Warp-Refine Propagation: Semi-Supervised Auto-labeling via Cycle-consistency

Aditya Ganeshan\textsuperscript{1*} Alexis Vallet\textsuperscript{2} Yasunori Kudo\textsuperscript{2} Shin-ichi Maeda\textsuperscript{2} Tommi Kerola\textsuperscript{2} Rares Ambrus\textsuperscript{3} Dennis Park\textsuperscript{3} Adrien Gaidon\textsuperscript{3}  
\textsuperscript{1}Brown University  \textsuperscript{2}Preferred Networks, Inc.  \textsuperscript{3}Toyota Research Institute (TRI)

Abstract

Deep learning models for semantic segmentation rely on expensive, large-scale, manually annotated datasets. Labelling is a tedious process that can take hours per image. Automatically annotating video sequences by propagating sparsely labeled frames through time is a more scalable alternative. In this work, we propose a novel label propagation method, termed Warp-Refine Propagation, that combines semantic cues with geometric cues to efficiently auto-label videos. Our method learns to refine geometrically-warped labels and infuse them with learned semantic priors in a semi-supervised setting by leveraging cycle-consistency across time. We quantitatively show that our method improves label-propagation by a noteworthy margin of 13.1 mIoU on the ApolloScape dataset. Furthermore, by training with the auto-labelled frames, we achieve competitive results on three semantic-segmentation benchmarks, improving the state-of-the-art by a large margin of 1.8 and 3.61 mIoU on NYU-V2 and KITTI, while matching the current best results on Cityscapes.

1. Introduction

Semantic segmentation, i.e. assigning a semantic class to each pixel in an input image, is an integral task in understanding shapes, geometry, and interaction of components from images. The field has enjoyed revolutionary improvements thanks to deep learning [20, 51, 34]. However, obtaining a large-scale dataset with pixel-level annotations is particularly expensive: for example, labeling takes 1.5 hours on average per image in the Cityscapes dataset [11]. Despite the recent introduction of datasets that are significantly larger than their predecessors [11, 41, 10, 28], scarcity of labeled data remains a bottleneck when compared to other recognition tasks in computer vision [33, 40, 22].

In the common scenario where data is provided as videos with labels for sparsely subsampled frames, a prominent way to tackle data scarcity is label propagation (LP), which automatically annotates additional video frames by propagating labels through time [53, 3]. This intuitive idea to leverage motion-cues via temporal consistency in videos has been widely explored, using estimated motion [2, 12, 26], patch matching [3, 4], or predicting video frames [54]. However, as discussed in Zhu et al. [54], estimating dense motion fields across long periods of time remains notoriously difficult. Further, these methods are often sensitive to hyperparameters (e.g. patch size), cannot handle de-occlusion, or require highly accurate optical flow, thus limiting their applicability.

Another promising approach for obtaining large-scale annotation in semi-supervised settings is self-training (ST), in which a teacher model, trained to capture semantic cues, is used to generate additional annotations on unlabeled images [18, 19, 56, 55]. While there have been significant improvements in ST, various challenges still remain in controlling noise in pseudo-labels, such as heuristic decisions on confidence thresholds [35], class imbalance in pseudo-labels [13], inaccurate predictions for small segments, and misalignment of category definition between source and target domain.

To mitigate the drawbacks of LP and ST, we propose...
Warp-Refine Propagation (referred to as warp-refine), a novel method to automatically generate dense pixel-level labels for raw video frames. Our method is built on two key insights: (i) By combining motion cues with semantic cues, we can overcome the respective limitations of LP and ST, and (ii) By leveraging cycle-consistency across time, we can learn to combine these two complementary cues in a semi-supervised setting without sequentially-annotated videos.

