Probabilistic Warp Consistency for Weakly-Supervised Semantic Correspondences
-
Supplementary Material

Prune Truong Martin Danelljan Fisher Yu Luc Van Gool
Computer Vision Lab, ETH Zurich, Switzerland
{prune.truong, martin.danelljan, vangool}@vision.ee.ethz.ch i@yf.io

In this supplementary material, we provide additional details about our approach, experiment settings and results. In Sec. A, we first give general implementation details, that apply to all networks we considered. We follow by explaining the triplet image creation and the sampling process of our synthetic warps M_W in Sec. B.

In Sec. C, we then focus on the training procedure to obtain PWarpC-SF-Net and PWarpC-SF-Net* in more depth. We continue by explaining the training details of PWarpC-NC-Net and PWarpC-NC-Net* in Sec. D. We subsequently provide training details for PWarpC-CATs and PWarpC-DHPF in respectively Sec. E and F. In all aforementioned sections, we provide information about the architecture, its original training strategy, our proposed training approach comprising the sampled transformations M_W and weighting of our losses, as well as implementation details. We also provide additional ablative experiments.

We then follow by analysing the effect of the strength of the sampled warps M_W in Sec. G. In Sec. H, we also provide results on all four benchmarks PF-Pascal [4], PF-Willow [3], SPair-71K [18] and TSS [25], when the networks are trained or finetuned on SPair-71K instead of PF-Pascal. Finally we present more detailed quantitative and qualitative results in Sec. I. In particular, we extensively explain the evaluation datasets and set-up. We also analyze our approach in terms of robustness to view-point, scale, truncation and occlusion. Finally, we further evaluate our approach on the Caltech-101 dataset [8].

A. General implementation details

In this section, we provide implementation details, which apply to all our PWarpC networks.

Creating of ground-truth probabilistic mapping P_W: Here, we describe how we obtain the ground-truth probabilisitc mapping P_W from the known mapping M_W. We first rescale the mapping M_W to the same resolution as the predicted probabilistic mapping $\widehat{P}$. We then convert the mapping into a ground-truth probabilistic mapping P_W, following this scheme. For each pixel position i' in I', we construct the ground-truth 2D conditional probability distribution $P_W(\cdot|i') \in \mathbb{R}^{h_{I'} \times w_{I'}}, \in [0,1]$ by assigning a one-hot or a smooth representation of $M(\mathbf{i}')$. In the latter case, following [11], we pick the four nearest neighbours of $M(\mathbf{i}')$ and set their probability according to distance. Then we apply 2-D Gaussian smoothing of size 3 on that probability map. We then vectorize the two dimensions of $P_W(\cdot|i')$, leading to our final known warp probabilistic mapping $P_W \in \mathbb{R}^{h_I w_I \times h_{I'} w_{I'}}$. We will specify which representation we used, as either one-hot or smooth, for each loss and each network.

Conversion of P to correspondence set: The output of the model is a probabilistic mapping, encoding the matching probabilities for all pairwise match relating an image pair. However, for various applications, such as image alignment or geometric transformation estimation, it is desirable to obtain a set of point-to-point image correspondences $M_{I \leftarrow J}$ between the two images. This can be achieved by either performing a hard or soft assignment. In the former case, the hard assignment in one direction is done by just taking the most likely match, the mode of the distribution as,

$$M_{I \leftarrow J}(\mathbf{j}) = \arg\max_i \; P_{I \leftarrow J}(i|j) \tag{1}$$

In the latter case, the soft assignment corresponds to soft-argmax. It computes correspondences $M_{I \leftarrow J}(\mathbf{j})$ for individual locations $\mathbf{j}$ of image J, as the expected position in I according to the conditional distribution $P_{I \leftarrow J}(.|j)$,

$$M_{I \leftarrow J}(\mathbf{j}) = \sum_i \; \mathbf{i} \cdot P_{I \leftarrow J}(i|j) \tag{2}$$

Training details: All networks are trained with PyTorch, on a single NVIDIA TITAN RTX GPU with 24 GiB of memory, within 48 hours, depending on the architecture.

B. Triplet creation and sampling of warps M_W

B.1. Triplet creation

Our introduced learning approach requires to construct an image triplet (I, I', J) from an original image pair (I, J), where all three images must have training dimensions $s \times s$. We follow a similar procedure than in [28], further described here. The original training image pairs (I, J) are first resized to a fixed size $s_r \times s_r$, larger than the desired training image size $s \times s$. We then sample a dense mapping M_W of the same dimension $s_r \times s_r$, and create I' by warping image I with M_W, as $I' = I \circ M_W$. Each of the images of the resulting image triplet (I, I', J) are then centrally cropped to the fixed training image size $s \times s$. The central cropping is necessary to remove most of the black areas in I' introduced from the warping operation with large sampled mappings M_W as well as possible warping artifacts arising at the image borders. We then additionally apply appearance transformations to all images of the triplet, such as brightness and contrast changes.

B.2. Sampling of warps M_W

A question raised by our proposed loss formulations (8)-(9) is how to sample the synthetic warps M_W. During training, we randomly sample it from a distribution $M_W \sim p_W$, which we need to design. Here, we also follow a similar procedure than in [28].

In particular, we construct M_W by sampling homography, Thin-plate Spline (TPS), or affine-TPS transformations with equal probability. The transformations parameters are then converted to dense mappings of dimension $s_r \times s_r$. Then, we optionally apply horizontal flipping to the each dense mapping with a probability p_{flip}.

Specifically, for homographies and TPS, the four image corners and a 3×3 grid of control points respectively, are randomly translated in both horizontal and vertical directions, according to a desired sampling scheme. The translated and original points are then used to compute the corresponding homography and TPS parameters. Finally, the transformations parameters are converted to dense mappings. For both transformation types, the magnitudes of the translations are sampled according to a uniform distribution with a range σ_H. Note that for the uniform distribution, the sampling range is actually $[-\sigma_H, \sigma_H]$, when it is centered at zero, or similarly $[1 - \sigma_H, 1 + \sigma_H]$ if centered at 1 for example. Importantly, the image points coordinates are previously normalized to be in the interval $[-1, 1]$. Therefore σ_H should be within $[0, 1]$.

For the affine transformations, all parameters, *i.e.* scale, translations, shearing and rotation angles, are sampled from a uniform distribution with range equal to τ, t, α and α respectively. The affine scale parameters are sampled within $[1 - \tau, 1 + \tau]$ with center at 1, while for all other parameters, the sampling interval is centered at zero.

B.3. List of Hyper-parameters

In summary, to construct our image triplet (I, I', J), the hyper-parameters are the following:

(i) s_r, the resizing image size, on which is applied M_W to obtain I' before cropping.

(ii) s, the training image size, which corresponds to the size of the training images after cropping.

(iii) σ_H, the range used for sampling the homography and TPS transformations.

(iv) τ, the range used for sampling the scaling parameter of the affine transformations.

(v) t, the range used for sampling the translation parameter of the affine transformations.

(vi) α, the range used for sampling the rotation angle of the affine transformations. It is also used as shearing angle.

(vii) σ_{tps}, the range used for sampling the TPS transformations, used for the Affine-TPS compositions.

(viii) The probability of horizontal flipping p_{flip}.

B.4. Hyper-parameters settings

Geometric transformations : For all our PWarpC networks, the mappings M_W are created by sampling homographies, TPS and Affine-TPS transformations with equal probability. For simplicity, we also use the same range for all three types of transformations. In particular, we use a uniform sampling scheme with a range equal to $[-\sigma_H, \sigma_H]$, where $\sigma_H = \sigma_{tps} = 0.4$. For the affine transformations, we also sample all parameters, *i.e.* scale, translation, shear and rotation angles, from uniform distributions with ranges respectively equal to $\tau = 0.45$, $t = 0.25$, and $\alpha = \pi/12$ for both angles. We use these parameters when training on either PF-Pascal [4] or SPair-71K [18].

Probability of horizontal flipping: When training on PF-Pascal, we set the probability of horizontal flipping to $p_{flip} = 5\%$ for all our PWarpC networks, except for PWarpC-NC-Net and PWarpC-NC-Net*, for which we use $p_{flip} = 30\%$. For training on SPair-71K, we increase this value to $p_{flip} = 15\%$ for all our PWarpC networks, except for PWarpC-NC-Net and PWarpC-NC-Net*, for which we keep $p_{flip} = 30\%$.

Appearance transformations: For all experiments and networks, we apply the same appearance transformations to the image triplet (I, I', J). Specifically, we convert each image to gray-scale with a probability of 0.2. We then apply color transformations, by adjusting contrast, saturation, brightness, and hue. The color transformations are larger for the synthetic image I' then for the real images (I, J). For the synthetic image I', we additionally randomly invert the RGB channels. Finally, on all images of the triplet, we further use a Gaussian blur with a kernel between 3 and 7,

and a standard deviation sampled within $[0.2, 2.0]$, applied with probability of 0.2.

C. PWarpC-SF-Net and PWarpC-SF-Net*

We first provide details about the SF-Net [9] architecture. We also briefly review the training strategy of the original work. We then extensively explain our training approach and the corresponding implementation details, for both our weakly and strongly-supervised approaches, PWarpC-SF-Net and PWarpC-SF-Net* respectively. Finally, we provide additional method analysis for this architecture.

C.1. Details about SF-Net

Architecture: SF-Net is based on a pre-trained ResNet-101 feature backbone, on which are added convolutional adaptation layers at two levels. The predicted feature maps are then used to construct two cost volumes, at two resolutions. After upsampling the coarsest one to the same resolution, the two cost volumes are combined with a point-wise multiplication. While the resulting cost volume is the actual output of the network, it is converted to a flow field through a kernel sotf-argmax operation. Specifically, a fixed Gaussian kernel is applied on the raw cost volume scores to post-process them, before applying SoftMax to convert the cost volume to a probabilistic mapping. From there, the soft-argmax operation transposes it to a mapping.

For our PWarpC approaches, we do not use the Gaussian kernel to post-process the predicted matching scores. We simply convert the predicted cost volume into a probabilistic mapping through a SoftMax operation, following eq. (4) of the main paper. Also note that only the adaptation layers are trained.

Training strategy in original work: The original work employs ground-truth foreground object masks as supervision. From single images associated with their segmentation masks, they create image pairs by applying random transformations to both the original images and segmentation masks. Subsequently, they train the network with a combination of multiple losses. In particular, they enforce the forward-backward consistency of the predicted flow, associated with a smoothness objective acting directly on the predicted flow. These losses are further combined with an objective enforcing the consistency of the warped foreground mask of one image with the ground-truth segmentation mask of the other image.

C.2. PWarpC-SF-Net and PWarpC-SF-Net*: our training strategy

Warps W sampling: For the weakly-supervised version, we resize the image pairs (I, J) to $s_r \times s_r = 340 \times 340$, sample a dense mapping M_W of the same dimension and create I'. Each of the images of the resulting image triplet (I, I', J) is then centrally cropped to $s \times s = 320 \times 320$.

For the strongly-supervised version, we apply the transformations on images of the same size than the crop, *i.e.* $s_r \times s_r = s \times s = 320 \times 320$. This is to avoid cropping keypoint annotations.

When training on PF-Pascal, we apply 5% of horizontal flipping to sample the random mappings M_W, while it is increased to 15% when training on SPair-71K.

Weighting and details on the losses : We found it beneficial to define the known probabilistic mapping P_W with a one-hot representation for our PW-bipath loss (eq.9 of m.p.), while using a smooth representation instead for the PWarp-supervision (eq.8 of m.p.) loss and the keypoint L_{kp} objective in (eq.12 of m.p.). Each representation is described in Sec. A.

For the weakly-supervision version PWarpC-SF-Net, the weights in eq. 11 of m.p. are set to $\lambda_{\text{P-warp-sup}} = L_{\text{vis-PW-bi}}/L_{\text{P-warp-sup}}$ and $\lambda_{\text{PNeg}} = 1$.

For the strongly-supervised version, PWarpC-SF-Net*, we use the same weight $\lambda_{\text{P-warp-sup}} = L_{\text{vis-PW-bi}}/L_{\text{P-warp-sup}}$. We additionally set $\lambda_{\text{kp}} = (L_{\text{P-warp-sup}} + L_{\text{vis-PW-bi}})/L_{\text{kp}}$, which ensure that our probabilistic losses amount for the same than the keypoint loss L_{kp}. Moreover, the keypoint loss L_{kp} is set as the cross-entropy loss, for both PWarpC-SF-Net* and its baseline SF-Net*.

