Meta-Auxiliary Learning for Future Depth Prediction in Videos

Huan Liu[12], Zhixiang Chi[1], Yuanhao Yu[1], Yang Wang[13], Jun Chen[2], Jin Tang[1]
[1] Huawei Noah's Ark Lab, [2] McMaster University, [3] Concordia University
{huan.liu3, zhixiang.chi, yuanhao.yu, tangjin}@huawei.com
yang.wang@concordia.ca, chenjun@mcmaster.ca

Abstract

We consider a new problem of future depth prediction in videos. Given a sequence of observed frames in a video, the goal is to predict the depth map of a future frame that has not been observed yet. Depth estimation plays a vital role for scene understanding and decision-making in intelligent systems. Predicting future depth maps can be valuable for autonomous vehicles to anticipate the behaviours of their surrounding objects. Our proposed model for this problem has a two-branch architecture. One branch is for the primary task of future depth prediction. The other branch is for an auxiliary task of image reconstruction. The auxiliary branch can act as a regularization. Inspired by some recent work on test-time adaption, we use the auxiliary task during testing to adapt the model to a specific test video. We also propose a novel meta-auxiliary learning that learns the model specifically for the purpose of effective test-time adaptation. Experimental results demonstrate that our proposed approach outperforms other alternative methods.

1. Introduction

We consider the problem of future depth prediction in videos. Given a sequence of consecutive frames in a video, the goal is to predict the depth map of a future frame that has not been observed yet (Fig. 1). Depth estimation from images or videos has been widely studied in computer vision. Recently, we have witnessed the tremendous success of monocular depth estimation [13,14,27,32,55]. However, current methods mainly focus on estimating depth on data that have been observed. However, in many real-world applications, we actually need to predict depth maps of future frames for decision-making. For example, if an autonomous vehicle can correctly anticipate the future depth of other vehicles in the scene, it can use this information to take proactive actions to avoid possible damage.

There has been a line of work on predicting future information in videos, such as future RGB frame [21,33,43], future semantic segmentation [20,31,40], future trajectory [6], future actions [12], etc. However, future depth prediction has not been studied before. This paper represents the first work on this topic.

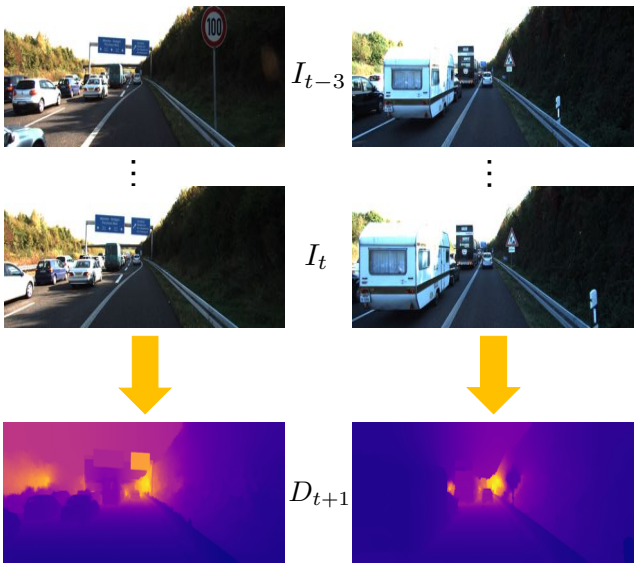

Figure 1: Illustration of the future depth prediction problem. Each column corresponds to a video. Given a few observed frames (e.g. at time steps $t-3$, ... t), the goal is to predict the depth map of a future frame (e.g. at time $t+1$) that has not been observed yet. Future depth prediction can be used by autonomous systems for better planning and decision making.

A naive solution of future depth prediction is to treat it as a purely supervised learning problem. The limitation of this approach is that the learned model tends to overfit to videos used for training and do not generalize well to unseen videos, especially when there is a large domain gap between training and testing videos. In this paper, we propose a meta-auxiliary learning approach [3, 28] for future depth prediction. Our method has the following characteristics. First of all, instead of treating future depth prediction as a purely supervised problem, we add an auxiliary task

that is complementary to the primary task of future depth prediction. These two tasks share the backbone for feature extractions and are learned together. The auxiliary task acts as a regularization that helps learn feature representations that are useful for the primary task. In this paper, we choose image reconstruction as the auxiliary task. Second, since our auxiliary task is self-supervised and does not require any manual labels, we can use it to perform test-time adaptation [45] to adapt the model for a specific test video. Finally, we propose a meta-auxiliary learning approach for training the model to facilitate effective test-time adaptation. Our proposed method significantly outperforms other alternatives on several benchmark datasets.

The contributions of this paper are manifold. First, we introduce a new problem called the future depth prediction in a video. Instead of predicting the depth maps of observed frames, the goal is to predict the depth map of a future frame that has not been observed yet. A reliable solution to this problem can be used in many real-world applications, such as autonomous driving. Second, instead of directly solving the future depth prediction as a purely supervised problem, we propose to use image reconstruction as an auxiliary task in the model. This auxiliary learning acts as a regularization and improves the performance of the primary task. In addition, since the auxiliary task is self-supervised and does not require any manual labels, we can use the auxiliary task to perform test-time adaptation. Finally, we propose a novel meta-auxiliary learning approach to learn the model in a way that enables effective test-time adaptation. Experimental results demonstrate that our proposed approach outperforms other alternative methods.

