Self-supervised Monocular Depth Estimation for Dynamic Objects with Ground Propagation

Huan Li Matteo Poggi Fabio Tosi Stefano Mattoccia
University of Bologna, Italy

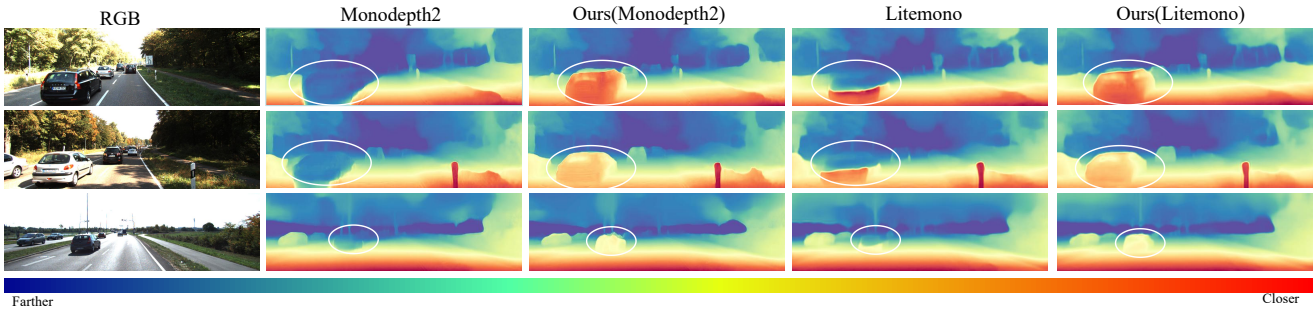

Fig. 1. **Handling dynamic objects with ground propagation.** Our solution effectively boosts the performance of existing self-supervised models such as Monodepth2 [1] and Lite-Mono [2].

***Abstract*— Self-supervised single-view depth estimation, trained on video sequences, faces significant challenges when dynamic objects are present in the training data, as they violate the basic multi-view geometry assumptions used to compute photometric losses. We propose a novel approach that leverages the relationship between the depth of moving objects and their ground contact points. By iteratively propagating ground features to moving targets in perceptual layers, we recalibrate the depth of dynamic entities while preserving details. Our method maintains the end-to-end training paradigm without additional networks or complex training procedures. Our experiments demonstrate that our method achieves state-of-the-art performance when estimating depth for dynamic objects and attains superior generalization compared to existing approaches.**

I. Introduction

Monocular depth estimation, allowing for the reconstruction of 3D structures from a single image, has become a crucial task in computer vision. Recent neural network advancements have significantly improved this field, with data-driven approaches [3]–[5] achieving impressive accuracy and generalization. These methods leverage diverse forms of supervision, including depth obtained from multi-view stereo images, sensor data, and synthetic datasets. However, acquiring such comprehensive depth labels remains a significant challenge, often proving expensive and labor-intensive. Alternatively, self-supervised monocular depth estimation eliminates the need for depth labels by using view synthesis signals derived from image reprojection across frames during training, exploiting the estimated scene depth and camera pose. Two main variants exist: (1) using stereo pairs with known camera positions [6], and (2) the more challenging case of unconstrained monocular videos [7], where camera motion is also estimated. The latter is particularly appealing as it only requires a single moving camera for data collection, enabling training on arbitrary sequences and potentially enhancing generalization to unseen environments. However, when using monocular videos, the presence of dynamic objects during training can be problematic because it violates the multi-view consistency assumption, which requires that objects remain static across frames.

While some methods attempt to address this by masking out moving objects during training [8], this can lead to poor generalization for dynamic objects as the network receives insufficient supervisory signal on them. Alternatively, other approaches incorporate auxiliary 2D optical flow or scene flow estimation [9]–[11] to disambiguate dynamic objects. While these techniques can improve performance, they introduce additional complexity and training challenges due to the larger network architectures.

Another line of research exploits the geometric assumption that depth for dynamic objects is generally consistent with the distance of their ground contact points from the camera. Leveraging this idea, some methods incorporate ground information [12]–[14] into object depth estimation by applying ground consistency loss functions or ground cumulative convolution. However, these approaches may struggle when estimating depth for objects closer to the camera, where surfaces vary vertically, as the ground consistency assumption becomes less reliable. Consequently, they often rely on depth labels for supervision or require a supplementary fine-tuning stage to refine results, which precludes end-to-end training.

Building on the idea of using ground information for dynamic object depth estimation, we propose a novel approach that addresses the limitations of previous methods while allowing for end-to-end training. Our method is based on the observation that in the decoder of a depth network, the activated feature maps across different channels can be categorized into depth-aware and detail-aware feature maps: the former provides information concerning the depth distribution in the scene and its smooth behavior, the latter highlights discontinuities and high-frequency details. We

argue that, by acting on the former category, we can propagate features extracted from ground regions up to the moving objects, yielding the decoder to predict a depth consistent with their ground contact point.