Specifically, our method first constructs an initial estimate by directly combining labels generated via motion cues and semantic cues. This initial estimate, containing erroneous conflict resolution and faulty merges, is then rectified by a separate refinement network. The refinement network is trained in a semi-supervised setting via a novel cycle-consistency loss. This loss compares the ground-truth labels with their cyclically propagated version created by propagating the labels forward-and-backward through time in a cyclic loop ($t \rightarrow t + k \rightarrow t$). Our loss is built on the observation that as our auto-labeling method is bi-directional, it can be used to generate different versions of each annotated frame. Once this network is trained, it is used to correct errors caused by propagation of variable length. In Fig. 2 we show a qualitative comparison of our method against prior-arts, demonstrating drastic improvements in label quality.

With quantitative analysis on a large scale autonomous driving datasets (ApolloScape [41]), we concretely establish the superior accuracy of our method against previous state-of-the-art auto-labeling methods. Such an analysis of different methods has been starkly missing from prior works [54, 35, 26]. As shown in Fig. 1, we observe that warp-refine accurately propagates labels for significantly longer time intervals, with a notable average $13.1 \text{mIoU}$ improvement on ApolloScape compared to previous work. Further, it accurately labels rare-classes such as ‘Bicycle’ and thin-structures such as ‘Poles’ (cf. Section 4.3). As a result, by training single-frame semantic segmentation models with the additional data labeled by our method, we achieve state-of-the-art performance on KITTI [1], NYU-V2 [27] and Cityscapes [11] benchmarks (cf. Section 4.5).

In summary, our main contributions are: 1) A novel algorithm, termed Warp-Refine Propagation, that produces significantly more accurate pseudo-labels, especially for frames distant in time; 2) A novel loss function, based on the cycle-consistency of learned transformations, to train our method in a semi-supervised setting; and 3) A quantitative analysis on the quality and utility of different auto-labeling methods on multiple diverse datasets. To the best of our knowledge, our work is the first to utilize both semantic and geometric understanding for the task of video auto-labeling.

2. Related Work

Self-training (ST). The approach of applying a mature teacher networks to unlabeled images and using the predicted
labels to supervise student networks has received increasing attention. Xie et al. [45] introduces a framework for ST for controlling noise in pseudo-labels to exceed the teacher network. Chen et al. [9] extend it for semantic segmentation. Re-voting [23]. The key criteria for success in LP is the predicted labels, for usage in a self-training framework. In generation of ground-truth labels, not feature representation or segmentation [12, 38, 30]. Our chief distinction is the propagation of ground-truth labels to prevent error accumulation.

methods combining semantic and geometric cues have been proposed in the past for other tasks such as future-frame prediction [21] and video-segmentation [12, 38, 30]. Our chief distinction is the propagation of ground-truth labels, not feature representation or predicted labels, for usage in a self-training framework. In Sec. 4.4, we provide quantitative analysis and report that our method is successful in propagating ground-truth labels accurately across long intervals of time.

Cycle-consistency. The concept of cycle-consistency has been previously utilized for learning object embeddings [44], one-shot semantic segmentation [42] and video interpolation [32]. Our work is inspired by work using cycle-consistency for learning a robust tracker [44]. However, we differ in that we address the noisy nature of our tracking/geometric modelling method itself.

3. Warp-Refine Propagation

We first present the notation used throughout our paper. This is followed by the description of two recursive algorithms for propagating dense pixel labels, followed by the proposed method to train the denoising models by leveraging cyclic consistency.

3.1. Notation

Given a labeled video frame \((I_t, L_t)\) and its adjacent raw frame \(L_{t+k'}\), we aim to create approximated labels \(\hat{L}_{t+k'}\) for \(1 \leq k' \leq K\). To this end, we introduce two greedy propagation algorithms. They are greedy in that the optimal solution for \(\hat{L}_{t+k'}\), namely \(\hat{L}_{t+k'}^*\), is obtained by applying a recursive propagation step to the (approximately optimal) solution for the previous frame, \(\hat{L}_{t+k'-1}^*\):

\[
\hat{L}_k^* = \Psi(\hat{L}_{k-1}^*, I_{k-1}) , \quad t + 1 \leq k \leq t + K, \quad (1)
\]

\[
\hat{L}_t^* = L_t . \quad (2)
\]

We introduce two algorithms, warp-inpaint and warp-refine, that grow in complexity of \(\Psi\). Notably, the approach of Zhu et al. [54] can be included in this framework, when only video-prediction algorithm [31] based motion vectors are used for defining \(\Psi\).