Implementation details: For our weakly-supervised PWarpC-SF-Net, we set the initial learnable parameter z, corresponding to the unmatched state $\emptyset$ for our occlusion modeling, at $z = 0$.

For both the weakly and strongly-supervised approaches, the SoftMax temperature, corresponding to equation (4) of the main paper, is set to $\tau = 1.0/50.0$, the same than originally used in the baseline for soft-argmax. The hyper-parameter used in the estimation of our visibility mask $\widehat{V}$ (eq. 9 of the main paper) is set to $\gamma = 0.7$ and to $\gamma = 0.2$ when trained on PF-Pascal or SPair-71K respectively. This is because in SPair-71K, the objects are generally much smaller than in PF-Pascal.

For training, we use similar training parameters as in baseline SF-Net [9]. We train with a batch size of 16 for maximum 100 epochs. The learning rate is set to 3.10^{-5} and halved after 50. We optionally finetune the networks on SPair-71K for an additional 20 epochs, with an initial learning rate of 1.10^{-5}, halved after 10 epochs. The networks are trained using Adam optimizer [7] with weight decay set to zero.

C.3. Additional analysis

Here, we first analyse the effect of the kernel applied in the original SF-Net baseline [9] before converting the predicted cost volume to a probabilistic mapping representation. We also provide the ablation study of our strongly-

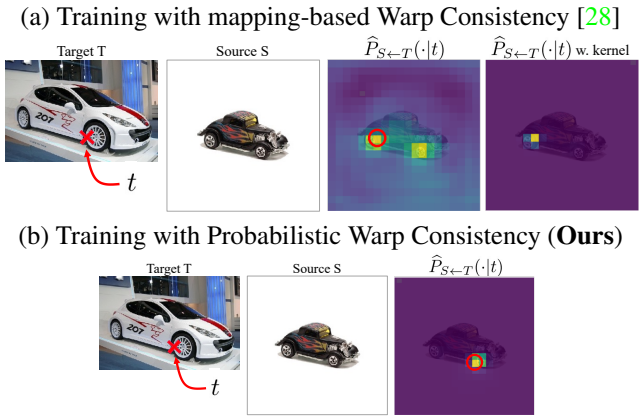


Figure 1. In (a), SF-Net is trained using the mapping-based Warp Consistency approach [28], after converting the cost volume to a mapping through soft-argmax [9]. It predicts ambiguous matching scores, struggling to differentiate between the car wheels. After applying the kernel, the mode of the distribution corresponds to the wrong wheel. Also note that the kernel is extremely important in that case to post-process the multi-hypothesis distribution. Our probabilistic approach (b) instead directly predicts a Dirac-like distribution, whose mode is correct.

supervised PWarpC-SF-Net*. Note that the ablation study of the weakly-supervised SF-Net is provided in Tab. 2 of the main paper. Finally, we show the impact of different losses on negative image pairs, *i.e.* depicting different object classes.

Effect of kernel: Baseline SF-Net relies on a kernel soft-argmax strategy to convert the predicted cost volume to a mapping. In particular, the kernel is applied on the cost volume before applying SoftMax (eq.4 of m.p.), which transposes it to a probabilistic mapping. From there, soft-argmax is used to obtain a mapping. Nevertheless, we observe that this kernel is extremely important in order to post-process the matching scores. This is shown in Fig. 1. In contrast, our approach Probabilistic Warp Consistency, which directly acts on the predicted dense matching scores, produces clean, Dirac-like conditional distributions, without relying on any post-processing operations.

Ablation study for strongly-supervised PWarpC-SF-Net*: In Tab. 1, we analyse key components of our approach PWarpC-SF-Net*. From the strongly-supervised

	PF-Pascal		PF-Willow		Spair-71K	TSS
	α_{img}		α_{bbox}		α_{bbox}	α_{img}
Methods	0.05	0.10	0.05	0.10	0.10	0.05
SF-Net*	**78.7**	**92.9**	43.2	72.5	27.9	73.8
+ Vis-aware PW-bipath (eq. 9 of m.p.)	77.1	91.6	**47.8**	**77.9**	31.1	80.3
+ PWarp-supervision (eq. 8 of m.p.)	78.3	92.2	47.5	77.7	**32.5**	**84.2**

Table 1. Ablation study for strongly-supervised PWarpC-SF-Net*. We incrementally add each component. We measure the PCK on the PF-Pascal [4], PF-Willow [3], SPair-71K [18] and TSS [25] datasets. The evaluation results are computed using ground-truth annotations at original resolution.

		PF-Pascal		PF-Willow		Spair-71k	TSS
		α_{img}		α_{bbox}		α_{bbox}	α_{img}
	Methods	0.05	0.10	0.05	0.10	0.10	0.05
I	SF-Net baseline (soft-argmax)	59.0	84.0	46.3	74.0	24.0	75.8
II	SF-Net baseline (argmax)	60.3	81.3	43.7	71.0	26.9	74.1
III	Vis-PW-bipath + PWarp-sup	63.0	84.9	47.0	76.9	30.7	83.5
IV	(III) + PNeg (eq. 10 of m.p.) (**PWarpC-SF-Net**)	**65.6**	**87.9**	**47.3**	**78.2**	**33.8**	**84.1**
V	(III) + Max-score [22]	63.7	81.2	44.6	71.6	31.8	77.3
VI	(III) + Min-entropy [19]	59.4	76.7	41.8	67.9	28.8	73.2

Table 2. Comparison of different losses applied on negative image pairs, *i.e.* depicting different object classes, when associated with our introduced PW-bipath and PWarp-supervised losses on positive image pairs. We use SF-Net as baseline network. The evaluation results are computed using the annotations at original resolution.

baseline SF-Net*, adding our probabilistic PW-bipath objective leads to a significant improvement on the PF-Willow, SPair-71K and TSS datasets. Further including our PWarp-supervision objective results in additional gains on SPair-71K and TSS.

Comparisons to alternative negative losses: In Tab. 2, we compare combining our PW-bipath and PWarp-supervision objectives on image pairs of the same label, with different losses on images pairs showing different object classes, *i.e.* on negative image pairs. In the version denoted as (IV), we introduce our explicit occlusion modeling (Sec. 4.3 of the main paper), trained with our probabilistic negative loss L_{PNeg}. In (V) and (VI), we instead combine our probabilistic objectives on the positive image pairs (III), with an additional objective, minimizing the max scores or the negative entropy of the cost volume respectively. While it brings a small improvement with respect to version (III), the resulting network performances in (V) and (VI) are far lower than when trained with our final combination (eq. 11 of m.p.), which corresponds to version (IV).

D. PWarpC-NC-Net and PWarpc-NC-Net*

In this section, we first provide details about the NC-Net architecture. We also briefly review the training strategy of the original work. We then extensively explain our training approach and the corresponding implementation details, for both our weakly and strongly-supervised approaches, PWarpC-NC-Net and PWarpC-NC-Net* respectively. Finally, we extensively ablate our approach for this architecture.

D.1. Details about NC-Net

Architecture: In [22], Rocco *et al.* introduce a learnable consensus network, applied on the 4D cost volume constructed between a pair of feature maps. Specifically, they process the cost volume with multiple 4D convolutional layers, to establish a strong locality prior on the relationships between the matches. The cost volume before and after applying the 4D convolutions is also processed with a soft mutual nearest neighbor filtering.

Training strategy in original work: The baseline NC-Net is trained with a weakly-supervised strategy, using image-level class labels as only supervision. Their proposed objective maximizes the mean matching scores over all hard assigned matches from the predicted cost volume constructed between images pairs of the same class, while minimizing the same quantity for image pairs of different classes. By retraining the NC-Net architecture with this strategy, we nevertheless found the training process to be quite unstable, multiple training runs leading to substantially different performance.

D.2. PWarpC-NC-Net and PWarpC-NC-Net*: our training strategy

Warps W sampling: For the weakly-supervised version, we resize the image pairs (I, J) to $s_r \times s_r = 430 \times 430$, sample a dense mapping M_W of the same dimension and create I'. Each of the images of the resulting image triplet (I, I', J) is then centrally cropped to $s \times s = 400 \times 400$.

For the strongly-supervised version, we apply the transformations on images of the same size than the crop, *i.e.* $s_r \times s_r = s\times = 400 \times 400$. This is to avoid cropping keypoint annotations.

As for the random mapping M_W, we apply 30% of horizontal flipping. We found increasing the percentage of horizontal flipping for our PWarpC-NC-Net and PWarpC-NC-Net* networks to be beneficial compared to the other networks, in order to help stabilize the learning.

Weighting and details on the losses : For all losses, we use a smooth representation for the known probabilistic mapping P_W (see Sec. A).

In general, we found the PWarp-supervision objective (eq.8 of m.p.) to be slightly harmful for the PWarpC-NC-Net networks, and therefore did not include it in our final weakly and strongly-supervised formulations. This is particularly the case when finetuning the features, which is the setting we used for our final PWarpC-NC-Net and PWarpC-NC-Net*. This is likely due to the network 'overfitting' to the synthetic image pairs and transformations involved in the PWarp-supervision loss, at the expense of the real images considered in the PW-bipath (eq. 9 of m.p.) and PNeg (eq. 10 of m.p.) objectives.

As a result, for the weakly-supervision version PWarpC-NC-Net, the weights in eq. 11 of m.p. are set to $\lambda_{\text{P-warp-sup}} = 0$ and $\lambda_{\text{PNeg}} = 1$. For the strongly-supervised version, PWarpC-NC-Net*, we use the same weight $\lambda_{\text{P-warp-sup}} = 0$. We additionally set $\lambda_{\text{kp}} = (L_{\text{P-warp-sup}} + L_{\text{vis-PW-bi}})/L_{\text{kp}}$, which ensure that our probabilistic losses amount for the same than the keypoint loss L_{kp}. Moreover, the keypoint loss L_{kp} is set as the cross-entropy loss, for both PWarpC-NC-Net* and its baseline NC-Net*.

Implementation details: For PWarpC-NC-Net, we set the initial learnable parameter z, corresponding to the unmatched state $\emptyset$ for our occlusion modeling at $z = 10$. This is to ensure that it is in the same range than the cost volume, at initialization.

The SoftMax temperature, corresponding to equation 4 of the main paper, is set to $\tau = 1.0$, the same than originally used in the baseline loss. The hyper-parameter used in the estimation of our visibility mask $\hat{V}$ (eq. 9 of the main paper) is set to $\gamma = 0.2$. Indeed, for NC-Net, we found that using a more restrictive threshold, as compared to the other networks which use $\gamma = 0.7$ (when trained on PF-Pascal), is beneficial to stabilize the training. It offers a better guarantee that the PW-bipath loss (eq. 9 of m.p.) is *only* applied in common visible object regions between the triplet.

Similarly to baseline NC-Net [22], we train in two stages. In the first stage, we only train the consensus neighborhood network while keeping the ResNet-101 feature backbone extractor fixed. We further finetune the last layer of the feature backbone as well as the consensus neighborhood network in a second stage. These two stages are used to train on PF-Pascal [4], our final PWarpC-NC-Net and PWarpC-NC-Net* approaches, as well as strongly-supervised baseline NC-Net*.

For training, we use similar training parameters as in baseline NC-Net. We train with a batch size of 16, which is reduced to 8 when the last layer of the backbone feature is finetuned. During the first training stage on PF-Pascal, we train for a maximum of 30 epochs with a learning rate set to a constant of $5 \cdot 10^{-4}$. During the second training stage on

		PF-Pascal		**PF-Willow**		**Spair-71k**	**TSS**
		α_{img}		α_{bbox}		α_{bbox}	α_{img}
	Methods	0.05	0.10	0.05	0.10	0.10	0.05
I	NCNet baseline (Max-score) [22]	60.5	82.3	44.0	72.7	28.8	77.7
II	PW-bipath	diverged					
III	+ Visibility mask	**64.7**	83.8	**45.4**	75.9	32.8	82.7
IV	+ PWarp-supervision	61.7	79.2	45.1	73.8	35.6	85.4
III	PW-bipath Visibility mask	**64.7**	83.8	**45.4**	75.9	32.8	82.7
V	+ PNeg	62.0	82.2	**45.4**	**76.2**	33.2	87.9
VI	+ ft features (**PWarpC-NC-Net**)	64.2	**84.4**	45.0	75.9	35.3	**89.2**
VII	ft features from scratch	63.7	82.9	44.9	76.1	**35.7**	87.4
VI	**PWarpC-NC-Net**	**64.2**	**84.4**	**45.0**	**75.9**	**35.3**	**89.2**
I	Max-score (NC-Net baseline)	60.5	82.3	44.0	72.7	28.8	77.7
VIII	Min-entropy [19]	55.6	79.2	42.0	72.3	25.4	78.4
IX	Warp Consistency [28]	59.1	75.0	44.6	70.1	35.0	87.0
III	PW-bipath Visibility mask	64.7	**83.8**	**45.4**	75.9	32.8	82.7
V	(III) + PNeg (Ours)	62.0	82.2	**45.4**	**76.2**	**33.2**	**87.9**
X	(III) + Max-score	**62.9**	82.1	**45.4**	74.2	31.3	79.0
XI	(III) + Min-entropy	60.8	78.5	44.8	71.4	31.5	78.6

Table 3. In the top part, we conduct an ablation study for PWarpC-NC-Net. There, we incrementally add each component. In the middle part, we then compare our Probabilistic Warp Consistency objective to alternative weakly-supervised losses. In the bottom part, we compare the impact of combining different losses on non-matching pairs with our PW-bipath objective, applied on image pairs of the same class. We measure the PCK on the PF-Pascal [4], PF-Willow [3], SPair-71K [18] and TSS [25] datasets. The evaluation results are computed using ground-truth annotations at original resolution.