To effectively propagate ground features, we first identify depth-aware feature maps that closely relate to the final depth map's structure. We do this by calculating the cosine similarity between these maps and depth pseudo-labels derived from the ground. We then focus propagation only on the highest-scoring maps. This process is repeated multiple times to ensure ground feature propagation even when large, moving objects are present. Our strategy is simple, yet effective at solving the problem and can be easily integrated into any state-of-the-art monocular depth estimation network.

In summary, our main contributions are:

- We propose ground propagation, a novel method for dealing with moving objects when training self-supervised monocular depth estimation models, as shown in Fig. 1.
- Our method is compatible with existing models and requires no additional network parameters.
- Experiments on KITTI and DrivingStereo datasets highlight that our strategy improves the accuracy of any baseline model and achieves state-of-the-art results for dynamic objects.

II. Related Work

We review the literature trends relevant to our work.

Monocular Depth Estimation. Monocular depth estimation has progressed significantly since the early days of handcrafted features [15]. Deep learning revolutionized the field, with supervised methods making remarkable advancements [16]–[19]. However, the need for large labeled datasets led to the development of self-supervised techniques. Pioneered by Garg et al. [20] with stereo pairs, and extended to monocular video by Zhou et al. [7], self-supervised approaches have inspired new efforts based on feature-based reconstructions [21], [22], semantic segmentation integration [23], [24], and proxy depth representations [25], [26]. Architectural advancements have been crucial, evolving from standard CNNs to sophisticated designs incorporating multiscale fusion [27] and attention mechanisms [28]. Transformer-based models [29], [30] have further improved accuracy, while lightweight architectures [31]–[33] address real-time applications.

The challenge of "in-the-wild" generalization led to affine-invariant models [4], [34]–[36], exploiting diverse datasets for cross-domain performance. Recent trends include incorporating camera intrinsics [37], [38], applying diffusion models [39], [40], and addressing non-Lambertian surfaces [41], [42].

Handling Dynamic Objects. Moving objects challenge self-supervised monocular depth estimation by violating the rigid scene assumption. Some approaches [6], [43]–[45] mask out moving objects, but this reduces effectiveness in motion-dense scenes. Other methods [8], [10], [46]–[49] incorporate motion networks to estimate object motion, yet struggle with complex or sparse motions. A promising approach leverages the geometric principle that object depth should align with its ground contact point. Moon et al. [13] proposed a ground-contacting disparity smoothness loss, but struggled with near-camera objects, requiring an additional refinement stage. Our method differs by using ground depth consistency without auxiliary motion networks. We apply an iterative propagation algorithm on depth-related feature maps, allowing ground depth to propagate throughout objects. This approach facilitates end-to-end training without additional parameters, ensuring compatibility with various monocular depth estimation networks.

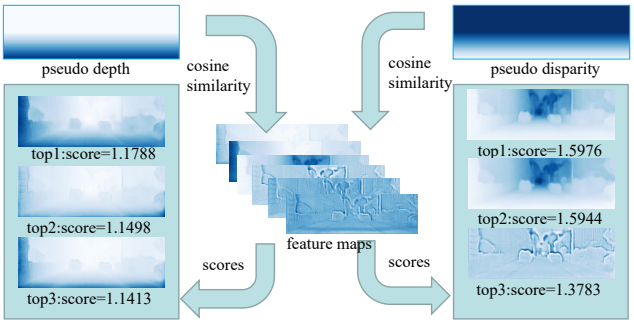

Fig. 2: **Feature maps ranking.** We construct pseudo depth/disparity maps to rank the feature maps and identify depth-related maps, according to the cosine similarity with the former. Pseudo depths/disparities are generated to recall the depth/disparity distribution in correspondence with the ground plane.

III. Background

Retrieving depth from a single image is an ill-posed problem. However, [7] demonstrated that multi-view geometry from unlabeled monocular video sequences can be leveraged during training to enable self-supervised depth estimation, assuming that the scenes are mostly rigid, with scene appearance changes dominated by camera motion. This approach uses view synthesis as a supervision signal by minimizing the photometric error between the original and reconstructed images from video sequences, eliminating the need for ground truth depth or pose information. Instead, it employs two neural networks: one for estimating depth (which is used at test time) and another for camera pose prediction.

More specifically, given a pixel with homogeneous coordinates p_t in the target view, camera intrinsics K, predicted depth $\hat{D}_t$, and estimated relative pose $\hat{T}_{t\to s}$, the projected coordinates p_s in the source view are computed as:

$$p_s \sim K\hat{T}_{t\to s}\hat{D}_t(p_t)K^{-1}p_t \tag{1}$$

Differentiable bilinear sampling [50] is then used to reconstruct the target view. Various reconstruction losses, such as absolute difference or structural similarity (SSIM), can be employed to minimize the photometric distances between the warped image and the target image.

