Object Concepts Emerge from Motion

Haoqian Liang[1], **Xiaohui Wang**[1], **Zhichao Li**, **Ya Yang**[1], **Naiyan Wang**
[1]Beijing University of Posts and Telecommunications
lianghq@bupt.edu.cn, nekomio@bupt.edu.cn, leeisabug@gmail.com,
yangya@bupt.edu.cn, winsty@gmail.com

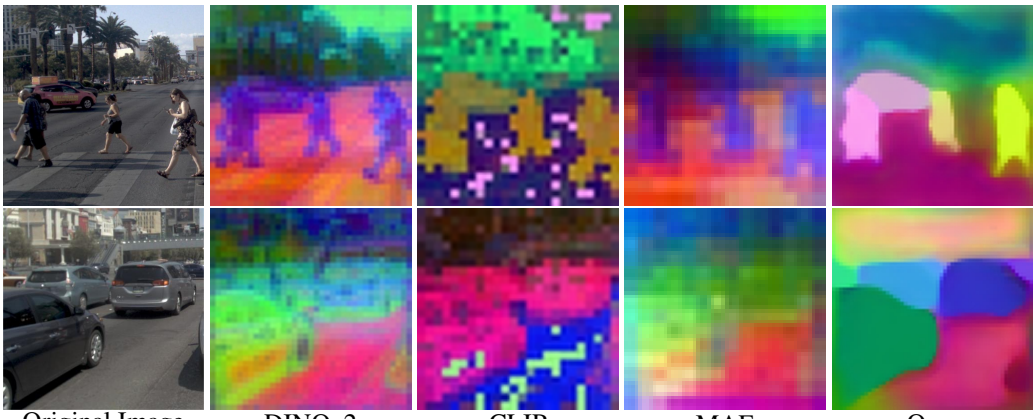

Figure 1: Comparisons with feature maps learned by our method and different visual foundation models. Our method focuses on the unity of object instance, in contrast to other methods emphasize on object class more.

Abstract

Object concepts play a foundational role in human visual cognition, enabling perception, memory, and interaction in the physical world. Inspired by findings in developmental psychology—where infants are shown to acquire object understanding through observation of motion—we propose a biologically inspired framework for learning object-centric visual representations in an unsupervised manner. We were inspired by the insight that motion boundary serves as a strong signal for object-level grouping, which can be used to derive pseudo-instance supervision from raw videos. Concretely, we generate motion-based instance masks using off-the-shelf optical flow and clustering algorithms, and use them to train visual encoders via contrastive learning. Our framework is fully label-free and does not rely on camera calibration, making it scalable to large-scale unstructured video data. We evaluate our approach on three downstream tasks spanning both low-level (monocular depth estimation) and high-level (3D object detection and occupancy prediction) vision. Our models outperform previous supervised and self-supervised baselines and demonstrate strong generalization to unseen scenes. These results suggest that motion-induced object representations offer a compelling alternative to existing vision foundation models, capturing a crucial but overlooked level of abstraction: the visual instance. The implementation can be found here: https://github.com/yulemao/Object_Concepts_Emerge_from_Motion

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Physical AI aims to develop intelligent agents capable of perceiving and interacting with the physical world. A fundamental cognitive capacity required for such agents is the ability to recognize and understand the concept of "object"—a core unit of perception and reasoning. In the human visual system, the importance of object concepts is well-established in neuroscience. As noted by Kellman and Spelke [27], *"this cognitive ability not only supports object recognition and classification, but also plays a crucial role in spatial perception, memory formation, and the interaction between objects and their environment."* Understanding how object concepts are formed and represented in biological systems provides critical insights for building more robust and generalizable visual agents in artificial systems.

However, *what makes an object look like an object?* This is a non-trivial question, as objects can vary drastically in appearance, shape, and motion patterns. Early studies in developmental psychology [27] have demonstrated that the ability to perceive object unity is not innate, but learned during infancy. Infants begin to exhibit evidence of understanding object cohesion from around two months of age, with robust performance observed by four months. These findings suggest that object perception is a learned capacity grounded in sensory experience. Subsequent research [24, 40] has shown that motion cues—particularly common or coherent motion—serve as a powerful signal for infants to infer object boundaries and unity. As the visual system matures, this dynamic understanding is gradually internalized into the ventral visual stream [29, 53], enabling object recognition from static visual inputs alone. Inspired by this developmental trajectory, our work aims to design an unsupervised computational model that mimics this learning process: *beginning with motion-based interactions and evolving toward abstract, appearance-based object concepts.*

Recently, learning universal visual representations through self-supervised or weakly supervised paradigms has gained significant attention, due to their strong performance across a wide range of vision tasks. Among self-supervised approaches, notable examples include the DINO [7, 41] and MAE [23, 64] families, which rely on self-distillation and self-reconstruction mechanisms, respectively, to learn robust feature representations. Another influential direction leverages web-scale image-text pairs, as exemplified by CLIP [45], to align visual and language representations. To better understand what these models capture, we compare the low-dimensional PCA projections of features extracted by DINO, CLIP, and our model (see Fig. 1). We observe that DINO and CLIP tend to focus on semantic *categories*. However, neither method captures the concept of a semantic *instance*—a distinct, coherent object entity—adequately. We argue that existing visual foundation models overlook this crucial level of abstraction, which is fundamental for understanding the physical world.

In this work, we propose a biologically inspired framework for learning visual features that encode object-level semantics. As a first step, we explore this approach in outdoor driving scenarios, which provide rich motion cues arising from both ego-motion and independently moving objects. The key observation that inspires our method is that motion boundaries often align with object boundaries (detailed in Sec. 3.1), which echoes the discoveries in neuroscience that common motion is crucial to the early development of object unity. Based on this, we employ an off-the-shelf optical flow estimation algorithm, followed by a simple clustering technique, to generate pseudo instance masks without human supervision. These instance labels are then used to supervise representation learning via a contrastive objective. Importantly, unlike previous approaches [74, 4], our method does not require camera calibration parameters, allowing it to scale to large and diverse unlabeled video datasets.