PF-Pascal, the learning rate is reduced to $1 \cdot 10^{-4}$ and the network trained for an additional 30 epochs.

We optionally further finetune the networks on SPair-71K [18] for 10 epochs, with the same learning rate equal to $1 \cdot 10^{-4}$. Note that in this setting, the last layer of the feature backbone is also finetuned. The networks are trained using Adam optimizer [7] with weight decay set to zero.

D.3. PWarpC-NC-Net: ablation study and comparison to previous works

Similarly to Sec. 5.4 of the main paper for PWarpC-SF-Net, we here provide a detailed analysis of our weakly-supervised approach PWarpC-NC-Net.

Ablation study: In the top part of Tab. 3, we analyze key components of our weakly-supervised approach. The version denoted as (II) is trained using our PW-bipath objective (eq. 7 of m.p.), without the visibility mask. NC-Net trained with this loss diverged. With the NC-Net architecture, we found it crucial to extend our loss with our visibility mask (eq. 7 of m.p.), resulting in version (III). We believe applying our PW-bipath loss on all pixels (II) confuses the NC-Net network, by enforcing matching even in *e.g.* non-matching background regions. Note that version (III) trained with our visibility aware PW-bipath objective (eq. 9 of m.p.) already outperforms the baseline (I) on all datasets and for all thresholds. Further adding the PWarp-supervision loss (eq. 8 of m.p.) in (IV) leads to worse results than (III) on the PF-Pascal and PF-Willow datasets, despite bringing an improvement on SPair-71K and TSS. To obtain a final network achieving competitive results on all four datasets, we therefore do not include the PWarp-supervision objective (eq. 8 of m.p.) in our final formulation.

From (III), including our occlusion modeling, *i.e.* the unmatched state and its corresponding probabilistic negative loss (eq. 10 of m.p.) in (V) leads to notable gains on the PF-Willow, SPair-71K and TSS datasets. In (VI), we further finetune the last layer of the feature backbone with the neighborhood consensus network in a second training stage. It leads to substantial improvements on all datasets, except for PF-Willow, where results remain almost unchanged.

From (VI) to (VII), we compare finetuning the feature backbone in a second training stage (VI), or directly in a single training stage (VII). The former leads to better performance on the PF-Pascal dataset. As a result, version (VI) corresponds to our final weakly-supervised PWarpC-NC-Net, trained with two stages on PF-Pascal.

Comparison to other losses: In the middle part of Tab. 3, we compare our Probabilistic Warp Consistency approach to previous weakly-supervised alternatives. The baseline NC-Net, corresponding to version (I), is trained with maximizing the max scores of the predicted cost volumes for matching images. It leads to significantly worse results than our approach (VI) on all datasets and threshold. The same conclusions apply to version (VIII), trained with minimizing the cost volume entropy for matching images. Finally, we compare our probabilistic approach (VI) to the mapping-based Warp Consistency method, corresponding to (IX). While Warp Consistency (IX) achieves good performance on the SPair-71K and TSS datasets, it leads to poor results on the PF-Pascal and PF-Willow datasets.

Comparison of objectives on negative image pairs: Finally, in the bottom part of Tab. 3, we compare multiple alternative losses applied on image pairs depicting different object classes. In particular, we combine our visibility-aware PW-bipath loss (III) with either our introduced probabilistic negative loss (eq. 10 of m.p.), minimizing the maximum scores [22] or maximizing the cost volume entropy [19] in respectively (V), (X) and (XI). Our probabilistic negative loss (eq. 10 of m.p.) leads to significantly better results on the PF-Willow, SPair-71K and TSS datasets. We believe it is because it enables to explicitly model occlusions and unmatched regions through our extended probabilistic formulation, including the unmatched state.

E. PWarpC-CATs

In this section, we first briefly review the CATs architecture and the original training strategy. We then provide details about the integration of our probabilistic approach into this architecture. Finally, we analyse the key components of our resulting strongly-supervised networks PWarpC-CATs and PWarpC-CATs-ft-features.

E.1. Details about CATs

Architecture: CATs [2] finds matches which are globally consistent by leveraging a Transformer architecture applied to slices of correlation maps constructed from multi-level features. The Transformer module alternates self-attention layers across points of the same correlation map, with inter-correlation self-attention across multi-level dimensions.

Training strategy in original work: While the final output of the CATs architecture is a cost volume, the latter is converted to a dense mapping by transposing into a probabilistic mapping with SoftMax, and then applying soft-argmax. The network is then trained with the End-Point Error objective, by leveraging the keypoint match annotations.

E.2. PWarpC-CATs: our training strategy

Warps W sampling: We apply the transformations on images with dimensions $s_r \times s_r = s \times s = 256 \times 256$. We do not further crop central images to avoid cropping keypoint annotations.

When training on PF-Pascal, we apply 5% of horizontal flipping to sample the random mappings M_W, while it is increased to 15% when training on SPair-71K.

Weighting and details on the losses : We define the known probabilistic mapping P_W with a one-hot representation for our PW-bipath and PWarp-supervision losses (8)-(9) of the main paper (see Sec. A).

To obtain PWarpC-CATs, we set the weights in eq. 12 of m.p. as $\lambda_{\text{P-warp-sup}} = L_{\text{vis-PW-bi}}/L_{\text{P-warp-sup}}$ and $\lambda_{\text{kp}} = (L_{\text{P-warp-sup}} + L_{\text{vis-PW-bi}})/L_{\text{kp}}$, which ensure that our probabilistic losses amount for the same than the keypoint loss L_{kp}.

To obtain PWarpC-CATs-ft-features, where the ResNet-101 backbone feature is additionally finetuned, we found the PWarp-supervision objective (eq. 8 of m.p.) to be slightly harmful, and therefore did not include it in this case. This is consistent with the findings of PWarpC-NC-Net and PWarpC-NC-Net*, for which the PWarp-supervised objective was also found harmful when the feature backbone is finetuned. This is likely due to the network 'overfitting' to the synthetic image pairs and transformations involved in the PWarp-supervision loss, at the expense of the real images considered in the PW-bipath (eq. 9 of m.p.) objectives. As a result, for the PWarpC-CATs-ft-features version, we set the weights in eq. 12 of m.p. as $\lambda_{\text{P-warp-sup}} = 0$ and $\lambda_{\text{kp}} = (L_{\text{P-warp-sup}} + L_{\text{vis-PW-bi}})/L_{\text{kp}}$.

Moreover, to be consistent with the baseline CATs, the keypoint loss L_{kp} is set as End-Point-Error loss, after converting the probabilistic mapping to a mapping through soft-argmax.

Implementation details: The softmax temperature, corresponding to equation 4 of the main paper, is set to $\tau = 0.02$, the same than originally used in the baseline. The hyper-parameter used in the estimation of our visibility mask $\widehat{V}$ (eq. 9 of the main paper) is set to $\gamma = 0.7$ and to $\gamma = 0.2$ when trained on PF-Pascal or SPair-71K respectively. This is because in SPair-71K, the objects are generally much smaller than in PF-Pascal.

For training, we use similar training parameters as in baseline CATs. We train with a batch size of 16 when the feature backbone is frozen, and reduce it to 7 when finetuning the backbone. The initial learning rate is set to $3 \cdot 10^{-6}$ for the feature backbone, and $3 \cdot 10^{-5}$ for the rest of the architecture. It is halved after 80, 100 and 120 epochs and we train for a maximum of 150 epochs. We use the same training parameters when training on either PF-Pascal or SPair-71K. The networks are trained using AdamW optimizer [14] with weight decay set to 0.05.

E.3. Ablation study

In Tab. 4, we analyse the key components of our strongly-supervised approaches PWarpC-CATs (top part) and PWarpC-CATs-ft-features (bottom part). From the CATs baseline, which is trained with the End-Point Error (EPE) objective while keeping the backbone feature frozen, adding our visibility-aware PW-bipath loss (eq. 9 of m.p.) leads to a subtantial gain on the PF-Willow and TSS dataset. Further including our PWarp-supervision objective results in improved performance on PF-Willow, SPair-71K and TSS. For the versions with finetuning the feature backbone (bottom part of Tab. 4), our visibility-aware PW-bipath objective brings major gains on PF-Willow, SPair-71K and TSS. However, further adding the PWarp-supervision leads to a small drop in performance on all datasets. For this reason, we use the combination of the EPE loss with our visibility-aware PW-bipath objective to train our final PWarpC-CATs-ft-features.

	PF-Pascal		PF-Willow		Spair-71k	TSS
	α_{img}		α_{bbox}		α_{bbox}	α_{img}
Methods	0.05	0.10	0.05	0.10	0.10	0.05
CATs baseline (EPE)	67.3	**88.6**	41.6	68.9	22.1	74.8
+ Vis-aware-PW-bipath	**68.1**	88.5	44.0	70.6	21.4	76.3
+ PWarp-supervision (**PWarpC-CATs**)	67.1	88.5	**44.2**	**71.2**	**23.3**	**82.4**
CATs-ft-features (EPE)	**79.8**	92.7	45.2	73.2	26.8	78.4
+ Vis-aware-PW-bipath (**PWarpC-CATs-ft-features**)	**79.8**	92.6	**48.1**	**75.1**	**27.9**	**88.7**
+ PWarp-supervision	79.6	92.4	46.7	74.4	26.0	**88.7**

Table 4. Ablation study for PWarpC-CATs and PWarpC-CATs-ft-features. We incrementally add each component. We measure the PCK on the PF-Pascal [4], PF-Willow [3], SPair-71K [18] and TSS [25] datasets. The evaluation results are computed using ground-truth annotations at original resolution.

F. PWarpC-DHPF

As in previous sections, we first review the DHPF [19] architecture and its original training strategy. We then provide training details for our strongly-supervised PWarpC-DHPF. Finally, we provide an ablation study for our approach applied to this architecture.

F.1. Details about DHPF

Architecture: DHPF learns to compose hypercolumn features, *i.e.* aggregation of different layers, on the fly by selecting a small number of relevant layers from a deep convolutional neural network. In particular, it proposes a gating mechanism to choose which layers to include in the hypercolumn. The hypercolumns features are then correlated, leading to the final output cost volume.

Training strategy in original work: The original work proposes both a weakly and strongly-supervised approach. The weakly-supervised approach is trained with minimizing the cost volume entropy computed between image pairs depicting the same class, while maximizing it for pairs depicting a different semantic content.

The strongly-supervised approach is instead trained with the cross-entropy loss, after converting the keypoint match annotations to probability distributions. In both cases, the authors also include a layer selection loss. It is a soft constraint to encourage the network to select each layer of the feature backbone at a certain rate.

F.2. PWarpC-DHPF: our training strategy

Warps W sampling: We apply the transformations on images with dimensions $s_r \times s_r = s \times s = 240 \times 240$. Similarly to PWarpC-CATs, we do not further crop central images to avoid cropping keypoint annotations.

When training on PF-Pascal, we apply 5% of horizontal flipping to sample the random mappings M_W, while it is increased to 15% when training on SPair-71K.

Weighting and details on the losses : We define the known probabilistic mapping P_W with a smooth representation for our PW-bipath and PWarp-supervision losses (8)-(9) of the main paper (see Sec. A).

