Abstract

The key challenge in learning dense correspondences lies in the lack of ground-truth matches for real image pairs. While photometric consistency losses provide unsupervised alternatives, they struggle with large appearance changes, which are ubiquitous in geometric and semantic matching tasks. Moreover, methods relying on synthetic training pairs often suffer from poor generalisation to real data.

We propose Warp Consistency, an unsupervised learning objective for dense correspondence regression. Our objective is effective even in settings with large appearance and viewpoint changes. Given a pair of real images, we first construct an image triplet by applying a randomly sampled warp to one of the original images. We derive and analyze all flow-consistency constraints arising between the triplet. From our observations and empirical results, we design a general unsupervised objective employing two of the derived constraints. We validate our warp consistency loss by training three recent dense correspondence networks for the geometric and semantic matching tasks. Our approach sets a new state-of-the-art on several challenging benchmarks, including MegaDepth, RobotCar and TSS. Code and models are at github.com/PruneTruong/DenseMatching.

1. Introduction

Finding dense correspondences between images continues to be a fundamental vision problem, with many applications in video analysis [44], image registration [48, 42], image manipulation [7, 25], and style transfer [19, 24]. While supervised deep learning methods have achieved impressive results, they are limited by the availability of ground-truth annotations. In fact, collecting dense ground-truth correspondence data of real scenes is extremely challenging and costly, if not impossible. Current approaches therefore resort to artificially rendered datasets [4, 14, 45, 13], sparsely computed matches [5, 55], or sparse manual annotations [3, 34, 10]. These strategies lack realism, accuracy, or scalability. In contrast, there is a virtually endless source of unlabelled image and video data, which calls for the design of effective unsupervised learning approaches.

Photometric objectives, relying on the brightness constancy assumption, have prevailed in the context of unsupervised optical flow [35, 57, 31]. However, in the more general case of geometric matching, the images often stem from radically different views, captured at different occasions, and under different conditions. This leads to large appearance transformations between the frames, which significantly undermine the brightness constancy assumption. It is further invalidated in the semantic matching task [25], where the images depict different instances of the same object class. As a prominent alternative to photometric objectives, warp-supervision [50, 49, 36, 32], also known as self-supervised learning [37, 40, 34], trains the network on synthetically warped versions of an image. While benefiting from direct supervision, the lack of real image pairs often leads to poor generalization to real data.

We introduce Warp Consistency, an unsupervised learning objective for dense correspondence regression. Our loss
models based on brightness constancy and spatial smoothness losses [35, 57]. The predominant technique mainly relies on photometric losses, e.g. Charbonnier penalty [57], census loss [31], or SSIM [54, 52]. Such losses are often combined with forward-backward consistency [31] and edge-aware smoothness regularization [53]. Occlusion estimation techniques [16, 31, 53] are also employed to mask out occluded or outlier regions from the objective. Recently, several works [27, 28, 26] use a data distillation approach to improve the flow predictions in occluded regions. However, all aforementioned approaches rely on the assumption of limited appearance changes between two consecutive frames. While this assumption holds to a large degree in optical flow data, it is challenged by the drastic appearance changes encountered in geometric or semantic matching applications, as visualised in Fig. 2.

Unsupervised geometric matching: Geometric matching focuses on the more general case where the geometric transformations and appearance changes between two frames may be substantial. Methods either estimate a dense flow field [32, 50, 49, 41] or output a cost volume [39, 55], which can be further refined to increase accuracy [38, 22, 47]. The later approaches train the feature embedding, which is then used to compute dense similarity scores. Recent works further leverage the temporal consistency in videos to learn a suitable representation for feature matching [6, 15, 51]. Our work focuses on the first class of methods, which directly learn to regress a dense flow field. Recently, Xen et al. [41] use classical photometric and forward-backward consistency losses to train RANSAC-Flow. They partially alleviate the sensitivity of photometric losses to large appearance changes by pre-aligning the images with Ransac. Several methods [32, 50, 49] instead use a warp-supervision loss. By posing the network to regress a randomly sampled warp during training, a direct supervisory signal is obtained, but at the cost of poorer generalization abilities to real data.

