2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

MAL: Motion-Aware Loss with Temporal and Distillation Hints for Self-Supervised Depth Estimation

Yue-Jiang Dong[1] Fang-Lue Zhang[2] Song-Hai Zhang[1*]
https://github.com/YuejiangDong/MAL

***Abstract*— Depth perception is crucial for a wide range of robotic applications. Multi-frame self-supervised depth estimation methods have gained research interest due to their ability to leverage large-scale, unlabeled real-world data. However, the self-supervised methods often rely on the assumption of a static scene and their performance tends to degrade in dynamic environments. To address this issue, we present Motion-Aware Loss, which leverages the temporal relation among consecutive input frames and a novel distillation scheme between the teacher and student networks in the multi-frame self-supervised depth estimation methods. Specifically, we associate the spatial locations of moving objects with the temporal order of input frames to eliminate errors induced by object motion. Meanwhile, we enhance the original distillation scheme in multi-frame methods to better exploit the knowledge from a teacher network. MAL is a novel, plug-and-play module designed for seamless integration into multi-frame self-supervised monocular depth estimation methods. Adding MAL into previous state-of-the-art methods leads to a reduction in depth estimation errors by up to 4.2% and 10.8% on KITTI and CityScapes benchmarks, respectively.**

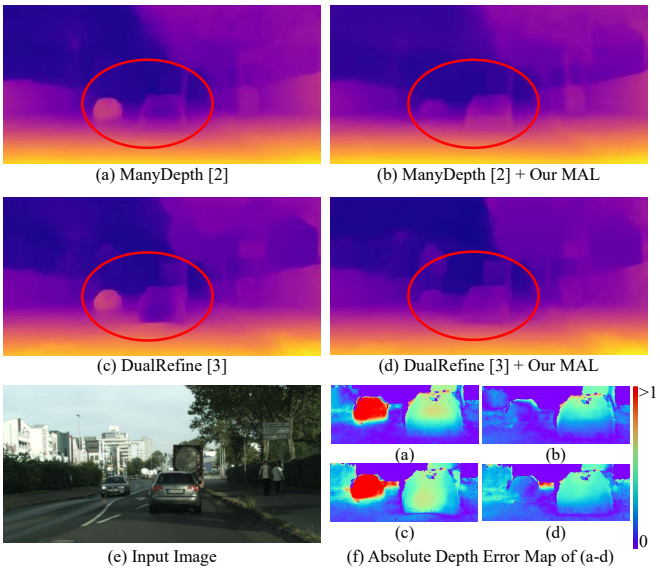

Fig. 1. **Qualitative Demonstration of Our MAL's Effectiveness on CityScapes [12] Dataset**. MAL is designed for multi-frame depth estimation methods (a, c). It's a plug-and-play module (b, d) aimed at improving depth perception (f), especially for moving objects, in dynamic scenes (e).

I. INTRODUCTION

Accurate depth information is crucial for autonomous vehicles and robots to perceive and interact with environments in a manner akin to human cognition. Recent strides in deep learning methodologies have yielded remarkable progress in training networks to autonomously infer depth directly from RGB images. Expanding on this progress, a surge of interest has emerged in leveraging extensive, unlabeled real-world data, driving the pursuit of self-supervised methodologies employing monocular videos as input [1], [2], [3].

Early methods employ self-supervision by making the foundational assumption of a static scene and framing the depth estimation task as a cross-view consistency problem [1], where the difference between the current frame and the reprojected frame from its neighbor serves as an image reprojection loss function. Recent state-of-the-art techniques employ multiple frames as input [3], [4], [5], incorporating a reprojection and matching process at feature level across adjacent frames for a better scene geometry understanding.

Despite the advancements mentioned above, challenges still exist in dynamic scenes due to the violation of the static scene assumption. Moving objects introduce errors in feature matching and image reprojection loss computation. Some methods [2], [3], [5], [4], [6] use teacher-student distillation with a single-frame depth network as a teacher to alleviate errors in feature matching, but errors in loss remain. Other approaches employ optical flow [7] or 3D motion fields [8], [9], [10], [6] to model object motion, or rely on semantic segmentation to separate foreground and background objects [4], [11]. However, these techniques often introduce complex algorithms into the network's forward pass, posing integration challenges with existing self-supervised depth estimation approaches. Our research aims to tackle these enduring challenges, enhancing the effectiveness of self-supervised depth estimation in the presence of dynamic elements while minimizing additional integration and inference costs.