2. Related Work

2.1. Depth Estimation

There has been extensively worked on estimating depth maps from monocular images. Most current state-of-the-art depth estimation methods use the deep learning framework. Eigen *et al*. [8] propose a two-scale structure for global depth estimation and local depth refinement. Alhashim *et al*. [1] show that better depth estimation results can be achieved with a more powerful design based on DenseNet [19]. Some works also explore the possibility of boosting the mapping ability of neural networks using statistical learning techniques. For example, Fu *et al*. [11] leverages ordinal regression to learn the ordinal relations of the scene. In addition, Ma *et al*. [32] achieves depth estimation and object detection by a unified framework. Video-based depth estimation methods often integrate camera motion and multi-view reconstruction from video sequences [49, 54].

2.2. Future Prediction

There has been a line of research on predicting information of future frames in videos. Early work [21, 33, 43] focuses on predicting the raw RGB values of future frames without explicitly modeling scene dynamic or low-level details. In recent years, to disentangle the variation from video representations, a number of works have focused on carefully designing loss functions and neural network structures. For example, [47, 50] separate motion and content from video by two-stream architecture. Some works [4, 35] predict future frames conditioned on the extra variables, such as odometry or robot state.

Another line of research reformulates the video prediction task as predicting other semantic information instead of raw pixels. Examples include future semantic segmentation [5, 31, 40], future human poses [46, 50], etc. The work in [18] proposes to predict future ego-motion, semantic map, depth map and optical flow jointly in a probabilistic manner.

2.3. Meta Learning

Meta-learning, also known as learning to learn, has been shown to be effective for solving various problems [7, 42, 53], especially the few-shot learning problem [2, 16, 25, 30, 44]. There have been many different meta-learning paradigms [15, 51] in the literature. Optimization-based approaches [23, 38], in particular MAML [9], have been widely used for fast model adaptation. MAML uses nested optimization to learn a good initialization of model parameters for fast adaptation to new tasks.

In addition to few-shot classification, meta-learning has also been successfully applied for dense prediction tasks, such as super-resolution [36, 42], video interpolation [7], image dehazing [26], etc. The goal is to perform internal learning on every test image/video to utilize the unique statistical information to improve the generalization. The test-time adaptation requires supervision from the test data, which can be easily obtained by further downsampling the input image or frame rate in videos for those tasks. However, for future depth prediction, such surrogate training pairs do not exist at test time. So our problem is more challenging.

Our work closely relates to meta-auxiliary learning in [3, 28]. The work in [28] aims to generate optimal auxiliary labels to improve the primary image classification branch. In contrast, our proposed framework achieves test-time adaptation via the auxiliary reconstruction task. Recently, [3] propose to use meta-auxiliary learning for the dynamic scene blurring with test-time adaptation. In this paper, we further explore the possibility of using meta-auxiliary learning in dealing with the rarely researched problem, i.e., future depth prediction. Besides the noticeable difference between [3] and ours in the treated problem,

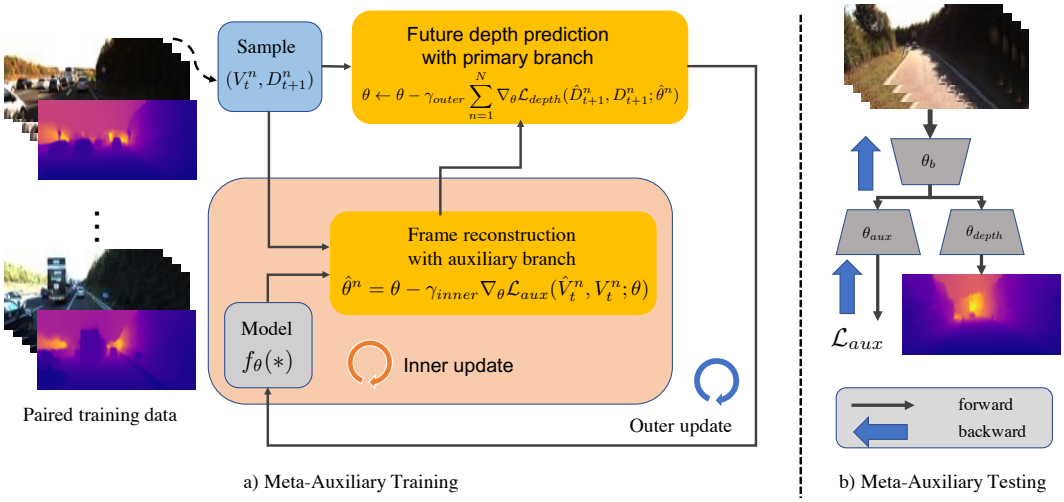

Figure 2: Overview of our proposed meta-auxiliary learning framework for future depth prediction. During the meta-training state (left), we have a collection of paired training data. Each pair consists of a sequence of frames as the input and the depth map of a future frame as the output. The meta-training process involves a nested-loop. In the inner loop, we sample a batch of training data. For each instance in the batch, we update the model parameter via the auxiliary task branch to obtain a model adapted to this instance. We then evaluate the adapted model using the loss for the primary branch. In the outer loop, we update the global model in a way that minimize the primary losses in the batch. After meta-training, we have obtained a model has been learned specifically for effective adaption to a new video. During meta-testing, we are given a new video. We use the auxiliary branch to obtain the adapted model for this test video, then use the adapted model for predictions in the remaining frames of this test video.

our results also provide evidence that meta-auxiliary learning can be used in dealing with sequential data.