IV. Proposed Framework

Beyond the multi-view consistency principle underlying self-supervised monocular depth estimation, we argue another

geometric constraint exists: object depth should align with its ground contact point. As this prior holds for any independent objects in the scene, stationary or moving, we can exploit it to properly deal with the dynamic objects violating the multi-view consistency principle. Purposely, we develop a strategy comprising three key steps: i) depth-aware features selection, ii) iterative ground propagation, and iii) clipping normalization.

1) Depth-aware Features Selection.: In the latent space, certain feature maps are responsible for extracting high-frequency information, while others are more sensitive to low-frequency information – i.e., task-specific information. For monocular depth estimation, we identify the latter as depth-aware feature maps. We argue that, by acting on these specific cues, we can propagate information from the ground contact points to dynamic objects.

To identify these feature maps among the different channels extracted by a particular layer after inference, we first generate a pseudo disparity/depth map. As shown in Fig. 2, this map reflects the depth/disparity distribution corresponding to the ground plane. We compute the cosine similarity between them and the feature maps, identifying as depth-aware feature maps those with the highest scores – as shown in Fig. 2 on the feature maps extracted by the 5th layer in the decoder in MonoDepth2 [1]. Acting on these features enables us to propagate cues from ground contact points to dynamic objects.

2) Iterative Ground Propagation.: To adjust the predictions for moving objects, we iteratively propagate ground features to dynamic targets within depth-aware feature maps, making them consistent with the ground contact points. This process is formalized by Eq. 2:

$$f^r_{i,j} = M_{i,j} * f^{r-1}_{i,j+1} + (1 - M_{i,j}) * f^0_{i,j}, r = 1, 2, 3 \cdots, n \quad (2)$$

where $f^r_{i,j}$ denotes feature values located at pixel (i, j) during the r-th iteration, and M represents an objects mask obtained from an off-the-shelf semantic segmentation network. Accordingly, at any iteration, the features of the moving objects will be replaced by those below them.

Fig. 3 visually demonstrates the effect of ground propagation. As the number of iterations increases, the ground features gradually propagate to the entire object, enforcing consistency with ground contact points, and thereby revising the original incorrect disparity prediction. Since this iterative ground propagation occurs solely on depth-aware feature maps, for moving objects whose surfaces are not perfectly perpendicular to the ground (e.g, vehicles approaching the camera) or whose ground contact points are occluded, they can effectively preserve depth details without propagating to other objects.

3) Clipping Normalization.: We argue that naïvely overwriting object features with ground features might be too radical – e.g., when applied to segmented objects that are not moving – and it is strictly necessary only in the presence of a large difference between the two – i.e., when the objects are moving. Purposely, we retain information from the original

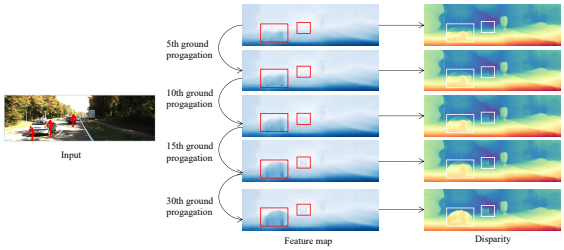

Fig. 3: **Ground Propagation in action**. We iteratively perform ground propagation on the feature map of the 5th layer of the depth decoder for 30 steps. The corresponding outputs of reverse depth are shown in the rightmost image column.

features f^0 by replacing the updated features f^n with a weighted sum between the two:

$$f^n = \min(w, 1) * f^n + (1 - \min(w, 1)) * f^0 \quad (3)$$

with w being computed according to clipping normalization, by dividing the absolute difference between the two by the C percent of its maximum value, as formulated as Eq. 4.

$$w_{i,j} = \frac{\left| f^n_{i,j} - f^0_{i,j} \right|}{\max(|f^n - f^0|) * C}, 0 < C \leq 1 \quad (4)$$

Accordingly, we preserve the reliable features learned according to the multi-view consistency principle and correct only the outlier values according to ground depth consistency.

V. Experiments

In this section, we collect the outcome of our experiments to support the effectiveness of our proposed ground propagation strategy with respect to existing solutions.

A. Implementation Details

We apply our strategy to two self-supervised monocular depth estimation frameworks: Monodepth2 [1] and Lite-Mono [33]. We implement the two variants of our framework in Pytorch, starting from the existing codebases of both models and training them following the original training schedules. Specifically, MonoDepth2 and Lite-Mono variants are trained respectively for 20 and 35 epochs on the KITTI dataset with batch size set to 12. For the remaining training hyper-parameters, losses, and optimizer, we adhered to the original settings detailed in the respective papers [1], [33]. In our experiments, we use a single RTX 3090 GPU and process images at 640×192 resolution. Overall, the network training requires about 15 hours and we adopt the same data augmentation detailed in [1]. Regarding ground propagation, the Monodepth2 variant applies it on the 2nd, 3rd, 4th, and 5th decoder layers for 4, 8, 16, and 32 iterations respectively. The Lite-Mono variant implements ground propagation on the 1st, 2nd, and 3rd decoder layers, using 8, 16, and 32 iterations. Given any layer, we run ground propagation on the $\frac{1}{8}$ and $\frac{1}{16}$ feature maps having the highest cosine similarity with respect to the predicted depth map, for the Monodepth2