In this paper, we propose Motion-Aware Loss (MAL), a plug-and-play module designed for multi-frame self-supervised depth estimation from monocular videos. The primary aim is to enhance depth estimation in dynamic scenes through a novel approach to loss computation. We leverage temporal coherence in adjacent frames of monocular videos to address errors from moving objects in the image reprojection loss and enhance distillation in the teacher-student network to mitigate errors in the feature matching process. In a group of three consecutive frames, the antecedent and subsequent frames exhibit a symmetrical correspondence with

*corresponding author
[1]Yue-Jiang Dong and Song-Hai Zhang are with the Department of Computer Science and Technology, Tsinghua University, Beijing, China {dongyj21@mails., shz@}tsinghua.edu.cn
[2]Fang-Lue Zhang is with School of Engineering and Computer Science, Victoria University of Wellington, New Zealand fanglue.zhang@vuw.ac.nz

regard to the central frame. Assuming uniform linear motion due to the short time interval between adjacent frames, we perform positional adjustments on dynamic elements and reconstruct occluded regions utilizing the symmetrical frame. This correction helps eliminate errors introduced by object motion in loss. Meanwhile, previous methods [2], [3], [4], [5], [6] confine distillation operations to regions where the difference between the output of teacher network and the depth of the lowest feature matching cost in student network exceeds a prescribed threshold, and straightforwardly adopt the teacher's output as the distillation target. To further reduce errors caused by object motion in feature matching, we propose an extension of distillation across the entire image domain, advocating the utilization of the loss value as a criterion to select the more accurate depths between the outputs of the two networks as the distillation target.

This approach brings two key benefits. Firstly, this module is exclusively confined to the training phase, guaranteeing real-time inference efficiency. Secondly, as the modifications are restricted to the loss calculation stage, this module can be effortlessly and swiftly integrated into existing methods without the need for changes in the base model.

Our main contributions are:

- We propose Motion-Aware Loss (MAL), a plug-and-play module to enhance multi-frame self-supervised depth estimation methods. It operates at the loss computation level, ensuring improved results without incurring additional computational overhead during inference.
- In MAL, we propose to leverage the temporal motion information inherent in neighboring frames and employ a new distillation scheme that spans the entire depth map. This strategic combination leads to notable enhancements in depth estimations, particularly in dynamic scenes.
- We integrated our MAL module into multiple multi-frame self-supervised depth estimation methods. Notably, we observed up to a remarkable 4.2% improvement on KITTI and an impressive 10.8% enhancement on CityScapes benchmarks, underscoring its efficacy.

II. Related Work

A. Self-Supervised Depth Estimation

Self-supervised depth estimation initially emerged as a technique for stereo pairs, where the estimated depth is constrained by a novel view synthesis process [13]. In this context, two images of the same scene are captured from different positions, and one image can be synthesized with the other using the estimated depth based on Structure from Motion. This framework was later adapted to monocular settings, where monocular video sequences serve as input [1]. A pose network is concurrently trained with the depth prediction network to model camera ego-motion. Previous advancements in this field include handling object occlusions [14], ensuring scale consistency across frames [15], [16], [17], [18], [19], and improving network architectures [20].

B. Self-Supervised Depth Estimation in Dynamic Scenes

Monocular videos usually contain dynamic objects, which violate the static-scene assumption inherent to self-supervised depth estimation methodologies. To address this issue, some methods explicitly model pixel-wise motion using optical flow [7] or 3D motion fields [8], [9], [10]. Others leverage semantic cues [9], [21], [22]. They distinguish moving objects from the background and model object-level motion. Similarly, our MAL module also leverages instance segmentation information to address dynamic scenes.

C. Multi-Frame Self-Supervised Depth Estimation

Depth estimation from a single image is inherently challenging due to its ill-posed nature [23]. Consequently, recent research in self-supervised depth estimation has focused on multi-frame methods, which utilize multiple images during inference. ManyDepth [2] introduces a feature matching scheme based on cost volume construction to leverage geometric information between frames. Building upon this approach, recent advancements integrate attention mechanisms into cost volume construction [5] and employ deep equilibrium models to improve the depth and pose estimates [3]. DynamicDepth [4] leverages instance segmentation results to handle object motion by adjusting the positions of moving objects in input frames. However, DynamicDepth requires both the teacher network's estimated depth and the input frame modification during inference. Dyna-DepthFormer [6] utilizes self- and cross-attention modules to aggregate multi-frame and design a 3D motion field jointly trained with the depth network to handle moving objects. In contrast, our MAL module is solely involved in loss computation, ensuring that the real-time inference performance remains unaffected.

III. Method

A. Framework of Self-Supervised Depth Estimation

Here, we revisit the multi-frame self-supervised depth estimation methodology [2]. The framework (Fig. 2) consists of a teacher depth network, a student depth network, and a shared pose network. The teacher network generates a depth map from a single frame, while the student network uses two consecutive frames to predict the latter frame's depth map. Both networks share the same architecture, with a key distinction: the student constructs a cost volume to match features between adjacent frames in the encoder. This feature matching process is absent in the teacher network. The shared pose network estimates camera ego-motion as a six-dimensional vector, encompassing three dimensions for rotation angles and three for translation, and is used by both the teacher and student networks.