3. Proposed Method

In this section, we present our approach for future depth prediction. We first introduce the architecture of our model. Our model has a two-branch architecture that jointly solves two related tasks. These two tasks share the backbone features. Given a sequence of observed frames, the primary task aims to predict the depth map of a future frame. In addition to the primary branch, our model has another branch that solves an auxiliary task complementary to the primary task. These two tasks can be jointly learned. The auxiliary task can act as a regularization. We choose to use image reconstruction as the auxiliary task. Given a test image, we can update the model parameters using the auxiliary task since it is self-supervised and does not require manual labels. To avoid catastrophic forgetting, we then propose a meta-auxiliary learning scheme for effective test-time model adaptation. See Fig. 2 for an overview of our approach.

3.1. Model Architecture

Our model has a two-branch architecture (see Fig. 3). The input to the network consists of several consecutive frames in a video. In this paper, we assumption that the input consists of four frames denoted as $(I_{t-3}, I_{t-2}, I_{t-1}, I_t)$. Given the observed four consecutive frames, the primary branch is used to predict the future depth map D_{t+1} at the next time step $t+1$. The auxiliary branch is a self-supervised task that reconstructs the observed frames. These two branches share a backbone network for feature extraction. In the following, we describe the details of these two branches.

Primary Task Learning: Given the input frames $(I_{t-3}, I_{t-2}, I_{t-1}, I_t)$, we first use a 2DCNN backbone to extract spatial features from each frame. The 2DCNN backbone can be selected from any off-the-shelf image classification network, such as VGG [41], ResNet [17] and DenseNet [19]. In our implementation, we adopt VGG19 as our backbone network. The last three fully convolutional layers in VGG19 are removed. We then use a 3DCNN module to encode the features of these four frames from the last convolutional layer of our 2DCNN backbone. For computational reasons, we use a single 3D convolution layer with a kernel size of 4 along the temporal dimension. The 3DCNN module allows us to capture temporal information among the sequence of frames. Lastly, we follow U-NET architecture [39] to construct our decoder. As the decoder upsamples features from low scales to original input resolution, we add an additional convolution layer to output depth maps at

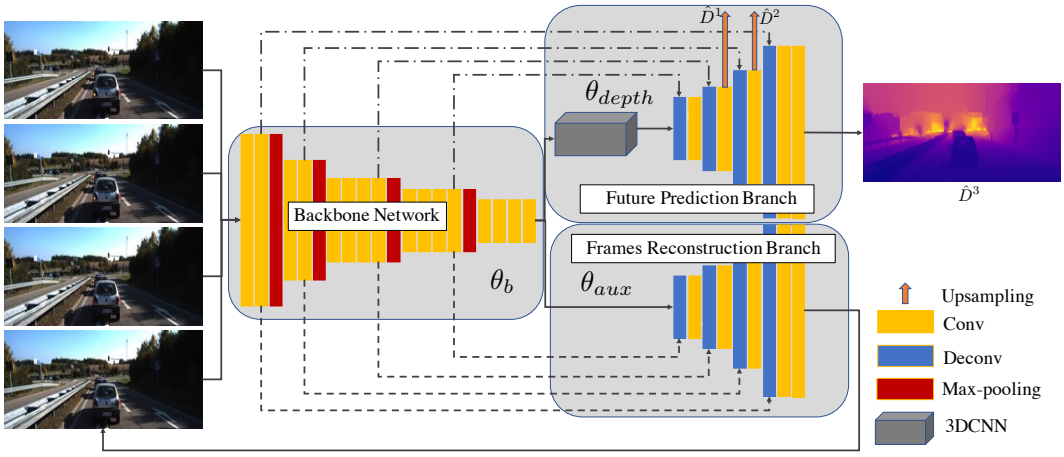

Figure 3: Illustration of our model architecture. Our model has a two-branch architecture. Given a sequence of observed frames, a 3DCNN-based backbone network (with parameters θ_b) is used to extract features. The primary branch (with parameters θ_{depth}) is used to predict the depth map of a future frame. The auxiliary branch (with parameters θ_{aux}) is used to reconstruct the original observed frames from the features extracted by the backbone network. The auxiliary branch can act as a regularization. Note that the auxiliary branch is self-supervised since it does not require any labels.

three different scales. Inspired by the method in [14] that improves the multi-scale formulation, we upsample all the lower resolution depth maps to the input resolution.

We then apply L1 loss and SSIM loss [52] to supervise the primary task branch as:

$$\mathcal{L}_{depth} = \frac{1}{3}\sum_{s=1}^{3}\Big(L_1(\hat{D}^s, D^s) + \alpha \mathbf{SSIM}(\hat{D}^s, D^s)\Big) \quad (1)$$

where $\hat{D}^s$ and D^s denote the predicted future depth of the network and the ground-truth depth map at scale s for the frame at $t+1$, respectively. Here α is a hyperparameter that controls the relative weighting of these two losses.