To obtain PWarpC-DHPF, we set the weights in eq. 12 of m.p. as $\lambda_{\text{P-warp-sup}} = L_{\text{vis-PW-bi}}/L_{\text{P-warp-sup}}$ and $\lambda_{\text{kp}} = (L_{\text{P-warp-sup}} + L_{\text{vis-PW-bi}})/L_{\text{kp}}$, which ensure that our probabilistic losses amount for the same than the keypoint loss L_{kp}.

Moreover, in the strongly-supervised baseline DHPF, they train with a keypoint loss L_{kp} corresponding to the cross-entropy with the ground-truth keypoint matches converted to one-hot probabilistic mapping representations. We nevertheless found that the baseline is slightly improved when the ground-truth keypoint matches are instead converted to smooth probability distributions. We denote this version as DHPF* and compare it to our final PWarpC-DHPF in Tab. 5. As a result, for our PWarpC-DHPF, we set the keypoint loss L_{kp} in eq. 12 of m.p. to the cross-entropy with a smooth representation of the ground-truth keypoint match distributions. Finally, for fair comparison, we add the layer selection loss used in baseline DHPF to our strongly-supervised loss (eq. 12 of m.p.).

Implementation details: The softmax temperature, corresponding to equation 4 of the main paper, is set to $\tau = 1$, as in the baseline loss. Note that following the baseline DHPF, we apply gaussian normalization on the cost volume before applying the SoftMax operation to convert it to a probabilistic mapping. The hyper-parameter used in the estimation of our visibility mask $\widehat{V}$ (eq. 9 of the main paper) is set to $\gamma = 0.7$ and to $\gamma = 0.2$ when trained on PF-Pascal or SPair-71K respectively. This is because in SPair-71K, the objects are generally much smaller than in PF-Pascal.

For training, we use similar training parameters as in baseline DHPF. We train on PF-Pascal with a batch size of 6 for a maximum of 100 epochs. The initial learning rate is set $3 \cdot 10^{-2}$ and halved after 50 epochs. We optionally further finetune the network on SPair-71K, with an additional 10 epochs and a constant learning rate of $1 \cdot 10^{-2}$. The networks are trained using SGD optimizer [23].

F.3. Ablation study

In Tab. 5, we conduct ablative experiments on PWarpC-DHPF. Training with the cross-entropy loss using a smooth representation of the ground-truth in DHPF* leads to slightly better results than DHPF on PF-Pascal and SPair-71K. For this reason, we use it as baseline. Further including our visibility-aware PW-bipath loss and PWarp-supervision leads to incremental gains on PF-Willow and SPair-71K.

	PF-Pascal		**PF-Willow**		**Spair-71k**	**TSS**
	α_{img}		α_{bbox}		α_{bbox}	α_{img}
Methods	0.05	0.10	0.05	0.10	0.10	0.05
DHPF baseline (CE with one-hot)	77.3	**91.7**	44.8	70.6	27.5	72.2
DHPF* (CE with smooth)	**78.1**	90.7	44.7	70.1	27.9	74.02
+ Vis-aware-PW-bipath	76.3	90.7	47.3	73.6	28.0	73.7
+ PWarp-supervision (**PWarpC-DHPF**)	77.7	**91.7**	**47.7**	**74.3**	**28.6**	**74.3**

Table 5. Ablation study for PWarpC-DHPF. We incrementally add each component. We measure the PCK on the PF-Pascal [4], PF-Willow [3], SPair-71K [18] and TSS [25] datasets. The evaluation results are computed using ground-truth annotations at original resolution.

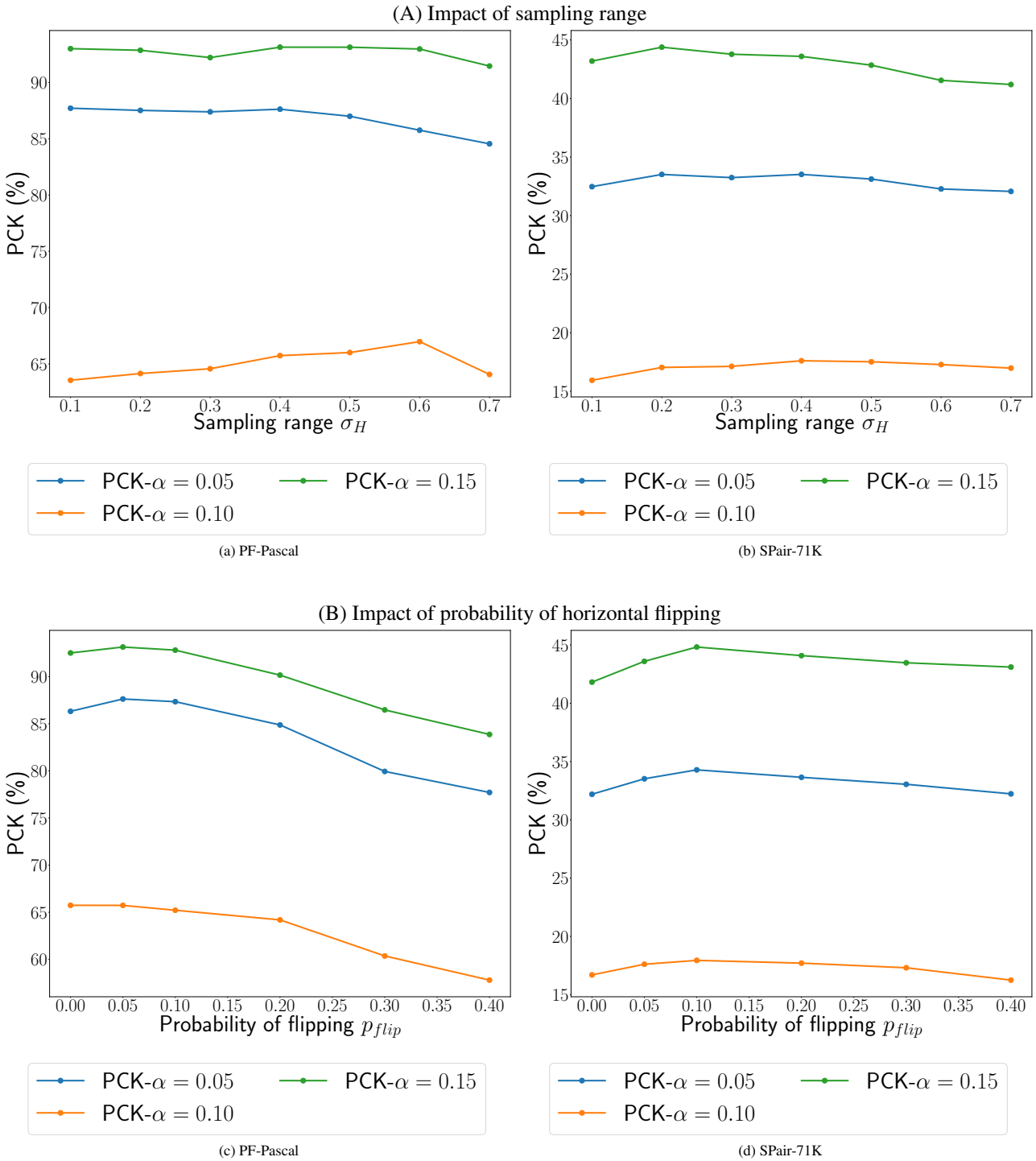
G. Analysis of transformations W

In this section, we analyse the impact of the sampled transformations' strength on the performance of the corresponding trained PWarpC networks. As explained in Sec. B, the strength of the warps M_W is mostly controlled by the range σ_H, used to sample the base homography, TPS and Affine-TPS transformations. The probability of horizontal flipping p_{flip} also has a large impact. We thus analyse the effect of the sampling range σ_H and the probability of horizontal flipping p_{flip} on the evaluation results of the corresponding PWarpC networks. In particular, we provide the analysis for our weakly-supervised PWarpC-SF-Net. The trend is the same for the other PWarpC networks.

While we choose a specific distribution to sample the transformations parameters used to construct the mapping M_W, our experiments show that the performance of the trained networks according to our proposed Probabilistic Warp Consistency loss is relatively insensitive to the strength of the transformations M_W, if they remain in a reasonable bound. We present these experiments in Fig. 2.

In Fig. 2 (A), we analyse the impact of the sampling range on the performance of PWarpC-SF-Net. Any range within $[0.1, 0.7]$ leads to similar performance, for $\alpha = 0.1$ and for $\alpha = 0.15$. Only for $\alpha = 0.05$ on PF-Pascal, increasing the range up to 0.6 leads to better results, with a drop for $\sigma_H = 0.7$. We select $\sigma_H = 0.4$ in our final setting.

We then look at the impact of the probability of horizontal flipping in Fig. 2 (B). On PF-Pascal, increasing the probability of flipping up to 5% leads to an increase in performance. Increasing it further nevertheless results in a gradual drop in performance. The trend is the same on SPair-71K, except that the best results are achieved for $p_{flip} = 10\%$. We therefore set $p_{flip} = 5\%$ for our final PWarpC networks.



(a) PF-Pascal (b) SPair-71K

(c) PF-Pascal (d) SPair-71K

Figure 2. Impact of the strength of the transformations M_W, on the performance of the weakly-supervised PWarpC-SF-Net network. We look at the PCK for α thresholds in $\{0.05, 0.1, 0.15\}$ obtained on the PF-Pascal [4] and SPair-71K [18] datasets, for different sampling ranges σ_H and probability of horizontal flipping p_{flip}, used to create the synthetic transformations M_W during training.

H. Results when training on SPair-71K

In this section, we analyse the performance of our PWarpC networks when trained or finetuned on SPair-71K instead of PF-Pascal. In Tab. 6, we provide results on the PF-Pascal, PF-Willow, Spair-71K and TSS, when networks are trained on the SPair-71K dataset. It extends Tab. 1 of the main paper, where models were instead trained on the

	Methods	Reso	PF-Pascal PCK @ α_{img} 0.05	0.10	0.15	PF-Willow PCK @ α_{bbox} 0.05	0.10	0.15	Spair-71k PCK @ α_{bbox} 0.05	0.10	TSS PCK @ α_{img}, $\alpha = 0.05$ FG3DCar	JODS	Pascal	Avg.
S	$\text{HPF}_{\text{res101}}$ [17]	max 300	-	-	-	-	-	-	-	28.2	-	-	-	-
	$\text{SCOT}_{\text{res101}}$ [13]	max 300	-	-	-	-	-	-	-	35.6	-	-	-	-
	$\text{CHM}_{\text{res101}}$ [16]	240	-	-	-	-	-	-	-	46.3	-	-	-	-
	$\text{PMD}_{\text{res101}}$ [12]	-	-	-	-	-	-	-	-	37.4	-	-	-	-
	$\text{PMNC}_{\text{res101}}$ [10]	-	-	-	-	-	-	-	-	50.4	-	-	-	-
	$\text{MMNet}_{\text{res101}}$ [30]	224×320	-	-	-	-	-	-	-	40.9	-	-	-	-
	$\text{DHPF}_{\text{res101}}$ [19]	240	52.6 †	75.4 †	84.8 †	37.4 †	63.9 †	77.0 †	20.7 †	37.3	-	-	-	-
	$\text{CATs}_{\text{res101}}$ [2]	256	45.3 †	67.7 †	77.0 †	31.8 †	56.8 †	69.1 †	21.9 †	42.4	-	-	-	-
	$\text{CATs-ft-features}_{\text{res101}}$ [2]	256	54.4 †	74.1 †	81.9 †	39.7 †	66.3 †	78.3 †	27.9 †	49.9	-	-	-	-
	$\text{CATs-ft-features}_{\text{res101}}$ [2]	ori †	57.7	75.2	82.9	43.5	69.1	80.8	27.1	48.8	88.9	73.9	57.1	73.3
	PWarpC-CATs-ft-features$_{\text{res101}}$	ori	58.8	77.4	84.6	*46.4*	*73.6*	*85.0*	28.2	48.4	91.1	**85.8**	*69.1*	*82.0*
	$\text{DHPF}_{\text{res101}}$ [19]	ori †	56.9	77.2	86.3	40.9	66.8	79.9	20.6	36.3	83.8	69.7	57.3	70.3
	PWarpC-DHPF$_{\text{res101}}$	ori	65.8	*85.5*	92.3	**47.6**	72.9	84.5	23.3	38.7	87.5	73.7	60.3	73.8
	NC-Net*$_{\text{res101}}$	ori	59.8	75.6	82.1	38.9	62.6	74.7	*29.1*	*50.7*	81.1	66.7	45.4	64.4
	PWarpC-NC-Net*$_{\text{res101}}$	ori	*67.8*	82.3	86.9	46.1	72.6	82.7	**31.6**	**52.0**	**93.0**	*84.6*	**70.6**	**82.7**
	SF-Net*$_{\text{res101}}$	ori	66.5	85.0	90.8	43.5	70.4	82.9	26.2	50.0	88.3	75.3	57.2	73.6
	PWarpC-SF-Net*$_{\text{res101}}$	ori	**72.1**	**89.6**	**93.5**	46.3	**75.2**	**87.0**	27.0	48.8	*92.5*	81.1	66.2	79.9
U	$\text{CNNGeo}_{\text{res101}}$ [20] (results from [18])	-	-	-	-	-	-	-	-	20.6	-	-	-	-
	$\text{A2Net}_{\text{res101}}$ [24] (results from [18])	-	-	-	-	-	-	-	-	22.3	-	-	-	-
M	$\text{SF-Net}_{\text{res101}}$ [9] (results from [10])	-	-	-	-	-	-	-	-	26.3	-	-	-	-
W	**PWarpC-SF-Net**$_{\text{res101}}$	ori	**64.5**	**86.9**	**92.6**	**47.1**	**78.1**	**89.9**	**18.6**	*37.1*	91.0	81.6	67.4	80.0
	$\text{WeakAlign}_{\text{res101}}$ [21] (results from [18])	-	-	-	-	-	-	-	-	20.9				
	$\text{DHPF}_{\text{res101}}$ [19]	240	46.1 †	78.1 †	88.4 †	34.9 †	66.2 †	82.5 †	12.4 †	27.7	-	-	-	-
	$\text{DHPF}_{\text{res101}}$ [19]	ori †	53.3	81.3	90.3	40.9	70.1	84.6	12.7	27.2				
	$\text{PMD}_{\text{res101}}$ [12]	-	-	-	-	-	-	-	-	26.5	-	-	-	-
	$\text{WarpC-SemGLU-Net}_{\text{vgg16}}$ [28]	ori	57.0 †	78.7 †	88.7 †	*46.1* †	72.8 †	84.9 †	12.8 †	23.5	**96.3** †	*84.2* †	*80.2* †	*86.9*
	$\text{NC-Net}_{\text{res101}}$ [22] (results from [18])	-	-	-	-	-	-	-	-	20.1	-	-	-	-
	PWarpC-NC-Net$_{\text{res101}}$	ori	*61.7*	*82.6*	*88.5*	43.6	*74.6*	*86.9*	*18.5*	**38.0**	*95.4*	**88.9**	**85.6**	**90.0**