1To avoid clutter in subscripts, we define \(k := t + k'\).
3.2. Warp-Inpaint

As commonly observed in online visual tracking [15], when Ψ relies purely on motion cues, the propagated labels are susceptible to propagation error (i.e. drifting) [54]. In addition, the pixels of new scene elements cannot be explained by labels in previous frames (e.g. cars entering the field of view). One way to address this is to re-initialize the semantic labels using a strong semantic segmentation model.

We therefore allow each pixel in \( \hat{L}_k \) to be derived either from motion cues encoded in the \((I_{k-1}, I_k)\) pair or from semantic cues computed solely from \(I_k\). Formally, we compute a version of \( \hat{L}_k \), namely \( \hat{L}_k^m \), by remapping \( \hat{L}_k \) using a transformation \( \phi_{k-1,k} \) learned to warp \( I_{k-1} \) onto \( I_k \).

We then blend \( \hat{L}_k^m \) with another version of \( \hat{L}_k \), namely \( \hat{L}_k^s \), which are semantic labels obtained by applying a pretrained semantic segmentation model \( g_\psi \) on \( I_k \):

\[
\hat{L}_k^m = \phi_{k-1,k}(\hat{L}_k) \tag{3},
\]

\[
\hat{L}_k^s = g_\psi(I_k) \tag{4},
\]

\[
\hat{L}_k = M \odot \hat{L}_k^m + (1 - M) \odot \hat{L}_k^s \tag{5},
\]

where \( \odot \) denotes pixel-wise multiplication. The \((x, y)\) value of the binary mixing coefficients \( M \) represents whether we trust the the estimated motion vector at the position \((x, y)\), compared with the semantic label computed by \( g_\psi \). We determine \( M \) by measuring the Euclidean distance of pixel values between \( I_k \) and \( \phi_{k-1,k}(I_{k-1}) \):

\[
M(x, y) = \| I_k(x, y) - \phi_{k-1,k}(I_{k-1})(x, y) \|_2 < \tau \tag{6},
\]

where \( \| \cdot \| \) is the indicator function. The motion vectors are obtained by applying a pretrained motion estimation model, \( \theta \), to neighboring image pairs: \( \phi_{k-1,k} = \theta(I_{k-1}, I_k) \). We let \( \Psi \) denote the entire propagation process (3) - (5), and \( \hat{L}_k^W \) denote the resulting pseudo-labels at the \( k \)-th frame:

\[
\hat{L}_k^W = \Psi^W(\hat{L}_k^R, I_{k-1}, I_{k-1}) \tag{7}.
\]

3.3. Warp-Refine

With \( \hat{L}_k^W \), we obtain an initial fusion of geometric and semantic cues. This estimate is however still subject to artifacts from imperfect motion estimation and semantic segmentation models (\( \theta \) and \( g_\psi \)). We therefore extend the recursive step to refine \( \hat{L}_k^W \) by applying a de-noising network \( \Omega_\lambda \) that aims to remove these artifacts:

\[
\hat{L}_k = \Omega_\lambda(\hat{L}_k^W) \tag{8}.
\]

Note that the goal of \( \Omega_\lambda \) is to mitigate particular types of errors caused by the propagation steps of \( \Psi^W \), specifically by \( \theta \) and \( g_\psi \), and the choice of \( \tau \), and in effect properly merge the semantic and geometric cues. The extended propagation process and generated pseudo-labels are denoted by \( \Psi^R \) and \( \hat{L}_k^R \), respectively: \( \hat{L}_k^R = \Psi^R(\hat{L}_k, I_{k-1}, I_{k-1}) \).