Semantic correspondences: Semantic matching poses additional challenges due to intra-class appearance and shape variations. Manual annotations in this context are ill-defined and ambiguous, making it crucial to develop unsupervised objectives. Methods rely on warp-supervision strategies [36, 37, 3, 40, 50], use proxy losses on the cost volume [12, 39, 37, 34], identify correct matches from forward-backward consistency of the cost volumes [17], or jointly learn semantic correspondence with attribute transfer [19] or segmentation [21]. Most related to our work are [58, 56, 59]. Zhou et al. [58] learn to align multiple images using 3D-guided cycle-consistency by leveraging the ground-truth matches between multiple CAD models. However, the need for 3D CAD models greatly limits its applicability in practice. In FlowWeb [59], the authors optimize online pre-existing pair-wise correspondences using the cycle consistency of flows between images in a collec-
tion. Unlike these approaches, we require pairs of images as unique supervision and propose a general loss formulation, learning to regress dense correspondences directly.

3. Method

3.1. Problem formulation and notation

We address the problem of finding pixel-wise correspondences between two images \( I \in \mathbb{R}^{h \times w \times 3} \) and \( J \in \mathbb{R}^{h \times w \times 3} \). Our goal is to estimate a dense displacement field \( F_{I \rightarrow J} \in \mathbb{R}^{h \times w \times 2} \), often referred to as flow, relating pixels in \( I \) to \( J \). The flow field \( F_{I \rightarrow J} \) represents the pixel-wise 2D motion vectors in the coordinate system of image \( I \). It is directly related to the mapping \( M_{I \rightarrow J} \in \mathbb{R}^{h \times w \times 2} \), which encodes the absolute location \( M_{I \rightarrow J}(x) \in \mathbb{R}^{2} \) in \( J \) corresponding to the pixel location \( x \in \mathbb{R}^{2} \) in image \( I \). It is thus related to the flow through \( M_{I \rightarrow J}(x) = x + F_{I \rightarrow J}(x) \). It is important to note that the flow and mapping representations are asymmetric. \( M_{I \rightarrow J} \) parametrizes a mapping from each pixel in image \( I \), which is not necessarily bijective.

With a slight abuse of notation, we interchangeably view \( F_{I \rightarrow J} \) and \( M_{I \rightarrow J} \) as either elements of \( \mathbb{R}^{h \times w \times 2} \) or as functions \( \hat{f}_{I \rightarrow J}, \hat{M}_{I \rightarrow J} : \mathbb{R}^{2} \rightarrow \mathbb{R}^{2} \). The latter is generally obtained by a bilinear interpolation of the former, and the interpretation will be clear from context when important.

We define the warping \( \Phi_{F}(T) \) of a function \( T : \mathbb{R}^{2} \rightarrow \mathbb{R}^{d} \) by the flow \( F \) as \( \Phi_{F}(T)(x) = T(x + F(x)) \). This is more compactly expressed as \( \Phi_{F}(T) = T \circ M_{F} \), where \( M_{F} \) is the mapping defined by \( F \) and \( \circ \) denotes function composition. Lastly, we let \( \mathbb{I} : \mathbb{R}^{2} \rightarrow \mathbb{R}^{2} \) be the identity map \( \mathbb{I}(x) = x \).

The goal of this work is to learn a neural network \( j_{\theta} \), with parameters \( \theta \), that predicts an estimated flow \( \hat{F}_{I \rightarrow J} = f_{\theta}(I, J) \) relating \( I \) to \( J \). We will consistently use the hat \( \hat{\cdot} \) to denote an estimated or predicted quantity. The straightforward approach to learn \( f_{\theta} \) is to minimize the discrepancy between the estimated flow \( \hat{F}_{I \rightarrow J} \) and the ground-truth flow \( F_{I \rightarrow J} \) over a collection of real training image pairs \((I, J)\).

However, such supervised training requires large quantities of densely annotated data, which is extremely difficult to acquire for real scenes. This motivates the exploration of unsupervised alternatives for learning dense correspondences.