An image reprojection loss is used to train the framework (Fig. 2 (e)). Denoting three consecutive frames as I_{t-1}, I_t, and I_{t+1}, we can project each pixel p_t in I_t onto $I_{t\pm1}$ based on structure-from-motion under a static-scene assumption:

$$p_{t\pm1} \sim K T_{t\to t\pm1} D_t(p_t) K^{-1} p_t \tag{1}$$

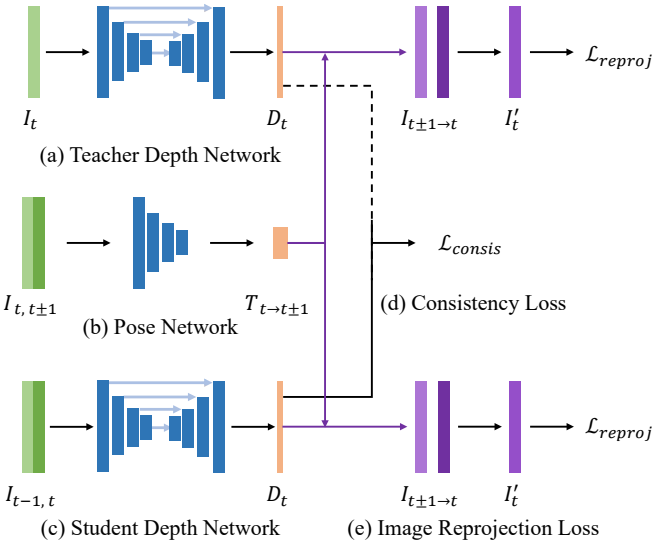

Fig. 2. **Framework of Multi-Frame Self-Supervised Depth Estimation**. The three sub-networks (a-c) are trained concurrently with both image reprojection loss (e) and consistency loss (d). The dotted line indicates that the gradients of the teacher network are not updated by the consistency loss.

where $p_{t\pm1}$ are pixels in $I_{t\pm1}$ and K is camera intrinsic matrix. $D_t(p_t)$ and $T_{t\to t\pm1}$ are depth and camera pose predicted by the network. Pixels in $I_{t\pm1}$ are sampled according to Eqn. (1) to reconstruct image at I_t's viewpoint:

$$I_{t\pm1\to t}[p_t] = I_{t\pm1} < p_{t\pm1} > \tag{2}$$

where $<>$ represents bilinear sampling.

The two reconstructed images, $I_{t\pm1\to t}$, are combined pixel-wisely to handle occlusions. The final reconstructed image is created by choosing the pixel from either $I_{t-1\to t}$ or $I_{t+1\to t}$ with the lower photometric error compared to I_t [14]:

$$I'_t = \mathcal{P}(I_{t-1\to t}, I_{t+1\to t}) \tag{3}$$

where $\mathcal{P}$ represents the pixel selection and I'_t is the final reconstructed image. The photometric difference between I'_t and I_t serves as the image reprojection loss.

The feature matching process in the student network involves projecting features of I_{t-1} to I_t's viewpoint with pixel correspondences in Eqn. (1). This projection employs a predefined set of uniformly distributed depth planes and seeks for depth to match the projected features with those of I_t. However, dynamic regions induce errors in this matching process, resulting in suboptimal depth estimates when training the student network solely with the image reprojection loss [2]. To address this limitation, an asymmetric distillation scheme is employed, transferring knowledge from the teacher network, which does not involve the feature matching process, to the student. An uncertainty mask, denoted as $\mathcal{M}$, is computed through pixel-wise comparisons between the predicted depths of the teacher (D_t) and the depth with the lowest matching cost (D_{cv}) [2]:

$$\mathcal{M} = \max(\frac{D_{cv} - D_t}{D_t}, \frac{D_t - D_{cv}}{D_{cv}}) > 1 \tag{4}$$

During training, the reliable area ($\neg\mathcal{M}$) is supervised by the image reprojection loss, while the unreliable area ($\mathcal{M}$) is instead supervised by a consistency loss, calculated as the L1 difference between depths predicted by the teacher and student. Besides, we use an edge-aware smoothness loss $\mathcal{L}_s$ with a weight of $\lambda_s = 1e-3$ as per standard practice [14]:

$$\mathcal{L}_s = |\partial_x d_t^*|e^{-|\theta_x I_t|} + |\partial_y d_t^*|e^{-|\theta_y I_t|} \tag{5}$$

where $d_t^* = d_t/\overline{d_t}$ is the mean-normalized inverse depth. The original loss of the student network can be formulated as:

$$\mathcal{L}_{ori} = \neg\mathcal{M} \cdot \mathcal{L}_{reproj} + \mathcal{M} \cdot \mathcal{L}_{consis} + \lambda_s \cdot \mathcal{L}_s \tag{6}$$