Figure 4: For training the refinement network, we generate cyclically propagated labels by applying forward-backward propagation on annotated frame using warp-inpaint transformation, and compare them with the ground-truth labels.

3.4. Learning

There are three sets of learnable parameters in warp-refine: in the motion estimation model \( \theta \), in the semantic segmentation model \( g_\psi \), and in the de-noising model \( \Omega_\lambda \). We use a constant pretrained motion estimation model [31] and semantic segmentation model [35, 48] for \( \theta \) and \( g_\psi \), respectively, and use a fixed value for \( \tau \). Here we describe how to train \( \Omega_\lambda \).

**Cycle-consistency.** Training a de-noising model in a fully-supervised setting typically requires a large dataset of noisy-clean pairs [16], which in our case is \((\hat{L}_k^R, I_k)\). To address the lack of \( L_k \) in our semi-supervised setting, we leverage the cycle-consistency inherent in our propagation mechanism. The cyclic propagation consists of two stages:

1. **forward:** we execute propagation steps of \( \Psi^W \) (i.e. Eq. 3 - Eq. 5) \( l \) times to obtain \( \hat{L}_l^W \).

2. **backward:** we execute \( l \) steps of inverted propagation of \( \Psi^W \) to obtain a cyclically propagated variant of \( L_l \), namely \( \hat{L}_0 \).

The inverted propagation step is similar to \( \Psi^W \), but executed in a reverse order:

\[
\hat{L}_m_{k-1} = \phi_{k-1,k}(\hat{L}_k) \tag{9},
\]

\[
\hat{L}_k = g_\psi(I_{k-1}) \tag{10},
\]

\[
\hat{L}_k = M \odot \hat{L}_k^m + (1 - M) \odot \hat{L}_k^s \tag{11}.
\]
where \( \hat{L}^k_t \) is set to the result of the forward stage, \( \hat{L}^W_t \), and motion vectors are computed in backward:

\[
\phi_{k,k-1} = f_\theta(I_k, I_{k-1}).
\] (12)

The cycle-consistency loss is computed by comparing the annotated labels \( L_t \) and its cyclically warped counterpart \( \hat{L}^?_t \) with de-noising applied: \( \mathcal{L} (L_t, \Omega (\hat{L}^?_t)) \). This is optimized via standard gradient-based methods during training.

Notably, the backward steps, and therefore the entire forward-backward process, chains the same set of transformations used in \( \Psi^W \) (i.e. \( f_\theta, g_\psi \) and the blending strategy). Therefore, the de-noising network trained with this cycle-consistency loss is expected to correct the errors in pseudo-labels generated by variable length of the \( \Psi^W \), which is the goal of \( \Omega_\lambda \). In practice, we use varying \( \lambda \) in training to maximize this generalizability. Fig. 4 presents a visual summary of our approach for learning label refinement via cycle-consistency of labels.

4. Experiments

4.1. Datasets

We present quantitative and qualitative experiments on four semantic segmentation benchmarks:

**NYU.** The NYU-Depth V2 dataset [27] consists of 1449 densely labeled images split into 795 training images and 654 testing images, which are randomly sampled from video sequences. Due to the high fps (20-30 fps) and slow camera movement, we sub-sample the video at 2fps and create sequences of variable lengths of up to 21 frames around the labeled frame, yielding 9786 frames for label propagation.

**KITTI.** The KITTI Vision Benchmark Suite [1] consists of 200 training and 200 test images with pixel-level annotation. For each of the training images, we use sequential frames (±10) from the scene-flow subset for label propagation.

**Cityscapes.** The Cityscapes dataset [11] is split into a train, validation, and test set with 2975, 500, and 1525 images, respectively. For each training image, we use the ±10 unlabeled neighboring frames provided as part of the dataset.