3.2. Unsupervised data losses

To develop our approach, we first briefly review relevant existing alternatives for unsupervised learning of flow. While there is no general agreement in the literature, we adopt a practical definition of unsupervised learning in our context. We call a learning formulation “unsupervised” if it does not require any information (i.e., supervision) other than pairs of images \((I, J)\) depicting the same scene or object. Specifically, unsupervised methods do not require any annotations made by humans or other matching algorithms. Photometric losses: Most unsupervised approaches train the network using a photometric loss \([57, 31, 53, 41]\). Under the photometric consistency assumption, it minimizes the difference between image \( I \) and image \( J \) warped according to the estimated flow field \( \hat{F}_{I \rightarrow J} \) as,

\[
L_{\text{photo}} = \rho(I, \Phi_{\hat{F}_{I \rightarrow J}}(J)) \quad \text{(1)}
\]

where, \( \rho(\cdot, \cdot) \) is a function measuring the difference between two images, e.g., \( L_{2} \) \([57]\), SSIM \([54]\), or census \([31]\).

Forward-backward consistency: By constraining the backward flow \( \hat{F}_{J \rightarrow I} \) to yield the reverse displacement of its forward counterpart \( \hat{F}_{I \rightarrow J} \), we achieve the forward-backward consistency loss \([31]\).

\[
L_{\text{fb}} = \| \hat{F}_{I \rightarrow J} + \Phi_{\hat{F}_{J \rightarrow I}}(\hat{F}_{I \rightarrow J}) \| \quad \text{(2)}
\]

While this results in a strong and direct training signal, warp supervision methods struggle to generalize to real image pairs \((I, J)\). This can lead to over-smooth predictions and instabilities in the presence of unseen appearance changes.

3.3. Warp consistency graph

We set out to find a new unsupervised objective suitable for scenarios with large appearance and view-point changes, where photometric based losses struggle. While the photometric consistency assumption is avoided in the forward-backward consistency (Fig. 3a) and warp-supervision (Fig. 3b) objectives, these methods suffer from
severe drawbacks in terms of degenerate solutions and lack of realism, respectively. To address these issues, we consider all possible consistency relations obtained from the three images involved in both aforementioned objectives. Using this generalization, we not only retrieve forward-backward and warp-supervision as special cases, but also derive a family of new consistency relations.

From an image pair (I, J), we first construct an image triplet (I, I', J) by warping I with a known flow-field W in order to generate the new image I' = ΦW(I). We now consider the full consistency graph, visualized in Fig. 3c, encompassing all flow-consistency constraints derived from the triplet of images (I, I', J). Crucially, we exploit the fact that the transformation F_{I'\rightarrow I} = W is known. The goal is to find consistency relations that translate to suitable learning objectives. Particularly, we wish to improve the network prediction between the real image pair (I, J). We therefore first explore the possible consistency constraints that can be derived from the graph shown in Fig. 3c. For simplicity, we do not explicitly denote visible or valid regions of the stated consistency relations. They should be interpreted as an equality constraint for all pixel locations x where both sides represent a valid, non-occluded mapping or flow.

Pair-wise constraints: We first consider the consistency constraints recovered from pairs of images, as visualized in Fig. 4e. From the pair (I, J), and analogously (J, I'), we recover the standard forward-backward consistency constraint \( I = M_{J\rightarrow I} \circ M_{I\rightarrow J} \), from which we derive (2). Furthermore, from the pair (I', I) we can derive the warp-supervision constraint (3) \( F_{I'\rightarrow I} = W \).\(^1\)

Bipath constraints: The novel consistency relations stem from constraints that involve all three images in the triplet (I, I', J). These appear in two distinct types, here termed bipath and cycle constraints, respectively. We first consider the former, which have the form \( M_{I'\rightarrow I} = M_{3\rightarrow 2} \circ M_{1\rightarrow 3} \). That is, we obtain the same mapping by either proceeding directly from image 1 to 2 or by taking the detour through image 3. We thus compute the same mapping by two different paths: 1 \( \rightarrow \) 2 and 1 \( \rightarrow \) 3 \( \rightarrow \) 2, from which we derive the name of the constraint. The images 1, 2, and 3 represent any enumeration of the triplet (I, I', J) that respects the direction I' \( \rightarrow \) I, specified by the known warp W. There thus exist three different bipath constraints, detailed in Sec. 3.4.

