1.283 ± 0.479
$\lambda_{IT} = 20, \lambda_s = 5, \lambda_o = 30, \beta = 2$, m=0	0.861 ± 0.107	0.839 ± 0.112	1.567 ± 0.728	1.207 ± 0.462
$\lambda_{IT} = 20, \lambda_s = 10, \lambda_o = 30, \beta = 4$, m=1	0.844 ± 0.120	0.826 ± 0.123	1.100 ± 0.716	1.121 ± 0.451
$\kappa = 0.7$ $\lambda_{IM} = 100, \lambda_{IT} = 20, \lambda_s = 10, \lambda_o = 30$				
$\beta = 4$, m=1	0.845 ± 0.120	0.826 ± 0.123	1.256 ± 0.975	1.041 ± 0.450
$\beta = 4$, m=0	0.848 ± 0.117	0.829 ± 0.121	1.270 ± 0.766	1.068 ± 0.442
$\beta = 2$, m=1	0.846 ± 0.120	0.826 ± 0.125	1.545 ± 0.946	1.020 ± 0.384
$\beta = 2$, m=0	0.839 ± 0.118	0.818 ± 0.122	1.088 ± 0.647	0.948 ± 0.352
200 epochs $\lambda_{IM} = 100, \lambda_{IT} = 20, \lambda_s = 10, \lambda_o = 30$, m=1				
$\kappa = 0.9, \beta = 2$	0.842 ± 0.124	0.821 ± 0.128	2.121 ± 0.736	1.002 ± 0.387
$\kappa = 0.5, \beta = 4$	0.835 ± 0.123	0.817 ± 0.126	1.194 ± 0.797	1.047 ± 0.412
$\kappa = 0.7, \beta = 4$	0.843 ± 0.126	0.825 ± 0.129	1.003 ± 0.649	1.045 ± 0.419
$\kappa = 0.9, \beta = 4$	0.836 ± 0.125	0.818 ± 0.128	1.441 ± 1.390	1.004 ± 0.384

reconstruction model output changes its actual task to predict $v_t - v_1$; hence $v_t - v_1 = \phi^{rec}(v_1, x_t)$.
The keypoint detector is trained to optimize three losses:
(1) L2 image reconstruction loss

$$\mathcal{L}_{\text{image}} = \sum_t ||v - \hat{v}||_2^2 \,, \tag{44}$$

where v is the true and $\hat{v}$ is the reconstructed image.
(2) Temporal separation loss penalizes the overlap between trajectories within a Gaussian radius σ_{sep}

$$\mathcal{L}_{\text{sep}} = \sum_k \sum_{k'} exp(-\frac{d_{kk'}}{\sigma_{\text{sep}}}) \,, \tag{45}$$

where $d_{kk'} = \frac{1}{T} \sum_t ||(x_{t,k} - \overline{x}_k) - (x_{t,k'} - \overline{x}_{k'})||_2^2$ is the distance between the trajectories of keypoints k and k'.
(3) Sparsity loss adds an L1 penalty on the keypoint intensity μ (the mean value of the corresponding feature map) to encourage keypoints to be sparsely active

$$\mathcal{L}_{\text{sparse}} = \sum_k |\mu_k| \,. \tag{46}$$

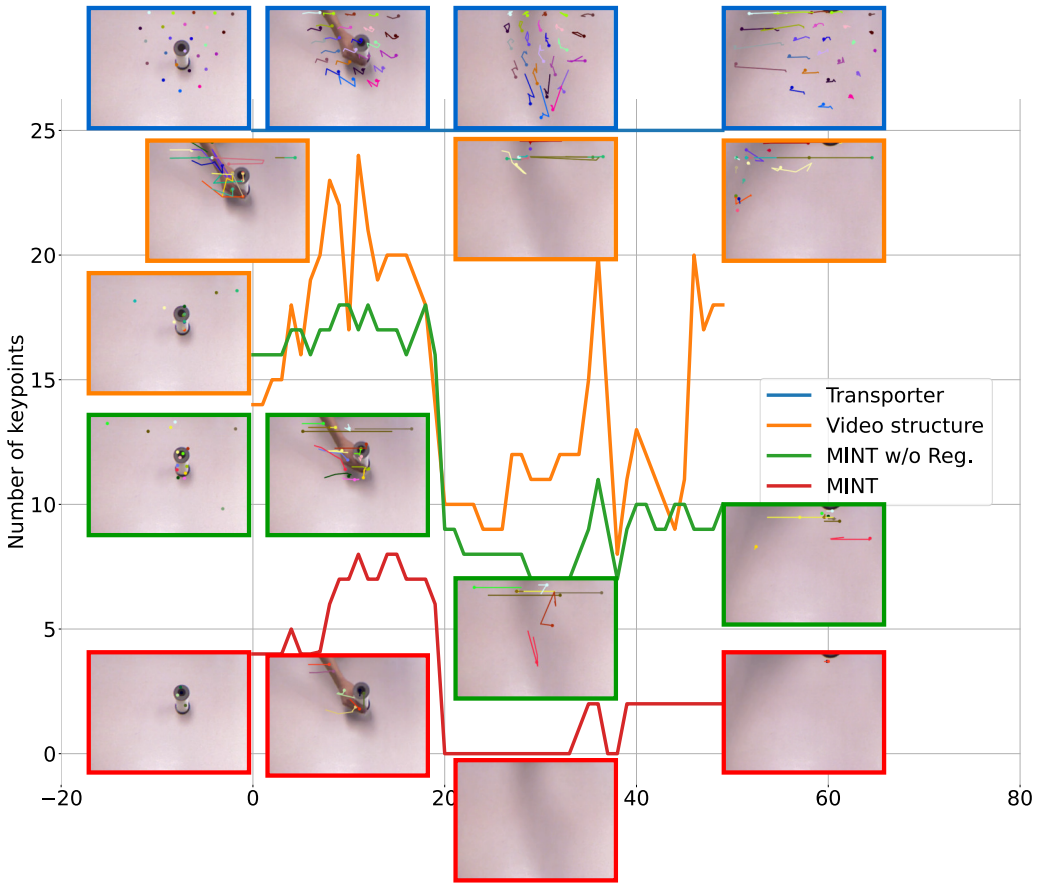

Figure 12. Come-and-go scenario. In a manipulation video from MIME, the hand enters after the start of the video and departs before the end. We plot the number of active keypoints w.r.t. timesteps. The results show the role of regularization in our Method MINT in improving resource assignment and economizing the number of active keypoints.

A keypoint is active if the intensity is higher than a specific threshold th_μ, the threshold is a hyperparameter that has to be tuned depending on the video.

Transporter (Kulkarni et al., 2019) is a neural network architecture for discovering keypoint representations in an unsupervised manner by transporting learned image features between video frames using the keypoint bottleneck. During training, spatial feature maps $\phi(x)$ and keypoint coordinates $\psi(x)$ are predicted for a source frame x_s and a target frame x_t using a ConvNet and KeyNet (Barroso-Laguna et al., 2019). The keypoint coordinates are transformed into Gaussian heatmaps $h_{\psi(x)}$.

A transported feature-map $\hat{\phi}(x_s, s_t)$ is generated by suppressing both sets of keypoint location in $\phi(x_s)$ and composing into the feature maps around the keypoints from x_t:

$$\hat{\phi}(x_s, s_t) \triangleq (1 - h_{\psi(x)_t}) \cdot (1 - h_{\psi(x)_t}) \cdot \phi(x_s) + h_{\psi(x_t)} \cdot \phi(x_t) \ . \tag{47}$$

An additional refiner net learns to map the transported features maps into an image $\hat{x}_t$. The learning objective is reconstructing the target image x_t from the process. Hence, the Transporter optimizes the L2 reconstruction error $\mathcal{L} = ||x_t - \hat{x}_t||_2^2$.

Transporter-modified is a modified version of the transporter baseline (Kulkarni et al., 2019). The original implementation of the method has two potential bottlenecks: (1) the feature maps $\phi(s)$ have a receptive field of size 24 for each position, for an input of size 128x128; and (2) the resolution of the feature maps between which the features are transported is 32x32. For fair comparison to our method, which uses an entropy region of size 3x3, we modified the network architecture of ConvNet $\phi(x)$ to have (1) a receptive field of 7 and (2) a feature map of size 122x122. We call the new architecture Transporter-modified.

The experimental results show that the Transporter-modified model outperforms the original Transporter in the quantitative

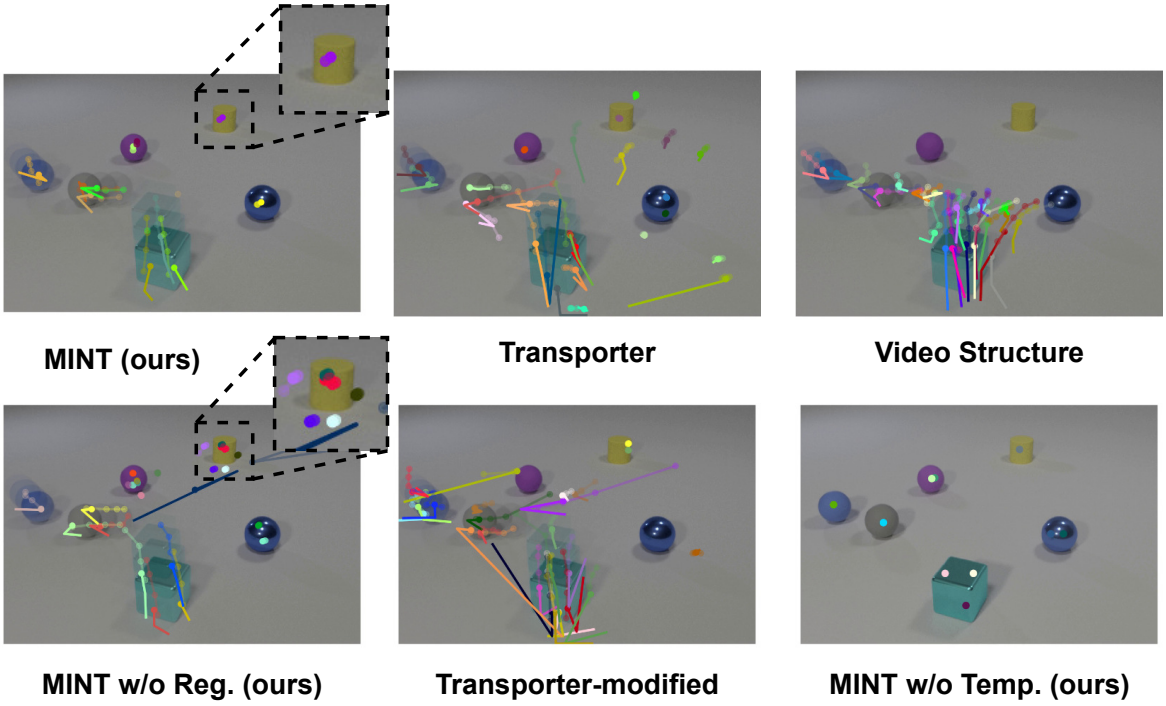

Figure 13. Qualitative results on CLEVRER dataset for Task I (object detection and tracking) and Task II (learning dynamics). We include results for all baselines described in Appendix E.3.

evaluation on CLEVRER dataset (Yi et al., 2019) (*cf.* Table 1). We want to refer to the visual results in Figure 13 that show that the keypoints detected by the original Transporter (top middle image) are more stable than those detected by the Transporter-modified (bottom middle image). We argue this behavior is due to the smaller receptive field leading the model to assign keypoints to features instead of objects, and thus keypoints jump to similar features in different objects.

E.4. Interaction Network Architecture

The interaction network (IN) (Battaglia et al., 2016) is a model developed for learning the interaction relations between physical objects to infer the physics of the environment. The interaction network treats the objects as nodes of a graph, with the relations as edges. In our case, we use the keypoints as object nodes, with the coordinates, status, and positional encoding as features. We form a fully connected graph of the keypoints, with no relational features for edges.

The interaction network used in our experiments has two sub-models; a relational model and an object model. The relational model uses the relational information and object attributes to predict the effects of all interactions. The object model uses the effects to update the features of the object. We encode node features before passing them to the interaction network. After one pass through the interaction network, we decode the features into coordinates for the prediction task, and we add another prediction head for the action decoding in the imitation learning task.

E.5. Imitation Learning Results

CNN-agent. The CNN agent is trained from scratch for every environment. Note that the state space for the CNN agent, the image pixels, is one order of magnitude higher than the keypoints' features. For fair comparison, we train the CNN agent longer (twice the epochs used for training the MINT-based agent to counteract for MINT's pretraining).