**ApolloScape.** The ApolloScape dataset [41] contains pixel-level annotations for sequentially recorded images, divided as 40960 training, and 8327 validation images, allowing evaluation of the accuracy of the propagated labels. We create continuous partitions of 21 frames, and use the central frame as a training data-point, and the adjacent frames (±10) for label propagation. This yields a train-subset of size containing 2005 and a label-propagation subset of 38095 images.

4.2. Implementation Details

Our approach consists of three components: a motion-estimation network \( f_\theta \), the semantic segmentation model \( g_\psi \), and a de-noising network \( \Omega_\lambda \). Following Zhu et al. [54], for the motion-estimation network \( f_\theta \), we use video-prediction model based on SDC-net [31]. For our task, video prediction performed better than warping with optical-flow [36] (cf. supplementary). For segmentation model \( g_\psi \), we adopt the architecture (MSA-HrNet-OCR) and training protocols outlined in Tao et al. [35]. Finally for \( \Omega_\lambda \), we use a pix2pix-style network [43, 29]. First, an encoder takes the warp-inpainted labels as the input. The encoding formed is then concatenated with OCR features [48] from \( g_\psi \), and passed through a decoder. For robust refinement, we perform cycle-consistency training with the range of propagation sampled till ±6. Note that the labels (\( L_t/L^W_t/L^?_t \)) are utilized as one-hot vectors.

**Single-frame semantic segmentation.** For training single frame semantic segmentation networks with the auto-labeled data, we use the same architecture and training protocol as followed for \( g_\psi \), with one important modification: independent of the amount of data generated by auto-labeling, we use a fixed epoch size (thrice the dataset size). This ensures that we do not under-train the baseline model (cf. supplementary). When training with the propagated data, we sample 70% of the epoch from the propagated data, and 30% from the manually annotated data. Unless specified otherwise, we sample additional data at time-frames \( t \pm n \) (\( n \in \{2, 4, 6, 8\} \)) (similar to [26]). We refer to the model trained with no additional data as the baseline model.

**Auto-labeling baselines.** We compare our methods warp-inpaint (\( \Psi^W \), cf. section 3.2) and warp-refine (\( \Psi^R \), cf. section 3.3) against existing auto-labeling techniques. We use the method proposed by Zhu et al. [54], which uses only \( f_\theta \) to generate the labels, and refer to this method as motion-only labeling. When using these labels, we use joint image-label propagation as recommended by [54]. Similarly, we also use the method proposed by Tao et al. [35], which generates labels using only \( g_\psi \), and refer to this method as semantic-only labeling. For semantic-only labeling, we use the best performing architecture (MSA-HRNet-OCR, trained on the manually annotated images), and use only the pixels with > 0.9 confidence, as recommended in [35].

4.3. Quality of Propagated Labels

We first provide extensive analysis of the our auto-labeling methods: warp-inpaint and warp-refine, and existing techniques motion-only and semantic-only. We evaluate the labels generated by these methods against the ground-truth labels provided in the ApolloScape dataset [41]. We focus on two crucial aspects: (1) long-range propagation, and (2) labeling of hard classes.

First, we compare the different auto-labeling methods across various propagation lengths. Fig. 1 reports the mean Intersection over Union (mIoU) between propagated labels and ground-truth labels, given various propagation methods and propagation lengths. We note that the motion-only and the semantic-only models show a clear trade-off with respect
to the propagation length: motion-only model produces more accurate labels over shorter ranges, while semantic-only model over longer ranges. Further, when propagating without refinement (i.e. warp-inpaint), the accuracy degrades for longer propagation lengths, even dropping below the semantic-only labeling. Finally, propagating with refinement (i.e. warp-refine) produces significantly cleaner labels than the others. Due to the refinement module, our method retains its accuracy even at large time-steps, attaining a large margin of 11.35 mIoU over the closest competing method at $t \pm 10$.