Method	M.N	Data	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Geo-Net [43]	✓	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Struct2Depth [8]	✓	K	0.141	1.026	5.290	0.215	0.816	0.945	0.979
SC-DepthV3 [51]		K	0.118	0.756	4.709	0.188	0.864	0.960	0.984
Dyna-DM [44]	✓	C+K	0.115	0.785	4.698	0.192	0.871	0.959	0.982
SGDepth [52]	✓	C+K	0.113	0.835	4.693	0.191	0.879	0.961	0.981
Insta-DM [53]	✓	K	0.112	0.777	4.772	0.191	0.872	0.959	0.982
Monodepth2 [1]		K	0.115	0.917	4.880	0.193	0.877	0.959	0.981
Dynamo-Depth (Monodepth2) [10]	✓	K	0.120	0.864	4.850	0.195	0.858	0.956	0.982
FGTO (Monodepth2) [13]		K	0.112	0.866	4.766	0.190	0.879	0.960	0.982
Ours (Monodepth2)		K	0.111	0.797	4.682	0.188	0.880	0.961	0.982
Lite-Mono [33]		K	0.107	0.765	4.561	0.183	0.886	0.963	0.983
Dynamo-Depth (Lite-Mono)	✓	K	0.112	**0.758**	**4.505**	0.183	0.873	0.959	**0.984**
Ours (Lite-Mono)		K	**0.106**	0.761	4.529	**0.181**	**0.888**	**0.964**	0.983

TABLE I: **Results on KITTI Eigen split [54] – raw LiDAR as ground truth.** Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Insta-DM	✓	0.091	0.506	3.997	0.141	0.907	0.981	0.995
SC-DepthV3		0.099	0.523	4.094	0.144	0.897	0.979	0.995
Dyna-DM	✓	0.092	0.494	3.898	0.140	0.907	0.980	0.995
SGDepth	✓	0.085	0.491	3.755	0.130	0.921	0.984	**0.996**
Monodepth2		0.090	0.545	3.942	0.137	0.914	0.983	0.995
Dynamo-Depth (Monodepth2)	✓	0.096	0.552	4.075	0.145	0.901	0.979	0.995
Ours (Monodepth2)		0.089	0.494	3.843	0.136	0.914	0.983	0.995
Lite-Mono		0.083	**0.455**	3.689	0.128	0.923	0.985	**0.996**
Dynamo-Depth (Lite-Mono)	✓	0.088	0.463	3.692	0.131	0.917	0.984	**0.996**
Ours (Lite-Mono)		**0.081**	0.458	**3.603**	**0.124**	**0.928**	**0.986**	**0.996**

TABLE II: **Results on KITTI Eigen split [54] – improved ground truth [55].** Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

and Lite-Mono variants respectively. We set the clipping rate to 0.3 for both. This process occurs during both training and testing phases.

For evaluation, we compute the seven standard metrics (Abs Rel, Sq Rel, RMSE, RMSE log, $\delta_1 < 1.25$, $\delta_2 < 1.25^2$, $\delta_3 < 1.25^3$) proposed in [54] and used by most works in the literature. In each table, we highlight with **bold** or underline the best and second-best results respectively.

B. Datasets

We conduct experiments on two popular driving datasets.

KITTI [57]. The KITTI stereo dataset contains 61 scenes, with a typical image size of 1242×375, captured using a stereo rig mounted on a moving car equipped with a LiDAR sensor. Following previous works in this field [1], [33], we use the image split of Eigen *et al.* [54], which consists of 39810 monocular triplets for training and 4424 for validation. To compare with the existing solutions, we evaluate the depth performance on the test split of [54] either using raw LiDAR (697 images) or improved ground truth labels [55] (652 images).

DrivingStereo [56]. It is a large-scale stereo dataset depicting autonomous driving scenarios. Among several sequences, we use the four image splits made available on the website, each made of 500 frames collected under different weather conditions, respectively *foggy*, *cloudy*, *rainy* and *sunny*. We use this dataset to evaluate the generalization capacity of existing solutions and ours.

C. Depth Evaluation

We start by evaluating the overall accuracy of depth maps predicted by our model and existing ones. For all evaluations, we apply median scaling [7] relative to the ground truth to recover the metric scale, which is typically lost in self-supervised training on monocular videos [7].