Table 6. PCK [%] obtained by different state-of-the-art methods on the PF-Pascal [4], PF-Willow [3], SPair-71K [18] and TSS [25] datasets. All approaches are trained or finetuned on the training set of Spair-71K. **S** denotes strong supervision using key-point annotation, **M** refers to using ground-truth object segmentation mask, **U** is fully unsupervised requiring only single images, and **W** refers to weakly supervised with image class labels. Each method evaluates with ground-truth annotations resized to a specific resolution. However, using different ground-truth resolution leads to slightly different results. We therefore use the standard setting of evaluating on the original resolution (**ori**) and gray the results computed at a different resolution. When needed, we re-compute metrics of baselines using the provided pre-trained weights, indicated by †.

PF-Pascal dataset.

Weakly-supervised: In the bottom part of Tab. 6, we compare approaches trained with a weakly-supervised approach. Our PWarpC-SF-Net and PWarpC-NC-Net trained on PF-Pascal were further finetuned on SPair-71K with our Probabilistic Warp Consistency objective 11 of the main paper. Note that baselines SF-Net and NC-Net were obtained by finetuning on SPair-71K the original models trained on PF-Pascal, with their respective original training strategies. Our weakly-supervised approaches PWarpC-SF-Net and PWarpC-NC-Net lead to a particularly impressive improvement compared to their respective baselines, with 41% (+ 10.8) and 89.1% (+ 17.9) relative (and absolute) gains. As a result, PWarpC-SF-Net and PWarpC-NC-Net set a new state-of-the-art on respectively the PF-Willow and PF-Pascal datasets, and the SPair-71K and TSS datasets, across all unsupervised (U), weakly-supervised (W) and mask-supervised (M) approaches trained on SPair-71K.

Strongly-supervised: In the top part of 6, we report results of models trained with a strongly-supervised approach, leveraging keypoint match annotations. While training on SPair-71K with our approach leads to similar results than the baselines on SPair-71K, our PWarpC networks show drastically better generalization properties to PF-Pascal, PF-Willow and TSS. Our strongly-supervised PWarpC-NC-Net* sets a new state-of-the-at on SPair-71K and TSS, across all strongly-supervised approaches trained on SPair-71K. Our PWarpC-SF-Net* also obtains state-of-the-art results on the PF-Pascal and PF-Willow datasets.

I. Detailed results when trained on PF-Pascal

In this section, we first provide additional details on the validation datasets and experimental setting in Sec. I.1. In Sec. I.2, we then analyze the robustness of our approach to different variation factors, *i.e.* occlusion, truncation, scale and view-point, on the SPair-71K dataset. Subsequently, we show additional prediction examples of the unmatched state for our weakly-supervised approaches in Sec. I.3. We follow by analysing our approach in terms of robustness of the predicted confidence scores in Sec. I.4. In Sec. I.6, we fur-

	Methods	Reso	View-point			Scale			Truncation				Occlusion				
			easy	medi	hard	easy	medi	hard	none	src	trg	both	none	src	trg	both	All
U	CNNGeo (from [18])	-	25.2	10.7	5.9	22.3	16.1	8.5	21.1	12.7	15.6	13.9	20.0	14.9	14.3	12.4	18.1
	A2Net (from [18])	-	27.5	12.4	6.9	24.1	18.5	10.3	22.9	15.2	17.6	15.7	22.3	16.5	15.2	14.5	20.1
M	SF-Net	ori †	32.0	15.5	10.0	28.4	22.0	13.2	27.0	20.1	20.0	18.7	26.6	18.5	18.9	18.0	24.0
W	**PWarpC-SF-Net**	ori	*41.9*	*24.2*	*20.7*	*39.1*	*31.8*	*18.8*	*36.3*	*29.7*	*30.4*	**28.4**	*36.5*	*27.7*	*27.9*	*24.7*	*33.5*
	WeakAlign (from [18])	-	29.4	12.2	6.9	25.4	19.4	10.3	24.1	16.0	18.5	15.7	23.4	16.7	16.7	14.8	21.1
	NC-Net (from [18])	-	34.0	18.6	12.8	31.7	23.8	14.2	29.1	22.9	23.4	21.0	29.0	21.1	21.8	19.6	26.4
	NC-Net	ori †	37.6	19.4	13.8	34.7	26.0	14.9	31.7	25.2	25.1	*23.5*	31.5	23.4	24.3	20.9	28.8
	PWarpC-NC-Net	ori	**42.6**	**27.1**	**24.6**	**40.8**	**33.4**	**20.8**	**38.5**	**30.1**	**32.8**	**28.4**	**38.1**	**29.1**	**31.2**	**25.9**	**35.3**
S	DHPF	ori †	34.5	20.0	15.4	32.4	25.7	14.7	31.1	22.5	22.7	22.1	30.2	21.8	22.9	18.7	27.5
	PWarpC-DHPF	ori	35.8	21.0	16.7	33.5	26.6	16.5	32.2	23.8	24.5	21.7	31.5	22.7	23.9	20.2	28.6
	CATS	ori †	29.7	13.8	9.56	26.4	20.2	11.6	25.2	17.8	18.1	17.3	24.3	17.2	18.4	16.3	22.1
	PWarpC-CATs	ori	30.7	15.1	11.2	28.2	21.0	11.6	26.7	19.1	18.4	18.0	25.8	17.8	19.0	16.6	23.3
	CATs-ft-features	ori †	35.6	17.0	12.8	31.5	24.9	14.7	29.9	22.6	22.3	22.4	29.7	20.5	21.6	18.5	26.8
	PWarpC-CATs-ft-features	ori	35.6	19.6	15.7	33.7	25.3	14.2	32.0	22.5	22.9	21.1	31.2	21.0	22.1	19.2	27.9
	SF-Net*	ori	36.8	18.6	12.2	32.8	25.8	16.0	30.1	25.4	25.0	23.7	30.6	22.7	23.2	18.4	27.9
	PWarpC-SF-Net*	ori	41.5	*22.8*	*18.1*	*38.1*	*30.6*	18.2	*35.6*	28.6	28.9	26.2	*35.4*	*26.4*	27.4	*25.3*	*32.5*
	NC-Net*	ori	*42.0*	22.3	15.4	37.5	30.3	*19.7*	34.9	*28.8*	*30.0*	*26.3*	35.2	26.2	*28.1*	23.8	32.4
	PWarpC-NC-Net*	ori	**45.4**	**27.7**	**24.7**	**42.6**	**35.2**	**22.5**	**40.3**	**32.5**	**33.7**	**29.7**	**40.0**	**30.5**	**32.2**	**29.6**	**37.1**

Table 7. PCK analysis for state-of-the-art approaches, by variation factors on SPair-71K. The variation factors include view-point, scale, truncation, and occlusion with various difficulty levels. All models in this table use ResNet101 as the backbone, and are trained on the training set of PF-Pascal. **S** denotes strong supervision using keypoint match annotations, **M** refers to using ground-truth object segmentation mask, **U** is fully unsupervised requiring only single images, and **W** refers to weakly-supervised with image-level class labels. Each method evaluates with ground-truth annotations resized to a specific resolution. However, using different ground-truth resolutions leads to slightly different results. We therefore use the standard setting of evaluating on the original resolution (**ori**). When needed, we re-compute metrics of baselines using the provided pre-trained weights, indicated by †. For each of our PWarpC networks, we compare to its corresponding baseline within the dashed-lines. Best and second best results are in red and blue respectively.

ther compare state-of-the-art methods on the Caltech-101 dataset. Finally, we provide extensive qualitative comparisons in Sec I.7.

I.1. More details on datasets and metrics

Evaluation metrics: For evaluation, we adopt the standard evaluation metric, percentage of correct keypoints (PCK). Given a set of M predicted and ground-truth keypoint $\left\{\widehat{k}\right\}_{m=1}^{M}$ and $\{k\}_{m=1}^{M}$, the PCK for the corresponding image pair is calculated as PCK $= \frac{1}{M}\sum_{m=1}^{M} \mathbb{1}\left[\left\|\widehat{k}_m - k_m\right\| \leqslant \alpha_\tau \cdot \max(h_s^\tau, w_s^\tau)\right]$. Here, h_s and w_s are either the dimensions of the source image or the dimensions of the object bounding box in the source image.
PF-Pascal contains 1341 image pairs from 20 categories. Images have dimensions randing from 102×300 to 300×300. We use the splits proposed in [5] where training, validation and test sets respectively contain 700, 300 and 300 image pairs. In line with [5], we report the PCK with respect to the dimensions of the source image.
PF-Willow comprises 900 images from 4 categories with small variations in view-point and scale, and 10 keypoint annotations per pair. Images have dimensions ranging from 153×300 to 300×300. Due to the absence of real bounding box annotations in PF-WILLOW, the evaluation threshold of a bounding box, $\max(w_s^{\text{bbox}}, h_s^{\text{bbox}})$, is computed using the two furthest key-point positions to approximate a bounding box that tightly wraps the object. However, note that a few previous works sometimes use a different bounding box definition, which loosely covers the object by using only a single keypoint position. Since this definition is not as accurate as the former, we do not report the results using this bounding box definition.

SPair-71K is a highly challenging dataset, comprising 70958 image pairs from 18 categories with extreme and diverse viewpoint and scale variations. Images have dimensions ranging from 188×312 to 500×500. The dataset contains rich annotations for each image pair, *e.g.* keypoints, scale difference, truncation and occlusion difference, and a clear data split. In line with previous works, we report the PCK with respect to source bounding box dimensions.

TSS is the only dataset proving dense flow field annotations for the foreground object in each pair. It contains 400 image pairs, divided into three groups: FG3DCAR, JODS, and PASCAL, according to the origins of the images. Images have dimensions ranging from 237×250 to 600×800. Evaluation is done on 800 pairs, by also exchanging source and target images. The PCK is computed with respect to source image size.