Next, we quantify the IoU of difficult classes and the overall mIoU across all propagation lengths in Fig. 5. Notably, both prior methods for auto-labeling fail on thin structures such as ‘Poles’ - motion-only causes over-labeling due to drifting, while semantic-only causes under-labeling due to low-confidence predictions (such regions are frequently masked out). Further, we note that semantic-only labeling severely fails at estimating the ‘Ignore’ class, in contrast to motion-only labeling which yields accurate estimates. As the ‘Ignore’ class consists of widely varying objects, it is difficult to model it as a semantic class (typically ‘Ignore’ class includes regions labeled as ‘Others’ and ‘Unlabeled’; i.e. labels which do not have any semantic definition). Therefore, following the recommended protocol \( [35] \), for semantic-only labeling we estimate ‘Ignore’ regions via probability thresholding \( [35] \) (cf. Sec. 4.2). Despite the careful selection of this threshold parameter, semantic-only labeling fails to accurately label the ‘Ignore’ class. Our methods (warp-inpaint, and warp-refine) effectively combine motion-cues with semantic-cues, thus overcoming this drawback, and properly estimate the ‘Ignore’ class. Overall, our method again yields an impressive margin of 13.12 mIoU over the closest prior-art (averaged across all propagation lengths).

Finally, we also present qualitative results in Fig. 6, noting that our propagated labels do not suffer from error propagation compared to motion-only, while achieving higher accuracy on the rare classes (e.g. bikes) compared to semantic-only. Fig. 7 also presents a failure case for our method, caused due to the concurrent failure of both semantic and motion cues for the ‘Rider’.

### 4.4. Utility of Propagated Labels

We now demonstrate that the significant improvements in auto-labeling directly translate to superior performance for single-frame semantic segmentation models trained with our generated data. We perform our experiments on NYU-V2, ApolloScape and Cityscapes. Following Zhu et al. \( [54] \), for each experiment, we perform three runs and report the mean (\( \mu(mIoU) \)) and sample standard deviation (\( \sigma(mIoU) \)). Our analysis is summarised in Table 1.

**NYU-V2 & ApolloScape.** Training with warp-refine labels consistently yields better results. On NYU-V2, our labels yield an average improvement of 1.54 mIoU, compared to only 0.35 mIoU for the semantic-only labeling method. Similarly, on ApolloScape, warp-refine labels increase performance by 1.11 mIoU, whereas the closest auto-labeling baseline (semantic-only) yields a benefit of only 0.47. Finally, we note that training with motion-only labels consistently leads to a drop in performance.

Since ApolloScape contains the ground-truth annotation of the all the provided images, we also provide evaluation using the ground-truth labels instead of the labels generated via auto-labeling. This acts as an oracle propagation model, and we treat it as an empirical upper-bound on the benefits from label propagation. Using the ground-truth instead of propagated labels (at \( (t \pm \{2, 4, 6, 8\}) \)) yields a benefit of
work shows that using geometric cues can further improve
Table 1: We analyse the performance of semantic-segmentation models trained with auto-labelled data. Across the three datasets, we observe that warp-refine labels consistently induce larger improvements than labels from other methods. The results for semantic-only, warp-inpaint and warp-refine are computed by sampling auto-labelled frames at time steps \( t \pm \{2, 4, 6, 8\} \). Following Zhu et al. [54], for motion-only, we only sample frames at time steps \( t \pm 2 \). We report the average performance (\( \mu(mIoU) \)), sample standard deviation (\( \sigma(mIoU) \)) and average improvement over baseline (\( \Delta(mIoU) \)) by performing three runs of each experiment.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \mu(mIoU) )</td>
<td>( \sigma(mIoU) )</td>
<td>( \Delta(mIoU) )</td>
</tr>
<tr>
<td>Baseline</td>
<td>50.58</td>
<td>0.25</td>
<td>-</td>
</tr>
<tr>
<td>motion-only</td>
<td>50.38</td>
<td>0.13</td>
<td>-0.20</td>
</tr>
<tr>
<td>semantic-only</td>
<td>50.93</td>
<td>0.17</td>
<td>+0.35</td>
</tr>
<tr>
<td>warp-inpaint</td>
<td>51.28</td>
<td>0.19</td>
<td>+0.70</td>
</tr>
<tr>
<td>warp-refine</td>
<td>52.12</td>
<td>0.13</td>
<td>+1.54</td>
</tr>
<tr>
<td>Oracle (GT)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 7: Qualitative comparison: We show a failure case for our method warp-refine on Cityscapes [11]. Our method can fail when both semantic and motion cues fail (e.g. ‘Rider’ is mislabeled).