Cycle constraints: The last category of constraints is formulated by starting from any of the three images in Fig. 4d and composing the mappings in a full cycle. Since we return to the starting image, the resulting composition is equal to the identity map. This is expressed in a general form as \( I = M_{3\rightarrow 1} \circ M_{2\rightarrow 3} \circ M_{1\rightarrow 2} \), where we have proceeded in the cycle 1 \( \rightarrow \) 2 \( \rightarrow \) 3 \( \rightarrow \) 1. Again constraining the direction I' \( \rightarrow \) I, we obtain three different constraints, as visualized in Fig. 4d. Compared to the bipath constraints, the cycle variants require two consecutive warping operations, stemming from the additional mapping composition. Each warp reduces the valid region and introduces interpolation noise and artifacts in practice. Constraints involving fewer warping operations are thus desirable, which is an advantage of the class of bipath constraints. In the next parts, we therefore focus on the later class to find a suitable unsupervised objective for dense correspondence estimation.

3.4. Bipath constraints

As mentioned in the previous section, there exist three different bipath constraints that preserve the direction of the known warp W. These are stated in terms of mappings as,

\[
M_{I'\rightarrow I} = M_{I\rightarrow I'} \circ M_{W} \quad (4a)
\]
\[
M_{J\rightarrow I} = M_{W} \circ M_{J\rightarrow I'} \quad (4b)
\]
\[
M_{W} = M_{J\rightarrow I'} \circ M_{I\rightarrow J} \quad (4c)
\]

From (4), we can derive the equivalent flow constraints as,

\[
F_{I'\rightarrow I} = W + \Phi_{W}(F_{I'\rightarrow I}) \quad (5a)
\]
\[
F_{J\rightarrow I} = F_{J\rightarrow I'} + \Phi_{F_{J\rightarrow I'}}(W) \quad (5b)
\]
\[
W = F_{I'\rightarrow J} + \Phi_{F_{I'\rightarrow J}}(F_{J\rightarrow I}) \quad (5c)
\]

Each constraint is visualized in Fig. 4a, b and c respectively. At first glance, any one of the constraints in (5) could be used as an unsupervised loss by minimizing the error between the left and right hand side. However, by separately

---

\(^{1}\)While \( I = M_{I\rightarrow I'} \circ M_{W} \) and \( I = M_{W} \circ M_{I\rightarrow I'} \) are also possible, they offer no advantage over standard warp-supervision: \( M_{I'\rightarrow I} = M_{W} \).
analyzing each constraint in (4)-(5), we will find them to have radically different properties which impact their suitability as an unsupervised learning objective.

**I’-J-bipath:** The constraint (4a), (5a) is derived from the two possible paths from I’ to J (Fig. 4a). While not obvious from (5a), it can be directly verified from (4a) that this constraint has a degenerate trivial solution. In fact, (4a) is satisfied for any W by simply mapping all inputs x to a constant pixel location c ∈ R² as ŜI’→J(x) = ŜI→J(x) = c. In order to satisfy this constraint, the network can thus learn to predict the same flow Ř = c – I for any input image pair.

**JI-bipath:** From the paths J → I in Fig. 4b, we achieve the constraint (4b), (5b). The resulting unsupervised loss is formulated as

\[ L_{J→I} = \| \hat{F}_{J→I'} + \Phi_{\hat{F}_{J→I'}}(W) - \hat{F}_{J→I} \|. \]  

Unfortunately, this objective suffers from another theoretical disadvantage. Due to the cancellation effect between the estimated flow terms ŒJ→I’ and ŒJ→I, the objective (6) is insensitive to a constant bias in the prediction. Specifically, if a small constant bias b ∈ R² is added to all flow predictions in (6), it can be shown that the increase in the loss (6) is approximately bounded by \( \| \Phi_{\hat{F}_{J→I'}}(DWb) \| \). Here, the bias error b is scaled with the Jacobian DW of the warp W. Since a smooth and invertible warp W implies a generally small Jacobian DW, the change in the loss will be negligible. The resulting insensitivity of (6) to a prediction bias is further confirmed empirically by our experiments. We provide derivations in the suppl. A.1. To further understand and compare the bipath constraints (5), it is also useful to consider the limiting case of reducing the magnitude of the warps \( ||W|| \) → 0. By setting W = 0 it can be observed that (6) becomes zero, i.e. no learning signal remains.

**W-bipath:** The third bipath constraint (4c), (5c) is derived from the paths I’ → I, which is determined by W (Fig. 4c). It leads to the W-bipath consistency loss,

\[ L_W = \| \hat{F}_{I'→J} + \Phi_{\hat{F}_{I'→J}}(\hat{F}_{J→I}) - W \|. \]  

We first analyze the limiting case \( ||W|| \) → 0 by setting W = 0, which leads to standard forward-backward consistency (2) since I’ = I. The W-bipath is thus a direct generalization of the latter constraint. Importantly, by randomly sampling non-zero warps W, degenerate solutions are avoided, effectively solving the one fatal issue of forward-backward consistency objectives. In addition to avoiding degenerate solutions, W-bipath does not experience cancellation of prediction bias, as in (6). Furthermore, compared to warp-supervision (3), it enables to directly learn the flow prediction ŒJ→I between the real pair (I, J). In the next section, we therefore develop our final unsupervised objective based on the W-bipath consistency.

### 3.5. Warp consistency loss

In this section, we develop our warp consistency loss, an unsupervised learning objective for dense correspondence estimation, using the consistency constraints derived in Sec. 3.3 and 3.4. Specifically, following the observations in Sec. 3.4, we base our loss on the W-bipath constraint.

**W’-bipath consistency term:** To formulate an objective based on the W’-bipath consistency constraint (5c), we further integrate a visibility mask V ∈ [0, 1]ₜₓₗ. The mask V takes a value V(x) = 1 for any pixel x where both sides of (4c), (5c) represent a valid, non-occluded mapping, and V(x) = 0 otherwise. The loss (7) is then extended as,

\[ L_{W\text{-vis}} = \| \hat{V} \cdot (\hat{F}_{I'→J} + \Phi_{\hat{F}_{I'→J}}(\hat{F}_{J→I}) - W) \|. \]  

Since we do not know the true V, we replace it with an estimate \( \hat{V} \). While there are different techniques for estimating visibility masks [16, 31, 53], we base our strategy on [31]. Specifically, we compute our visibility mask as,

\[ \hat{V} = 1 \left[ \| \hat{F}_{I'→J} + \Phi_{\hat{F}_{I'→J}}(\hat{F}_{J→I}) - W \|_2^2 < \alpha_2 \right] + \alpha_1 \left( \| \hat{F}_{I'→J} \|_2^2 + \| \Phi_{\hat{F}_{I'→J}}(\hat{F}_{J→I}) \|_2^2 + \| W \|_2^2 \right). \]

\[ \text{Here,} \ 1[\cdot] \text{ takes the value} \ 1 \text{ or} \ 0 \text{ if the input statement is true or false, respectively. The scalars} \ \alpha_1 \text{ and} \ \alpha_2 \text{ are hyper-parameters controlling the sensitivity of the mask estimation.} \]

For the warp operation \( \Phi_{\hat{F}_{I'→J}}(\hat{F}_{J→I}) \), we generally found it beneficial not to back-propagate gradients through the flow \( \hat{F}_{I'→J} \) used for warping. We believe that this better encourages the network to directly adjust the flow \( \hat{F}_{J→I} \), rather than ‘move’ the flow vectors using the warp \( \Phi_{\hat{F}_{I'→J}} \).

**Warp-supervision term:** In addition to our W’-bipath objective (8), we use the warp-supervision (3), found as a pairwise constraint in our consistency graph (Fig. 4e). Benefiting from the strong and direct supervision provided by the synthetic flow W, the warp-supervision term increases convergence speed and helps in driving the network towards higher accuracy. Further, by the direct regression loss against the flow W, which is smooth by construction, it also acts as a smoothness constraint. On the other hand, through the W’-bipath loss (8), the network learns the realistic motion patterns and appearance changes present between real images (I, J). As a result, both loss terms are mutually beneficial. From a practical perspective, the warp-supervision loss can be integrated at a low computational and memory cost, since the backbone feature extraction for the three images I, I’, J can be shared between the two loss terms.

**Adaptive loss balancing:** Our final unsupervised objective combines the losses (8) and (3) as \( L = L_{W\text{-vis}} + \lambda L_{\text{warp}} \). This raises the question of how to set the trade-off \( \lambda \). In-
3.6. Sampling warps W

The key element of our warp consistency objective is the sampled warp W. During training, we randomly sample it from a distribution W ∼ pW, which we need to design. As discussed in Sec. 3.4, the W-bipath loss (8) approaches the forward-backward consistency loss (2) when the magnitude of the warps decreases ||W|| → 0. Exclusively sampling too small warps W ≈ 0 therefore risks biasing the prediction towards zero. On the other hand, too large warps would render the estimation of $\hat{F}_{1 \rightarrow 2}$ challenging and introduce unnecessary invalid image regions. As a rough guide, the distribution pW should yield warps of similar magnitude as the real transformations ||F_{1 \rightarrow 2}||, thus giving similar impact to all three terms in (8). Fortunately, as analyzed in the supplementary Sec. G, our approach is not sensitive to these settings as long as they are within reasonable bounds.

We construct W by sampling homography, Thin-plate Spline (TPS) and affine-TPS transformations randomly, following a procedure similar to previous approaches using warp-supervision [36]. (i) Homographies are constructed by randomly translating the four image corner locations. The magnitudes of the translations are chosen independently through Gaussian or uniform sampling, with standard-deviation or range equal to $\sigma_H$. (ii) For TPS, we randomly jitter a $3 \times 3$ grid of control points by independently translating each point. We use the same standard deviation or range $\sigma_H$ as for our homographies. (iii) To generate larger scale and rotation changes, we also compose affine and TPS. We first sample affine transformations by selecting scale, rotation, translation and shearing parameters according to a Gaussian or uniform sampling. The TPS transform is then sampled as explained above and the final synthetic flow W is a composition of both flows.

To make the warps W harder, we optionally also compose the flow obtained from (i), (ii) and (iii) with randomly sampled elastic transforms. Specifically, we generate an elastic deformation motion field, as described in [43] and apply it in multiple regions selected randomly. Detailed settings are provided in the supplementary Sec. C, D and E.

4. Experiments

We evaluate our unsupervised learning approach for three dense matching networks and two tasks, namely GLU-Net [50] and RANSAC-Flow [41] for geometric matching, and SemanticGLU-Net [50] for semantic matching. We extensively analyze our method and compare it to earlier unsupervised objectives, defining a new state-of-the-art on multiple datasets. Further results, analysis, visualizations and implementation details are provided in the supplementary.

4.1. Method analysis

We first perform a comprehensive analysis of our approach. We adopt GLU-Net [50] as our base architecture. It is a 4-level pyramidal network operating at two image resolutions to estimate dense flow fields.

Experimental set-up for GLU-Net: We slightly simplify the GLU-Net [50] architecture by replacing the dense decoder connections with standard residual blocks, which drastically reduces the number of network parameters with negligible impact on performance. As in [50], the feature extraction network is set to a VGG-16 [2] with ImageNet pre-trained weights. We train the rest of the architecture from scratch in two stages. We first train GLU-Net using our unsupervised objective, described in Sec. 3.5, but without the visibility mask $\hat{V}$. As a second stage, we add the visibility mask and employ stronger warps W, with elastic transforms. For both stages, we use the training split of the MegaDepth dataset [23], which comprises diverse internet images of 196 different world monuments.

Datasets and metrics: We evaluate on standard datasets with sparse ground-truth, namely RobotCar [29, 20] and MegaDepth [23]. For the latter, we use the test split of [41], which consists of 19 scenes not seen during training. Images in Robotcar depict outdoor road scenes and are particularly challenging due to their many textureless regions. MegaDepth images show extreme viewpoint and appearance variations. In line with [41], we use the Percentage of Correct Keypoints at a given pixel threshold T (PCK-T) as the evaluation metric (%). We also employ the 59 sequences of the homography dataset HPatches [1]. We evaluate with the Average End-Point-Error (AEPE) and PCK.

Warp consistency graph losses: In Tab. 1 we empirically compare the constraints extracted from our warp consistency graph (Sec. 3.3). All networks are trained with only the first stage, on the same synthetic transformations W. Since we observed it to give a general improvement, we stop gradients through the flow used for warping (but not
the flow that is warped). The $I'J$-bipath (II) and $JI$-bipath (III) losses lead to a degenerate solution and a large predicted bias respectively, which explains the very poor performance of the networks. The cycle loss ($V$) obtains much better results but does not reach the performance of the $W$-bipath constraint (IV). We only show the cycle starting from $I'$ here ($V$), since it performs best among all cycle losses (see suppl. A.3). While the warp-supervision loss ($I$) results in a better accuracy on all datasets (PCK-1 and PCK-5 for HPatches), it is significantly less robust to large view-point changes than the $W$-bipath objective (IV), as evidenced by results in PCK-10 and AEPE. These two losses have complementary behaviors and combining them (VIII) leads to a significant gain in both accuracy and robustness. Combining the warp-supervision loss ($I$) with $I'J$-bipath (II) in (VI) or with $JI$-bipath (III) in (VII) instead results in drastically lower performance than (VIII). The cycle loss ($V$) with the warp-supervision ($I$) in (IX) is also slightly worse.

**Ablation study:** In Tab. 2 we analyze the key components of our approach. We first show the importance of not back-propagating gradients in the warp operation. Adding the warp-supervision objective with constant weights of $\lambda = 1$ increases both the network's accuracy and robustness for all datasets. Further using adaptive loss balancing (Sec. 3.5) provides a significant improvement in accuracy (PCK-1) for MegaDepth with only minor loss on other thresholds. Including our visibility mask $\hat{V}$ in the second stage training drastically improves all metrics for all datasets. Finally, further sampling harder transformations results in better accuracy, particularly for PCK-1 on MegaDepth. We therefore use this as our standard setting in the following experiments, where we denote it as WarpC.

**Comparison to alternative losses:** Finally, in Tab. 3 we compare and combine our proposed objective with alternative losses. The census loss [31] (I), popular in optical flow, does not have sufficient invariance to appearance changes and thus leads to poor results on geometric matching datasets. The SSIM loss [54] (II) is more robust to the changes than the native losses. The census loss [31] (I), popular in optical matching methods. Further combining SSIM with the forward-backward consistency loss (III) leads to a small improvement. Compared to SSIM (III) on MegaDepth, our WarpC approach (VI) achieves superior PCK-5 (+7.8%) and PCK-10 (+10.2%) at the cost of a slight reduction in sub-pixel accuracy. Furthermore, our approach demonstrates superior generalization capabilities by outperforming all other alternatives on the RobotCar and HPatches datasets. For completeness, we also evaluate the combination (VII) of our loss with the photometric SSIM loss. This leads to improved PCK-1 on MegaDepth but degrades other metrics compared to WarpC (VI). Nevertheless, adding WarpC significantly improves upon SSIM (II) for all thresholds and datasets. Moreover, combining the warp-supervision (IV) with the forward-backward loss in (V) leads to an improvement compared to (IV). It is however significantly worse than combining the warp-supervision with our $W$-bipath loss in (VI), which can be seen as a generalization of the forward-backward loss. Finally, we compare with using the sparse ground-truth supervision provided by SfM reconstruction of the MegaDepth training images. Interestingly, training the dense prediction network from scratch with solely sparse annotations (VIII) leads to inferior performance compared to our unsupervised objective (VI). Lastly, we fine-tune (IX) our proposed network (VI) with sparse annotations. While this leads to a moderate gain on MegaDepth, it comes at the cost of worse generalization properties on RobotCar and HPatches.

### 4.2. Geometric matching

Here, we train the recent GLU-Net [50] and RANSAC-Flow [41] architectures with our unsupervised learning approach and compare them against state-of-the-art dense geometric matching methods.

**Experimental setup for GLU-Net:** We follow the training procedure explained in Sec. 4.1 and refer to the resulting model as WarpC-GLU-Net. The original GLU-Net [50] is trained using solely the warp-supervision (3) on a different training set. For fair comparison, we also report results of our altered GLU-Net architecture when trained on MegaDepth with our warp distribution. This corresponds to setting (IV) in Tab. 3, which we here call GLU-Net*. 