	Methods	View-point			Scale			Truncation				Occlusion				
		easy	medi	hard	easy	medi	hard	none	src	trg	both	none	src	trg	both	All
I	SF-Net	32.0	15.5	10.0	28.4	22.0	13.2	27.0	20.1	20.0	18.7	26.6	18.5	18.9	18.0	24.0
II	PW-bipath (eq. 7 of m.p.)	35.0	20.3	17.4	33.6	25.9	14.2	31.7	23.1	23.4	21.8	30.7	21.9	23.8	21.0	28.0
III	+ visibility mask (eq. 9 of m.p.)	37.4	19.0	13.3	33.6	26.6	15.8	31.7	24.7	24.4	22.1	31.3	22.3	24.2	21.4	28.5
IV	+ PWarp-supervision (eq. 8 of m.p.)	38.0	22.2	18.9	35.9	28.5	16.5	33.7	25.9	26.8	25.3	33.4	24.6	25.3	22.9	30.7
V	+ PNeg (eq. 10 of m.p.) (**PWarpC-SF-Net**)	**41.9**	**24.2**	**20.7**	**39.1**	**31.8**	**18.8**	**36.3**	**29.7**	**30.4**	**28.4**	**36.5**	**27.7**	**27.9**	**24.7**	**33.5**
V	**PWarpC-SF-Net** (Ours)	**41.9**	**24.2**	**20.7**	**39.1**	**31.8**	**18.8**	**36.3**	**29.7**	**30.4**	**28.4**	**36.5**	**27.7**	**27.9**	**24.7**	**33.5**
VI	Mapping Warp Consistency [28]	34.4	18.2	14.0	31.7	24.7	14.1	30.5	22.3	21.2	19.3	29.4	20.9	21.8	18.3	26.6
VII	PWarp-supervision only (eq. 8 of m.p.)	35.3	21.6	16.6	33.4	26.9	16.2	31.1	24.4	26.5	23.2	31.4	23.0	23.3	21.2	27.9
VIII	Max-score [22]	34.0	20.8	15.9	33.0	25.3	14.2	30.0	24.7	24.2	22.4	30.3	21.6	22.5	20.4	24.6
IX	Min-entropy [19]	28.3	17.5	12.3	27.7	20.4	11.9	25.3	19.3	19.9	19.8	25.4	18.2	18.1	15.5	20.6

Table 8. Ablation study (top part) and comparison to alternative weakly-supervised or unsupervised losses (bottom part). We compare the PCK by variation factors on SPair-71K. The variation factors include view-point, scale, truncation, and occlusion with various difficulty levels.

I.2. Robustness to specific challenges

To better understand the performance of our training approach under complex conditions, we report the results according to different variation factors with various difficulty levels. In particular, the SPair-71k dataset contains diverse variations in view-point, scale, truncation and occlusion. In addition to the keypoint match annotations, the dataset also provide specific annotations for each of the variation factors, with different levels of difficulty. We are particularly interested in the occlusion setting.

Comparison to state-of-the-art: In Tab. 7, we compare state-of-the-art approaches by variation factor on the SPair-71K dataset. Both our weakly-supervised (W) approaches PWarpC-NC-Net and PWarpC-SF-Net bring a significant improvement compared to their respective baselines, for all variation factor and all difficulty levels. Specifically for occlusion, our PWarpC-SF-Net and PWarpC-NC-Net bring an absolute gain of 8.7% and 5.9%, from their respective baselines SF-Net and NC-Net.

As for strongly-supervised approaches (S), each of our PWarpC network shows an improvement compared to its baseline, for all four variation factors.

Ablation study: In Tab. 8 top part, we show the impact of the key components of our weakly-supervised approach. From version (II), which corresponds to training with our PW-bipath objective without the visibility mask, to our final PWarpC-SF-Net, denoted as (V), incrementally adding each of our losses brings a significant improvement for all variation factors and levels of difficulty. In particular, in (V), our explicit occlusion modeling, *i.e*. through introducing an unmatched state and our probabilistic negative loss, leads to a particularly impressive gain compared to (IV), of 3.3% and 1.8% for truncation and occlusion respectively.

Comparison to alternative weakly-supervised cost volume losses: In Tab. 8 bottom part, we further compare our final weakly-supervised PWarpC-SF-Net to alternative weakly-supervised objectives. We first compare our probabilistic approach (V) to the mapping-based Warp Consistency [28], denoted as (VI). Here, note that to train with the Warp Consistency objective [28], the predicted cost volume is converted to a mapping, by applying kernel soft-argmax as in the original work [9]. Our probabilistic approach (V) obtains significantly better results than the mapping-based warp consistency (VI), for all variations factors and level of difficulties. Version (VII) corresponds to training the SF-Net architecture with only the PWarp-supervision objective. The performance is much worse than when trained with our Probabilistic Warp Consistency (V). Finally, both training with maximizing the max scores in (VIII), or minimizing the cost volume entropy in (IX) also lead to poor results compared to our approach (V) for all variation factors.

I.3. Example predictions for the unmatched state

In Fig. 5, we provide examples of the unmatched state predictions of our weakly-supervised approach PWarpC-NC-Net. The results are similar for PWarpC-SF-Net. Our probabilistic approach predicts Dirac-like distributions, whose mode are correct. Furthermore, through our explicit occlusion modeling approach (Sec. 4.3 of the paper), the network successfully identifies the object in most examples.

I.4. Confidence analysis

Most semantic matching architectures predict a cost volume as the final network output. The cost volume, after conversion to a probabilistic mapping through SoftMax, inherently encodes the confidence of each predicted tentative match. It is not the case when directly regression a mapping or flow output instead. Nevertheless, a confidence estimation for each of the predicted matches can in the case be obtained, by *e.g*. forward-backward consistency of the flow field [15]. In this section, we analyse the quality of the confidence predictions, when the networks are trained with our weakly-supervised Probabilistic Warp Consistency approach.

A common technique to assess the quality of a confidence estimate is to rely on sparsification and error curves.

Sparsification and error curves: To assess the quality of the uncertainty estimates, we rely on sparsification plots, in line with [1, 6, 26, 27, 29]. The pixels having the high-

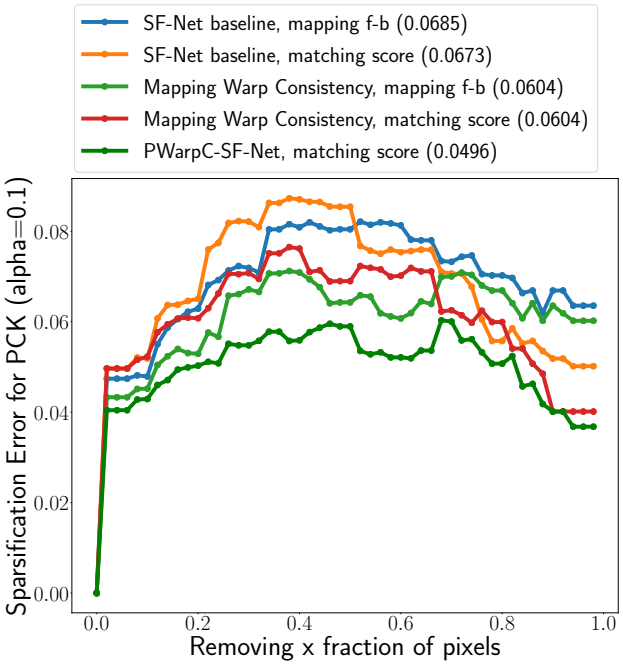


Figure 3. Sparsification error curves and AUSE on the PF-Pascal dataset, for our PWarpC-SF-Net, baseline SF-Net and SF-Net trained with the mapping-based Warp Consistency objective. For the two latter, the predicted cost volume is converted to a mapping before applying the loss. For this reason, we show two alternative confidence estimation schemes, based on the matching scores, or on the forward-backward consistency of the flow. For our approach PWarpC-SF-Net, our Probabilistic Warp Consistency objective is applied directly on the probabilistic mapping, therefore we directly use the probabilities of the hard assigned matches as confidence measures. Smaller AUSE is better.

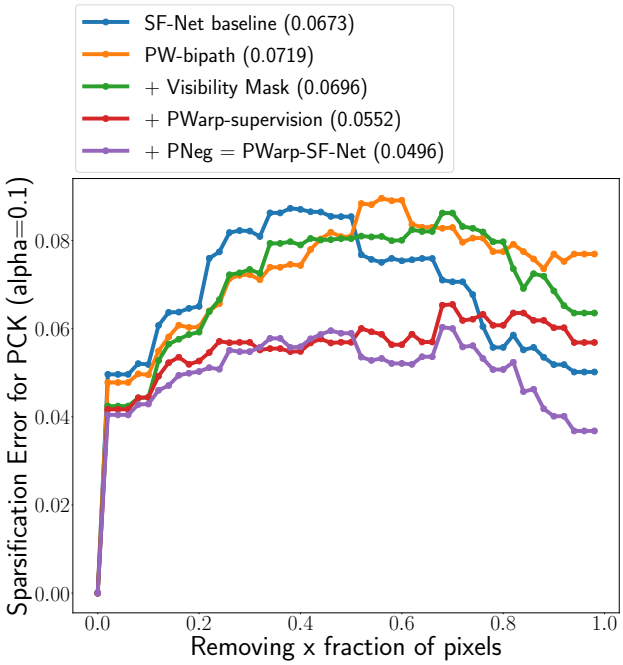


Figure 4. Ablation study for weakly-supervised PWarpC-SF-Net in terms of sparsification error curves and AUSE on the PF-Pascal dataset. As confidence measure, we use the probabilities of the hard assigned matches. Smaller AUSE is better.

est uncertainty are progressively removed and the PCK of the remaining pixels is plotted in the sparsification curve. These plots reveal how well the estimated uncertainty relates to the true errors. Ideally, larger uncertainty should correspond to larger errors. Gradually removing the predictions with the highest uncertainties should therefore monotonically improve the accuracy of the remaining correspondences. The sparsification plot is compared with the best possible ranking of the predictions, according to their actual errors computed with respect to the ground-truth flow. We refer to this curve as the oracle plot.

Note that, for each network the oracle is different. Hence, an evaluation using a single sparsification plot is not possible. To this end, we use the Sparsification Error, constructed by directly comparing each sparsification plot to its corresponding oracle plot by taking their difference. Since this measure is independent of the oracle, a fair comparison between different methods is possible. As evaluation metric, we use the Area Under the Sparsification Error curve (AUSE). We compute the sparsification error curve on each image pair. The final error curve is the average over all image pairs of the dataset.

The sparsification and error plots provide an insightful and directly relevant assessment of the uncertainty. In particular, the AUSE directly evaluates the ability to filter out inaccurate and incorrect correspondences, which is the main purpose of the uncertainty estimate.

Results: For this confidence analysis, we use SF-Net as baseline. In Fig. 3, we report the sparsification error curves and AUSE obtained by our approach PWarpC-SF-Net, the baseline SF-Net, and SF-Net trained with the mapping-based Warp Consistency [28] objective. For the baseline and the Warp Consistency versions, using directly the predicted matching scores of the hard-assigned matches or the forward-backward mapping as confidence measure leads to similar results. Nevertheless, our probabilistic PWarpC-SF-Net outperforms the other approaches in AUSE. It demonstrates the benefit of our probabilistic approach, acting directly on dense matching scores. In particular, it shows our approach can filter out inaccurate and incorrect correspondences better than previous methods, which is extremely important for usability in end tasks.

Ablation study: In Fig. 4, we show the impact of the key components of our weakly-supervised Probabilistic Warp Consistency approach, on the confidence estimation robustness. Adding the visibility mask, our PWarp-supervision loss and our occlusion modelling consistency improves the AUSE.