Our motion-guided learning paradigm naturally bridges low-level and high-level vision. We validate our method on three downstream tasks: monocular depth estimation (low-level), 3D object detection and occupancy prediction (high-level). Across all model sizes, our method consistently outperforms supervised pretraining on ImageNet-22K and other self-supervised learning methods, demonstrating the effectiveness of learning object-centric representations from motion cues. Moreover, we find that our features are complementary to those from existing foundation models such as DINO—fusing them leads to further performance gains. Interestingly, although our model is trained only on outdoor scenes, it generalizes well to unseen indoor environments. This suggests that the learned features capture object composition and structure, rather than merely memorizing training-set appearances.

To summarize, our key contributions are as follows:

- We propose a biologically inspired paradigm for object-centric visual representation learning. Motivated by studies of infant cognition, we are the first to demonstrate the effectiveness and scalability of using motion as an unsupervised supervisory signal on a large-scale dataset and modern model architectures.
- We introduce a computationally efficient framework that implements this paradigm using off-the-shelf optical flow and simple clustering. Our approach scales naturally to large-scale outdoor driving datasets without requiring camera calibration or manual labels, and supports models of varying capacities.
- We extensively evaluate our models on three downstream tasks—monocular depth estimation (low-level), 3D object detection and occupancy prediction (high-level). Our method outperforms supervised and self-supervised methods across all model sizes, and shows strong generalization to unseen indoor scenes, highlighting the robustness and transferability of the learned object-centric features.

2 Related Work

2.1 Object Discovery

The central aim of object discovery is the identification and localization of objects within visual data, including images and videos, without the prerequisite of instance-level annotations for specific object classes. This paradigm significantly mitigates the need for large-scale, high-quality labeled datasets. Early approaches to object discovery included methods based on object occurrence frequency [25, 26, 54], and techniques utilizing region proposals to select key object bounding boxes through combinatorial optimization [52, 55, 56, 62, 75]. More recently, researchers have proposed numerous learning-based methods built upon the Transformer architecture. These approaches leverage features obtained from powerful pre-trained image models (e.g., DINO) to identify and segment objects via graph-based or spectral clustering techniques [46, 50, 59, 60, 73]. Another line of research adopts an object-centric perspective, frequently utilizing scene generation or reconstruction methodologies to derive learning signals, that involve decomposing scenes into their constituent parts (e.g., objects, background) and learning their respective representations [5, 18, 36, 38]. Similarly, another class of methods utilizes motion and multi-modal information as supervision, using the motion consistency of 2D or 3D points as a cue to distinguish objects from the background [34, 51, 61, 72]. To address the high demand for input dependencies of these works, our method only requires the simplest optical flow and clustering process to acquire object masks in an unsupervised manner. These masks subsequently serve as pseudo labels for single-image representation learning.

2.2 Visual Foundation Models

Visual foundation models aim to learn broadly applicable and transferable visual representations by pre-training on large-scale data. These learned general representations are intended to be transferred to downstream tasks by fine-tuning or prompting. Various self-supervised learning paradigms have been proposed for visual foundation models. Contrastive-learning-based methods pull together representations of different augmented views of the same image (positive samples) in an embedding space, while pushing apart representations of different images (negative samples) [8–10, 14, 22, 70]. Building upon this, subsequent self-distillation methods utilize "teacher" signals moving averaged by the student model itself for self-guided training, achieving excellent performance without relying on negative sample pairs [7, 12, 19, 41]. On the other hand, inspired by masked language modeling in natural language processing, masked-autoencoder-based methods learn representations by randomly masking portions of an image and training the model to reconstruct the masked content [2, 23, 44, 58, 64]. There is also some works that jointly learn the embeddings and predictions for more semantically rich and general features [1, 3]. Furthermore, despite different supervisory approaches, methods employing weak supervision, such as through text, have also made significant progress in the field of foundation models [45]. However, as illustrated in Fig. 1, all these pretrained models provide *semantic class* features rather than *semantic instance* features. We argue that semantic instance features could be beneficial for downstream tasks that require instance-level separation, such as object detection.

3 Method

To build an object-centric visual representation that generalizes across tasks and environments, we propose a biologically inspired learning framework centered on motion cues. Rooted in cognitive

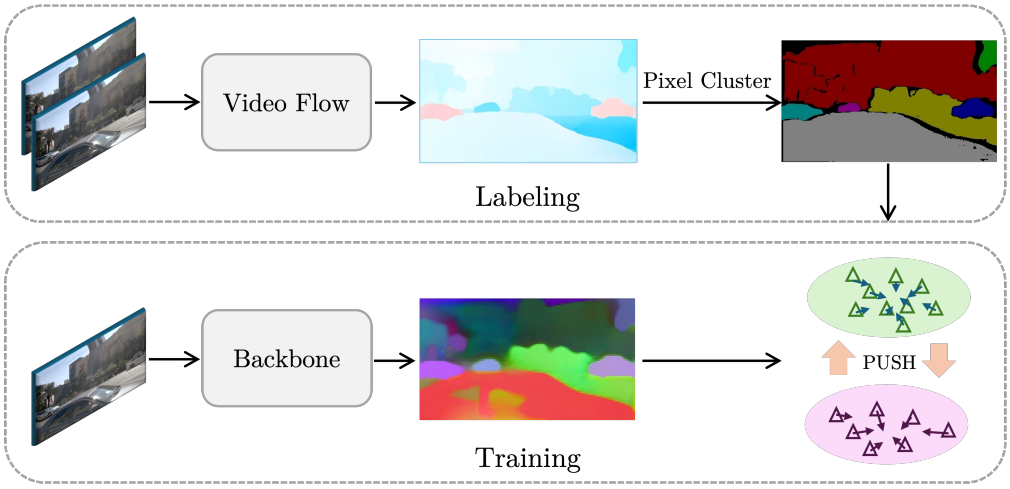

Figure 2: Pipeline of the proposed method.

developmental insights, our approach leverages the observation that coherent motion often indicates objecthood—an idea supported by infant perception studies and geometric reasoning in dynamic scenes. In this section, we introduce our method, which consists of three key components: (1) a geometric analysis revealing how motion boundaries correlate with object boundaries, (2) a data processing pipeline that extracts motion-induced pseudo-labels from large-scale video data, and (3) an unsupervised training objective designed to learn robust and transferable features from these labels. Together, these components form a scalable and calibration-free paradigm for learning object-level semantics from raw videos. The whole pipeline is shown in Fig. 2.

3.1 Geometric Insights

A central insight of our approach is that *motion boundaries often align with object boundaries*. It is obvious that if the object itself moves, its flow boundary can naturally serve as the object boundary. In this section, we provide a geometric and mathematical justification for why ego-motion can also be used to separate different objects under the assumption of rigid scenes.