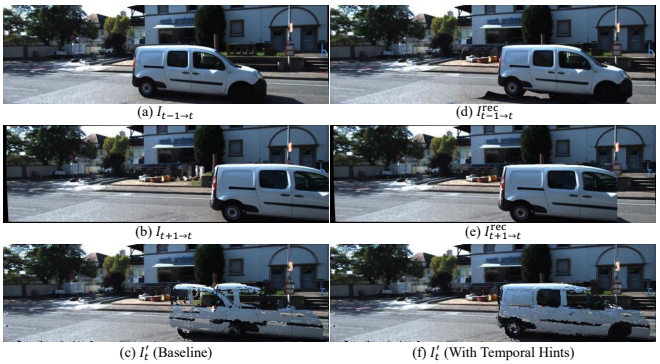

Fig. 3. **Temporal Hints**. Linking object positions to input frames' temporal order via a linear motion model, we align object positions (d-e) and significantly reduce motion-induced errors in the reconstructed image (f).

B. Temporal Hints

The reconstructed images $I_{t\pm1\to t}$ represent images captured at times $t \pm 1$ from the same viewpoint as I_t, as depicted in Fig. 3 (a-b). In regions containing moving objects in $I_{t\pm1\to t}$, photometric errors arise not from the estimated depth in $I_{t\pm1\to t}$, but rather from inherent geometric changes. These errors subsequently affect the quality of the final reconstructed image, as demonstrated in Fig. 3 (c). To tackle this challenge, we leverage temporal coherence in adjacent frames to adjust the positions of moving objects in $I_{t\pm1\to t}$.

We initiate the process by employing a pre-trained instance segmentation model [24] to identify moving objects, such as vehicles and pedestrians, within the scene. The parameters of this segmentation model don't update in training. To establish correspondences between instances across consecutive frames, we utilize the Hungarian algorithm. This algorithm leverages instance class labels and Intersection over Union (IoU) metrics between instance masks as the cost function.

When shooting monocular video, frames are captured with short time intervals. For instance, in the KITTI dataset [25], data is recorded at 10Hz, resulting in a time gap of 0.1 seconds between frames. Consequently, we approximate the position of dynamic objects at time t as the average of their positions in $I_{t\pm1\to t}$. Considering the possibility of dynamic objects moving out of the camera's field of view, which may result in truncated objects near image borders in $I_{t\pm1\to t}$, we implement a bounding-box-level object displacement calculation method to mitigate this issue.

We specifically concentrate on instances that are consistently present in both $t\pm1$ frames. This means that even if an object is partially truncated, either its left or right boundary

should remain intact from $t-1$ to $t+1$, as depicted in Fig. 3 (a-b). Hence, we approximate the horizontal displacement of the object as the maximum value between the displacement of its left boundary and right boundary from $t-1$ to $t+1$:

$$\Delta h^i_{t-1\to t+1} = \max(|l^i_{t+1} - l^i_{t-1}|, |r^i_{t+1} - r^i_{t-1}|) \quad (7)$$

Here l^i_t and r^i_t denote the left and right boundaries of instance i at time t. The vertical displacement is calculated in a similar manner using the top and bottom boundaries of the instance.

We align object positions in $I_{t\pm1\to t}$ with those at time t using the calculated displacement. This translation may uncover previously occluded areas. We leverage the symmetry between $t\pm1$ for restoration. Specifically, an area obscured by a moving object at time $t+1$ but exposed at time t cannot be covered by the same object at time $t-1$. Therefore, we use pixels from $I_{t-1\to t}$ to fill in $I^{\text{rec}}_{t+1\to t}$, and vice versa.

Due to the potential errors in the displacement calculations above, we introduce the motion-rectified image $I^{\text{rec}}_{t\pm1\to t}$ as an additional input in the final image reconstruction process:

$$I'_t = \mathcal{P}(I_{t-1\to t}, I_{t+1\to t}, I^{\text{rec}}_{t-1\to t}, I^{\text{rec}}_{t+1\to t}) \quad (8)$$

Similar to Eqn. (3), here, $\mathcal{P}$ represents the pixel selection operation and I'_t denotes the resulting reconstructed image. This approach effectively mitigates errors caused by object motion in both the reconstructed image and subsequent reprojection loss calculations.