	Methods	Reso	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	dog	horse	moto	person	plant	sheep	train	tv	all
U	CNNGeo (from [18])	-	21.3	15.1	34.5	12.8	31.2	26.3	24.0	30.6	11.6	24.3	20.4	12.2	19.7	15.6	14.3	9.6	28.5	28.8	18.1
	A2Net (from [18])	-	20.8	17.1	37.4	13.9	33.6	29.4	26.5	34.9	12.0	26.5	22.5	13.3	21.3	20.0	16.9	11.5	28.9	31.6	20.1
M	SF-Net	ori †	24.1	15.7	43.3	14.6	35.6	20.8	13.8	47.7	14.9	26.7	24.2	13.4	19.4	20.7	15.2	14.2	28.8	34.9	24.0
W	**PWarpC-SF-Net**	ori	**38.8**	*27.6*	*58.3*	*18.9*	**41.3**	**30.3**	*21.7*	*56.2*	**20.1**	**38.3**	*33.8*	*20.0*	*28.6*	*24.2*	*21.7*	*18.2*	*42.2*	**60.0**	*33.5*
	WeakAlign (from [18])	-	23.4	17.0	41.6	14.6	37.6	28.1	26.6	32.6	12.6	27.9	23.0	13.6	21.3	22.2	17.9	10.9	31.5	34.8	21.2
	NC-Net (from [18])	-	24.0	16.0	45.0	13.7	35.7	25.9	19.0	50.4	14.3	32.6	27.4	19.2	21.7	20.3	20.4	13.6	33.6	40.4	26.4
	NC-Net	ori †	25.9	18.1	45.6	16.7	39.7	26.6	20.0	52.7	15.5	33.4	30.6	18.4	24.4	23.5	24.2	16.0	36.1	47.3	28.8
	PWarpC-NC-Net	ori	*37.4*	**28.8**	**60.8**	**22.9**	*40.5*	29.4	**22.8**	**60.1**	19.5	37.8	**38.4**	**27.9**	**32.1**	**29.7**	**29.2**	**20.2**	**44.5**	50.0	**35.3**
S	DHPF	ori †	23.1	21.6	56.0	16.6	36.6	21.7	15.8	49.6	16.5	31.3	34.8	*19.2*	25.0	25.8	20.3	14.8	31.7	31.5	27.5
	PWarpC-DHPF	ori	25.2	23.8	62.0	17.1	35.1	23.5	16.8	51.2	16.9	34.7	*34.6*	19.1	25.6	25.3	18.1	16.0	31.6	37.0	28.6
	CATs-ft-features	ori †	23.7	18.7	49.5	16.3	37.3	20.8	14.6	47.1	17.7	32.5	30.3	15.2	22.4	22.6	20.2	15.4	34.7	37.7	26.8
	PWarpC-CATs-ft-features	ori	24.5	21.3	56.3	16.6	35.0	23.7	16.0	54.0	15.3	34.5	36.2	14.6	21.1	19.8	17.3	15.4	39.4	37.6	27.9
	CATs	ori †	22.5	15.0	41.9	14.0	34.2	19.5	14.1	40.7	13.8	24.9	24.2	13.6	17.2	16.8	13.6	13.1	27.9	27.1	22.1
	PWarpC-CATs	ori	24.2	14.9	44.5	14.4	34.4	21.0	15.2	44.4	13.6	27.6	26.1	14.0	17.4	16.2	15.5	12.9	31.1	28.2	23.3
	SF-Net*	ori	26.1	21.6	48.7	16.7	*39.6*	23.1	17.8	52.6	17.7	32.2	31.9	15.7	21.8	*27.1*	22.5	16.3	31.9	35.5	27.9
	PWarpC-SF-Net*	ori	*33.8*	*28.3*	*56.1*	*18.6*	38.9	**30.4**	20.5	56.3	*19.3*	*36.8*	32.4	18.4	*28.9*	26.1	*23.4*	*18.6*	*42.2*	*53.1*	*32.5*
	NC-Net*	ori	28.8	24.0	53.6	19.2	41.1	27.8	*21.4*	*61.1*	18.8	38.5	35.4	22.9	25.2	25.9	28.1	20.7	41.3	45.3	32.4
	PWarpC-NC-Net*	ori	**40.1**	**31.0**	**65.5**	**23.4**	**43.1**	*29.4*	**21.9**	**61.8**	**21.4**	**41.2**	**39.2**	**28.1**	**32.0**	**30.8**	**30.0**	**22.5**	**43.9**	**58.2**	**37.1**

Table 9. Per-class PCK ($\alpha_{bbox} = 0.1$) results on SPair-71K. All models in this table use ResNet101 as the backbone, and are trained on the training set of PF-Pascal. **S** denotes strong supervision using keypoint match annotations, **M** refers to using ground-truth object segmentation mask, **U** is fully unsupervised requiring only single images, and **W** refers to weakly-supervised with image-level class labels. Each method evaluates with ground-truth annotations resized to a specific resolution. However, using different ground-truth resolutions leads to slightly different results. We therefore use the standard setting of evaluating on the original resolution (**ori**). When needed, we re-compute metrics of baselines using the provided pre-trained weights, indicated by †. For each of our PWarpC networks, we compare to its corresponding baseline within the dashed-lines. Best and second best results are in red and blue respectively.

I.5. Detailed results when trained on PF-Pascal

For completeness, we provide the results per category on the SPair-71K dataset in Tab. 9. All approaches are trained on the PF-Pascal dataset. It corresponds to results provided in Tab. 1 of the main paper.

I.6. Additional results on Caltech

Here, we additionally evaluate our approach on the Caltech dataset.

Caltech-101 contains images depicting 101 diverse object classes. Each image comes with a ground-truth foreground object segmentation mask. Although originally introduced for the image classification task, this dataset was adopted for assessing semantic alignment. Particularly, the predicted dense correspondences relating the target to the source image are used to warp the ground-truth segmentation mask of the source towards the target. The overlap between the warped source segmentation mask and the ground-truth target segmentation mask is then measured, and used as a proxy to assess the quality of the predicted dense correspondences. We follow the standard set-up, according to which the evaluation is performed on 1515 semantically related image pairs, *i.e.* 15 pairs for each of the 101 object categories of the dataset. The semantic alignment is evaluated using two different metrics: the label transfer accuracy (LT-ACC) and the intersection-over-union (IoU). They both measure the overlap between the annotated foreground object segmentation masks, with former putting more emphasis on the background class and the latter on the foreground object.

Note that compared to other benchmarks described above, the Caltech-101 dataset provides image pairs from more diverse classes, enabling us to evaluate our method under more general correspondence settings.

Results: In Tab. 10, we present results of semantic networks on the Caltech dataset. All approaches are trained on the PF-Pascal dataset. For both weakly-supervised (W) and strongly-supervised (S) approaches, our PWarpC networks show significantly better performance than their respective baselines. The only exception is our weakly-supervised PWarpC-SF-Net, which obtains worse results than its baseline SF-Net. This is because on Caltech-101, the evaluation is conducted by warping the foreground mask of the source image, according to the predicted dense flow or mapping. In that case, a smooth flow is very beneficial. Baseline SF-Net explicitly enforces smoothness of the predicted flow fields as part of the training strategy [9], which explains its good performance. On the other hand, we do not specifically enforce any smoothness priors, which is why PWarpC-SF-Net lacks in performance compared to SF-Net for this particular dataset and evaluation. Nevertheless, on all other datasets (see Tab. 1 of m.p.), where the metrics are based directly on the predicted matches, our approach PWarpC-SF-Net significantly outperforms baseline SF-Net. Moreover, our PWarpC-DHPF sets a new state-of-the-art on Caltech-101 across all approaches, independently of their level of supervision.

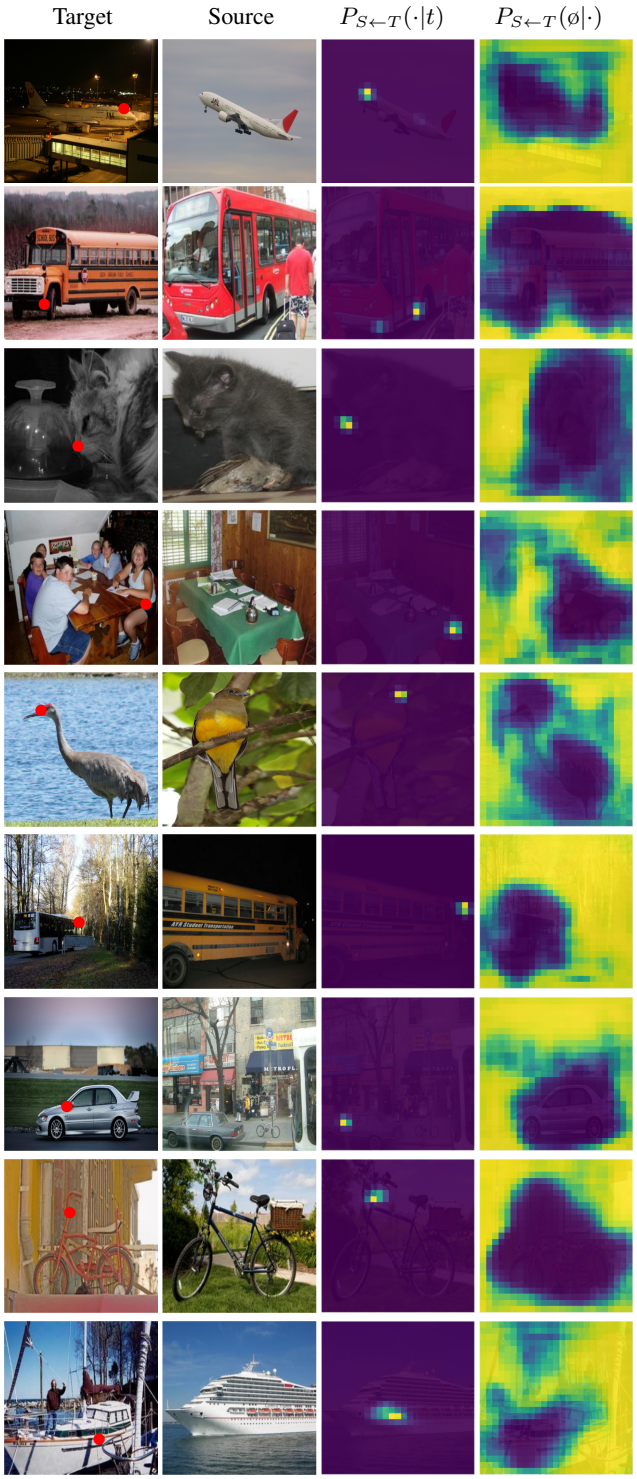


Figure 5. Examples of the predicted probability distribution relating the target to the source image, given a pixel location t (red dot) in the target image. We also show the predicted unmatched state for all pixels of the target image. Yellow and purple corresponds respectively to values of 1 and 0. Here, we use our weakly-supervised PWarpC-NC-Net for the predictions.

Sup.	Methods	Reso	LT-ACC ↑	IoU ↑
S	SCNet$_{\text{VGG16}}$ [5] (from [17])	-	0.79	0.51
	HPF$_{\text{res101}}$ [17]	ori	*0.87*	*0.63*
	DHPF$_{\text{res101}}$ [19]	240	0.87	0.62
	CATs$_{\text{res101}}$ [2]	ori †	0.84	0.58
	PWarpC-CATs$_{\text{res101}}$	ori	0.85	0.60
	CATs-ft-features$_{\text{res101}}$ [2]	ori †	0.84	0.59
	PWarpC-CATs-ft-features$_{\text{res101}}$	ori	0.86	0.60
	DHPF$_{\text{res101}}$ [19]	ori †	*0.87*	0.62
	PWarpC-DHPF$_{\text{res101}}$	ori	**0.88**	**0.64**
	NC-Net*$_{\text{res101}}$	ori	0.81	0.54
	PWarpC-NC-Net*$_{\text{res101}}$	ori	*0.87*	0.62
	SF-Net*$_{\text{res101}}$	ori	0.85	0.59
	PWarpC-SF-Net*$_{\text{res101}}$	ori	*0.87*	0.62
U	CNNGeo$_{\text{res101}}$ [20] (from [21])	-	0.83	0.61
	A2Net$_{\text{res101}}$ [24]	-	0.80	0.57
M	SF-Net$_{\text{res101}}$ [9]	ori †	**0.87**	**0.64**
W	**PWarpC-SF-Net**$_{\text{res101}}$	ori	*0.86*	0.61
	WeakAlign$_{\text{res101}}$ [21]		0.85	*0.63*
	DHPF$_{\text{res101}}$ [19]	240	0.86	0.61
	DHPF$_{\text{res101}}$ [19]	ori †	**0.87**	*0.63*
	NC-Net$_{\text{res101}}$ [22]	-	0.85	0.60
	NC-Net$_{\text{res101}}$ [22]	ori †	0.85	0.58
	PWarpC-NC-Net$_{\text{res101}}$	ori	*0.86*	0.61

Table 10. State-of-the-art comparison on the Caltech-101 dataset. All approaches are trained on the PF-Pascal dataset. **S** denotes strong supervision using keypoint annotation, **M** refers to using ground-truth object segmentation mask, **U** is fully unsupervised requiring only single images, and **W** refers to weakly-supervised with image-level class labels. Each method evaluates with ground-truth annotations resized to a specific resolution. However, using different ground-truth resolutions leads to slightly different results. We therefore use the standard setting of evaluating on the original resolution (**ori**) and gray the results computed at a different size. When needed, we re-compute metrics of baselines using the provided pre-trained weights, indicated by †. For each of our PWarpC network, we compare to its corresponding baseline within the dashed-lines. Best and second best results are in red and blue respectively.