The CNN feature extractor consists of 5 convolution blocks, each consisting of a 2D convolutional layer with a ReLU activation function and a batch normalization layer. The input to the model is a sequence of 4 color images stacked over the channel axis; hence, the input size is $12 \times 96 \times 96$. The layers have 64, 128, 128, 128, and 128 filters, with a kernel size of 3 and stride of 2, except the initial layer, which has a kernel size of 5 and stride of 1. The output of the last block is flattened and passed to a linear layer to provide the final features. A policy model uses the features to infer the actions. The policy

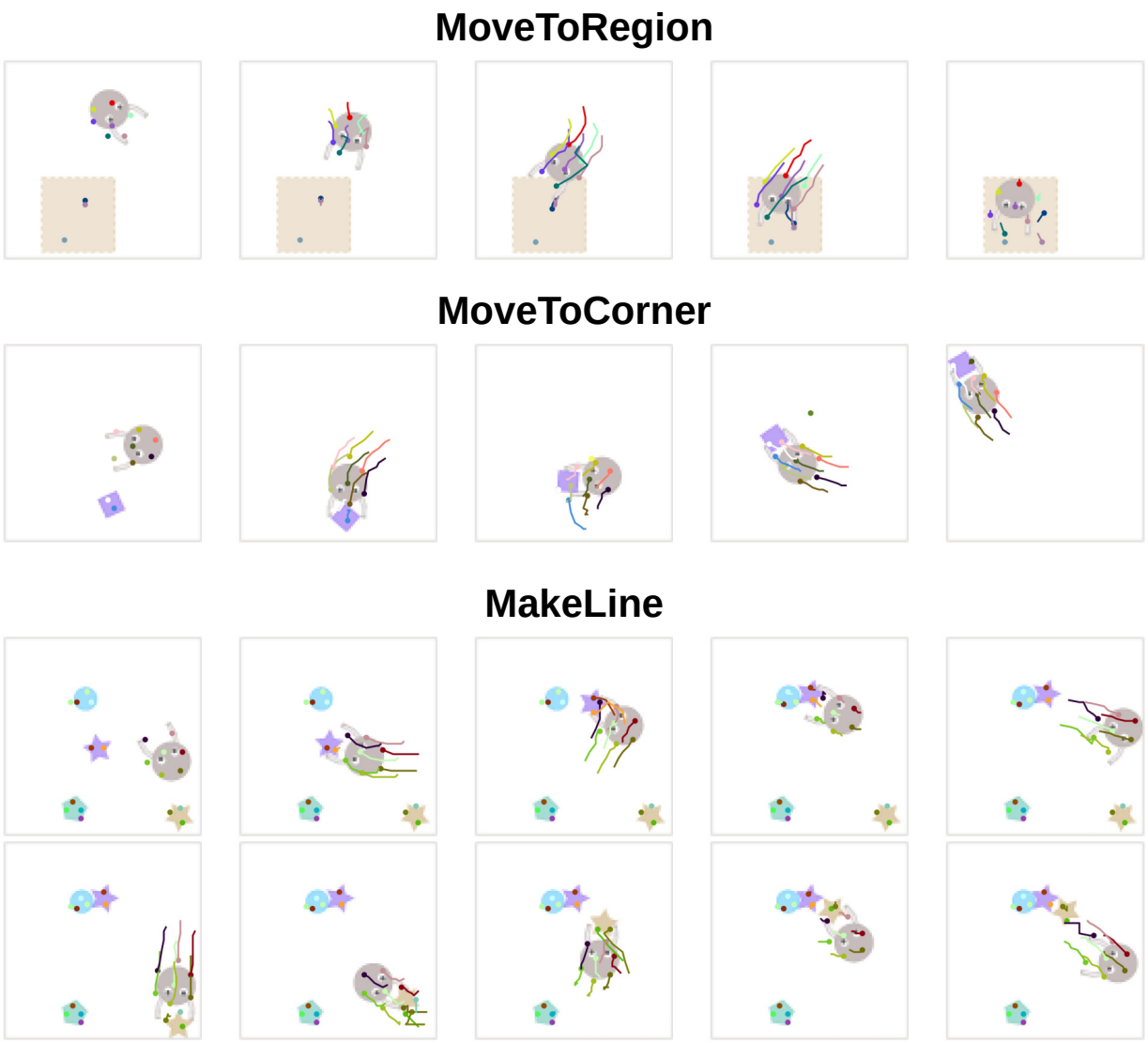

Figure 14. Rollouts from MINT agent in MAGICAL (Toyer et al., 2020) dataset.

model is a multi-layer perceptron with 4 linear layers of sizes 128, 64, 32, and 32. The output matches the action dimension of MAGICAL environment which is 18.

We also provide visualizations of the learned policies of the MINT-based agent on the three environments of MAGICAL in Figure 14. The visualization shows that MINT can assign reasonable keypoints for the agent and all the objects in the environment. The imitation agent can solve the first two tasks **MoveToRegion** and **MoveToCorner**, but it struggles with the last task **MakeLine**. The agent receives a score of 1.0 when it sorts all 4 objects in one line, while it gets a score of 0.5 for putting 3 out of 4 in one line. Our imitation agent could (occasionally) sort only 3 out of 4 in the depicted environment (which led to an overall 0.2 mean score over 5 seeds – Table 3), despite being able to assign keypoints to all objects. The results suggest that there is a problem in encoding the relational features between keypoints, hindering the agent to reason upon getting the right locations. We argue that further investigation of the appropriate model to pool information from the keypoints is necessary to solve this most challenging task, but this is out of scope of the current work.

E.6. Additional Video Results

We provide additional video results on the website of our project: `https://sites.google.com/view/mint-kp`.

F. Code

Our code is available under an open-source license at: https://github.com/iROSA-lab/MINT .
We provide instructions to run the code, with sample datasets to reproduce the results in the paper.

The implementation of video structure (Minderer et al., 2019)[9] uses outdated libraries. Due to compatibility reasons, we reimplemented their code in PyTorch with our best effort. We adapted the implementation of Transporter from Li et al. (2020) [10] into the codebase of MINT.[11]

G. Additional Related Work Discussion

Information-theoretic approaches in machine learning. Information-theoretic principles proved advantageous in training and understanding machine learning models (Yu et al., 2021). Different information measures aim to describe a random variable's behavior due to a probability density function. The probability density function is normally unknown, and machine learning methods usually estimate it (Pardo, 2018; Avi-Aharon et al., 2020). In our method, we use kernel density estimation (KDE) (Parzen, 1962) to estimate the probability density function for a region of pixels. Various information-theoretic quantities were used in machine learning for different applications; examples are the cross-entropy loss for classification (Good, 1992; Goodfellow et al., 2016), maximum entropy regularization in reinforcement learning (Peters et al., 2010; Haarnoja et al., 2018), mutual information for self-supervised learning and interpretability (Rakelly et al., 2021; Zhang et al., 2018), and KL divergence for training deep energy models (Yu et al., 2020). Our approach uses Shannon's definition of entropy (Shannon, 2001) to compute the local image entropy. With image entropy, we estimate joint entropy, conditional entropy, and mutual information and develop our information-theoretic losses.
Temporal information plays an important role for many downstream tasks. Recent methods in neural video compression (Li et al., 2022) propose to estimate spatial-temporal intra-frame entropy over quantized latent representation. In our work, we opted to use inter-frame entropy estimation with temporal losses that encourage the model to attend to temporal information changes, which proved to provide a strong inductive bias for keypoint detection.
The information maximization principle (InfoMax) (Linsker, 1988) treats the neural network as an information channel and aims to maximize the information transferred through the network. Recent methods in computer vision and natural language processing use the InfoMax principle for self-supervised learning (Oord et al., 2018; Hjelm et al., 2018; Kong et al., 2019). Our method adopts the treatment of the neural network as an information channel in the information maximization loss and extends it to treat the keypoints as transmitters of information, while being completely unsupervised.

Glossary

CV	Computer Vision. 1, 2, 9
DOP	percentage of the detected object. 6, 7, 22, 23, 25
IM	information maximization. 23
IN	Interaction Network. 7, 8
ISE	image spatial entropy. 1, 2, 3, 5, 8, 9, 17, 19, 20
IT	information transportation. 4, 5, 20, 22
MCE	masked conditional entropy. 3, 4, 20, 22
ME	masked entropy. 3, 4, 5, 6, 22
MI	mutual information. 5
MINT	Maximum Information keypoiNTs. 1, 2, 5, 6, 7, 8, 9, 22, 23, 24, 26, 27, 28
MME	monkey model entropy. 17
PoI	Points of Interest. 1, 8, 9
RAK	redundant keypoint assignment. 6, 7, 22, 23, 25
TOP	percentage of tracked objects. 6, 7, 22, 23, 25
UAK	unsuccessful keypoint assignment. 6, 7, 22, 23, 25

[9] https://github.com/google-research/google-research/tree/master/video_structure
[10] https://github.com/pairlab/v-cdn
[11] Our baselines implementation is available in our codebase https://github.com/iROSA-lab/MINT