Let $\mathbf{p} = (u, v)$ denote a pixel in the image domain, and $D(u, v)$ its corresponding depth. Assuming a pinhole camera model with intrinsic matrix $\mathbf{K}$, $\mathbf{p}$ is the pixel projected by the 3D point $\mathbf{P}$. Under rigid motion, the 3D scene point undergoes a transformation via camera pose change $(\mathbf{R}, \mathbf{t}) \in SE(3)$, resulting in a new image projection $\mathbf{p}'$ in the next frame. Projecting $\mathbf{P}'$ back into the image plane yields:

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \sim \mathbf{K} \cdot \mathbf{P}' = \mathbf{K} \left(\mathbf{R} \cdot D(u, v) \cdot \mathbf{K}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \mathbf{t} \right) \tag{1}$$

The optical flow is then computed as the pixel displacement. This means that the optical flow $\mathbf{F}(u, v)$ is a function of the depth $D(u, v)$, the camera motion $(\mathbf{R}, \mathbf{t})$, and the camera intrinsics $\mathbf{K}$. We summarize this dependency as:

$$\mathbf{F}(u, v) = \begin{bmatrix} u' - u \\ v' - v \end{bmatrix} = \phi(D(u, v); \mathbf{R}, \mathbf{t}, \mathbf{K}) \tag{2}$$

Taking the spatial gradient of the flow field gives:

$$\nabla \mathbf{F}(u, v) = \frac{d\phi}{dD} \cdot \nabla D(u, v) \tag{3}$$

This expression indicates that discontinuities in the flow field—i.e., motion boundaries—can arise from large gradients in the depth map. Under the assumption of rigid motion, these motion boundaries serve as reliable proxies for object boundaries. This geometric insight underpins our approach of utilizing motion cues to derive instance-level supervision.

This concept aligns with foundational theories in computer vision. David Marr, in his seminal book [39], articulated that:

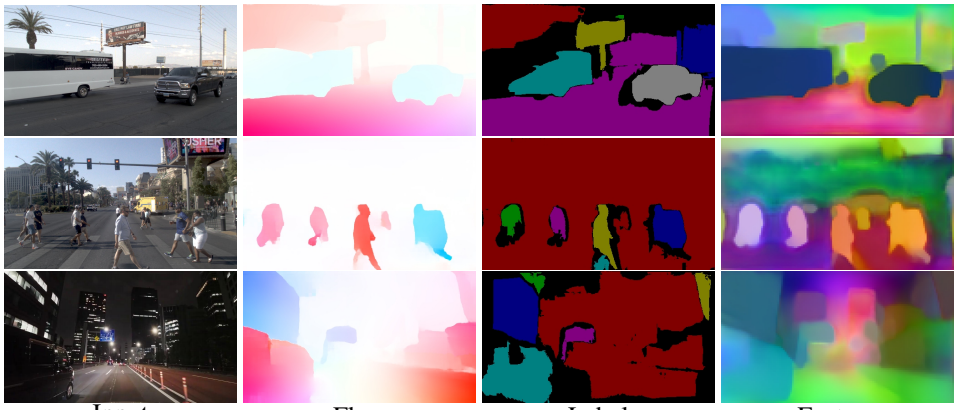

Figure 3: Examples of the pseudo-label generation results and the output features.

"...the velocity field of motion in the image varies continuously almost everywhere, and if it is ever discontinuous at more than an isolated point, then a failure of rigidity (like an object boundary) is present in the outside world. In particular, if the direction of motion is ever discontinuous at more than one point—along a line, for example—then an object boundary is present."

A notable advantage of our method is its independence from camera calibration. The necessary geometric information is inherently encoded within the optical flow, enabling us to train on large-scale, uncalibrated video datasets. This approach enhances scalability and broadens the applicability of our framework across diverse real-world scenarios.

3.2 Data Processing

Data Sources. We use two datasets in our approach: OpenDV-YouTube [67] and nuPlan [20]. Both datasets provide a large amount of high-quality and diverse unlabeled video data. OpenDV-YouTube contains videos collected from more than 244 cities all over the world, resulting in a total of 1747 hours of front-view videos. nuPlan provides 8 different camera views. It collects 1200 hours of driving data from 4 cities, 120 hours of which were recorded with 8 different camera views. We merged the two datasets and obtained approximately 2,700 hours of raw video data in total.

Optical Flow Estimation. We apply the pipeline of VideoFlow [49] to extract optical flow information from videos. The model takes five frames as input and outputs the optical flow for the middle three frames. We sample each clip with 0.3s intervals, and for each clip, we select the first frame within two consecutive frames as input, which spans 1 second.

Pixel Cluster. For all optical flow data generated by VideoFlow, we perform a simple Breadth-First Search (BFS) to cluster objects. Our algorithm takes the optical flow, the forward-backward consistency check result, and two thresholds θ_f and θ_s as input. For each pixel that satisfies the forward-backward consistency check, all neighboring pixels with a flow difference smaller than θ_f are considered to belong to the same object. θ_s is the minimum number of pixels to form a cluster. Pseudo-code of our algorithm is provided in the appendix.

Results. We set two thresholds to $\theta_f = 1.5, \theta_s = 100$. Fig. 3 shows some example results. As illustrated in the pseudo-label visualizations, the proposed algorithm successfully segments objects exhibiting significant movement, such as moving cars and pedestrians. Furthermore, because objects at different depths exhibit different apparent motion in the image even when they are stationary, the proposed algorithm is also able to segment foreground instances such as trees and signs. The pseudo-labels exhibit under segmentation due to weak motion cues or errors in optical flow estimation. Such cases are explicitly handled in the design of the loss function. We retained all samples with at least two pseudo-labels(i.e., at least one foreground cluster) and successfully obtained a total of 48 million images along with their corresponding pseudo-labels for pre-training.

3.3 Pre-training

Overall Structure. Our network architecture follows the design proposed in [28]. The network takes one image as input. Due to the need for high-resolution feature maps, we choose backbone

networks(*e.g.* , ResNet [21], Swin [35]) whose computational cost scales linearly with input size. These features are then processed by a Feature Pyramid Network (FPN). Similar to the semantic segmentation branch in [28], the information from all levels of the FPN pyramid is merged into a simple output. The resulting feature map has a spatial resolution of 1/4 the input size and a channel dimension of 64. A 2× bilinear upsampling is then applied, and each feature vector is normalized to yield the final output features.