2.73 mIoU. Notably, training with warp-refine labels attains about 40% of this empirical upper-bound.

Cityscapes-val. While warp-refine labels yield performance superior to prior methods, we notice that the gap between semantic-only and warp-refine is smaller on Cityscapes (while still being statistically significant). It is likely due to performance saturation: across the three datasets, we observe that as the performance of the baseline model increase, (1) the utility of semantic-only labels increases (as they are more accurate), and (2) the utility of warp-refine labels decreases (as labels in adjacent images become less useful). Despite this drawback, warp-refine labels are still significantly more effective than the prior arts. Labeling additional data via a teacher model has been recently used in many self-training approaches [45, 9]. Our work shows that using geometric cues can further improve the labels, leading to increased performance.

Propagating over longer time-horizons. The efficacy of the propagated labels depends on the how far the ground-truth labels can be accurately propagated. This is critical as there is little information gain in the immediate neighbors of an annotated frame. In Figure 8, we observe that training semantic segmentation models with warp-refine labels sampled from long time ranges yields a large benefit. This demonstrates that the ability of warp-refine to accurately propagate labels onto remote frames is critical for effectively improving semantic-segmentation models. Note that training with motion-only labels only degrade the performance further as the propagation length increases. In contrast, our method shows no degradation, even while sampling labels from [−10, 10].

4.5. Semantic Segmentation Benchmarks

Finally, we tabulate the state-of-the-art performance achieved by our method on three semantic segmentation benchmarks: NYU-V2, KITTI, and Cityscapes-test. As the evaluation rules for KITTI and Cityscapes explicitly state that the test-split should not be used for ablative study, we only evaluate our final model, i.e. MSA-HrNet-OCR [35] trained with warp-refine labels.
NYU-V2. We compare our model with the best reported scores on the dataset in Table 3. Our method yields an improvement of 1.8 mIoU over the prior state-of-the-art, while also attaining favorable statistics for the other metrics, notably, an increase of 3.5% in class-wise mean pixel-accuracy (mean-acc). Due to the semantic complexity of NYU-V2, additional data is decisively beneficial. Specifically, long-tail classes such as ‘Bag’, ‘White-board’ and ‘Shower-curtain’ yield an average benefit of 7.29 IoU over the baseline.

KITTI. We report our performance on the KITTI dataset in Table 4. We show a significant increase over the previous state-of-the-art method (Chen et al. [9]). The optimal way to combine the propagated labels with manual annotations is to use a model trained on ApolloScape. Further, labels generated from warp-refine are shown to be useful for improving single-frame semantic-segmentation models. By training semantic segmentation models with warp-refine labels, we achieve state-of-the-art performance on NYU-V2 (+1.8 mIoU), KITTI (+3.6 mIoU) and Cityscapes (+0.1 mIoU). The optimal way to combine the propagated labels with manual annotations as well as weaker sources of supervision (i.e. coarse labels) remains an unsolved problem, which we aim to address in future work.

Acknowledgments. This work was supported by Woven Core, Inc.
References


[52] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain


