This is the authors version of an article accepted for publication in
Computer Vision and Image Understanding (CVIU).

Monocular Per-Object Distance Estimation with Masked Object Modeling

Aniello Panariello**, Gianluca Mancusi, Fedy Haj Ali, Angelo Porrello, Simone Calderara, Rita Cucchiara

University of Modena and Reggio Emilia, Via Vivarelli 10, Modena, Italy

ABSTRACT

Per-object distance estimation is critical in surveillance and autonomous driving, where safety is crucial. While existing methods rely on geometric or deep supervised features, only a few attempts have been made to leverage self-supervised learning. In this respect, our paper draws inspiration from Masked Image Modeling (MiM) and extends it to **multi-object tasks**. While MiM focuses on extracting global image-level representations, it struggles with individual objects within the image. This is detrimental for distance estimation, as objects far away correspond to negligible portions of the image. Conversely, our strategy, termed **Masked Object Modeling** (**MoM**), enables a novel application of masking techniques. In a few words, we devise an auxiliary objective that reconstructs the portions of the image pertaining to the objects detected in the scene. The training phase is performed in a single unified stage, simultaneously optimizing the masking objective and the downstream loss (*i.e.*, distance estimation). We evaluate the effectiveness of MoM on a novel reference architecture (DistFormer) on the standard KITTI, NuScenes, and MOTSynth datasets. Our evaluation reveals that our framework surpasses the SoTA and highlights its robust regularization properties. The MoM strategy enhances both zero-shot and few-shot capabilities, from synthetic to real domain. Finally, it furthers the robustness of the model in the presence of occluded or poorly detected objects. Code is available at `https://github.com/apanariello4/DistFormer`

1. Introduction

The Computer Vision community has a long-standing commitment to estimating the *third dimension*, namely the distance of a target object from the camera (or *observer*) when projected onto the image plane. In this respect, humans continuously practice such a capability. For example, when approaching a stop sign, the driver visually assesses the remaining distance to the sign and adjusts the car's velocity accordingly. However, distance estimation through human perception is often rough and qualitative; its precision depends on the skills of the subject and on its health status, which can be altered by the consumption of drugs and alcohol. Additionally, external factors such as high vehicle speed, the terrain (Sinai et al., 1998), or adverse weather conditions can further worsen distance perception.

Machine vision development has facilitated automating tasks requiring precise distance estimation. Initial efforts leveraged the pinhole model and standard projective transformations. Under this framework, **geometric** methods (Gökçe et al., 2015; Haseeb et al., 2018; Tuohy et al., 2010) manage to learn a linear relation between the perceived size of the object (*e.g.*, bounding box height) and its distance. Such methods assume consistent object sizes within classes, which does not hold in practice (*e.g.*, the heights of children and adults may vary significantly). Modern **feature-based** approaches (Zhu and Fang, 2019; Jing et al., 2022; Li et al., 2022; Mancusi et al., 2023) exploit fine-grained visual information regarding the target objects and the context of the scene. To do so, the entire image is fed to a global encoder (*e.g.*, a Convolutional Neural Network (CNN) (Simonyan and Zisserman, 2015; He et al., 2016)). Then, to enable multi-target evaluation, **Region of Interest (RoI)** pooling techniques have been adopted to provide a fixed-size representation for each target object.

Per-object distance estimation shares similarities with dense depth estimation. However, it offers a more targeted approach by predicting distance values for each detected object rather than for every pixel in an image. This focused method proves advantageous, particularly in domains like autonomous driving and object tracking. By discerning distances at an object level, computational resources are efficiently allocated, allow-

**Corresponding author
e-mail: `aniello.panariello@unimore.it` (Aniello Panariello)

ing quicker inference. Moreover, this approach prioritizes objects of interest, enhancing the precision of tasks such as collision avoidance and object tracking. Finally, in per-object distance estimation, even the distance of partially occluded objects can be predicted, a crucial capability for ensuring robustness in complex real-world scenarios.

Recently, **Masked Image Modeling** (MiM) gained popularity to pre-train models through a *pretext task*. For example, Masked Autoencoders (MAEs) (He et al., 2022) reconstruct an input image from a portion of it, leveraging an asymmetric encoder-decoder design. These networks exhibit appealing properties, including reduced training memory usage, improved accuracy in downstream tasks, and better adaptation to new scenarios(Gandelsman et al., 2022). Moreover, reconstruction-based methods have also been explored for tasks like anomaly detection, where reconstruction errors are leveraged to assess whether a target has been accurately reconstructed or deviates from expected patterns (Wang et al., 2023b, 2024, 2023a). In light of these advantages, we consider employing MiM techniques for video surveillance tasks (like ours) desirable, as they often involve high-resolution images and limited data availability due to privacy concerns. While some prior works have explored MiM approaches for tasks such as tracking and detection, they primarily focus on using MAEs solely for pre-training (Bielski and Favaro, 2022; Li et al., 2023a) and for **single target** scenarios (Wu et al., 2023; Zhao et al., 2023). In this respect, the core contribution of our work is to extend the application of MiM to problems featuring multiple targets, such as multi-object distance estimation.

The original MAEs framework poorly supports downstream tasks requiring multiple outputs (one for each target object), such as distance estimation. As shown in Fig. 1 (left), standard MiM models randomly drop patches from the input image without discriminating between instance- and background-related patches. While it allows the model to learn a global image representation, it will likely be biased toward the most present patterns, *i.e*., the background. By doing so, not enough importance will be placed on the objects of interest of the downstream task (Han et al., 2024).

To address such a limitation and allow MAEs to perform multi-object analysis, we propose a novel masking strategy skewed toward fine-grained details and applied at the instance level. As shown in Fig. 1 (right), we apply two main changes. Firstly, we move the masking operation right after the RoI extraction layer and teach the decoder to reconstruct only the areas related to each instance (*e.g*., cars or pedestrians). This way, we drive the learned features to a localized understanding of target objects. Secondly, while MAEs are generally used only for pre-training in a two-stage fashion, we propose a joint approach that optimizes the reconstruction and downstream losses (*i.e*., the regression of distances) with a shared backbone. In a sense, we exploit masking to enforce regularization beyond reducing the memory training footprint.

Eventually, we plug such a novel masking strategy into a **hybrid architecture**, termed DistFormer, which gracefully combines CNNs and Transformer layers. Our model balances local and global information, addressing the limitations of existing methods. We validated the proposed approach by conducting extensive experiments on KITTI (Geiger et al., 2012), NuScenes (Caesar et al., 2020), and MOTSynth (Fabbri et al., 2021). Our findings show that, when employed to predict the distances of several objects from the camera, our method delivers impressive performance, surpassing the state-of-the-art by a wide margin (-57% on KITTI, -32% on NuScenes and -27% on MOTSynth in RMSE). However, its advantages do not stop at an improvement in terms of accuracy.

Contributions. In this work, we identify and outline the key innovations and advancements introduced by our approach, focusing on its novel methodologies, unified optimization strategies, and state-of-the-art performance. Below, we provide a detailed summary of the main contributions:

- **Instance-Level Masking:** We introduce a novel masking strategy at the instance-level that reconstructs regions associated with detected objects, ensuring the learned features are fine-grained and tailored for downstream tasks such as distance estimation. As shown in Fig. 1 (right), this masking occurs post-RoI extraction, focusing the decoder on reconstructing only target-relevant areas.

- **Joint Optimization Framework:** Our unified training approach integrates self-supervised reconstruction and downstream task optimization in a single stage, leveraging a shared backbone. This joint approach not only regularizes training but also enhances robustness to occlusions and domain shifts, as discussed in Sec. 4.4 and illustrated in Fig. 4, where our method achieves improved RMSE and $\delta_{<1.25}$ scores under varying training set sizes.

- **State-of-the-Art Performance and Transferability:** As shown in Sec. 4.3, our method achieves significant gains in per-object distance estimation across benchmarks (KITTI, NuScenes, MOTSynth), with up to a -57% reduction in RMSE. Furthermore, it excels in zero-shot transfer capabilities, effectively bridging synthetic and real-world domains, as explored in Sec. 4.4.