C. Distillation Hints

Besides the image reprojection loss computation, object motion also introduces errors in the feature matching progress inherent to the student network design. These errors cannot be easily mitigated by temporal hints alone because the feature matching occurs at the encoder, and the errors can propagate to subsequent parts of the student network.

To rectify these errors, we expand the distillation process within the region $\mathcal{M}$ mentioned in Eqn. (4) to cover the entire image, thereby maximizing the utilization of knowledge from the teacher network. We fuse the depth predictions from both the teacher and student networks on a pixel-wise basis, selecting the depth with a lower image reprojection loss to generate the target distillation depth map D_{td}. The distillation loss is computed as:

$$\mathcal{L}_{distil} = \neg\mathcal{M} \cdot ||D_s - D_{td}||_1 \quad (9)$$

where D_s represents the depth predicted by the student depth network. Similar to the settings in ManyDepth [2], the distillation process is unidirectional, and the teacher network does not update during the backward propagation of $\mathcal{L}_{distil}$.

D. Loss Balancing

The training of the student network is constrained with two loss terms in total: $\mathcal{L}_{ori}$, as computed by Eqn. (6), and $\mathcal{L}_{distil}$, computed by Eqn. (9):

$$\mathcal{L} = w_1 \cdot \mathcal{L}_{ori} + w_2 \cdot \mathcal{L}_{distil} \quad (10)$$

To maintain an effective balance between the loss terms, we apply the multi-loss rebalancing algorithm (MLRA) [26]. Initially, each loss weight is set to 1/2, and these weights are iteratively updated during training based on the descending rate of each respective loss term. The hyperparameter λ dictates whether the algorithm prioritizes rapidly descending loss terms or slower ones.

IV. Experiments

A. Dataset

1) KITTI: KITTI [25] is the standard benchmark for self-supervised depth estimation evaluation. It is an autonomous driving dataset featuring urban scenes. Following the established practice in previous work, we use the data split of Eigen [32] and the data pre-processing to remove static frames established by [1], resulting in 39,810 monocular triplets for training, 4,424 for validation, and 697 for testing.

2) CityScapes: CityScapes [12] is also a popular benchmark including numerous dynamic scenes with multiple moving objects [21]. It is a notable benchmark for algorithms dealing with dynamic objects [4], [8], [10], [22]. We follow the protocol in previous work [2], [4] and evaluate 1,525 images.

B. Experiment Setup

Currently, there are five prominent multi-frame self-supervised depth estimation methods in the literature [2], [5], [4], [3], [6]. These methods all incorporate image reprojection loss and the teacher-student distillation scheme in their architectures, which theoretically aligns them with our MAL module. However, as Dyna-DepthFormer[6] is not open-sourced and the training configuration file for DepthFormer[5] is not yet publicly accessible, we have chosen to evaluate MAL using ManyDepth [2], DynamicDepth [4], and DualRefine [3] as our baseline frameworks. Notably, DualRefine currently stands as the top-performing model on the KITTI benchmark, while DynamicDepth leads among multi-frame methods on the CityScapes benchmark, with the exception of the most recent Dyna-DepthFormer.

Since self-supervised learning predicts relative depth, we adhere to the single-image median scaling and cap depth values at 80 meters during evaluations, as is standard in the field [14]. We assess the depth predictions using established depth evaluation metrics [27], including Absolute Relative Error (AbsRel), Squared Relative Error (SqRel), Root Mean Squared Error (RMSE), Root Mean Squared Log Error (RMSElog), and accuracy within specified thresholds (δ).

For ManyDepth+MAL and DynamicDepth+MAL, we fine-tune the official models provided by the authors from their respective GitHub repositories, employing a batch size of 24 and 12 with a learning rate of 1e-4 and 1e-5 respectively. DualRefine+MAL undergoes fine-tuning on KITTI with a batch size of 16 and a learning rate of 1e-5. In the case of CityScapes, where no pre-trained DualRefine model is available, we train DualRefine+MAL from scratch, utilizing a batch size of 16 and a learning rate of 1e-4 for 10 epochs. We employ the Adam optimizer [33] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ across all experiments. λ of MLRA is set to 3 and linearly decreases to -3.

TABLE I

DEPTH ESTIMATION RESULTS ON KITTI EIGEN SPLIT [27].