Results on KITTI. We evaluate our models on the established KITTI Eigen split [54], comprising 697 images paired with raw LiDAR scans. Although these scans produce several outliers when projected on the image plane, we use them to allow a fair comparison with existing works, before moving to more accurate experiments with the improved ground truth [55]. Table I collects the outcome of this evaluation, involving several existing frameworks for self-supervised monocular depth estimation, including those specifically designed to handle dynamic objects, such as Dynamo-Depth [10] and From-Ground-To-Objects (FGTO). These latter are grouped at the bottom of the table, depending on the backbone they deploy – either MonoDepth2 or Lite-Mono. In each block, our solution consistently outperforms the original model and achieves more accurate results compared to both Dynamo-Depth and FGTO. Notably, Dynamo-Depth fails to improve the overall accuracy of MonoDepth2 and Lite-Mono, despite

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Insta-DM	✓	0.217	3.961	12.156	0.296	0.694	0.885	0.950
SC-DepthV3		0.198	2.589	9.685	0.259	0.708	0.914	0.971
Dyna-DM	✓	0.214	4.068	11.766	0.277	0.720	0.898	0.957
SGDepth	✓	0.166	2.231	9.590	0.237	0.770	0.928	0.972
Monodepth2		0.173	2.582	9.753	0.239	0.771	0.929	0.973
Dynamo-Depth (Monodepth2)	✓	0.181	2.966	10.364	0.245	0.766	0.923	0.971
Ours (Monodepth2)		0.169	2.503	9.645	0.234	0.777	0.932	0.975
Lite-Mono		0.160	2.318	9.338	0.225	0.794	0.937	0.976
Dynamo-Depth (Lite-Mono)	✓	0.179	3.169	10.562	0.236	0.778	0.926	0.973
Ours (Lite-Mono)		**0.156**	**2.165**	**9.043**	**0.221**	**0.801**	**0.941**	**0.978**

TABLE III: **Results on DrivingStereo [56] dataset.** Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Insta-DM	✓	0.130	1.088	5.242	0.179	0.818	0.949	0.987
SC-DepthV3		0.157	1.381	5.553	0.186	0.791	0.949	0.989
Dyna-DM	✓	0.123	0.935	4.797	0.162	0.845	**0.966**	**0.992**
SGDepth	✓	0.146	1.642	5.743	0.185	0.814	0.950	0.983
Monodepth2		0.147	1.731	6.003	0.188	0.815	0.951	0.981
Dynamo-Depth (Monodepth2)	✓	0.158	1.583	5.916	0.199	0.783	0.941	0.984
Ours (Monodepth2)		0.125	1.082	5.276	0.177	0.834	0.955	0.988
Lite-Mono		0.133	1.207	5.263	0.175	0.828	0.956	0.987
Dynamo-Depth (Lite-Mono)	✓	0.147	1.280	5.283	0.184	0.808	0.947	0.985
Ours (Lite-Mono)		**0.117**	**0.912**	**4.734**	**0.161**	**0.862**	0.964	0.989

TABLE IV: **Dynamic Objects Evaluation: results on KITTI Eigen split [54] – improved ground truth [55].** Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

Method	M.N	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Insta-DM	✓	0.245	5.578	13.097	0.286	0.620	0.861	0.944
SC-DepthV3		0.268	5.045	12.655	0.279	0.575	0.857	0.958
Dyna-DM	✓	0.244	5.769	12.696	0.271	0.655	0.884	0.951
SGDepth	✓	0.210	4.102	12.703	0.278	0.627	0.850	0.943
Monodepth2		0.208	4.143	12.266	0.262	0.664	0.881	0.957
Dynamo-Depth (Monodepth2)	✓	0.245	5.341	12.771	0.280	0.628	0.864	0.950
Ours (Monodepth2)		0.185	3.376	11.854	0.252	0.676	0.878	0.959
Lite-Mono		0.197	3.483	11.366	0.244	0.680	0.897	0.968
Dynamo-Depth (Lite-Mono)	✓	0.214	3.959	11.261	0.241	0.683	**0.911**	**0.972**
Ours (Lite-Mono)		**0.173**	**2.713**	**10.578**	**0.227**	**0.710**	**0.911**	**0.974**

TABLE V: **Dynamic Objects Evaluation: results on DrivingStereo [56] dataset.** Any network processes 192×640 images (except Dyna-DM, SC-Depthv3, and Insta-DM, processing 256×892 images). For each method, we report the use of additional motion networks to deal with dynamic objects (M.N).

enhancing results for dynamic objects, as we will appreciate in the remainder. FGTO, conversely, succeeds in this regard but introduces a two-stage training protocol. Eventually, our strategy further improves the results while maintaining a single-stage training paradigm.

Results on KITTI – Improved Ground Truth. We repeat the same evaluation using the improved ground truth labels provided by [55], which reduces the number of testing images to 652. Table II summarizes the results from this evaluation. In general, we can observe lower errors compared to the previous experiments, thanks to the absence of outliers in these improved ground truth labels. In particular, we highlight once again the comparison between the two baseline models, MonoDepth2 and Lite-Mono, the Dynamo-Depth variants and ours[1]. We can observe a trend similar to the one observed in the raw LiDAR evaluation, with Dynamo-Depth being not capable of improving over the baseline models, whereas our models consistently do.

Results on DrivingStereo. Finally, to assess the generalization capability of the models, we evaluate under four different weather conditions, including *foggy*, *cloudy*, *rainy*

[1]Unfortunately, we cannot compare with FGTO [13] as the authors did not use improved ground truth and the code is not publicly available at the time of this submission.

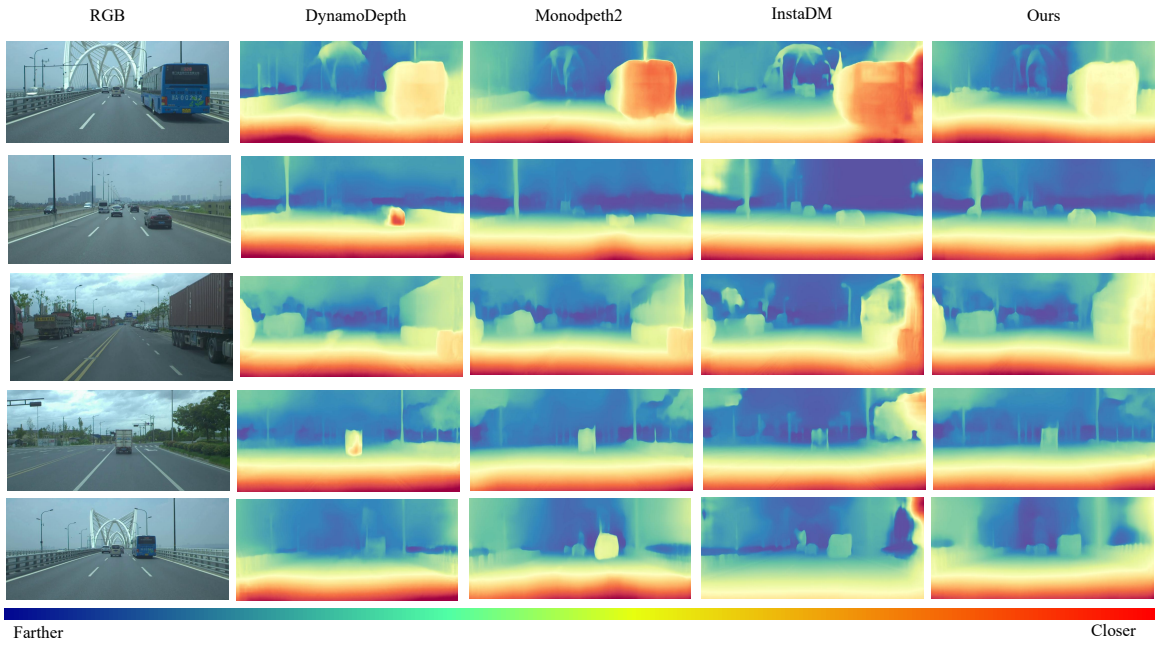

Fig. 4: **Qualitative Results – DrivingStereo dataset [56].** While existing approaches often fail at properly perceiving moving objects, ours predicts consistent depth maps also in the presence of these latter.

Method	Selected Features	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Ours (Monodepth2)	All	0.115	0.810	4.766	0.196	0.874	0.960	0.980
	$\frac{1}{2}$	0.116	0.829	4.820	0.195	0.872	0.959	0.980
	$\frac{1}{4}$	**0.112**	**0.787**	**4.699**	**0.189**	0.878	**0.961**	**0.982**
	$\frac{1}{8}$	**0.112**	0.826	4.779	0.190	**0.879**	**0.961**	**0.982**
	$\frac{1}{16}$							

(a)

Method	Clipping rate	Abs Rel↓	Sq Rel↓	RMSE↓	RMSElog↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Ours (Monodepth2)	0.8	0.113	0.808	4.752	0.191	0.876	0.960	0.981
	0.5	0.112	0.819	4.720	0.190	**0.880**	0.960	0.981
	0.3	**0.111**	**0.797**	**4.682**	**0.188**	**0.880**	**0.961**	**0.982**
	0.1	0.113	0.824	4.744	0.190	0.879	0.960	0.982

(b)

TABLE VI: **Ablation Studies on KITTI Eigen split [54] – raw LiDAR as ground truth.** We evaluate the contributions by (a) Ground Propagation and (b) Clipping Normalization with Monodepth2 as the baseline.

and *sunny*, from the DrivingStereo [56] dataset. Table III collects the outcome of this evaluation, conducted by applying KITTI-trained models to DrivingStereo without fine-tuning. This experiment reveals a more significant performance gap between existing methods – such as Insta-DM, SC-DepthV3, Dyna-DM, SGDepth, and Dynamo-Depth – and our solution. Notably, our Lite-Mono variant achieves a substantial improvement over both these methods and the original Lite-Mono backbone.