2. Related Works

Object Distance Estimators. Estimating object distances from a single RGB image (*i.e*., monocular distance estimation) is a crucial task for many computer vision applications (Alhashim and Wonka, 2018; Godard et al., 2017; Ranftl et al., 2021; Godard et al., 2019; Ranftl et al., 2020; Lee et al., 2019; Zhu and Fang, 2019; Haseeb et al., 2018). One of the approaches to this task is to perform per-object distance estimation. Early works leverage the object's geometry to find its distance. However, these works did not take into account any visual features. Among these works, the Support Vector Regressor (SVR) (Gökçe et al., 2015) finds the best-fitting hyperplane given the geometry of the bounding boxes. The Inverse Perspective Mapping (IPM) (Mallot et al., 1991; Rezaei et al., 2015) improves results by adopting an iterative approach that converts image points to bird's-eye view coordinates. Nonetheless, this method introduces distortion on the image, making it

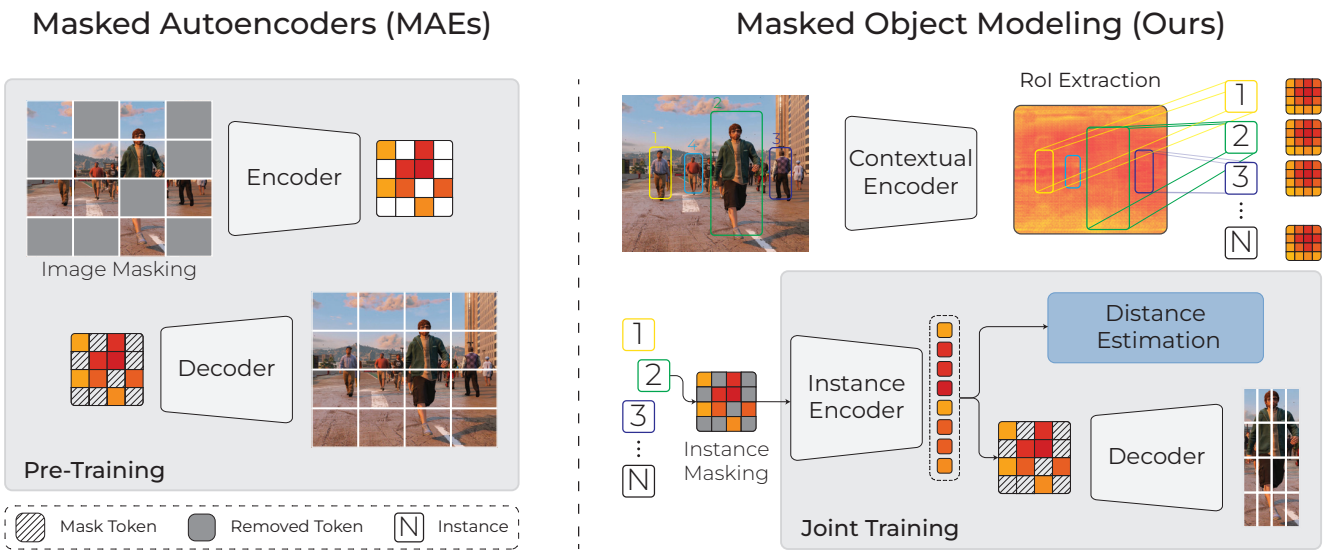

Fig. 1: Standard MAEs (left) *vs.* MoM (right). We defer masking until after the RoI-based extraction, allowing us to mask out tokens of single instances rather than the entire input image. The downstream task is jointly optimized with the reconstruction.

challenging to predict distance for objects that are either distant or on curved roads.

Successive methods (Haseeb et al., 2018; Gökçe et al., 2015) exploit the object bounding box geometry to infer its distance through deep neural networks, effectively improving upon pure algorithmic techniques. These approaches are limited when the target objects are of different classes *e.g.*, vehicles and people. A car and a person with similar bounding boxes will have very different distances from the camera. Zhu *et al.* (Zhu and Fang, 2019) has made a notable improvement by introducing a structure inspired by Faster R-CNN (Ren et al., 2015) and extracting visual features with a *RoIPool* (Girshick, 2015) operation. This approach captures significant visual attributes, which the distance regressor then uses to predict the distance of objects. More recently, DistSynth (Mancusi et al., 2023) leveraged multiple frames from a sequence to consistently predict distances over time.

A different approach in this field is monocular per-pixel depth estimation (Eigen et al., 2014; Godard et al., 2019; Lee et al., 2019; Ranftl et al., 2020; Li et al., 2023b), where the goal is to predict a depth map starting from a single RGB image. In (Eigen et al., 2014), the authors propose employing two deep networks to predict and refine. More recently, (Lee et al., 2019) used multi-resolution depth maps to construct the final map. However, these works have a high computational cost and are difficult to implement in a real-time system such as autonomous driving. Moreover, translating the depth map into object distance is nontrivial due to occlusions and the looseness of bounding boxes. Our approach, instead, has reduced computational requirements and can predict distance for partially occluded objects.

3. Method

DistFormer, shown in Fig. 2, includes three main modules: the Contextual Encoder, the Local Encoder, and the Global Encoder. Firstly, the **Contextual Encoder** $f(\mathbf{x};\theta_f)$ (Sec. 3) produces a feature map from an image $\mathbf{x}$, encoding visual features. Such a network is a CNN built upon Feature Pyramid Networks (Lin et al., 2017), which extracts high-level features and retains fine-grained details.

Secondly, given the bounding box for each instance, we extract per-object representations with standard region-based pooling to obtain a structured grid of activations for each target object, denoted as **latent patches** (or tokens). Then, we apply our masking strategy to these latent patches, treating each instance independently. Unmasked tokens are fed to the **Local Encoder** $LE(f(\cdot);\theta_L)$ (Sec. 3), which further enhances local visual reasoning and promotes the extraction of localized fine-grained details. Specifically, it performs self-attention between latent patches of the same object, disregarding other objects. Notably, the Local Encoder interacts with the **Decoder** network and, based on that, receives a self-supervised training signal (**MoM**, Sec. 3.1).

Unlike standard MAEs, our approach jointly optimizes the pretext and downstream tasks (*i.e.*, multi-target distance estimation) in a single training stage. In this respect, there is a third and final component, the **Global Encoder** (Sec. 3), trained to predict the distance of the objects. As our task benefits by focusing on mutual distances, the Global Encoder applies self-attention to representations from distinct objects. This multi-object analysis plays an essential role also in human perception, as stated by the *adjacency principle* (Gogel, 1963). Indeed, an object's apparent size or position in the field of view is determined by the size or distance cues between it and adjacent objects. Detailed pseudocode for the training and testing phases of the proposed architecture is provided in Algorithms A and B, respectively, in the appendix.

Contextual Encoder. We feed our backbone f with an RGB frame $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$ where C is the number of channels and (H, W) are the frame resolution. We adopt a CNN as our backbone, specifically utilizing ConvNeXt pre-trained on ImageNet-

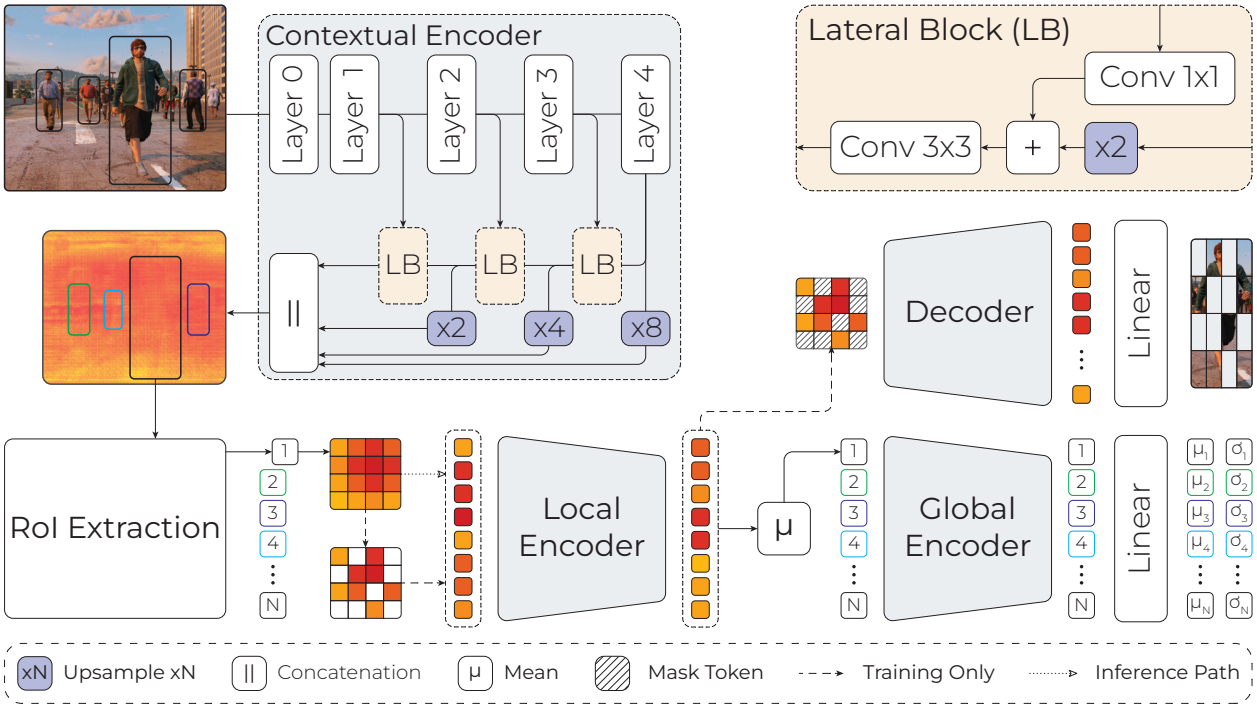

Fig. 2: DistFormer's overview. The image passes through the Contextual Encoder and RoI extraction to obtain per-object representations, which the Local and Global Encoders then process. Finally, we predict a Gaussian modeling distance and uncertainty.