Training Loss. Based on the labels derived from the optical flow and the output feature map, we design a simple yet effective loss function. As discussed in Sec. 3.2, the pseudo-labels can only extract a subset of the instances, making it inappropriate to cluster all background regions together. Since the number of background pixels is usually significantly larger than that of instance pixels, we treat the label with the highest pixel count as the background. The loss between any two background pixels is ignored. The loss function between two pixels i and j is defined as follows:

$$L(i,j) = \begin{cases} {\|F_i - F_j\|_2}^2, & y_i = y_j \neq 0 \\ \max\{m - \|F_i - F_j\|_2, 0\}^2, & y_i \neq y_j \\ 0, & y_i = y_j = 0 \end{cases} \tag{4}$$

where F_i and F_j are the feature vectors of the final output feature map of the network, m is a margin parameter, y represents the instance label derived from the optical flow. $y = 0$ denotes the background. We set the margin parameter m to 1.0 in our implementation.

The total loss over all sampled pixel pairs is defined as:

$$L_{total} = \frac{1}{N} \sum_{i,j} L(i,j). \tag{5}$$

4 Experiments

To validate the effectiveness of our method across the vision spectrum, we conduct comprehensive experiments on both low-level and high-level vision tasks. Our core hypothesis is that the model, by learning from low-level cues, develops an internal understanding of object composition, which subsequently benefits high-level semantic reasoning. Conversely, this object-centric representation also enhances performance on low-level tasks by providing richer contextual cues. We evaluate our models on three representative downstream tasks: monocular depth estimation (low-level), 3D object detection, and 3D occupancy prediction (high-level).

4.1 Implementation Details

We implement the proposed method using PyTorch [43] and mmPretrain [11]. We train models on Swin Transformer [35] (Tiny to Large) and ResNet-50 [21]. All Swin models use a window size of 7, while the B and L variants of SimMIM [64] and Semantic-SAM [30], which are used for comparison, adopt a larger window size of 12. This larger window is usually beneficial due to the increased context, at the expense of higher computational cost. AdamW optimizer [37] with a weight decay of 0.05 is adopted. All models are trained for 200 epochs using a cosine decay learning rate scheduler and 10 epochs of linear warm-up. The initial learning rate is set to 0.001 and batch size is set to 2048. All input images are cropped and resized to a resolution of 224×224. We employ a data augmentation strategy that includes random flipping, brightness, and gamma adjustment. We sample 200 labeled pixels from each image for training. We further fine-tune the models for 20 epochs with an initial learning rate of 2×10^{-5} and a weight decay of 10^{-4}. During fine-tuning, two random crops are extracted from each input image, and the loss is calculated both within each crop and between the two. This fine-tuning process further enhances the separation of distant objects in large images. All downstream models are trained with official open-sourced code for comparison. During fine-tuning on downstream tasks, only the pretrained weights of the backbone are utilized for a fair comparison.

4.2 Qualitative Results

The fourth column in Fig. 3 shows PCA projections of our model's features. Thanks to the generalization of the backbone network, the features reveal a key strength: the model distinguishes many objects not annotated in the pseudo-labels—such as distant cars, pedestrians, and even static structures like buildings and poles. This suggests that our model goes beyond mimicking pseudo-labels and learns a more general, object-centric representation. Fig. 4 further visualizes similarity maps from selected

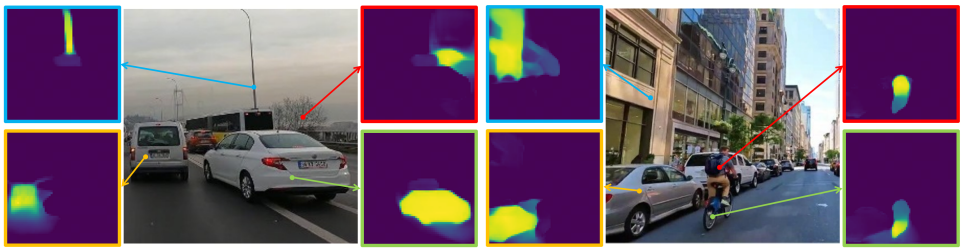

Figure 4: Similarity visualization for a set of reference points.

Table 1: Quantitative evaluation of DCDepth [57] on the KITTI Eigen split using different pretraining.

Method	Backbone	SILog ↓	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
ImageNet-22K	Swin-T	7.455	0.055	0.165	2.182	0.082	0.969	0.996	0.999
Semantic-SAM [30]	Swin-T	7.346	0.055	0.165	2.169	0.082	0.971	0.996	0.999
DINOv2 [41]	ViT-S	7.119	0.052	0.158	2.153	0.079	0.974	0.997	0.999
ImageNet-22K	Swin-L	6.891	0.051	0.145	2.044	0.076	0.977	0.997	0.999
Semantic-SAM [30]	Swin-L	6.713	0.049	0.137	2.007	0.074	0.979	**0.998**	**1.000**
SimMIM [64]	Swin-L	**6.542**	0.048	0.130	1.941	0.073	0.979	**0.998**	0.999
Ours	Swin-T	6.991	0.051	0.145	2.016	0.077	0.975	0.997	0.999
Ours	Swin-S	6.736	0.049	0.138	1.981	0.075	0.978	0.997	0.999
Ours	Swin-B	6.598	0.048	0.131	1.939	0.073	**0.981**	0.997	0.999
Ours	Swin-L	6.558	**0.047**	**0.129**	**1.929**	**0.072**	**0.981**	0.997	0.999

reference points. The sharp boundaries and clear object separation confirm that our features capture consistent, instance-level semantics, even without explicit supervision.

4.3 Monocular Depth Estimation

We evaluate our model on the KITTI dataset [16] using the standard Eigen split [15], with DCDepth [57] as the decoder. As shown in Tab. 1, our model consistently outperforms both supervised ImageNet-22K pretraining and models pretrained on the Semantic-SAM [30], which is a weakly supervised method utilizing large-scale pseudo segmentation annotations.

Our approach achieves superior performance across all backbone sizes. For instance, with Swin-Tiny, our model reduces the RMSE to 2.016 (compared to 2.169 from Semantic-SAM) and improves the δ_1 accuracy to 0.975. These results are even comparable to Swin-Large with ImageNet-22K pretraining. As the backbone scale increases, the performance of our method improves steadily.

As shown in Tab. 5, combining our features with DINO leads to consistent and significant performance improvements. While our method alone already outperforms using either DINO or ImageNet-22K pretrained features in isolation (rows 1–3), the best result is achieved when concatenating our features with DINO pretrained features (row 6), reaching the lowest SILog (6.796) and a competitive RMSE (2.014).

This highlights the complementary nature of the two representations: DINO focuses more on semantic category-level cues, while our method emphasizes instance-level object structure derived from motion cues. Fusing them allows the model to leverage both semantic context and object-centric information, leading to improved depth estimation performance.

Tab. 2 further shows the results on the official KITTI online leaderboard. Our method outperforms other methods in the primary metric (SILog) and also achieves competitive performance across the other evaluation metrics.

4.4 3D Object Detection

We evaluate our learned visual representations on the nuScenes dataset [6] for the 3D object detection task, using BEVFormer V2 [31, 66] as the detection framework. We compare our method against a diverse set of pretraining strategies, including supervised ImageNet-22K and COCO, as well as self-supervised approaches such as MoCo [10] and SimMIM [64].

As shown in Tab. 3, our approach achieves consistent and substantial improvements in both mean Average Precision (mAP) and NuScenes Detection Score (NDS) across multiple backbones. For instance, with a Swin-Tiny backbone, our model achieves an mAP of 43.01% and NDS of 52.41%,

Table 2: Quantitative results on the official split of KITTI dataset. All metrics reported here are from the KITTI online leaderboard.

Method	Backbone	Pretrain	SILog ↓	Abs Rel ↓	Sq Rel ↓	iRMSE ↓
NeW CRFs [71]	Swin-Large	ImageNet-22K	10.39	8.37	1.83	11.03
VA-DepthNet [32]	Swin-Large	ImageNet-22K	9.63	7.96	1.66	10.44
IEBins [48]	Swin-v2-Large	MIM [65]	9.84	7.82	1.60	10.68
NDDepth [47]	Swin-v2-Large	MIM [65]	9.62	**7.75**	1.59	10.62
DCDepth [57]	Swin-Large	Semantic-SAM [30]	9.60	7.83	**1.54**	**10.12**
DCDepth [57]	Swin-Large	Ours	**9.54**	7.76	1.55	10.37

Table 3: Quantitative evaluation of BEVFormerV2 [66] on nuScenes `val` set using different pretraining methods.

Method	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
COCO	Res50	51.82	41.99	66.89	28.14	39.15	38.34	19.28
ImageNet-1K	Res50	51.99	42.51	**65.90**	27.79	42.12	**37.70**	**19.20**
MoCo v3 [10]	Res50	52.42	42.94	67.13	27.70	**35.84**	39.91	19.95
Ours	Res50	**52.55**	**43.22**	66.30	**27.56**	37.76	38.47	20.53
ImageNet-22K	Swin-T	51.69	42.12	67.69	**28.07**	**38.39**	40.89	18.68
Ours	Swin-T	**52.41**	**43.01**	**65.81**	28.30	41.43	**37.23**	**18.15**
ImageNet-22K	Swin-S	53.62	45.22	**64.94**	27.75	**36.97**	40.07	20.18
Ours	Swin-S	**54.22**	**45.49**	65.22	**27.73**	37.61	**35.61**	**19.04**
ImageNet-22K	Swin-B	53.98	45.48	65.94	28.10	35.82	38.75	**19.01**
SimMIM [64]	Swin-B	54.03	45.18	63.81	**27.53**	38.02	37.04	19.22
Ours	Swin-B	**55.68**	**47.54**	**62.74**	27.84	**33.79**	**36.77**	19.81
ImageNet-22K	Swin-L	54.59	45.91	65.39	27.44	34.31	37.64	18.87
SimMIM [64]	Swin-L	54.98	46.52	64.80	28.06	33.72	**35.87**	20.36
Ours	Swin-L	**55.80**	**47.29**	**62.83**	**27.16**	**33.20**	36.50	**18.77**

outperforming the ImageNet-22K pretrained counterpart (mAP 42.12%, NDS 51.69%). As the backbone scales up to Swin-Large, our model further improves to 47.29% mAP and 55.80% NDS, still outperforming the compared supervised and self-supervised methods.

To compare with more methods based on ViT architectures whose computational cost are not affordable for high input resolution, we also tested various methods at a resolution of 704×256. As shown in Tab. 4, our Swin-based models achieve competitive or superior performance compared to DINOv2, while using significantly fewer parameters and lower computational costs. For instance, our model pretrained with the Swin-L backbone attains an NDS of 52.03% and an mAP of 41.79%. These results are comparable to those achieved by DINOv2 with the ViT-L backbone.

Notably, these improvements are not limited to Transformer-based architectures. With ResNet backbones such as R50, our model also outperforms COCO-supervised models, indicating that the benefit of our pretraining is architecture-agnostic. This broad compatibility with both convolutional and Transformer backbones highlights the generality of the learned features.

These gains can be attributed to the object-centric and geometry-aware priors introduced by our object-based visual representation. Unlike traditional supervised pretraining, our approach enables the model to internalize compositional structure and spatial relationships between objects. This proves particularly valuable in 3D detection tasks, where reasoning about object placement, extent, and occlusion is critical.

4.5 3D Occupancy Perception

We evaluate our method on the nuScenes validation set using SparseOcc [33] as the occupancy prediction framework. As shown in Tab. 6, our pre-trained models outperform both supervised (ImageNet-22K) and self-supervised (SimMIM) counterparts across all Swin backbone variants. Crucially, the strong performance of our method can be attributed to the underlying geometric insight described in Sec.3.1. By leveraging this property, our method is able to encode spatial structures that are semantically meaningful, even without direct instance-level annotations.

Table 4: Quantitative evaluation of BEVFormerV2 [66] on nuScenes `val` set using different pretraining methods at a resolution of 704×256

Method	Backbone	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
DINOv2	ViT-S	46.24	34.88	71.65	28.45	49.97	42.70	18.84
DINOv2	ViT-B	49.08	38.36	69.74	28.21	41.81	**42.40**	18.84
DINOv2	ViT-L	**51.91**	**42.05**	**65.04**	**27.35**	**36.45**	43.79	**18.51**
ImageNet-22K	Swin-T	47.42	36.34	70.90	28.40	48.36	**40.47**	**19.40**
Ours	Swin-T	**48.24**	**37.08**	**70.61**	**27.99**	**44.45**	40.54	19.45
ImageNet-22K	Swin-S	48.78	38.00	70.65	28.35	41.20	42.95	19.01
Ours	Swin-S	**50.87**	**40.23**	**68.88**	**27.85**	**40.45**	**36.67**	**18.61**
ImageNet-22K	Swin-B	50.42	40.71	68.56	**27.80**	40.60	42.81	19.52
Ours	Swin-B	**51.69**	**41.36**	**66.01**	28.03	**37.98**	**39.73**	**18.10**
ImageNet-22K	Swin-L	50.48	40.09	68.01	**27.90**	40.69	40.67	18.38
Ours	Swin-L	**52.03**	**41.79**	**66.10**	28.12	**36.84**	**40.13**	**17.51**

Table 5: Ablation studies on the KITTI depth estimation task. We evaluate the impact of different pretraining strategies: DINO refers to DINOv2 [41], and IN denotes ImageNet-22K [13] supervised pretraining.Ours and IN use Swin-T as the backbone, while DINO uses ViT-S.

Ours	DINO	IN	SILog ↓	AbsRel ↓	RMSE ↓
✓			**6.991**	**0.051**	**2.016**
	✓		7.119	0.052	2.153
		✓	7.455	0.055	2.182
	✓	✓	7.071	0.052	2.117
✓		✓	6.927	**0.050**	**2.009**
✓	✓		**6.796**	**0.050**	2.014

Table 6: Quantitative evaluation of SparseOcc [33] on nuScenes `val` set using different pretraining.

Method	BB	RayIoU	$\text{RayIoU}_{1m, 2m, 4m}$		
MoCo v3 [10]	R50	34.4	28.3	35.1	39.9
ImageNet-1K	R50	35.0	28.8	35.6	40.5
Ours	R50	**36.4**	**30.2**	**37.1**	**41.8**
ImageNet-22K	Sw-T	35.5	29.4	36.3	40.9
Ours	Sw-T	**37.0**	**31.1**	**37.8**	**42.2**
DINOv2 [41]	ViT-S	35.9	29.5	36.8	41.4
ImageNet-22K	Sw-S	36.8	30.4	37.6	42.3
Ours	Sw-S	**38.1**	**32.0**	**39.1**	**43.4**
DINOv2 [41]	ViT-B	37.1	31.0	37.9	42.4
ImageNet-22K	Sw-B	37.6	31.3	38.4	43.1
SimMIM [64]	Sw-B	38.0	31.7	38.7	43.4
Ours	Sw-B	**38.3**	**32.1**	**39.1**	**43.7**
DINOv2 [41]	ViT-L	**39.0**	**32.8**	**39.9**	**44.3**
ImageNet-22K	Sw-L	37.6	31.4	38.4	43.0
SimMIM [64]	Sw-L	38.6	32.6	39.4	43.7
Ours	Sw-L	38.7	32.6	39.5	43.8

5 Discussion and Future Work

5.1 Generalization to out of domain scenes

As a preliminary investigation, we train our method on outdoor driving videos. To demonstrate the generalization of our learned features, we also visualize the features on daily life videos from the Ego4D [17] and the robot manipulation dataset RTX [42], as shown in Fig. 5. Though not perfect, our model is capable of distinguishing different objects that did not appear in the training set. The first row shows some common scenes in indoor scenes. Our method can segment the unseen objects, including windows, tools, even cats, and hands. We observe similar results in the second row for robot manipulation. These results illustrate that our method does not overfit the objects in the training set, but indeed learns the essential composition of an object.

5.2 Further scaling up and extensions

We also attempted to apply our method to broader scenarios, such as ego-centric videos and unconstrained videos from the web. However, we found that the performance of our method is greatly limited by the performance of optical flow. Fortunately, we find that recent closely related work in monocular depth estimation [68, 69] improves significantly with the help of large-scale synthetic data. We hope a similar paradigm can also benefit the performance of optical flow.

Our method can be further extended to a temporal setting. Based on the compact object instance representation, we can easily endow the model with temporal prediction ability. It is essentially a world model that could predict the dynamics of the world. We will pursue these directions in our future work.

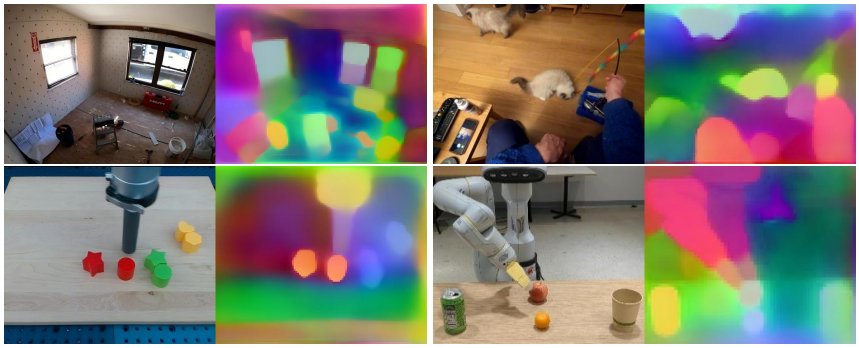

Figure 5: Examples of feature maps in out of domain scenes.

5.3 Improving precise localization ability

As seen in the visualization, our method focuses on the whole object, which means the features of the parts within an object are indistinguishable. This makes our method unsuitable for applications that need precise localization, such as keypoint matching. Our method can be improved by combining previous works that emphasizes local feature learning, such as CroCo [63], to get the best of two worlds.

6 Conclusions

In this work, we present a biologically inspired framework for learning object-centric visual representations, drawing motivation from developmental psychology studies on how infants acquire the concept of objects through motion cues. By leveraging the natural correlation between motion boundaries and object boundaries, our method derives instance-level pseudo labels from raw videos, enabling unsupervised representation learning without human annotations or camera calibration.

Through extensive experiments across three diverse vision tasks, we demonstrate that our approach not only matches but surpasses the supervised and self-supervised pretraining baselines. Our learned features capture object-level semantics that are complementary to those in existing vision foundation models such as DINO and MAE.

These results highlight the potential of integrating biologically inspired mechanisms—such as motion-guided grouping—into the design of scalable, general-purpose visual pretraining frameworks. We hope this work encourages further exploration of cognitive principles in building more robust and human-aligned vision systems.