Method	Test Frames	Semantic	W×H	*Errors*↓				*Accuracy*↑		
				AbsRel	SqRel	RMSE	$\text{RMSE}_{\log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Struct2Depth[21]	1	●	416×128	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Bian *et al.*[16]	1		416×128	0.137	1.089	5.439	0.217	0.830	0.942	0.975
Gordon *et al.*[28]	1	●	416×128	0.128	0.959	5.230	0.212	0.845	0.947	0.976
MonoDepth2 [14]	1		640×192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
InstaDM [22]	1	●	832×256	0.112	0.777	4.772	0.191	0.872	0.959	0.982
Packnet-SFM [20]	1		640×192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Wang *et al.*[19]	1		640×192	0.109	0.779	4.641	0.186	0.883	0.962	0.982
RM-Depth [10]	1		640×192	0.108	0.710	4.513	0.183	0.884	0.964	0.983
Johnston *et al.* [29]	1		640×192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
Guizilini *et al.*[30]	1	●	640×192	0.102	0.698	4.381	0.178	0.896	0.964	0.984
Wang [31]	2(-1,0)		640×192	0.106	0.799	4.662	0.187	0.889	0.961	0.982
DynamicDepth [4]	2(-1,0)	●	640×192	0.096	0.720	4.458	0.175	0.897	0.964	0.984
Dyna-DepthFormer [6]	2(-1,0)		640×192	0.094	0.734	4.442	0.169	0.893	0.967	0.983
DepthFormer [5]	2(-1,0)		640×192	0.090	0.661	4.149	0.175	0.905	0.967	0.984
ManyDepth [2]	2(-1,0)		640×192	0.098	0.770	4.459	0.176	0.900	0.965	0.983
+MAL	2(-1,0)	●	640×192	**0.094**	**0.732**	**4.425**	**0.174**	**0.906**	**0.966**	**0.983**
DualRefine [3]	2(-1,0)		640×192	0.087	0.698	4.234	0.170	0.914	0.967	0.983
+MAL	2(-1,0)	●	640×192	**0.087**	**0.690**	**4.227**	**0.169**	**0.915**	**0.968**	**0.983**

TABLE II

DEPTH ESTIMATION RESULTS ON CITYSCAPES [12].

Method	Test Frames	Semantic	W×H	*Errors*↓				*Accuracy*↑		
				AbsRel	SqRel	RMSE	$\text{RMSE}_{\log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Struct2Depth [21]	1	●	416×128	0.145	1.737	7.280	0.205	0.813	0.942	0.976
MonoDepth2 [14]	1		416×128	0.129	1.569	6.876	0.187	0.849	0.957	0.983
Gordon *et al.* [28]	1	●	416×128	0.127	1.330	6.960	0.195	0.830	0.947	0.981
Li *et al.* [8]	1		416×128	0.119	1.290	6.980	0.190	0.846	0.952	0.982
InstaDM [22]	1	●	832×256	0.111	1.158	6.437	0.182	0.868	0.961	0.983
RM-Depth [10]	1		416×128	0.100	0.839	5.774	0.154	0.895	0.976	0.993
Dyna-DepthFormer [6]	2(-1,0)		416×128	0.100	0.834	5.843	0.154	0.901	0.975	0.992
ManyDepth [2]	2(-1, 0)		416×128	0.114	1.193	6.223	0.170	0.875	0.967	0.989
+MAL	2(-1, 0)	●	416×128	**0.103**	**1.073**	**5.952**	**0.157**	**0.896**	**0.973**	**0.991**
DynamicDepth [4] (paper)	2(-1, 0)	●	416×128	0.103	1.000	5.867	0.157	0.895	0.974	0.991
Officially Provided Model	2(-1, 0)	●	416×128	0.104	1.011	5.987	0.159	0.890	0.972	0.991
+MAL	2(-1, 0)	●	416×128	**0.101**	**0.957**	**5.865**	**0.156**	**0.895**	**0.974**	**0.991**
DualRefine [3]	2(-1, 0)		416×128	0.111	1.248	6.035	0.164	0.896	0.971	0.989
+MAL	2(-1, 0)	●	416×128	**0.099**	**0.973**	**5.530**	**0.149**	**0.905**	**0.977**	**0.992**

TABLE III

ABLATION STUDY FOR MANYDEPTH ON CITYSCAPES [12].

Method	Loss Terms Combination	*Errors*↓				*Accuracy*↑		
		AbsRel	SqRel	RMSE	$\text{RMSE}_{\log}$	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
ManyDepth [2]	Original	0.114	1.193	6.223	0.170	0.875	0.967	0.989
+Temporal Hints	Original	0.111	1.182	6.127	0.165	0.882	0.970	0.990
+Distillation Hints	Sum Up	0.114	1.300	6.296	0.168	0.882	0.969	0.989
+Distillation Hints	MLRA [26]	0.111	1.179	6.083	0.164	0.883	0.970	0.990
+MAL	Sum Up	0.109	1.141	6.035	0.162	0.887	0.971	0.990
+MAL	MLRA	**0.103**	**1.073**	**5.952**	**0.157**	**0.896**	**0.973**	**0.991**

C. Evaluation Results

We evaluate our method on KITTI and CityScapes benchmarks and the results are shown in Table I and Table II. The *Test Frames* column indicates the number of input frames during inference. A value of 1 corresponds to a single-frame method, while 2 (-1, 0) signifies a multi-frame method that employs the current frame and its previous frame as input.