22k (He et al., 2016; Liu et al., 2022). While convolutional networks offer benefits such as translation invariance and hierarchical reasoning (Krizhevsky et al., 2017), the pooling layers and other resolution reduction techniques can hinder distance estimation, as distant objects may be represented by only a few pixels. To avoid such shortcomings, we employ Feature Pyramid Networks (FPN) (Lin et al., 2017), similar to previous works (Mancusi et al., 2023; Lang et al., 2019; Chen et al., 2018; Yang et al., 2018). FPN-based networks consist of a forward branch for downsampling feature maps and a backward branch that progressively upscales the output. The backward branch utilizes Lateral Blocks (LB) to upscale the feature maps from the forward pass, concatenated into a single feature map.

Local Encoder. Next, the goal is to extract fixed-size latent representations, one for each object. To do so, we start from the feature maps processed by the Contextual Encoder and then apply the *RoIAlign* (He et al., 2017) operation[1], which extracts the portions of the feature map covered by the target objects. Indicating with N the number of bounding boxes, this operation yields feature vectors $\mathcal{F}_{i|i\in\{1,\dots,N\}} \in \mathbb{R}^{c\times h\times w}$, where c is the number of channels of the feature map and (h, w) are the dimensions of the RoI quantization (8×8 in our experiments). To better encode the information of the target object, we employ a module termed Local Encoder (LE), which consists of the final 6 layers of a pre-trained ViT-B/16 model (Dosovitskiy et al., 2021). To feed it, we rearrange the feature map of each object $\mathcal{F}_i$ into a vector – *i.e.*, $\mathbb{R}^{c\times(h\times w)} \rightarrow \mathbb{R}^{c\times(h\cdot w)}$ – treating each pixel of the activation map as a token. Then, the LE performs self-attention on the object's tokens. Such an operation aims to encode informative intra-object features and to encourage the model to focus on the most critical portions of the objects, *e.g.*, not occluded.

Global Encoder. Since the Local Encoder is based on ViT layers, it outputs $h \cdot w$ tokens for each bounding box, which we aggregate along the token axis through global average pooling – $\mathbb{R}^{c\times(h\cdot w)} \rightarrow \mathbb{R}^{c\times 1}$ – and hence obtain a singleton representation. This representation is fed to the Global Encoder (GE), structured as a two-layered ViT architecture. Its function is to enhance the understanding of inter-object relationships within the scene. Similarly to the Local Encoder, the Global Encoder employs multiple layers of attention-based operations. However, it conducts self-attention between tokens corresponding to different objects. This operation enables each token to integrate insights from other objects, including partially occluded ones. Consequently, each object $\in \mathbb{R}^c$ is passed to a Multi-Layer Perceptron (MLP) to predict its distance.

3.1. Masked Object Modeling (MoM)

We devised a self-supervised learning approach called **Masked Object Modeling (MoM)** in our architecture. During training, only 50% of the input tokens are fed into the Local Encoder. Subsequently, we employ a two-layer Decoder network $D(\cdot, \theta_D)$ to reconstruct only the input image area covered by the bounding box. As in original MAEs (He et al., 2022), the unmasked tokens are taken directly from the encoder output, while a learned control token substitutes the absent masked tokens. Finally, the Masked Object Modeling (MoM) objective

[1]Compared to *RoIPool*, commonly used in this task (Zhu and Fang, 2019; Li et al., 2022), *RoIAlign* avoids misalignments thanks to a more accurate interpolation strategy.

Fig. 3: Corresponding reconstructions yielded by the Decoder network trained with **+ MoM** .

Table 1: Computational cost analysis on KITTI (all classes).

Masking	Time	FLOPs	RMSE
DistFormer			
+ all tokens	67.6ms	380 G	2.87
+ mask 30%	66.5ms	360 G	2.89
+ **mask 50%**	65.2ms	345 G	2.91
+ mask 80%	64.4ms	330 G	3.14

is:

$$\mathcal{L}_{\text{MoM}} = \mathbb{E}_{(\mathbf{x}_i, \mathcal{F}_i) \in \mathcal{X}} \left[\| D(LE(\mathcal{F}_i, \theta_{LE}), \theta_D) - \mathbf{x}_i \|_2^2 \right], \quad (1)$$

where $\mathbf{x}_i$ is the i-th object image portion and $\mathcal{X}$ the whole set of target objects.

Overall objective. Given the intrinsic uncertainty of the task, we opt to predict a Gaussian distribution over the expected distance instead of providing a punctual estimation. The mean of the distribution represents the distance, while its variance is the model *aleatoric uncertainty* (Bertoni et al., 2019; Der Kiureghian and Ditlevsen, 2009), which refers to the inherent noise contained in the observations. To do so, the MLP mentioned in Sec. 3 outputs two scalars for each object; on top of them, the supervised part of the overall training signal can be carried by lowering the Gaussian Negative Log Likelihood (GNLL) (Nix and Weigend, 1994) of ground-truth distances. The final objective is given by Eq. (1) and the GNLL:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{MoM}} + \mathcal{L}_{\text{GNLL}}, \quad (2)$$

where α is a hyper-parameter balancing the importance of the MoM objective.

MoM acts as object-level feature regularizer. Adopting the MoM objective provides several noteworthy advantages, as discussed below. In Sec. 4.4, we demonstrate its efficacy in enhancing zero-shot capabilities, synthetic-to-real transfer, and its robustness to noisy bounding boxes. In this respect, we argue that our masking strategy promotes **standardization** in the features learned by the Local Encoder, biasing the optimization toward **more stable cues**. To support this claim, we report in Fig. 3 some examples of reconstructions: the decoder seems to exclude non-essential details (*e.g.*, colors) deemed irrelevant to distance estimation. Therefore, by *prioritizing* task-related cues, our model gains resilience to unexpected and unimportant visual variations, which are peculiar issues in the presence of domain shifts.

MoM allows elastic input. The application of MoM enables the reduction of the number of latent tokens provided to the Local Encoder. Significantly, this reduction can occur at training time as advocated and inference time, *during evaluation*. As reported in Tab. 1, in fact, by leveraging masking at inference time, we can reduce both the wall-clock time and the memory footprint while still producing accurate distance estimates (→ low root mean squared error).

3.2. Comparison with related works

In Sec. 2, we pointed out similarities with dense depth estimation, which focuses on predicting depth maps of images. For example, works such as (Li et al., 2023b; Lee et al., 2019; Ranftl et al., 2020; Yang et al., 2024) commonly employ end-to-end transformer architectures. Nevertheless, there are several distinctions:

- **Memory.** We defer self-attention layers until the Region of Interest (RoI) stage, applying self-attention only to targets instead of processing the entire input patch set. This leads to a significant reduction in the memory footprint: while methods such (Li et al., 2023b) demand 8 V100 GPUs for training, our method requires only a single 2080 Ti GPU with the same batch size, aligning with sustainability constraints.
- **Speed.** Moreover, in our approach, the number of tokens for self-attention depends on the number of objects in the frame, thus enhancing scalability and flexibility (as discussed in Sec. 4.3).
- **Adaptability.** Due to its decoupled design, which separates detection from distance estimation, our model can easily adapt to new detectors.
- **Flexibility.** Per-object distance estimation enables predicting distances for partially occluded objects, a critical task for tracking and autonomous driving.

Regarding the accuracy in estimating distances, we will report the results of a current state-of-the-art dense depth estimator such as Depth Anything V2 (Yang et al., 2024) in Tab. 3.

4. Experiments

4.1. Datasets

NuScenes (Caesar et al., 2020) is a large-scale multi-modal dataset with data from 6 cameras, 5 radars, and 1 LiDAR. It comprises 1000 driving scenes collected from urban environments, with 1.4M annotated 3D bounding boxes across 10 object categories. Following (Li et al., 2022), we consider the object's center as the ground truth distance.

MOTSynth (Fabbri et al., 2021) is a large synthetic dataset for pedestrian detection, tracking, and segmentation in an urban environment comprising 764 videos of 1800 frames,

Table 2: Comparison on the NuScenes and MOTSynth datasets. (†) Uses GT poses.

	MOTSynth				NuScenes			
	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$ ↑	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$ ↑
SVR	54.67%	6.758	12.61	26.08%	57.65%	10.48	19.18	32.49%
DisNet	8.73%	0.266	2.507	94.15%	18.47%	1.646	8.270	76.60%
Zhu *et al.*	4.40%	0.116	2.131	98.71%	14.95%	1.244	7.507	84.54%
DistSynth	3.71%	0.073	1.567	99.13%	-	-	-	-
Monoloco†	3.59%	0.064	1.488	99.69%	-	-	-	-
DistFormer (no MoM)	3.36%	0.046	1.152	99.31%	11.16%	0.807	6.363	91.10%
DistFormer (+**MoM**)	**2.81%**	**0.037**	**1.081**	**99.70%**	**8.13%**	**0.533**	**5.092**	**95.33%**

Table 3: Evaluation on KITTI, following the setting in Zhu and Fang (2019).

	ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$ ↑
SVR	147.2%	90.14	24.25	37.90%
IPM	39.00%	274.7	78.87	60.30%
DisNet	25.30%	1.81	6.92	69.83%
Zhu *et al.*	54.10%	5.55	8.74	48.60%
+ classifier	25.10%	1.84	6.87	62.90%
DepthAnythingV2-B	27.37%	2.39	6.11	72.10%
DepthAnythingV2-L	27.22%	2.32	5.65	74.33%
DistFormer-RN	11.40%	0.39	3.42	91.98%
DistFormer (no MoM)	10.61%	0.34	3.17	93.43%
DistFormer (+**MoM**)	**10.39%**	**0.32**	**2.95**	**93.67%**

with different weather conditions, lighting, and viewpoints. Among other annotations, MOTSynth provides 3D coordinates of skeleton joints. Following (Mancusi et al., 2023), we select the distance from the head joint as the ground truth.

KITTI (Geiger et al., 2012) is a well-known benchmark for autonomous driving, object detection, visual odometry, and tracking. The object detection benchmark includes 7481 training and 7518 test RGB images with LiDAR point clouds. A total of 80 256 labeled objects, including pedestrians, cars, and cyclists, are present. Following the convention proposed in (Chen et al., 2018), we divide the train set into training and validation subsets with 3712 and 3768 images, respectively. We obtain ground truth distances for each object from the point cloud, following the strategy in (Zhu and Fang, 2019).

4.2. Experimental setting

Evaluation Setting. We adhere to the widely adopted benchmark (Zhu and Fang, 2019; Haseeb et al., 2018; Jing et al., 2022; Mancusi et al., 2023; Li et al., 2022). Namely, the model is supplied with ground truth bounding boxes (or poses) during inference, along with the input image, to disentangle the detector's performance from the distance estimator's.

Metrics. We rely on popular metrics of per-object distance estimation (Eigen et al., 2014; Zhu and Fang, 2019; Garg et al., 2016; Shu et al., 2020; Liu et al., 2015; Mancusi et al., 2023), such as the τ-**Accuracy** (δ_τ) (Ladicky et al., 2014) (*i.e.*, the maximum allowed relative error), the percentage of objects with relative distance error below a certain threshold ($< k\%$) (Li et al., 2022) and classical error ones (Zhu and Fang, 2019): absolute relative error (**ABS**), square relative error (**SQ**), root mean squared error in linear and logarithmic space (**RMSE** and **RMSE**$_{log}$), average localization error (**ALE**) and average localization of occluded objects error (Mancusi et al., 2023) (**ALOE**). See Appendix F for the equations.

Baselines. Our comparison includes **Geometric methods**, *i.e.*, SVR (Gökçe et al., 2015), IPM (Tuohy et al., 2010), Disnet (Haseeb et al., 2018), and Monoloco (Bertoni et al., 2019) exploits the human pose to infer the distance, and **Feature-based methods**, *i.e.*, Zhu *et al.* (Zhu and Fang, 2019), CenterNet (Duan et al., 2019), PatchNet (Ma et al., 2020), Jing *et al.* (Jing et al., 2022), and DistSynth (Mancusi et al., 2023).

Experimental details. We train every approach with ground truth bounding boxes except Monoloco (Bertoni et al., 2019), trained with ground truth human poses. For the experiments involving NuScenes and MOTSynth, we use the same ConvNeXt backbone for all methods. Since the code bases of other competitors are unavailable for KITTI, we also provide the results with a ResNet-50 (DistFormer-RN row) to provide a more fair comparison. We train end-to-end on an NVIDIA 2080 Ti for 24 hours on NuScenes and MOTSynth and 6 hours on KITTI, applying early stopping.

4.3. Distance Estimation

Tab. 2 and 3 present the results of our approach and previous work. Results on KITTI (Tab. 3) are from their respective papers (apart from DepthAnythingV2), while we implemented other works from scratch for NuScenes and MOTSynth (Tab. 2). We draw the following overall conclusions (further expanded in the following): *i)* our architecture DistFormer achieves state-of-the-art performance on the three datasets under consideration; *ii)* notably, the adaption of the MoM objective (**+ MoM**) furthers the accuracy of our approach with a remarkable and stable gain. In Sec. 4.5, we dissect such evidence through ablation studies to disentangle the merits of the various components involved in DistFormer.

NuScenes presents unique challenges due to its dynamic scenarios, complex traffic situations, and distances up to 150 meters. Despite these challenges, our proposed approach demonstrates robust performance, achieving state-of-the-art results across all metrics. The MOTSynth dataset, instead, focuses

Table 4: ALE and ALOE comparison on KITTI and MOTSynth (using ConvNeXt).

	MOTSynth				KITTI			
	ALE ↓	ALOE ↓			ALE ↓	ALOE ↓		
Method	0m-100m	30-50	50-75	75-100	0m-100m	30-50	50-75	75-100
Zhu *et al*.	1.127	1.29	1.44	1.57	2.084	1.86	2.19	2.21
DistSynth	0.835	1.08	1.15	1.41	-	-	-	-
DistFormer	0.675	0.81	0.88	1.07	1.909	1.76	2.00	2.12
No Global Enc.	0.711	0.86	0.96	1.13	1.994	1.92	1.94	2.03
+ MoM	**0.617**	**0.76**	**0.85**	**0.99**	**1.854**	**1.71**	**1.89**	**1.94**

on the pedestrian class. However, its extensive range of landscapes and viewpoints renders it a comprehensive benchmark. In Tab. 2, our proposed method shows a remarkable -27% in RMSE w.r.t. Monoloco and -49% w.r.t. Zhu *et al*.

Regarding KITTI (Tab. 3), we report the average results (**All**) on the three classes examined (*i.e*., cars, pedestrians, cyclists); we refer the reader to the supplementary materials for a class-wise detailed analysis. Our approach surpasses the state-of-the-art across all classes, except for the car class, which is on par. In this respect, we remark that the methods matching our performance leverage multiple input frames (*e.g*., Jing *et al*. (Jing et al., 2022)), or they are designed to handle the class *car*. In contrast, our approach generalizes over all classes without further adjustments. We also tested Depth Anything V2 (Yang et al., 2024) on KITTI using the original pretrained weights for metric depth estimation. Our method outperforms it in object-level distance estimation, underscoring the difference between per-object and dense distance estimation tasks. Additionally, our model ($\approx 195M$ parameters) runs 6× faster than the Base version ($\approx 97M$ parameters) and 20× faster than the Large version ($\approx 335M$ parameters) on the same GPU.

4.4. The Impact of Masked Object Modeling

MoM enhances transfer learning. In Sec. 3.1, we conjectured that our masking strategy encourages the Local Encoder to prioritize the most consistent patterns (*e.g*., shapes, but not appearance styles). This enables the model to suppress input variations that do not contribute valid information for estimating the distance of target objects. This reduced sensitivity to unimportant variations is advantageous in the case of domain shifts, as it enhances the robustness of the final distance predictor.

To investigate this aspect, we assess the model in the presence of a domain shift, moving from a synthetic scenario (*i.e*., MOTSynth) to a real-world one (*i.e*., KITTI and NuScenes)[2]. In more details: *i)* we train two models on MOTSynth, one with the MoM objective and the other without it; *ii)* then, we move to KITTI and NuScenes and compare the performance of the two models on the class *pedestrian* (*i.e*., the only one present in all datasets). The evaluation performs under two settings: **zero-shot**, without any model refinement on the target dataset, and with **fine-tuning**, allowing a few training steps on a variable number of examples from the target scenario.

Tab. 5 reports the results of the two models (without and **+ MoM**), benchmarked in the above-described evaluation protocol. Notably, there is an impressive gain from MOTSynth to KITTI in the zero-shot scenario ($+13\%$ in $\delta_{<1.25}$), showcasing that our masking strategy extracts features better aligned with real-world domains. Similarly, in the fine-tuning protocol, we note a $+12\%$ in $\delta_{<1.25}$ proving that the **MoM** provides a better starting point to train on new domains, and keeping such an objective (*i.e*., object-level reconstruction) further improve the transfer capabilities of our approach.

Furthermore, in Fig. 4, we report RMSE and $\delta_{<1.25}$ for the fine-tuning experiment with varying numbers of samples to adapt. Specifically, we note how the model with **+ MoM** is much faster to reach convergence and stable w.r.t. to standard fine-tuning, showcasing that such a strategy could also be employed to reduce the training time requirements for the fine-tuning phase.

MoM aids in handling occlusions. **MoM** yields an advantage even for handling partially occluded objects. Specifically, we evaluate the accuracy at different occlusion levels by evaluating the ALOE (Mancusi et al., 2023) metric on MOTSynth and KITTI (Tab. 4). The proposed masking strategy provides a stable and reliable improvement over standard training, showcasing its efficacy. **MoM**'s efficacy in addressing occlusions can be ascribed to its distinctive approach to object representation during model training, resulting in more discernible and stable representations, enabling the model to differentiate between objects and background elements effectively.

MoM yields robustness to noisy bounding boxes. While employing ground truth bounding boxes is a common practice in this setting, one might question the model's performance in cases where bounding boxes are predicted by a detector, potentially influenced by errors. To this end, we employed YOLOX (Ge et al., 2021) as a state-of-the-art detector for MOTSynth, allowing us to gauge the system's resilience in real-world scenarios. Our findings show that incorporating MoM improves the system's performance, nearly reaching the upper bound in terms of the $\delta_{<1.25}$ while achieving a notable reduction in RMSE. Additionally, we purposely perturbed the geometry of ground truth bounding boxes, such that the noisy box and the original one have at least IoU equal to r. This experiment simulates real-world conditions where the exact bounding box

[2]Notably, the intrinsic camera parameters, reported by the authors of these datasets, are very different.

Table 5: Masked Object Modeling impact in domain-shifts.

Masking	Zero-shot (no training) ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$↑	Fine-tuning ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$↑
	MOTSynth → KITTI							
-	18.51%	0.56	2.95	70.44%	6.05%	0.12	1.89	97.58%
+ MoM	17.56%	0.47	2.87	83.57%	5.42%	0.12	1.48	99.16%
	MOTSynth → NuScenes							
-	20.74%	1.93	9.10	44.07%	15.62%	1.10	6.27	80.42%
+ MoM	19.94%	1.74	8.74	46.70%	10.28%	0.64	5.22	92.23%

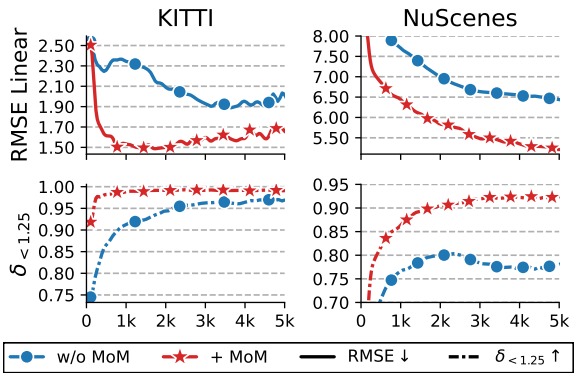

Fig. 4: Resulting performance on the class *pedestrian* after fine-tuning with varying training set sizes.

Table 6: Ablation of the backbone and modules on MOTSynth.

Contextual Enc.	Local Enc.	Global Enc.	ABS ↓	SQ ↓	RMSE ↓	$\delta_{<1.25}$ ↑
Ablative studies on the Contextual Encoder						
ViT-B/16	✓	ViT	7.88%	0.460	3.973	92.78%
	+MoM	ViT	6.81%	0.316	3.473	94.90%
ResNet34	✓	ViT	4.14%	0.107	2.078	98.93%
	+MoM	ViT	4.36%	0.094	1.826	98.94%
ResNet34-FPN	-	-	4.45%	0.102	1.975	98.91%
	✓	-	3.44%	0.056	1.363	99.53%
	-	ViT	3.30%	0.054	1.302	99.59%
	✓	ViT	3.15%	0.050	1.302	99.70%
	+MoM	GAT	3.49%	0.049	1.213	99.51%
	+MoM	ViT	3.00%	0.040	1.146	99.70%
Ablative studies on Local Encoder & MoM						
ConvNeXt-S-FPN	-	-	3.38%	0.055	1.289	99.31%
	✓	-	3.41%	0.055	1.275	99.34%
	-	ViT	3.38%	0.052	1.236	99.43%
	✓	ViT	3.36%	0.046	1.152	99.31%
	+MoM	-	3.31%	0.053	1.290	99.38%
	+MoM	GAT	3.26%	0.048	1.221	99.63%
ConvNeXt-S-FPN	+MoM	ViT	**2.81%**	**0.037**	**1.081**	**99.70%**

might be imprecise, showing the benefits of **MoM** (more details in supplementary materials).

4.5. Ablation studies

We herein report an extensive ablation study (Tab. 6) of our model on the MOTSynth dataset. The supplementary materials offer a similar analysis for NuScenes.

Local Encoder and MoM. The experiments with the ResNet34-FPN and the ConvNeXt-S-FPN highlight the role of the LE to process local cues. Indeed, its application (✓) improves all metrics. Moreover, our masking (+ MoM) further improves results, confirming our claims regarding its regularizing effect.

Global Encoder. Discarding the Global Encoder worsen the performance, confirming what reported in Sec. 3. We also assess the merits of the ViT layers by comparing them with Graph Attention Network (GAT) (Veličković et al., 2018; Brody et al., 2022). Notably, GAT leads to improvements w.r.t. not using a Global Encoder. However, ViT layers consistently outperform their GAT counterparts, underscoring their efficacy for multi-object analysis.

Contextual Encoder. The use of ResNet leads to lower results (especially when removing the FPN layers), indicating the significance of superior and larger feature maps. Notably, we observe a severe degradation when using the ViT-B/16[3], showcasing the efficacy of convolutional networks in extracting valuable feature for multi-object tasks.

5. Conclusion and Future Works

In this work, we propose DistFormer, an architecture designed for per-object distance estimation and a novel self-supervised objective termed Masked Object Modeling (MoM), which extends standard masking to multi-object analysis. The experimental outcomes indicate that DistFormer provides robust and reliable distance estimates. Higher-level tasks such as multi-object tracking could leverage the predictions of our approach, enabling the tracker to incorporate three-dimensional reasoning. Moreover, adding the MoM training objective provides strong regularizing benefits, ranging from better transfer capabilities to resilience to occlusions and noisy detection.

In future work, we aim to investigate the application of the MoM paradigm to a broader range of tasks such as pose estimation, object detection, and segmentation. Moreover, enhancing domain adaptability through techniques like meta-learning or synthetic-to-real transfer could broaden its applicability across diverse environments. Developing lightweight variants via pruning or quantization would make Masked Object Modeling even more suitable for resource-constrained systems.

Limitations & Societal Impact

The accuracy of DistFormer relies on the precision of the detector used to locate target objects. However, as it is agnostic to the detector, it allows for future-proofing and can adapt to novel and more accurate object detectors with no extensive modifications or network retraining. Nevertheless, the use of

[3]Due to its significant memory footprint at full resolution (*i.e.*, 720×1280), we resort to the standard resolution of 224×224.

deep techniques for distance estimation raises concerns regarding privacy and legal liabilities. To address these potential negative effects on society, it is imperative to carefully establish appropriate regulations, ethical guidelines, and promote social awareness.

Acknowledgements

Angelo Porrello was financially supported by the Italian Ministry for University and Research – through the ECOSISTER ECS 00000033 CUP E93C22001100001 project – and the European Commission under the Next Generation EU programme PNRR. Rita Cucchiara and Simone Calderara were financially supported by the EU Horizon project "ELIAS - European Lighthouse of AI for Sustainability" (No. 101120237).

References

Alhashim, I., Wonka, P., 2018. High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:1812.11941 .

Bertoni, L., Kreiss, S., Alahi, A., 2019. Monoloco: Monocular 3d pedestrian localization and uncertainty estimation, in: IEEE International Conference on Computer Vision.

Bielski, A., Favaro, P., 2022. Move: Unsupervised movable object segmentation and detection, in: Advances in Neural Information Processing Systems.

Brody, S., Alon, U., Yahav, E., 2022. How attentive are graph attention networks?, in: International Conference on Learning Representations Workshop.

Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Chen, L., Ai, H., Zhuang, Z., Shang, C., 2018. Real-time multiple people tracking with deeply learned candidate selection and person re-identification, in: IEEE International Conference on Multimedia and Expo.

Der Kiureghian, A., Ditlevsen, O., 2009. Aleatory or epistemic? does it matter? Structural safety .

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations Workshop.

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q., 2019. Centernet: Keypoint triplets for object detection, in: IEEE International Conference on Computer Vision.

Eigen, D., Puhrsch, C., Fergus, R., 2014. Depth map prediction from a single image using a multi-scale deep network. Advances in Neural Information Processing Systems .

Fabbri, M., Brasó, G., Maugeri, G., Ošep, A., Gasparini, R., Cetintas, O., Calderara, S., Leal-Taixé, L., Cucchiara, R., 2021. Motsynth: How can synthetic data help pedestrian detection and tracking?, in: IEEE International Conference on Computer Vision.

Gandelsman, Y., Sun, Y., Chen, X., Efros, A., 2022. Test-time training with masked autoencoders. Advances in Neural Information Processing Systems .

Garg, R., Bg, V.K., Carneiro, G., Reid, I., 2016. Unsupervised cnn for single view depth estimation: Geometry to the rescue, in: Proceedings of the European Conference on Computer Vision.

Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J., 2021. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 .

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Girshick, R., 2015. Fast r-cnn, in: IEEE International Conference on Computer Vision.

Godard, C., Mac Aodha, O., Brostow, G.J., 2017. Unsupervised monocular depth estimation with left-right consistency, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J., 2019. Digging into self-supervised monocular depth estimation, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Gogel, W.C., 1963. The visual perception of size and distance. Vision Research .

Gökçe, F., Üçoluk, G., Şahin, E., Kalkan, S., 2015. Vision-based detection and distance estimation of micro unmanned aerial vehicles. Sensors .

Han, Q., Zhang, G., Huang, J., Gao, P., Wei, Z., Lu, S., 2024. Efficient mae towards large-scale vision transformers, in: IEEE/CVF Winter Conference on Applications of Computer Vision.

Haseeb, M.A., Guan, J., Ristic-Durrant, D., Gräser, A., 2018. Disnet: a novel method for distance estimation from monocular camera. 10th Planning, Perception and Navigation for Intelligent Vehicles (PPNIV18), IROS .

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2022. Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Jing, L., Yu, R., Kretzschmar, H., Li, K., Qi, C.R., Zhao, H., Ayvaci, A., Chen, X., Cower, D., Li, Y., et al., 2022. Depth estimation matters most: improving per-object depth estimation for monocular 3d detection and tracking, in: International Conference on Robotics and Automation.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. Imagenet classification with deep convolutional neural networks. Communications of the ACM .

Ladicky, L., Shi, J., Pollefeys, M., 2014. Pulling things out of perspective, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O., 2019. Pointpillars: Fast encoders for object detection from point clouds, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Lee, J.H., Han, M.K., Ko, D.W., Suh, I.H., 2019. From big to small: Multi-scale local planar guidance for monocular depth estimation. arXiv preprint arXiv:1907.10326 .

Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y., 2023a. Mask dino: Towards a unified transformer-based framework for object detection and segmentation, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Li, Y., Chen, T., Kabkab, M., Yu, R., Jing, L., You, Y., Zhao, H., 2022. R4d: Utilizing reference objects for long-range distance estimation, in: International Conference on Learning Representations Workshop.

Li, Z., Chen, Z., Liu, X., Jiang, J., 2023b. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. Machine Intelligence Research .

Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Liu, F., Shen, C., Lin, G., Reid, I., 2015. Learning depth from single monocular images using deep convolutional neural fields. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S., 2022. A convnet for the 2020s, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W., 2020. Rethinking pseudo-lidar representation, in: Proceedings of the European Conference on Computer Vision.

Mallot, H.A., Bülthoff, H.H., Little, J., Bohrer, S., 1991. Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological cybernetics .

Mancusi, G., Panariello, A., Porrello, A., Fabbri, M., Calderara, S., Cucchiara, R., 2023. Trackflow: Multi-object tracking with normalizing flows, in: IEEE International Conference on Computer Vision.

Nix, D.A., Weigend, A.S., 1994. Estimating the mean and variance of the target probability distribution, in: Proceedings of the IEEE International Conference on Neural Networks.

Ranftl, R., Bochkovskiy, A., Koltun, V., 2021. Vision transformers for dense prediction, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V., 2020. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-

dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence .

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems .

Rezaei, M., Terauchi, M., Klette, R., 2015. Robust vehicle detection and distance estimation under challenging lighting conditions. IEEE Transactions on Intelligent Transportation Systems .

Shu, C., Yu, K., Duan, Z., Yang, K., 2020. Feature-metric loss for self-supervised learning of depth and egomotion, in: Proceedings of the European Conference on Computer Vision.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations Workshop .

Sinai, M.J., Ooi, T.L., He, Z.J., 1998. Terrain influences the accurate judgement of distance. Nature .

Tuohy, S., O'Cualain, D., Jones, E., Glavin, M., 2010. Distance determination for an automobile environment using inverse perspective mapping in opencv, in: IET Irish Signals and Systems Conference.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks, in: International Conference on Learning Representations Workshop.

Wang, D., Gao, L., Qu, Y., Sun, X., Liao, W., 2023a. Frequency-to-spectrum mapping gan for semisupervised hyperspectral anomaly detection. CAAI Transactions on Intelligence Technology 8, 1258–1273.

Wang, D., Zhuang, L., Gao, L., Sun, X., Huang, M., Plaza, A., 2023b. Bocknet: Blind-block reconstruction network with a guard window for hyperspectral anomaly detection. IEEE Transactions on Geoscience and Remote Sensing 61, 1–16.

Wang, D., Zhuang, L., Gao, L., Sun, X., Zhao, X., Plaza, A., 2024. Sliding dual-window-inspired reconstruction network for hyperspectral anomaly detection. IEEE Transactions on Geoscience and Remote Sensing .

Wu, Q., Yang, T., Liu, Z., Wu, B., Shan, Y., Chan, A.B., 2023. Dropmae: Masked autoencoders with spatial-attention dropout for tracking tasks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Yang, B., Luo, W., Urtasun, R., 2018. Pixor: Real-time 3d object detection from point clouds, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H., 2024. Depth anything v2. arXiv preprint arXiv: 2406.09414 .

Zhao, H., Wang, D., Lu, H., 2023. Representation learning for visual object tracking by masked appearance transfer, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition.

Zhu, J., Fang, Y., 2019. Learning object-specific distance from a monocular image, in: IEEE International Conference on Computer Vision.

Supplementary Material

Appendix A. Bounding Box Prior Through Centers Mask

To provide an additional signal on the objects, we feed the backbone with a further channel representing the centers of the bounding boxes. Specifically, we construct a heatmap h where we apply a fixed variance Gaussian over each center. Formally, given the bounding box $\mathbf{t}^k = (t_x^k, t_y^k, t_w^k, t_h^k)$, with $k \in \{1, \dots, K\}$, where K is the number of the bounding boxes in the frame, and the bounding box tuple represents the x and y coordinates of the top left corner (t_x^k, t_y^k) and its width and height (t_w^k, t_y^k), the heatmap h^k for a generic bounding box centered in $\mathbf{c}^k = (c_x^k, c_y^k) = (t_x^k + t_w^k/2, t_y^k + t_h^k/2)$ is given by:

$$h^k(\mathbf{u}) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{c}^k\|^2}{\sigma^2}\right),$$

where $\mathbf{u}$ is the generic (x, y) location of the heatmap. In a multi-object context, where centers may overlap, we aggregate the heatmaps h^k into a single heatmap h with a max operation:

$$h = \max_k \{h^k(\mathbf{u})\}.$$

In Tab. A, we present the ablation results of the centers' heatmap on MOTSynth. In particular, we show that adding such a signal leads to a slight improvement in all metrics.

Table A: Contribute of the centers mask on MOTSynth.

Centers	ABS ↓	SQ ↓	RMSE ↓	$\text{RMSE}_{\log}$ ↓	$\delta_{<1.25}$ ↑
	3.07%	0.051	1.266	0.048	99.52%
✓	**2.81%**	**0.037**	**1.081**	**0.043**	**99.70%**

Appendix B. Long-Range Distance Estimation

We additionally exploit Pseudo Long-Range KITTI and NuScenes (Li et al., 2022) to assess performance for **long-range objects** (*i.e.*, beyond 40 meters). The KITTI subset, comprises 2181 training images and 2340 validation images, with 4233 and 4033 vehicles, respectively. The NuScenes subset includes 18 926 training images with 59 800 target vehicles and 4017 validation images with 11 737 target vehicles. The hyperparameters used for these datasets are the same as Tab. 2 and 3; no additional tuning has been carried out.