**Experimental setup for RANSAC-Flow:** We addi-
4.3. Semantic matching

Finally, we evaluate our approach for the task of semantic matching by training SemanticGLU-Net [50], a version of GLU-Net specifically designed for semantic images, which includes multi-resolution features and NC-Net [39].

**Experimental set-up:** Following [37, 3], we only fine-tune a pre-trained network on semantic correspondence data.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Features</th>
<th>TSS [46]</th>
<th>PF-Pascal [9]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNNGeo [37]</td>
<td>ResNet-101</td>
<td>90.1</td>
<td>76.4</td>
</tr>
<tr>
<td>WeakAlign [37]</td>
<td>ResNet-101</td>
<td>90.3</td>
<td>76.4</td>
</tr>
<tr>
<td>RTNs [13]</td>
<td>ResNet-101</td>
<td>90.1</td>
<td>78.2</td>
</tr>
<tr>
<td>PARN [17]</td>
<td>ResNet-101</td>
<td>89.5</td>
<td>75.9</td>
</tr>
<tr>
<td>NC-Net [39]</td>
<td>ResNet-101</td>
<td>94.5</td>
<td>81.4</td>
</tr>
<tr>
<td>DCCNet [12]</td>
<td>ResNet-101</td>
<td>93.5</td>
<td>82.6</td>
</tr>
<tr>
<td>DHPP [34]</td>
<td>ResNet-101</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SAM-Net [19]</td>
<td>VGG-19</td>
<td>96.1</td>
<td>82.2</td>
</tr>
<tr>
<td>GLU-Net [50]</td>
<td>VGG-16</td>
<td>92.2</td>
<td>73.3</td>
</tr>
<tr>
<td>GLU-Net-GOCor [49]</td>
<td>VGG-16</td>
<td>94.6</td>
<td>77.9</td>
</tr>
<tr>
<td>SemanticGLU-Net [50]</td>
<td>VGG-16</td>
<td>94.4</td>
<td>75.5</td>
</tr>
<tr>
<td>WarpC-SemanticGLU-Net</td>
<td>VGG-16</td>
<td>97.1</td>
<td>84.7</td>
</tr>
</tbody>
</table>

**Datasets and metrics:** We first evaluate on the test set of PF-Pascal [9]. In line with [10], we report the PCK with a pixel threshold equal to \( \alpha \cdot \max(h_q, w_q) \), where \( h_q \) and \( w_q \) are the dimensions of the query image and \( \alpha = (0.05, 0.1) \). To demonstrate generalization capabilities, we also validate our trained model on TSS [46], which provides dense flow field annotations for the foreground object in each pair. We report the PCK for \( \alpha = 0.05 \). We also provide results on PF-Willow [8] and SPair-71K [33] in suppl. K.3.

**Results:** Results are reported in Tab. 5. Our approach WarpC-SemanticGLU-Net sets a new state-of-the-art on TSS by obtaining a remarkable improvement compared to previous works. On the PF-Pascal dataset, our method ranks first for the small threshold \( \alpha = 0.05 \) with a substantial 2% increase compared to second best method. It obtains marginally lower PCK (0.6%) than DCCNet [12] for \( \alpha = 0.1 \), but the later approach employs a much deeper feature backbone, beneficial on semantic images. Nevertheless, our unsupervised fine-tuning provides 16% and 11.1% gain, for each threshold respectively, over the baseline, demonstrating that our objective effectively copes with the radical appearance changes encountered in the semantic matching task. A visual example is shown in Fig. 5 bottom.

5. Conclusion

We propose an unsupervised learning objective for dense correspondences, particularly suitable for scenarios with large changes in appearance and geometry. From a real image pair, we construct an image triplet and design a regression loss based on the flow-constraints existing between the triplet. When integrated into three recent dense correspondence networks, our approach outperforms state-of-the-art for multiple geometric and semantic matching datasets.

**Acknowledgements:** This work was supported by the ETH Zurich Fund (OK), a Huawei Gift, the ETH Future Computing Laboratory (EFLC) financed by a gift from Huawei Technologies, Amazon AWS, and an Nvidia GPU grant.
References


[45] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cns for optical flow using pyramid, warping, and