Self-supervised Auxiliary Learning: Learning a primary task alongside a proper auxiliary task can force the model to capture more meaningful representations and refrain from learning spurious correlation that jeopardizes its generalization ability [34]. However, the auxiliary task should be carefully chosen to complement the primary task. Otherwise, the performance of the primary task would suffer from degradation. In our case, we require an auxiliary task to help the network learn features useful for future depth prediction. In addition, we would like the auxiliary task to be self-supervised so that we can use it for test-time adaptation [45].

In this paper, we propose to use image reconstruction [3, 29] as the auxiliary task. Image reconstruction is self-supervised and does not require any manual labels. In order to perform well in image reconstruction, the model will likely need to learn feature representations that capture the geometric and semantic information of the scene. Intuitively, these features will be useful for future depth prediction as well.

We design an image reconstruction branch similar to the depth prediction decoder based on the shared feature encoder. In the reconstruction branch, we only produce output images at the full resolution scale. The auxiliary task branch can be supervised by the L1 loss:

$$\mathcal{L}_{aux} = \frac{1}{4}\sum_{i=0}^{3}\left\|\hat{I}_{t-i} - I_{t-i}\right\|_1 \quad (2)$$

where $\hat{I}_{t-i}$ denotes the reconstructed frame at time $t-i$.

The overall loss function for our entire network is the linear combination of $\mathcal{L}_{depth}$ and $\mathcal{L}_{Aux}$ with hyperparameter $\beta \in (0, 1)$:

$$\mathcal{L}_{total} = \mathcal{L}_{depth} + \beta \mathcal{L}_{aux} \quad (3)$$

3.2. Meta-Auxiliary Learning

Although auxiliary learning can improve the performance of the primary task, we argue that the model jointly trained with Eq. 3 is sub-optimal for unseen data. Intuitively, a test-time adaptation strategy [45] would further improve the performance of the model. Inspired by [3, 28], we propose a test-time adaption approach using meta-auxiliary learning. The idea of test-time adaptation is to update the model parameters using the loss of the self-supervised auxiliary task during testing, so that the model fits better to the specific characteristic of the test data. However, we have found that naively applying the test-time adaption can sometimes cause catastrophic forgetting [10] that jeopardizes the performance of the primary task. We

propose using a meta-auxiliary learning scheme to learn the model parameters specifically for effective test-time adaptation.

Meta-auxiliary Training: The goal of meta-auxiliary training is to learn the model parameters so that they can be effectively used for test-time adaptation. We consider a pretrained baseline model parameterized by θ, where $\theta = \{\theta_b, \theta_{depth}, \theta_{aux}\}$. Here θ_b, θ_{depth} and θ_{aux} denote the parameters of the backbone network, the future depth prediction branch and the image reconstruction branch, respectively. For example, the pretrained baseline can be obtained by solving Eq. 3. Then we perform adaptation on the image reconstruction task given a training pair $\{(I^n_{t-3}, I^n_{t-2}, I^n_{t-1}, I^n_t), D^n_{t+1}\}$, where $n \in [1 : N]$ and N represents the batch size. For simplicity, we use V^n_t to denote $(I^n_{t-3}, I^n_{t-2}, I^n_{t-1}, I^n_t)$. The update can be written as:

$$\hat{\theta}^n = \theta - \gamma_{inner} \nabla_\theta \mathcal{L}_{aux}(\hat{V}^n_t, V^n_t; \theta) \tag{4}$$

where γ_{inner} denotes the learning rate. Intuitively, $\hat{\theta}^n = \{\hat{\theta}^n_b, \hat{\theta}^n_{depth}, \hat{\theta}^n_{aux}\}$ contains the adapted model parameters for the n-th training pair in the batch via the auxiliary task. Note that since the primary branch is not involved in $\mathcal{L}_{aux}$, the update in Eq. 4 will not change parameters of the primary branch, i.e. $\hat{\theta}^n_{depth} = \theta_{depth}$.

We would like the adapted parameters $\hat{\theta}^n$ to help enhance the prediction of future depth. Thus, we validate the "goodness" of the adapted parameters $\hat{\theta}^n$ using the loss of the primary task. Therefore, the meta-objective is defined as:

$$\min_\theta \sum_{n=1}^{N} \mathcal{L}_{depth}(\hat{D}^n_{t+1}, D^n_{t+1}; \hat{\theta}^n) \tag{5}$$

Note that $\mathcal{L}_{depth}$ is computed based on the adapted parameters $\hat{\theta}^n$, but the optimization is performed with respect to the original parameters θ. This is sensible because $\mathcal{L}_{depth}$ is also a function of θ, since $\hat{\theta}^n$ is obtained via θ in Eq. 4. Accordingly, the meta-objective in Eq. 5 can be minimized by the gradient descent as:

$$\theta \leftarrow \theta - \gamma_{outer} \sum_{n=1}^{N} \nabla_\theta \mathcal{L}_{depth}(\hat{D}^n_{t+1}, D^n_{t+1}; \hat{\theta}^n) \tag{6}$$

where γ_{outer} denotes the learning rate of this update. The full algorithm is outlined in Alg. 1.

Meta-auxiliary Testing: After the auxiliary meta-training, we have obtained the final model θ. During meta-testing, we have an unseen video. We firstly collect a few RGB frames from the unseen data and use Eq. 4 to conduct test-time training to obtain the adapted model parameters $\hat{\theta}$. Finally, we use the adapted model parameters $\hat{\theta}$ to perform predictions on the remaining frames in the test video. Since $\hat{\theta}$ is adapted to each test video, it can better fit the specific characteristic of the video.

Algorithm 1: Meta-Auxiliary Learning

Require: learning rate γ_{inner} and γ_{outer}
Output : meta-auxiliary learned model parameter θ

Initialize $\theta = \{\theta_b, \theta_{depth}, \theta_{aux}\}$ with pretrained model parameter using auxiliary learning.
while *not converge* **do**
  Sample a batch of training data $\{V^n_t, D^n_{t+1}\}_{n=1}^N$;
  for *each* V^n_t **do**
    Evaluate $\nabla_\theta \mathcal{L}_{aux}(\hat{V}^n_t, V^n_t; \theta)$ in Eq. 4.
    Compute adapted parameters $\hat{\theta}^n$:
    $\hat{\theta}^n = \theta - \gamma_{inner} \nabla_\theta \mathcal{L}_{aux}(\hat{V}^n_t, V^n_t; \theta)$
  Update:
  $\theta \leftarrow \theta - \gamma_{outer} \sum_{n=1}^{N} \nabla_\theta \mathcal{L}_{depth}(\hat{D}^n_{t+1}, D^n_{t+1}; \hat{\theta}^n)$

4. Experiments

In this section, we first introduce the dataset used in the experiments in Sec. 4.1. We then describe the details of our implementation in Sec. 4.2, introduce the evaluation metrics in Sec. 4.3, introduce several baseline methods used for comparison in Sec. 4.4, and present quantitative results in Sec. 4.5. Finally, we perform extensive ablation studies in Sec. 4.6 to gain further insights into our method.

4.1. Dataset

We use the KITTI Depth Prediction dataset [48] for evaluation. This dataset contains over 93k annotated depth maps with a resolution around 1241×376. Our proposed method is trained on the official training split and evaluated on the validation set. There are 138 videos in the training set and 13 videos in the validation set, respectively. In our implementation, we construct about 40k video clips for training and 3k video clips for evaluation. Each video clip contains 20 frames.

4.2. Implementation Details

Our method is implemented using the Pytorch library [37]. All experiments are conducted on Nvidia V100 GPUs. The Adam optimizer [22] is used during pre-training, meta-training and test-time adaptation. The input frames are resized to 512×256. Missing depth values are interpolated using the inpainting method in [24]. The upper bound of the depth values is 80 meters.

We first adopt Eq. 3 to pre-train the network. For both primary learning and auxiliary learning, the training is conducted on KITTI training set for 25 epochs. The learning rate is initially set to be 1e-4 and then reduced by a factor of 2 at epochs 15 and 20. The hyperparameter α is set to be 0.1 for balancing the L1 loss and SSIM loss. We let β equal to 0.001 in order to avoid auxiliary learning dominating the primary learning, since the image reconstruction task is eas-

Method	Error (lower is better)				Accuracy (higher is better)		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Current depth estimation							
DORN [11](Current)	**0.091**	**0.532**	**3.872**	**0.151**	**0.891**	**0.981**	**0.993**
DenseDepth [1] (Current)	0.109	0.648	4.414	0.169	0.868	0.969	0.990
Future depth prediction for one time step $t+1$							
DORN [11](Direct copy)	0.109	0.652	4.163	0.169	0.874	0.969	0.987
DenseDepth [1] (Direct copy)	0.122	0.849	4.702	0.193	0.848	0.954	0.982
Ours	**0.094**	**0.561**	**4.060**	**0.156**	**0.886**	**0.972**	**0.991**
Future depth prediction for three time steps $t+3$							
DORN [11](Direct copy)	0.129	1.089	5.102	0.214	0.838	0.936	0.979
DenseDepth [1] (Direct copy)	0.145	1.129	5.270	0.235	0.810	0.933	0.971
Ours	**0.121**	**0.720**	**4.958**	**0.199**	**0.844**	**0.951**	**0.982**

Table 1: Quantitative results of future depth prediction on the KITTI Depth Prediction dataset. We show the results of baselines, one time step ($t+1$) and three time steps ($t+3$) predictions. We compare with two state-of-the-art depth estimation methods (DORN [11] and [1]). We first report the current estimation results of the two baselines. Then we compare the two baselines by directly using the estimated depth of the last observed frame as the future depth prediction. Our proposed method outperforms all baselines for future depth prediction. Not surprisingly, the results show that three-time-step prediction is more challenging than the one-time-step prediction.

ier to converge. After pre-training, we then conduct meta-auxiliary training. During the meta-auxiliary training, we fix the learning rates γ_1 and γ_2 to be 2.5e-5. We perform five gradient updates in the inner update step.

4.3. Evaluation Metrics

The evaluation metrics used in this work are the same as those in [13]. Let d_i, $\hat{d}_i$ and N denote the ground truth disparity map, our estimate, and the total number of pixels in each image, respectively. The metrics are defined as: mean relative error (Abs Rel): $\frac{1}{N}\sum_{i=1}^{N}\frac{\|\hat{d}_i - d_i\|}{d_i}$; square relative error (Sq Rel): $\frac{1}{N}\sum_{i=1}^{N}\frac{\|\hat{d}_i - d_i\|^2}{d_i}$; root mean square error (RMSE): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{d}_i - d_i)^2}$; mean $\log 10$ square error (RMSE log): $\sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\log \hat{d}_i - \log d_i\|^2}$; accuracy with threshold $\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$: the percentage of $\hat{d}_i$ such that $\delta = max(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}) < 1.25, 1.25^2$ or 1.25^3.

4.4. Baselines

Since this paper is the first work on future depth prediction, there is no previous work that we can directly compare with. Nevertheless, we define several baseline methods for comparison as follows. To illustrate the effectiveness of our proposed future depth estimation, we compare with two state-of-the-art approaches for depth estimation, including DORN [11] and DenseDepth [1].

- *Current estimation*: In view of the fact that both DORN and DenseDepth are proposed for predicting the depths of observed images, we follow their original setting to train the two methods on the training set and perform the evaluation on the test set for current depth estimation. Specifically, given an image at time t, the two methods also predict the depth map at time t.

- *Direct copy*: However, these methods can only predict the depths of observed images. Therefore, for a fair comparison, the evaluation protocol is that we first predict the depth for the observed frame at time t, then directly copy it as the prediction at time $t+1$.

4.5. Experimental Results

We first show the experimental results of current depth estimation in Table 1 (top). Ideally, this experiment would provide the unachievable upper limit for future depth estimation, such as the performance of DORN. However, the experimental results also illustrate that not all the current depth estimation methods can be adopted to form this upper limit. For example, the performance of DenseDepth on current depth estimation is worse than that of ours in the case of predicting depth maps one time step into the future.

We then show quantitative results in Table 1 (middle rows) to compare our proposed method with baselines for one time step. From Table 1, we can observe that DORN and DenseDpeth perform worse than ours. This is because

Method	Error (lower is better)				Accuracy (higher is better)		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Impact of each component for one time step $t+1$							
Primary only	0.101	0.634	4.160	0.164	0.878	0.970	0.989
Multi-task	0.098	0.584	4.110	0.161	0.882	0.971	0.991
Multi-task + adaptation	0.112	0.593	4.154	0.164	0.876	0.969	0.988
Ours	**0.094**	**0.561**	**4.060**	**0.156**	**0.886**	**0.972**	**0.991**
Impact of each component for one time steps $t+3$							
Primary only	0.125	0.732	4.973	0.203	0.832	0.942	0.978
Multi-task	0.123	0.724	4.962	0.202	0.842	0.949	0.981
Multi-task + adaptation	0.126	0.745	4.993	0.203	0.828	0.939	0.977
Ours	**0.121**	**0.720**	**4.958**	**0.199**	**0.844**	**0.951**	**0.982**
Impact of batch size							
Batch size $N = 1$	0.097	0.578	4.095	0.160	0.883	0.971	0.991
Batch size $N = 3$	0.095	0.568	4.071	0.157	0.885	0.971	0.991
Batch size $N = 5$	0.094	0.561	4.060	0.156	0.886	0.972	0.991

Table 2: Ablation studies of our proposed method: (1) We compare with methods by removing various components of our proposed method (see Sec. 4.6 for details). We report the experimental results in both cases of predicting depth maps one and three time-step into the future. (2) The performance of our method when using different batch size N. Overall, a larger batch size gives better performance.

these methods are designed to predict depth maps of observed frames, not future frames. This shows that future depth prediction cannot be solved simply by copying the prediction from observed frames. Instead, we need to design algorithms specifically for the future depth prediction task.

We finally consider a more challenging scenario to predict future depth maps with three time steps ahead. Following the setting in future semantic segmentation prediction [31], we input four frames at time $\{t-9, t-6, t-3, t\}$ and predict future depth map at time $t+3$. The quantitative results are shown in Table 1 (bottom). Although the problem is harder, our proposed method still outperforms the two baselines.

4.6. Ablation Study

We perform ablation studies on the impact of the various factors in our method.

4.6.1 Impact of Each Component

We first study the influence of each component in our proposed method. To achieve this, we construct three methods as follows:

- *Primary only*: This method is the 3DCNN introduced in Section 3.1. There is no auxiliary branch in this method. After training, we directly evaluate the performance of the 3DCNN-based future depth prediction model on the test dataset.

- *Multi-task*: This method uses the two-branch architecture, but does not use meta-training or test-time adaptation. Instead, the model parameters are trained using the multi-task loss defined in Eq. 3. After training, we directly use the primary branch on test videos without adaptation. In this method, the auxiliary task is used only as regularization during training.

- *Multi-task + adaptation*: This method is similar to the previous one. The difference is that during testing, it applies the test-time adaptation using Eq. 4. The key difference between this method and our proposed one is that this method uses the multi-task loss in Eq. 3 during training, while our method uses meta-auxiliary learning.

Quantitative Results: From Table 2, we can make several observations. First, we can see that "Multi-task" performs better than "Primary only". This result shows the benefit of using the auxiliary task as regularization during training.

Second, "Multi-task + adaptation" actually performs worse than "Multi-task". This can be explained as a form of catastrophic forgetting. Although the test-time adaptation can improve the auxiliary task, it causes the performance of the primary task to drop. This is because the multi-task

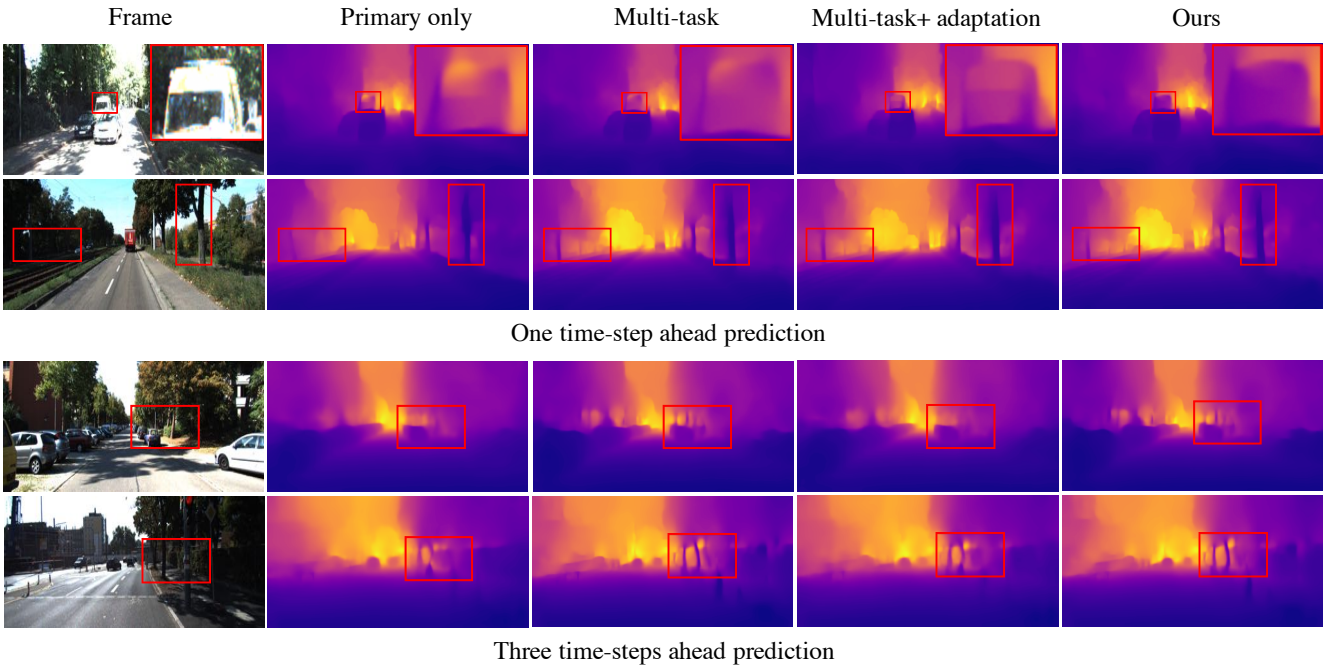

Figure 4: Qualitative examples: (top) one time step future depth prediction; (bottom) three time steps future depth prediction. Our model better captures the object boundaries highlighted by the bounding boxes. This is because the auxiliary task in our model implicitly captures geometric and semantic information specific to the test video since it tries to reconstruct frames in the video. By performing test-time adaption, the backbone network in our model is specifically tuned to the current video.

loss in Eq. 3 does not optimize the model to be effective for test-time adaptation.

Finally, our approach outperforms all other methods. The meta-auxiliary learning in our approach is specifically designed to learn a model that is ready for effective test-time adaptation.

Qualitative Results: To further illustrate the effectiveness of our meta-auxiliary learning approach for future depth estimation, we show some qualitative results in Figure 4. The top and bottom rows in Figure 4 show the results of one time step and three time steps predictions, respectively. It is interesting to note that the depth maps produced by the vanilla 3DCNN suffer from several problems (*e.g.*, failure to estimate accurate depth values on the leaves and object boundaries). In contrast, our method shows better qualitative results. We believe this is because the auxiliary task in our model implicitly captures geometric and semantic information specific to the test video since it tries to reconstruct frames in the video. By performing test-time adaption, the backbone network in our model is specifically tuned to the current video.

4.6.2 Impact of Batch Size

We then study the impact of the batch size N in Alg. 1. We use N = 1, 3 and 5 during the meta-auxiliary training. We demonstrate the quantitative results in Table 2 (bottom). Overall, we observe a large batch size can boost performance. One possible explanation is that a large batch size helps the model avoid overfitting to a particular video.

5. Conclusion

We first introduce the problem of future depth prediction in videos. We then propose a meta-auxiliary learning approach for addressing this problem. In addition to solving the primary task of future depth prediction, our model uses an additional branch to solve an auxiliary task of image reconstruction. The auxiliary task can be considered as regularization. The meta-auxiliary learning is used to learn a model so that it can be effectively adapted to new scenes. Experimental results show that our proposed method outperforms other alternatives both quantitatively and qualitatively.