As shown in Tab. B, DistFormer proves effective also on these benchmarks. The best competitor is R4D (Li et al., 2022), which needs additional input sensor data at inference time (*i.e.*, LiDAR), differently from our approach that instead requires only a single monocular image. We mainly ascribe the gains of our approach to the different mechanism employed to gather global/spatial information. While R4D builds upon the graph of pair-wise relationships between the target object and its references, we leverage self-attention to encode global relations among the whole set of objects in the scene.

Table B: Comparison on the Pseudo Long-Range KITTI and NuScenes datasets. Here $< k\%$ is accuracy below $k\%$ error.

	Dataset (Long Range)	LiDAR	Lower is better ABS	SQ	RMSE	Higher is better $< 5\%$	$< 10\%$	$< 15\%$
DisNet	KITTI	-	10.6%	1.55	10.4	37.1%	65.0%	77.7%
Zhu *et al.*	KITTI	-	8.7%	0.88	7.7	39.4%	65.8%	80.2%
Zhu *et al.*	KITTI	✓	8.9%	0.97	8.1	41.1%	66.5%	78.0%
R4D	KITTI	✓	7.5%	0.68	6.8	46.3%	72.5%	83.9%
Ours	KITTI	-	**5.2%**	**0.22**	**3.3**	**56.3%**	**88.3%**	**97.3%**
DisNet	NuScenes	-	10.7%	1.46	10.5	29.5%	58.6%	75.0%
Zhu *et al.*	NuScenes	-	8.4%	0.91	8.6	40.3%	66.7%	80.3%
Zhu *et al.*	NuScenes	✓	9.2%	1.06	9.2	37.7%	63.5%	77.2%
R4D	NuScenes	✓	7.6%	0.75	7.7	44.2%	71.1%	84.6%
Ours	NuScenes	-	**7.3%**	**0.65**	**6.8**	**47.3%**	**75.4%**	**88.6%**

Appendix C. KITTI pre-processing

To obtain the ground truth annotation for the KITTI (Geiger et al., 2012) dataset, we follow the setting proposed by Zhu *et al*. (Zhu and Fang, 2019). Specifically, for each object in the scene, we get all the point cloud points inside its 3D bounding box and sort them by distance. The chosen *keypoint* will be the n-th depth point where $n = 0.1 \cdot$ (number of points). After that, we remove objects marked with the *Don't Care* class and objects with a negative distance from the training set, which are objects behind the camera but still captured by the LiDAR.

Appendix D. NuScenes pre-processing

We utilized the preprocessing methodology outlined in (Li et al., 2022) for NuScenes (Caesar et al., 2020); specifically, we exploited the code provided by its authors[4] to convert the dataset into the KITTI format. This conversion allows us to leverage existing KITTI-specific code. It is worth noting that, unlike KITTI, where we adhere to the configuration proposed by (Zhu and Fang, 2019), for NuScenes, we adopt the Z component of the 3D bounding box's center as the annotation.

Appendix E. MOTSynth pre-processing

Since the MOTSynth (Fabbri et al., 2021) dataset was generated synthetically, its set of annotations covers all the pedestrians in the scene. On the one hand, we believe that such a variety could be beneficial and ensure good generalization capabilities; on the other hand, we observed that it hurts the performance, as some target annotations are highly *noisy* or extremely difficult for the learner. Thus we follow the filtering step from (Mancusi et al., 2023). Specifically, the dataset contains annotations even for completely occluded people (*e.g.*, behind a wall) or located very far from the camera (*e.g.*, even at 100 meters away). Hence, we discard these cases from the

[4] https://github.com/nutonomy/nuscenes-devkit/blob/master/python-sdk/nuscenes/scripts/export_kitti.py

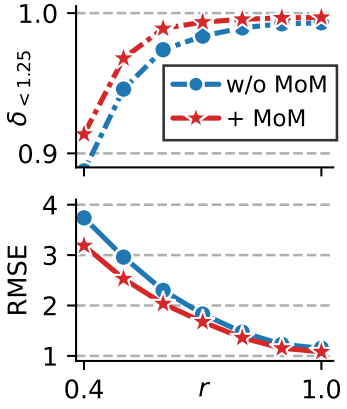

Fig. A: RMSE and $\delta_{<1.25}$ at varying bounding box noise r.

training and evaluation phases, performing a preliminary data-cleaning stage. Namely, in each experiment, we exclude pedestrians not visible from the camera viewpoint or located beyond the threshold used in (Mancusi et al., 2023) (*i.e.*, 70 meters).

Lastly, we sub-sample the official MOTSynth test set, keeping one out of 400 frames. This way, we avoid redundant computations and speed up the evaluation procedure.

Appendix F. Metrics

In the following, we present the equations for the standard distance estimation metrics used in our work.

$$\delta_\tau : \% \text{ of } d_i \; s.t. \max\left(\frac{d_i}{d_i^*}, \frac{d_i*}{d_i}\right) = \delta < \tau,$$

$$\text{ALE}_{[\tau_1:\tau_2]} = \frac{1}{N}\sum\nolimits_{d\in N_{[\tau_1:\tau_2]}} \left(|d - d^*|\,/d^*\right),$$

$$\text{ALOE}_{[\tau_1:\tau_2]} = \frac{1}{N}\sum\nolimits_{occl\in N_{[\tau_1:\tau_2]}} \left(|d - d^*|\,/d^*\right),$$

$$< \phi\%\text{-Accuracy} = \delta_{<\phi},$$

$$\text{ABS} = \frac{1}{N}\sum\nolimits_{d\in N} \left(|d - d^*|\,/d^*\right),$$

$$\text{SQ} = \frac{1}{N}\sum\nolimits_{d\in N} \left((d - d^*)^2\,/d^*\right),$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum\nolimits_{d\in N} \left((d - d^*)^2\right)},$$

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{N}\sum\nolimits_{d\in N} \left((\log d - \log d^*)^2\right)},$$

where d^* are the ground truth distances and d the predicted distances.

Appendix G. Class-Wise KITTI Results

We report in Tab. C the class-wise results for the KITTI dataset. Remarkably, our approach outperforms all previous methods by a wide margin but for the car class. However, we note that CenterNet (Duan et al., 2019), PatchNet (Ma et al., 2020), and Jing *et al.* (Jing et al., 2022) make use of additional training data other than being trained solely on the car category. Specifically, CenterNet and PatchNet incorporate

Table C: Experimental comparison on KITTI, following the setting in. (†) Requires additional training data.

		ABS↓	SQ↓	RMSE↓	$\delta_{<1.25}$ ↑
Car	SVR	149.4%	47.7	18.97	34.50%
	IPM	49.70%	1290	237.6	70.10%
	DisNet	26.49%	1.64	6.17	70.21%
	Zhu *et al.*	16.10%	0.61	3.58	84.80%
	CenterNet†	8.70%	0.43	3.24	95.33%
	PatchNet†	8.08%	0.28	2.90	95.52%
	Jing *et al.* †	**6.89%**	0.23	2.50	**97.60%**
	DistFormer (+ MoM)	9.97%	**0.22**	**2.11**	94.32%
Pedestrian	SVR	149.9%	34.56	21.68	12.90%
	IPM	34.00%	543.2	192.18	68.80%
	DisNet	7.69%	0.27	3.05	93.24%
	Zhu *et al.*	18.30%	0.65	3.44	74.70%
	DistFormer (+ MoM)	**5.67%**	**0.08**	**1.26**	**98.15%**
Cyclists	SVR	125.1%	31.61	20.54	22.60%
	IPM	32.20%	9.54	19.15	65.50%
	DisNet	12.13%	0.96	7.09	84.42%
	Zhu *et al.*	18.80%	0.92	4.89	76.80%
	DistFormer (+ MoM)	**8.01%**	**0.25**	**3.09**	**95.62%**
All	SVR	147.2%	90.14	24.25	37.90%
	IPM	39.00%	274.7	78.87	60.30%
	DisNet*	25.30%	1.81	6.92	69.83%
	Zhu *et al.*	54.10%	5.55	8.74	48.60%
	+ classifier	25.10%	1.84	6.87	62.90%
	DistFormer-RN	11.40%	0.39	3.42	91.98%
	DistFormer (no MoM)	10.61%	0.34	3.17	93.43%
	DistFormer (+ MoM)	**10.39%**	**0.32**	**2.95**	**93.67%**

Table D: Performance of DistFormer with predicted bounding boxes

	BB	RMSE↓	$\delta_{<1.25}$↑
DisNet	YOLO	2.840	90.80%
Zhu *et al.*	YOLO	1.820	98.74%
DistSynth	YOLO	1.781	98.83%
DistFormer	YOLO	1.768	98.95%
+ MoM	YOLO	1.498	99.56%
DistFormer	GT	0.813	99.85%

3D bounding box information. In contrast, Jing *et al.* leverage both 3D bounding boxes and multiple frames for training. Despite this, our method achieves comparable performance to these approaches, even surpassing them in some aspects (notably, achieving a -16% reduction in RMSE) while also having the capability to predict distances for all classes present in the KITTI dataset. We remark that the methods reported only for the car class do not release the results for other classes; thus, we do not report them.

Appendix H. MoM yields robustness to noisy bounding boxes

As stated in the main paper, we provide further details on our experiment with YOLOX as a detector for MOTSynth. We discuss how incorporating MoM improves the system's performance, nearly reaching the upper bound in terms of $\delta_{<1.25}$ while

Table E: Ablation of proposed modules on the NuScenes dataset (using the ConvNeXt backbone).

Local Enc.	Global Enc.	ABS ↓	SQ ↓	RMSE ↓	$\delta_{<1.25}$ ↑
-	-	8.71%	0.587	5.459	94.77%
✓	-	8.37%	0.554	5.210	95.06%
-	ViT	8.49%	0.558	5.341	94.98%
✓	ViT	8.18%	0.534	5.243	95.20%
+MoM	-	8.69%	0.568	5.170	95.14%
+MoM	GAT	10.06%	0.697	5.738	93.38%
+MoM	ViT	**8.13%**	**0.533**	**5.092**	**95.33%**

achieving a notable reduction in RMSE, see Tab. D. Additionally, we intentionally perturbed the geometry of ground truth bounding boxes, ensuring that the noisy box and the original one have at least an IoU equal to r. This experiment simulates real-world conditions where the exact bounding box might be imprecise, further demonstrating the benefits of MoM. Results reported in Fig. A

Appendix I. Implementation Details

The input of our contextual encoder, composed of a ConvNeXt and an FPN branch, is the full-resolution image. We extract the feature vectors of the objects from the feature map via *ROIAlign* (He et al., 2017) with a window of 8×8. Successively, we split the feature map in tokens, then we randomly mask 50% of the tokens and feed the unmasked ones to the Local Encoder (LE), which is composed of the last 6 layers of a ViT-B/16 pretrained on ImageNet. The output of the LE is used both in the MoM branch (during training only) and the distance regression branch. The MoM Decoder and the Global Encoder are 2-layer transformer encoders with 8 heads. We report in Tab. F the additional hyper-parameters for the different datasets.

Appendix J. Further Ablations

We present in Tab. E the ablations of the Local Encoder, Global Encoder, and MoM module on the NuScenes dataset, following Sec. 4.5. Similar to what we found on the MOTSynth dataset, each module also improves the baseline on the NuScenes dataset, specifically with neither the Local Encoder nor the Global Encoder. Furthermore, such results show the importance of using the Transformer attention mechanism as a Global Encoder since the GAT considerably reduces the performance, even below the baseline. Finally, combining all the modules leads to more remarkable performance, especially on the RMSE metric.

Table F: DistFormer hyperparameters.

config	MOTSynth	KITTI	NuScenes
ConvNeXt size	Small	Base	Small
input resolution	720×1280	375×1242	900×1600
optimizer	AdamW	AdamW	AdamW
base learning rate	1×10^{-4}	5×10^{-5}	5×10^{-5}
learning rate schedule	cosine annealing WR	cosine annealing WR	cosine annealing WR
weight decay	1×10^{-5}	2×10^{-5}	1×10^{-5}
MoM weight	$\alpha = 10$	$\alpha = 20$	$\alpha = 10$
MoM masking ratio	50%	50%	50%
optimizer momentum	$\beta_1, \beta_2 = (0.9, 0.999)$	$\beta_1, \beta_2 = (0.9, 0.999)$	$\beta_1, \beta_2 = (0.9, 0.999)$
batch size	2	4	2
dist loss delay epochs	0	20	0
dist loss warmup epochs	11	10	0
augmentation	RndLRFlip ($p = 0.5$)	RndLRFlip ($p = 0.5$)	RndLRFlip ($p = 0.5$)
	ColorJitter ($p = 0.25$)		
	GaussianBlur ($p = 0.25$)		
	RndGrayscale ($p = 0.2$)		
	RndAdjustSharpness ($p = 0.5$)		

Algorithm A DistFormer Training Phase

Input: Image $x \in \mathbb{R}^{C \times H \times W}$, bounding boxes $\{b_1, b_2, \ldots, b_N\}$, boxes RGB crops $\{x_1, x_2, \ldots, x_N\}$
Output: Distances $\{d_1, d_2, \ldots, d_N\}$ for N objects
Stage 1: Contextual Encoding
$\mathcal{F} \leftarrow \text{ContextualEncoder}(x)$ ▷ Extract feature map using ConvNeXt with FPN
Stage 2: Region of Interest (RoI) and Local Encoding
for $i = 1$ to N **do**
    $\mathcal{F}_i \leftarrow \text{RoIAlign}(\mathcal{F}, b_i)$ ▷ Extract features for bounding box b_i
    $T_i \leftarrow \text{Tokenize}(\mathcal{F}_i)$ ▷ Split features into tokens
    $H_i \leftarrow \text{LocalEncoder}(T_i)$ ▷ Process intra-object features with ViT layers
    $\hat{H}_i \leftarrow \text{AvgPool}(H_i)$ ▷ from $\mathbb{R}^{c \times (h \cdot w)}$ to $\mathbb{R}^{c \times 1}$
end for
Stage 3: Masked Object Modeling
for $i = 1$ to N **do**
    $H_i^{\text{masked}} \leftarrow \text{Mask}(H_i, \text{ratio} = 50\%)$ ▷ Randomly mask tokens over h and w
    $x_i^{\text{reconstructed}} \leftarrow \text{Decoder}(H_i^{\text{masked}})$ ▷ Reconstruct masked object regions
end for
$\mathcal{L}_{\text{MoM}} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \|x_i^{\text{reconstructed}} - x_i\|^2$ ▷ Compute reconstruction loss as in Eq. (1)
Stage 4: Global Encoding
$H_{\text{global}} \leftarrow \text{Concatenate}(\{\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_N\})$ ▷ Aggregate object-level tokens
$H_{\text{global}} \leftarrow \text{GlobalEncoder}(H_{\text{global}})$ ▷ Process inter-object relationships with ViT
Stage 5: Distance Prediction
for $i = 1$ to N **do**
    $d_i, \sigma_i \leftarrow \text{MLP}(H_{\text{global},i})$ ▷ Predict distance d_i and uncertainty σ_i
end for
Final Training Objective
$\mathcal{L} \leftarrow \alpha \mathcal{L}_{\text{MoM}} + \mathcal{L}_{\text{GNLL}}$ ▷ Combine losses as explained in Eq. (2)

Algorithm B DistFormer Inference Phase

Input: Image $x \in \mathbb{R}^{C \times H \times W}$, bounding boxes $\{b_1, b_2, \ldots, b_N\}$
Output: Distances $\{d_1, d_2, \ldots, d_N\}$ for N objects
Stage 1: Contextual Encoding
$\mathcal{F} \leftarrow \text{ContextualEncoder}(x)$ ▷ Extract feature map using ConvNeXt with FPN
Stage 2: Region of Interest (RoI) and Local Encoding
for $i = 1$ to N **do**
    $\mathcal{F}_i \leftarrow \text{RoIAlign}(\mathcal{F}, b_i)$ ▷ Extract features for bounding box b_i
    $T_i \leftarrow \text{Tokenize}(\mathcal{F}_i)$ ▷ Split features into tokens
    $H_i \leftarrow \text{LocalEncoder}(T_i)$ ▷ Process intra-object features with ViT layers
    $\hat{H}_i \leftarrow \text{AvgPool}(H_i)$ ▷ from $\mathbb{R}^{c \times (h \cdot w)}$ to $\mathbb{R}^{c \times 1}$
end for
Stage 3: Global Encoding
$H_{\text{global}} \leftarrow \text{Concatenate}(\{\hat{H}_1, \hat{H}_2, \ldots, \hat{H}_N\})$ ▷ Aggregate object-level tokens
$H_{\text{global}} \leftarrow \text{GlobalEncoder}(H_{\text{global}})$ ▷ Process inter-object relationships with ViT
Stage 4: Distance Prediction
for $i = 1$ to N **do**
    $d_i, \sigma_i \leftarrow \text{MLP}(H_{\text{global},i})$ ▷ Predict distance d_i and uncertainty σ_i
end for