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *CVPR*, 2023.

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BETR pre-training of image transformers. In *ICLR*, 2022.

[3] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *TMLR*, 2024.

[4] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *NeurIPS*, 2019.

[5] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.

[11] MMPreTrain Contributors. OpenMMLab's pre-training toolbox and benchmark. `https://github.com/open-mmlab/mmpretrain`, 2023.

[12] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[14] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *ICCV*, 2021.

[15] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[17] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4D: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[18] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.

[19] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020.

[20] K. Tan et al. H. Caesar, J. Kabzan. NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles. In *CVPR ADP3 workshop*, 2021.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[23] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.

[24] Scott P Johnson and Susan A Johnson. Development of perceptual completion originates in early infancy. *Psychological Science*, 14(6):553–559, 2003.

[25] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[26] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *CVPR*, 2012.

[27] Philip J Kellman and Elizabeth S Spelke. Perception of object unity in young infants. *Infant Behavior and Development*, 11(2):161–180, 1983.

[28] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019.

[29] Dwight J Kravitz, Kadharbatcha S Saleem, Chris I Baker, and Mortimer Mishkin. A new neural framework for visuospatial processing. *Nature Reviews Neuroscience*, 12(4):217–230, 2011.

[30] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, Lei Zhang, and Jianfeng Gao. Segment and recognize anything at any granularity. In *ECCV*, 2024.

[31] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: learning bird's-eye-view representation from LiDAR-camera via spatiotemporal transformers. *TPAMI*, 47(3):2020–2036, 2024.

[32] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. VA-DepthNet: A variational approach to single image depth prediction. In *ICLR*, 2023.

[33] Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang, Tianyu Li, Jia Zeng, Li Chen, Hongyang Li, and Limin Wang. Fully sparse 3D occupancy prediction. In *ECCV*, 2024.

[34] Runtao Liu, Zhirong Wu, Stella Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. In *NeurIPS*, 2021.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[36] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

[37] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[38] Rundong Luo, Hong-Xing Yu, and Jiajun Wu. Unsupervised discovery of object-centric neural fields. *arXiv preprint arXiv:2402.07376*, 2024.

[39] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. 1982.

[40] Amy Needham. Object exploration and object knowledge in young infants: A view from developmental psychology. *Cognition, Brain, and Consciousness*, 2001.

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *TMLR*, 2023.

[42] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open X-embodiment: Robotic learning datasets and RT-X models. In *ICRA*, 2024.

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[44] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. BEiT v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[46] Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *ICCV*, 2023.

[47] Shuwei Shao, Zhongcai Pei, Weihai Chen, Xingming Wu, and Zhengguo Li. NDDepth: Normal-distance assisted monocular depth estimation. In *ICCV*, 2023.

[48] Shuwei Shao, Zhongcai Pei, Xingming Wu, Zhong Liu, Weihai Chen, and Zhengguo Li. IEBins: Iterative elastic bins for monocular depth estimation. In *NeurIPS*, 2023.

[49] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. VideoFlow: Exploiting temporal cues for multi-frame optical flow estimation. In *ICCV*, 2023.

[50] Oriane Siméoni, Gilles Puy, Huy V Vo, Simon Roburin, Spyros Gidaris, Andrei Bursuc, Patrick Pérez, Renaud Marlet, and Jean Ponce. Localizing objects with self-supervised transformers and no labels. In *BMVC*, 2021.

[51] Silky Singh, Shripad Deshmukh, Mausoom Sarkar, and Balaji Krishnamurthy. Locate: self-supervised object discovery via flow-guided graph-cut and bootstrapped self-training. In *BMVC*, 2023.

[52] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104:154–171, 2013.

[53] Leslie G Ungerleider and Mortimer Mishkin. Two cortical visual systems. *Analysis of Visual Behavior*, 1982.

[54] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[55] Huy V Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Pérez, and Jean Ponce. Unsupervised image matching and object discovery as optimization. In *CVPR*, 2019.

[56] Van Huy Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. In *NeurIPS*, 2021.

[57] Kun Wang, Zhiqiang Yan, Junkai Fan, Wanlu Zhu, Xiang Li, Jun Li, and Jian Yang. DCDepth: Progressive monocular depth estimation in discrete cosine domain. In *NeurIPS*, 2024.

[58] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *CVPR*, 2023.

[59] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *CVPR*, 2022.

[60] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. TokenCut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *TPAMI*, 45(12):15790–15801, 2023.

[61] Yuqi Wang, Yuntao Chen, and Zhao-Xiang Zhang. 4D unsupervised object discovery. In *NeurIPS*, 2022.

[62] Xiu-Shen Wei, Chen-Lin Zhang, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Unsupervised object discovery and co-localization by deep descriptor transformation. *PR*, 88:113–126, 2019.

[63] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. CroCo: Self-supervised pre-training for 3D vision tasks by cross-view completion. In *NeurIPS*, 2022.

[64] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *CVPR*, 2022.

[65] Zhenda Xie, Zigang Geng, Jingcheng Hu, Zheng Zhang, Han Hu, and Yue Cao. Revealing the dark secrets of masked image modeling. In *CVPR*, 2023.

[66] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. BEVFormer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023.

[67] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized predictive model for autonomous driving. In *CVPR*, 2024.

[68] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[69] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *NeurIPS*, 2024.

[70] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *ECCV*, 2022.

[71] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected CRFs for monocular depth estimation. In *CVPR*, 2022.

[72] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. In *NeurIPS*, 2020.

[73] Xin Zhang, Jinheng Xie, Yuan Yuan, Michael Bi Mi, and Robby T Tan. HEAP: unsupervised object discovery and localization with contrastive grouping. In *AAAI*, 2024.

[74] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[75] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions, and the experiments in Sec. 4 are conducted to support these contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of our work are discussed in Sec. 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All details are specified in Sec. 3 and Sec. 4. Pseudo-code is provided in appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and data associated with this paper will be made publicly available at the link provided in the abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and test details are specified in Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We report evaluation metrics following prior work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All training and test details are specified in Sec. 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research followed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and related prior work are properly cited in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets at this time.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Pseudo-codes for Pixel Cluster

For all optical flow data generated by VideoFlow, we perform a simple Breadth-First Search(BFS) to segment moving objects. Alg. 1 provides a pseudocode description of our algorithm. The algorithm takes the optical flow, the forward-backward consistency check result, and two thresholds θ_f and θ_s as input. θ_f is used to determine when the optical flow of two adjacent pixels, being sufficiently close, is considered to belong to the same object. θ_s controls the minimum number of pixels that an object should have.

Algorithm 1 Pixel Cluster

Input: flow(optical flow), valid(consistency check), θ_f, θ_s
Initialization:$n \leftarrow 0, v[i][j] \leftarrow false, S \leftarrow \emptyset$
for $x \leftarrow 1$ to H **do**
  for $y \leftarrow 1$ to W **do**
    if $v[x][y] = true$ or valid$[x][y] = false$ **then**
      continue
    end if
    $Q \leftarrow$ empty queue, $C \leftarrow \emptyset$
    Enqueue$(Q, (x, y))$
    while $Q \neq \emptyset$ **do**
      $(x, y) \leftarrow$ Dequeue(Q)
      $C \leftarrow C \cup \{(x, y)\}$
      for (i, j) in (x, y)'s 4 neighbors **do**
        if $||$flow$[i][j]$, flow$[x][y]||_2 \leq \theta_f$ and $v[i][j] = false$ and valid$[x][y] = true$ **then**
          $v[i][j] = true$
          Enqueue$(Q, (i, j))$
        end if
      end for
    end while
    if $|C| \geq \theta_s$ **then**
      $S \leftarrow S \cup \{C\}$
    end if
  end for
end for
Output: S

B Data Augmentation Details

All input images are first randomly resized to a resolution between 512×288 and 1024×576. They are then randomly cropped to 224×224. During cropping, up to 10 attempts are made to ensure that the cropped region contains at least two distinct labels. Afterward, each image has a 50% chance of being horizontally flipped. Additionally, gamma, brightness, and color augmentations are applied with a 50% probability, each sampled within the range of (0.9, 1.1).

C More Qualitative Results

Fig. 6 shows additional qualitative results of the pseudo-label generation and the visualizations of the output features. As illustrated in the pseudo-label visualizations, the proposed algorithm successfully segments objects exhibiting significant movement, as well as foreground instances exhibiting motion patterns distinct from the background. The feature visualizations shows that the model distinguishes many objects not annotated in the pseudo-labels. This suggests our model goes beyond mimicking pseudo-labels, but learning a more general, object-centric representation.

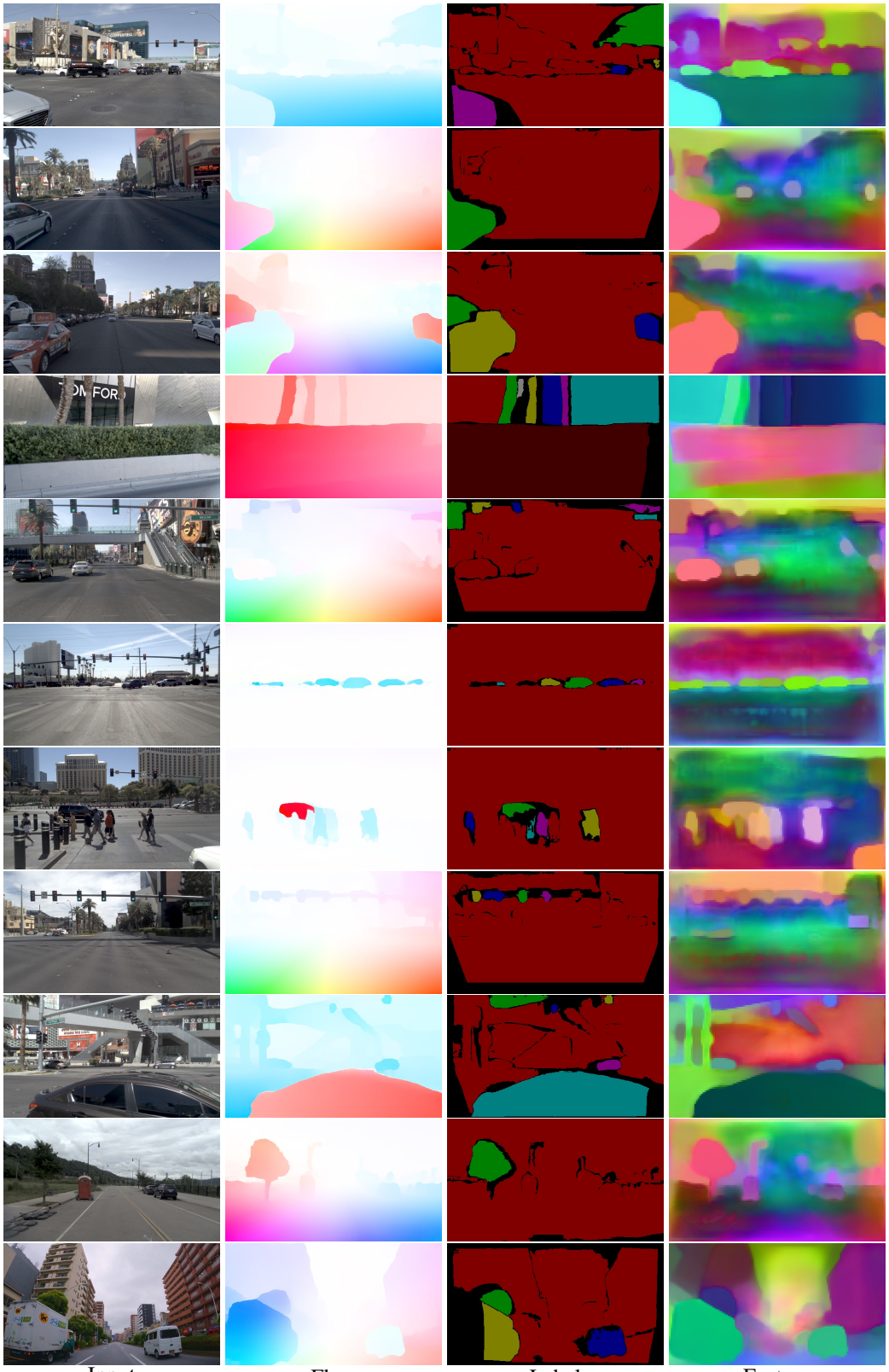

Figure 6: Examples of the pseudo-label generation results and the output features.