According to statistics from previous work [4], dynamic category objects (such as vehicles, pedestrians, cyclists) account for only 0.34% of the pixels in the KITTI dataset,

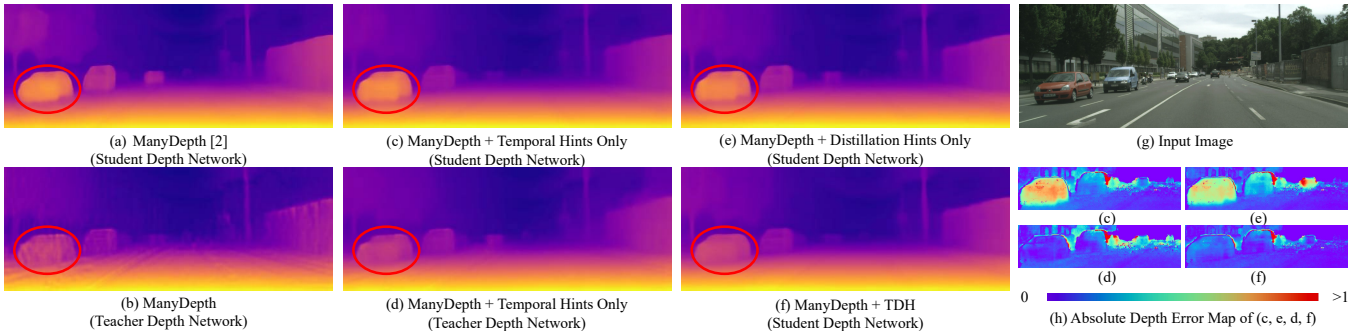

Fig. 4. **Qualitative Analysis of the Indispensability of Both the Temporal and Distillation Hints**. Please refer to Section IV-E for a detailed analysis.

and most of the vehicles are stationary. Hence, previous state-of-the-art methods that specifically target dynamic objects [4], [6], [10] show a relative minor advantage on KITTI compared to CityScapes, where dynamic scenes are more prevalent. As for MAL, ManyDepth shows an improvement of up to 4.2%, while adding MAL to DualRefine leads to higher $\delta < 1.25$ and $\delta < 1.25^2$, indicating a larger proportion of accurate inliers.

Meanwhile, on CityScapes, our MAL consistently enhances all seven depth evaluation metrics of ManyDepth and DualRefine as shown in Table II. Specifically, ManyDepth's depth estimation results can be improved by up to 10.1%, and DualRefine demonstrates an enhancement of up to 10.8%.

Further, we assess the impact of MAL on existing dynamic-scene-oriented methods, which also employ image reprojection loss and the teacher-student framework and is thus applicable for MAL. Since the code of DynaDepthFormer [6] and pre-computed masks of DynamicDepth [4] for KITTI are not publicly available, we experiment with DynamicDepth+MAL on CityScapes. Despite DynamicDepth's pre-optimized architecture for dynamic scenes, compared to the model officially provided by the authors of DynamicDepth, applying MAL yields a noticeable 5.34% decrease in SqRel and an increase from 89.0% to 89.5% in the accuracy metric $\delta < 1.25$, indicating a higher percentage of accurate inliers and a reduced proportion of outliers.

Our MAL offers a substantial performance improvement for existing multi-frame methods, achieving results comparable to state-of-the-art approaches. Importantly, MAL optimizes the algorithm at the loss level, making it easy to integrate into these established methods. Compared to other methods like RM-Depth [10], DynamicDepth [4], and DynaDepthFormer [6], which are designed with specific network forward pass algorithms to address dynamic objects, MAL exhibits greater portability.

D. Ablation Study

We conduct an ablation study on CityScapes to dissect the contributions of each component of our MAL (Table III). Our baseline is ManyDepth [2]. It is worth noting that our MAL enhances depth perception, even in the absence of MLRA (as demonstrated in the fourth row of Table III). MLRA plays a role in automatically generating more sensible weights, thereby providing an additional boost in performance.

E. Qualitative Analysis

Here we elucidate the indispensable roles of each component in our MAL. We disable distillation hints, only enable temporal hints, and employ the original loss function in Eqn. (6). In this case, it is noteworthy that moving objects can induce errors in the cost-volume-based feature matching process in the encoder of the student depth network, which may propagate and degrade its final output despite the temporal hints. Fig. 4 (c, h) manifest an obvious error in the student's depth prediction for the car marked by the red circle. Conversely, the teacher network provides a notably more accurate prediction for this car (Fig. 4 (d, h)), outperforming the baseline (Fig. 4 (b)). This highlights the effectiveness of our temporal hints in enhancing depth perception for dynamic objects in cases where the feature matching process is absent. Moreover, it implies the traditional distillation scheme may not fully exploit the information in the teacher network to improve the student's performance.

Further, we enable both the temporal and the distillation hints. The depth prediction of this car becomes much better (Fig. 4 (f, h)). Meanwhile, we disable the temporal hints and enable the distillation hints only (Fig. 4 (e)). Even in the absence of temporal hints, the depth estimation for the circled car is superior to the case only with temporal hints (Fig. 4 (h)(c) and (h)(e)). This underscore the efficacy of our distillation hints in facilitating a more effective information transfer from the teacher network to the student, compensating for the errors from the feature matching process in the student. However, with only distillation hints, the cars positioned behind the red-circled car show less accurate depth estimation (Fig. 4 (h)(c), indicating that both the temporal and distillation hints are indispensable.

V. Conclusion

In this paper, we present MAL, a plug-and-play module designed to augment depth perception, especially in dynamic scenes, using temporal and distillation hints. MAL can be seamlessly integrated with multi-frame self-supervised depth estimation methods and functions at the loss computation level, ensuring no additional inference time overhead. Our experimental results demonstrate that incorporating MAL into established multi-frame methods yields substantial improvements in depth estimation performance across the KITTI and CityScapes benchmarks.

REFERENCES

[1] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, Jul 2017, p. 6612–6619. [Online]. Available: http://ieeexplore.ieee.org/document/8100183/

[2] J. Watson, O. M. Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The Temporal Opportunist: Self-Supervised Multi-Frame Monocular Depth," in *Computer Vision and Pattern Recognition (CVPR)*, 2021.

[3] A. Bangunharcana, A. Magd, K.-S. Kim, *et al.*, "Dualrefine: Self-supervised depth and pose estimation through iterative epipolar sampling and refinement toward equilibrium," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 726–738.

[4] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li, "Disentangling object motion and occlusion for unsupervised multi-frame monocular depth," *arXiv preprint arXiv:2203.15174*, 2022.

[5] V. Guizilini, R. Ambruș, D. Chen, S. Zakharov, and A. Gaidon, "Multi-frame self-supervised depth with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 160–170.

[6] S. Zhang and C. Zhao, "Dyna-depthformer: Multi-frame transformer for self-supervised depth estimation in dynamic scenes," *arXiv preprint arXiv:2301.05871*, 2023.

[7] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1983–1992.

[8] 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.

[9] 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.

[10] 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.

[11] 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*. Springer, 2020, pp. 582–600.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[13] 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.

[14] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," October 2019.

[15] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, Jun 2018, p. 5667–5675. [Online]. Available: https://ieeexplore.ieee.org/document/8578692/

[16] J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," *Advances in neural information processing systems*, vol. 32, 2019.

[17] Y. Chen, C. Schmid, and C. Sminchisescu, "Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct 2019, p. 7062–7071. [Online]. Available: https://ieeexplore.ieee.org/document/9010956/

[18] L. Tiwari, P. Ji, Q.-H. Tran, B. Zhuang, S. Anand, and M. Chandraker, "Pseudo rgb-d for self-improving monocular slam and depth prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*. Springer, 2020, pp. 437–455.

[19] L. Wang, Y. Wang, L. Wang, Y. Zhan, Y. Wang, and H. Lu, "Can scale-consistent monocular depth be learned in a self-supervised scale-invariant manner?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 727–12 736.

[20] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3d packing for self-supervised monocular depth estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2485–2494.

[21] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Unsupervised monocular depth and ego-motion learning with structure and semantics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[22] S. Lee, S. Im, S. Lin, and I. S. Kweon, "Learning monocular depth in dynamic scenes via instance-aware projection consistency," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 1863–1872.

[23] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[24] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1290–1299.

[25] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[26] J.-H. Lee and C.-S. Kim, "Multi-loss rebalancing algorithm for monocular depth estimation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*. Springer, 2020, pp. 785–801.

[27] 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.

[28] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8977–8986.

[29] A. Johnston, Carneiro, Gustavo, *et al.*, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2020, pp. 4756–4765.

[30] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=ByxT7TNFvH

[31] J. Wang, G. Zhang, Z. Wu, X. Li, and L. Liu, "Self-supervised joint learning framework of depth estimation via implicit cues," *arXiv preprint arXiv:2006.09876*, 2020.

[32] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.