Limitation and future works. The optimization at the test-time can potentially create difficulties for deploying on edge devices. As future work, we would like to explore a non-optimization based method for efficient test-time adaption.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018.

[2] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafscil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14166–14175, 2022.

[3] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9137–9146, June 2021.

[4] Silvia Chiappa, Sébastien Racaniere, Daan Wierstra, and Shakir Mohamed. Recurrent environment simulators. *arXiv preprint arXiv:1704.02254*, 2017.

[5] Hsu-kuang Chiu, Ehsan Adeli, and Juan Carlos Niebles. Segmenting the future. *IEEE Robotics and Automation Letters*, 5(3):4202–4209, 2020.

[6] Chiho Choi and Behzad Dariush. Looking to relations for future trajectory forecast. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[7] Myungsub Choi, Janghoon Choi, Sungyong Baik, Tae Hyun Kim, and Kyoung Mu Lee. Scene-adaptive video frame interpolation via meta-learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9444–9453, 2020.

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[10] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018.

[12] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[13] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017.

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[15] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *International Conference on Learning Representations*, 2018.

[16] Li Gu, Zhixiang Chi, Huan Liu, Yuanhao Yu, and Yang Wang. Improving protonet for few-shot video object recognition: Winner of orbit challenge 2022. *arXiv preprint arXiv:2210.00174*, 2022.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[18] Anthony Hu, Fergal Cotter, Nikhil Mohan, Corina Gurau, and Alex Kendall. Probabilistic future prediction for video scene understanding. In *European Conference on Computer Vision*, pages 767–785. Springer, 2020.

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017.

[20] Xiaojie Jin, Huaxin Xiao, Xiaohui Shen, Jimei Yang, Zhe Lin, Yunpeng Chen, Zequn Jie, Jiashi Feng, and Shuicheng Yan. Predicting scene parsing and motion dynamics in the future. In *Advances in Neural Information Processing Systems*, 2017.

[21] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, 2017.

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[23] Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pages 2927–2936. PMLR, 2018.

[24] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH*, pages 689–694. 2004.

[25] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. *arXiv preprint arXiv:2207.11213*, 2022.

[26] Huan Liu, Zijun Wu, Liangyan Li, Sadaf Salehkalaibar, Jun Chen, and Keyan Wang. Towards multi-domain single image dehazing via test-time training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5831–5840, June 2022.

[27] Huan Liu, Junsong Yuan, Chen Wang, and Jun Chen. Pseudo supervised monocular depth estimation with teacher-student network. *arXiv preprint arXiv:2110.11545*, 2021.

[28] Shikun Liu, Andrew J Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. *Advances in Neural Information Processing Systems*, 2019.

[29] Kaiyue Lu, Nick Barnes, Saeed Anwar, and Liang Zheng. From depth what can you see? depth completion via auxiliary image reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11306–11315, 2020.

[30] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, pages 125–141. Springer, 2020.

[31] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 648–657, 2017.

[32] Jun Ma, ChuYue Yu, YiWei Xia, XunHuan Ren, Viktar Yurevich Tsviatkou, and Anatoliy Antonovich Boriskevich. Framework for estimating distance and detecting object on mono-camera. In *2022 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pages 1–5. IEEE, 2022.

[33] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations*, 2015.

[34] Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. Auxiliary learning by implicit differentiation. *International Conference on Learning Representations*, 2021.

[35] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. *Advances in neural information processing systems*, 28, 2015.

[36] Seobin Park, Jinsu Yoo, Donghyeon Cho, Jiwon Kim, and Tae Hyun Kim. Fast adaptation to super-resolution networks via meta-learning. *European Conference on Computer Vision*, 2020.

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 2019.

[38] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. *International Conference on Learning Representations*, 2016.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.

[40] Seyed shahabeddin Nabavi, Mrigank Rochan, and Yang Wang. Future semantic segmentation with convolutional lstm. In *British Machine Vision Conference*, 2018.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[42] Jae Woong Soh, Sunwoo Cho, and Nam Ik Cho. Meta-transfer learning for zero-shot super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3516–3525, 2020.

[43] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning*, 2015.

[44] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412, 2019.

[45] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, pages 9229–9248. PMLR, 2020.

[46] Jilin Tang, Haoji Hu, Qiang Zhou, Hangguan Shan, Chuan Tian, and Tony QS Quek. Pose guided global and local gan for appearance preserving human video prediction. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 614–618. IEEE, 2019.

[47] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *IEEE/CVF conference on computer vision and pattern recognition*, pages 1526–1535, 2018.

[48] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision*, pages 11–20. IEEE, 2017.

[49] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5038–5047, 2017.

[50] Ruben Villegas, Jimei Yang, Seunghoon Hong, Xunyu Lin, and Honglak Lee. Decomposing motion and content for natural video sequence prediction. *ICLR*, 2017.

[51] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *Advances in Neural Information Processing Systems*, 2016.

[52] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[53] Tao Zhong, Zhixiang Chi, Li Gu, Yang Wang, Yuanhao Yu, and Jin Tang. Meta-dmoe: Adapting to domain shift by meta-distillation from mixture-of-experts. *arXiv preprint arXiv:2210.03885*, 2022.

[54] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *European Conference on Computer Vision*, pages 822–838, 2018.

[55] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.