D. Depth Evaluation – Dynamic Objects

We now focus on dynamic objects, by measuring the accuracy of existing solutions and ours at estimating their depth. Purposely, we use a pre-trained semantic segmentation network [58] to segment cars in the testing images and compute the error metrics only over them during evaluation.

Results on KITTI – Improved Ground Rruth. We start this additional evaluation on the KITTI dataset, using the improved ground truth [55]. Table IV presents the results obtained by evaluating only pixels corresponding to cars. We can observe significantly higher error metrics and lower accuracy compared to those in Table II, confirming that the moving objects pose a major challenge to self-supervised monocular depth estimation frameworks. We can appreciate how many of the existing solutions, such as Insta-DM and Dyna-DM, are indeed more effective than MonoDepth2 and Lite-Mono on dynamic objects. Dynamo-Depth improves over the MonoDepth2 baseline but struggles when applied to the Lite-Mono backbone. In contrast, our strategy is effective when applied to both and achieves the best overall results.

Results on DrivingStereo. We also evaluate the accuracy of estimated depth over dynamic objects on the DrivingStereo dataset. Table V reports the outcome of this experiment, confirming once again that our models achieve the best results over dynamic objects. The superior accuracy achieved by our method in this setting can also be perceived qualitatively, as in Fig. 4. Here, we can appreciate how Monodepth2 fails at predicting the correct depth for moving objects in five

examples from the DrivingStereo dataset. While DynamoDepth and InstaDM occasionally compensate for these errors, they cannot fully resolve the issue. In contrast, our approach consistently produces satisfactory depth predictions.

E. Ablation Study

We conclude our experiments with ablation studies. Table VI reports, from top to bottom, analyses of (a) the number of features selected for ground propagation and (b) the clipping rate. By focusing on the former aspect (a), we can observe that the most favorable outcomes are achieved when selecting the top $\frac{1}{8}$ of the features map, whereas increasing or decreasing this selection yields drops in accuracy.

Concerning the latter (b), we apply different clipping rate values to determine if retaining part of the original features can further improve the results. We found that setting the clip rate to 0.3 consistently enhances the performance, with other values showing no significant improvements.

VI. Conclusion

In this paper, we presented a novel strategy for dealing with dynamic objects in self-supervised monocular depth estimation. By propagating the features from their ground contact points directly up to the moving objects, we can re-calibrate the depth-aware features, guiding the decoder to predict consistent depths across all image regions, whether static or dynamic. Experiments on KITTI and DrivingStereo datasets supported the effectiveness of our proposal and its superior accuracy compared to existing solutions, achieved without adding any new network parameters and while maintaining an end-to-end training procedure.

Acknowledgment. We sincerely thank the scholarship supported by China Scholarship Council (CSC).

References

[1] C. Godard, O. M. Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3828–3838.

[2] C. Zhang, W. Yin, B. Wang, G. Yu, B. Fu, and C. Shen, "Hierarchical normalization for robust monocular depth estimation," vol. 35, 2022.

[3] B. Ke, A. Obukhov, S. Huang, N. Metzger, R. C. Daudt, and K. Schindler, "Repurposing diffusion-based image generators for monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[4] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 371–10 381.

[5] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, 2022.

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 270–279.

[7] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised Learning of Depth and Ego-Motion from Video," *Conference on Computer Vision and Pattern Recognition*, pp. 6612–6619, 2017.

[8] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[9] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155. PMLR, 16–18 Nov 2021, pp. 1908–1917. [Online]. Available: https://proceedings.mlr.press/v155/li21a.html

[10] Y. Sun and B. Hariharan, "Dynamo-depth: Fixing unsupervised depth estimation for dynamical scenes," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[11] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. L. Yuille, "Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 2624–2641, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52987932

[12] W. Han, J. Yin, and J. Shen, "Self-supervised monocular depth estimation by direction-aware cumulative convolution network," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 8579–8589.

[13] J. Moon, J. L. G. Bello, B. Kwon, and M. Kim, "From-ground-to-objects: Coarse-to-fine self-supervised monocular depth estimation of dynamic objects with ground contact prior," *arXiv preprint arXiv:2312.10118*, 2023.

[14] X. Yang, Z. Ma, Z. Ji, and Z. Ren, "Gedepth: Ground embedding for monocular depth estimation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[15] A. Saxena, M. Sun, and A. Y. Ng, "Make3d: Depth perception from a single still image," in *Proc. AAAI*, 2008.

[16] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," 2014.

[17] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," *International Conference on 3D Vision*, pp. 239–248, 2016.

[18] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "Neural window fully-connected crfs for monocular depth estimation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3906–3915.

[19] Z. Li, Z. Chen, X. Liu, and J. Jiang, "Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation," *Machine Intelligence Research*, pp. 1–18, 2023.

[20] R. Garg, V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue," in *European Conference on Computer Vision*, 2016, pp. 740–756.

[21] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. M. Reid, "Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction," *Conference on Computer Vision and Pattern Recognition*, pp. 340–349, 2018.

[22] J. Spencer, R. Bowden, and S. Hadfield, "DeFeat-Net: General monocular depth via simultaneous unsupervised representation learning," in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 390–14 401.

[23] P. Zama Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Geometry meets semantics for semi-supervised monocular depth estimation," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer, 2019, pp. 298–313.

[24] V. Guizilini, A. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," *Conference on Computer Vision and Pattern Recognition*, pp. 2482–2491, 2020.

[25] J. Watson, M. Firman, G. Brostow, and D. Turmukhambetov, "Self-supervised monocular depth hints," *International Conference on Computer Vision*, vol. 2019-Octob, pp. 2162–2171, 2019.

[26] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," *Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9791–9801, 2019.

[27] S. M. H. Miangoleh, S. Dille, L. Mai, S. Paris, and Y. Aksoy, "Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9685–9694.

[28] A. Johnston and G. Carneiro, "Self-Supervised Monocular Trained Depth Estimation Using Self-Attention and Discrete Disparity Volume," in *Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4755–4764.

[29] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 12 179–12 188.

[30] C. Zhao, Y. Zhang, M. Poggi, F. Tosi, X. Guo, Z. Zhu, G. Huang, Y. Tang, and S. Mattoccia, "Monovit: Self-supervised monocular depth estimation with a vision transformer," *International Conference on 3D Vision*, 2022.

[31] M. Poggi, F. Aleotti, F. Tosi, and S. Mattoccia, "Towards real-time unsupervised monocular depth estimation on cpu," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 5848–5854.

[32] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6101–6108.

[33] N. Zhang, F. Nex, G. Vosselman, and N. Kerle, "Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 18 537–18 546.

[34] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, "Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[35] A. Eftekhar, A. Sax, J. Malik, and A. Zamir, "Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans," 2021, pp. 10 786–10 796.

[36] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *arXiv preprint arXiv:2406.09414*, 2024.

[37] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen, "Metric3D: Towards zero-shot metric 3d prediction from a single image," 2023.

[38] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambruș, and A. Gaidon, "Towards zero-shot scale-aware monocular depth estimation," 2023.

[39] Y. Ji, Z. Chen, E. Xie, L. Hong, X. Liu, Z. Liu, T. Lu, Z. Li, and P. Luo, "DDP: Diffusion model for dense visual prediction," 2023.

[40] S. Saxena, A. Kar, M. Norouzi, and D. J. Fleet, "Monocular depth estimation using diffusion models," *arXiv preprint arXiv:2302.14816*, 2023.

[41] A. Costanzino, P. Zama Ramirez, M. Poggi, F. Tosi, S. Mattoccia, and L. Di Stefano, "Learning depth estimation for transparent and mirror surfaces," in *The IEEE International Conference on Computer Vision*, 2023, iCCV.

[42] F. Tosi, P. Zama Ramirez, and M. Poggi, "Diffusion models for monocular depth estimation: Overcoming challenging conditions," in *European Conference on Computer Vision (ECCV)*, 2024.

[43] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *CVPR*, 2018.

[44] K. Saunders, G. Vogiatzis, and L. J. Manso, "Dyna-dm: Dynamic object-aware self-supervised monocular depth maps," in *2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2023, pp. 10–16.

[45] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ByxT7TNFvH

[46] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova, "Unsupervised monocular depth learning in dynamic scenes," in *Conference on Robot Learning*. PMLR, 2021, pp. 1908–1917.

[47] S. Lee, F. Rameau, F. Pan, and I. S. Kweon, "Attentive and contrastive learning for joint depth and motion field estimation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4862–4871.

[48] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li, "Disentangling object motion and occlusion for unsupervised multi-frame monocular depth," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXII*. Springer, 2022, pp. 228–244.

[49] T.-W. Hui, "Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1675–1684.

[50] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[51] L. Sun, J.-W. Bian, H. Zhan, W. Yin, I. Reid, and C. Shen, "Sc-depthv3: Robust self-supervised monocular depth estimation for dynamic scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.

[52] M. Klingner, J.-A. Termöhlen, J. Mikolajczyk, and T. Fingscheidt, "Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance," in *European Conference on Computer Vision (ECCV)*, 2020.

[53] S. Lee, S. Im, S. Lin, and I. S. Kweon, "Learning monocular depth in dynamic scenes via instance-aware projection consistency," *CoRR*, vol. abs/2102.02629, 2021. [Online]. Available: https://arxiv.org/abs/2102.02629

[54] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[55] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity Invariant CNNs," *International Conference on 3D Vision*, pp. 11–20, 2017.

[56] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi, and B. Zhou, "Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[57] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[58] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.