I.7. Qualitative results

In Fig. 6 and 7, we show example predictions of baseline SF-Net compared to our weakly-supervised approach PWarpC-SF-Net on the PF-Pascal, PF-Willow and SPair-71K datasets. Similarly, we present example predictions for NC-Net and PWarpC-NC-Net in Fig. 8-9. Our weakly-supervised Probabilistic Warp Consistency approach finds significantly more correct matches than the baseline in both cases, for a variety of object classes.

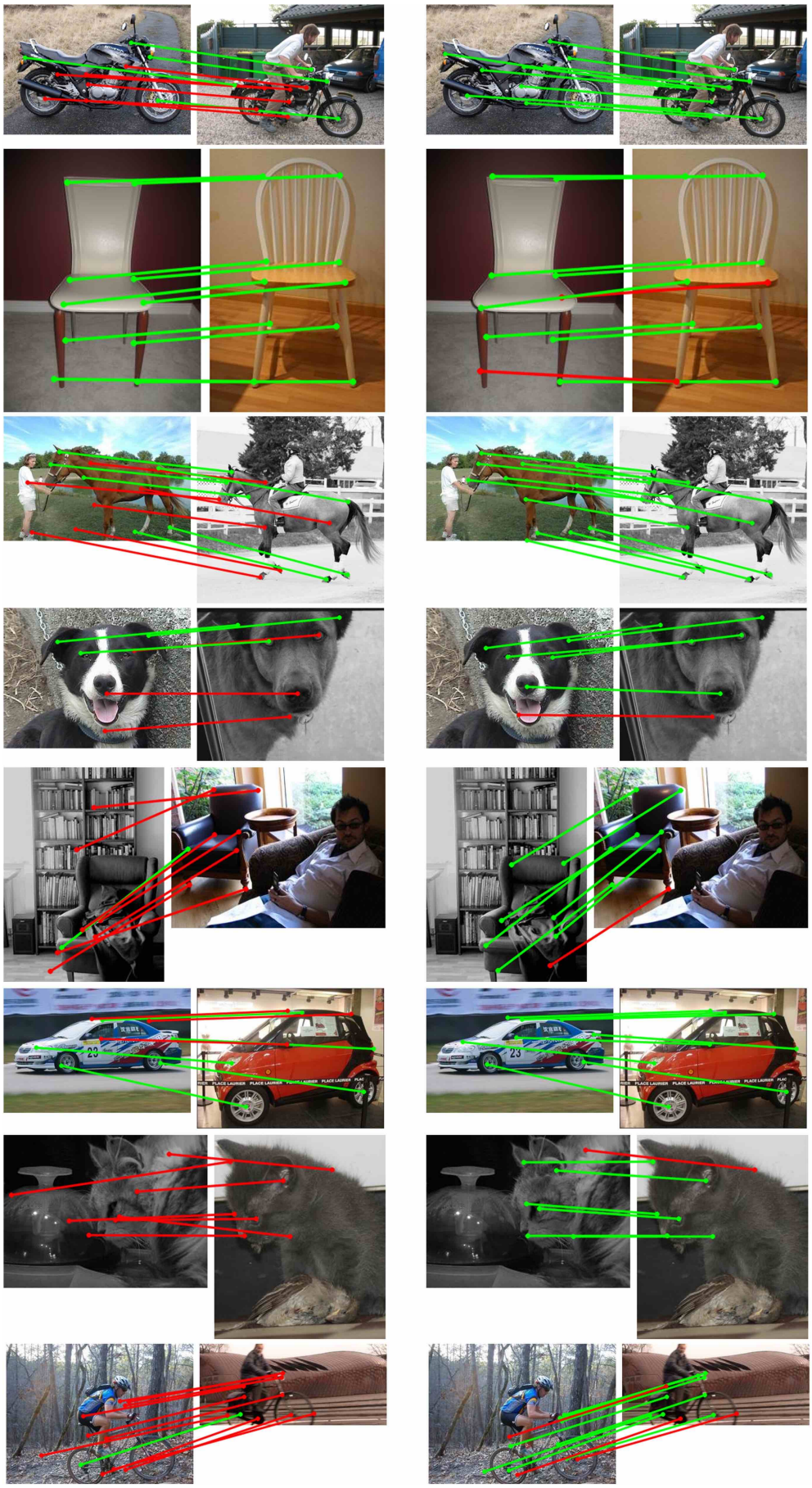


Figure 6. Example predictions on PF-Pascal and PF-Willow, of baseline SF-Net [9] (**left**) compared to our weakly-supervised PWarpC-SF-Net (**right**). Green and red line denotes correct and wrong predictions, respectively, with respect to the ground-truth.

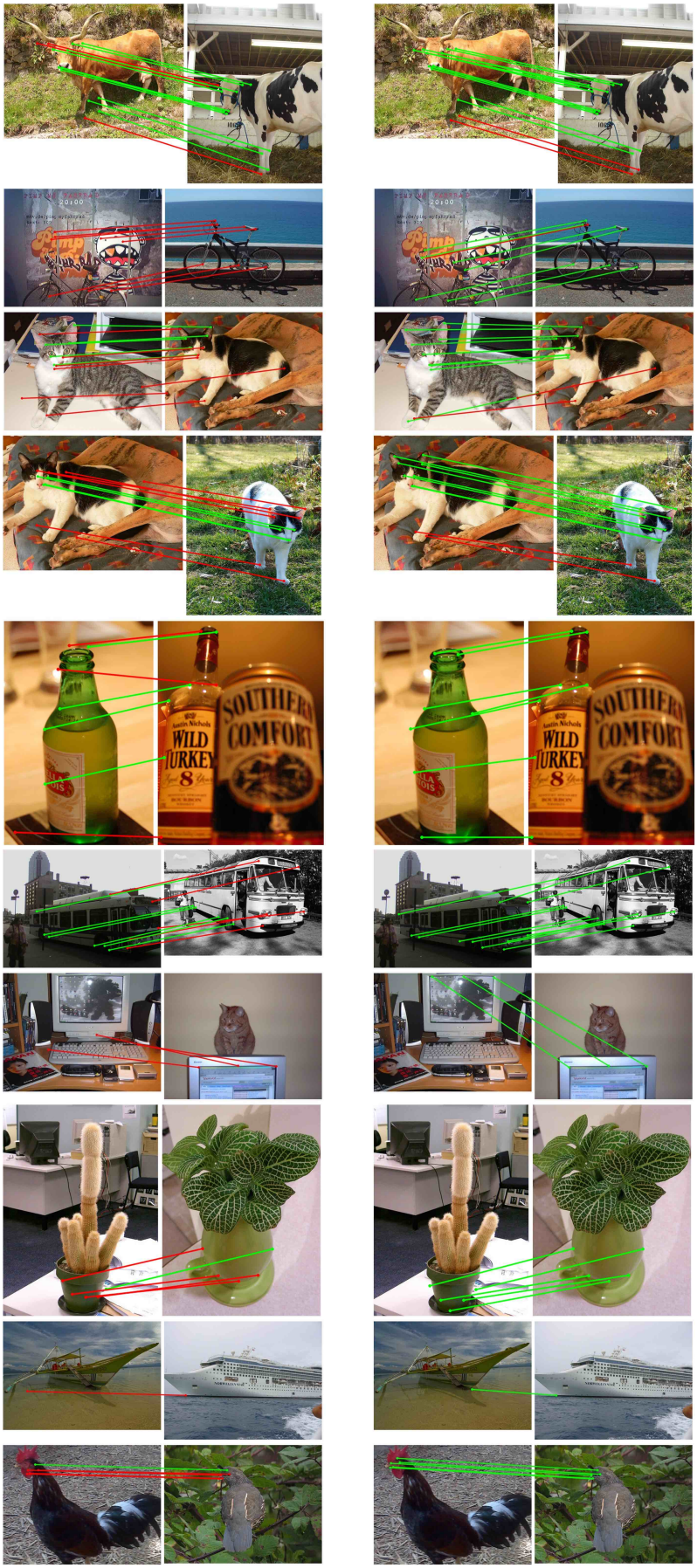


Figure 7. Example predictions on SPair-71K, of baseline SF-Net [9] (**left**) compared to our weakly-supervised PWarpC-SF-Net (**right**). Green and red line denotes correct and wrong predictions, respectively, with respect to the ground-truth.

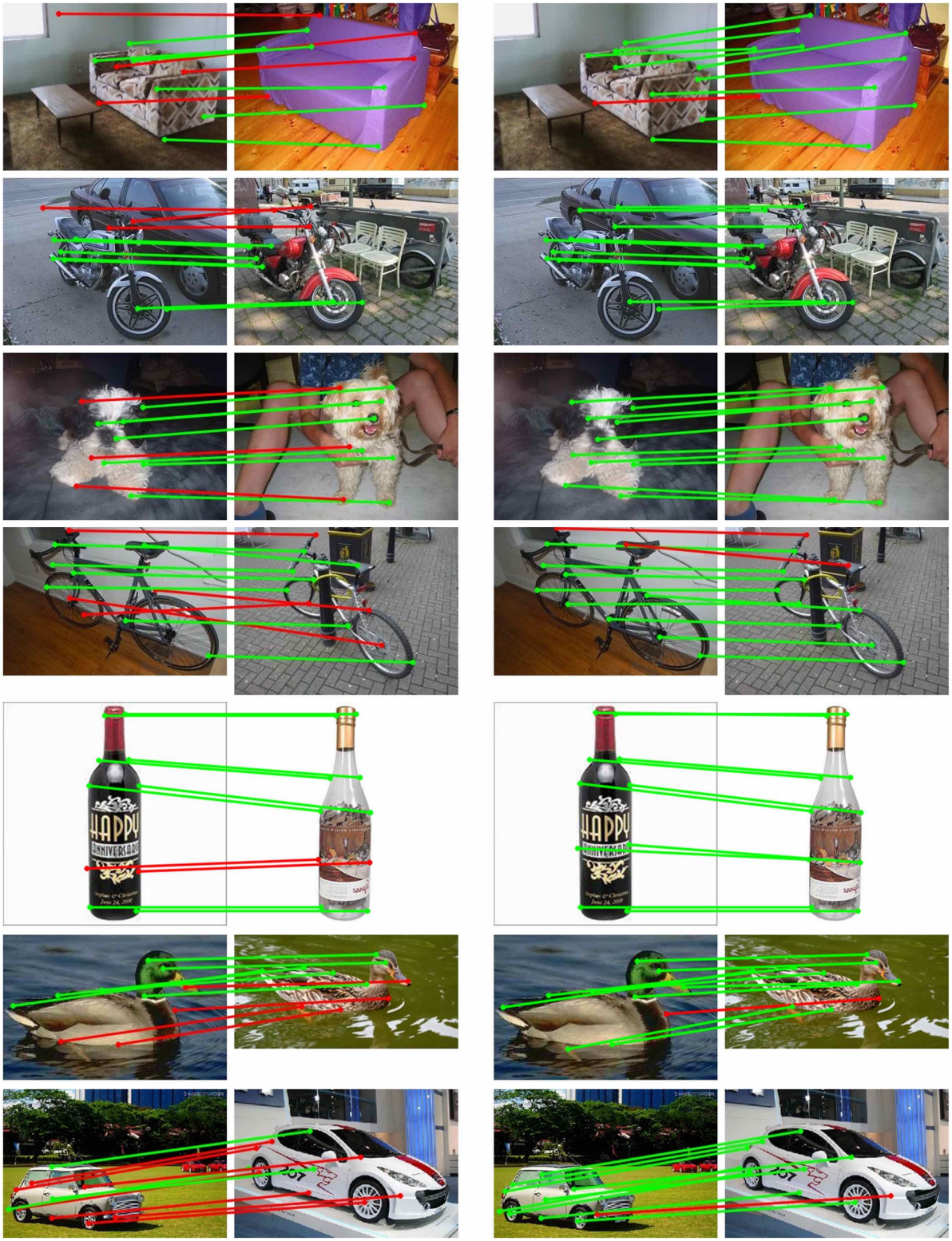


Figure 8. Example predictions on PF-Pascal and PF-Willow, of baseline NC-Net [22] (**left**) compared to our weakly-supervised PWarpC-NC-Net (**right**). Green and red line denotes correct and wrong predictions, respectively, with respect to the ground-truth.

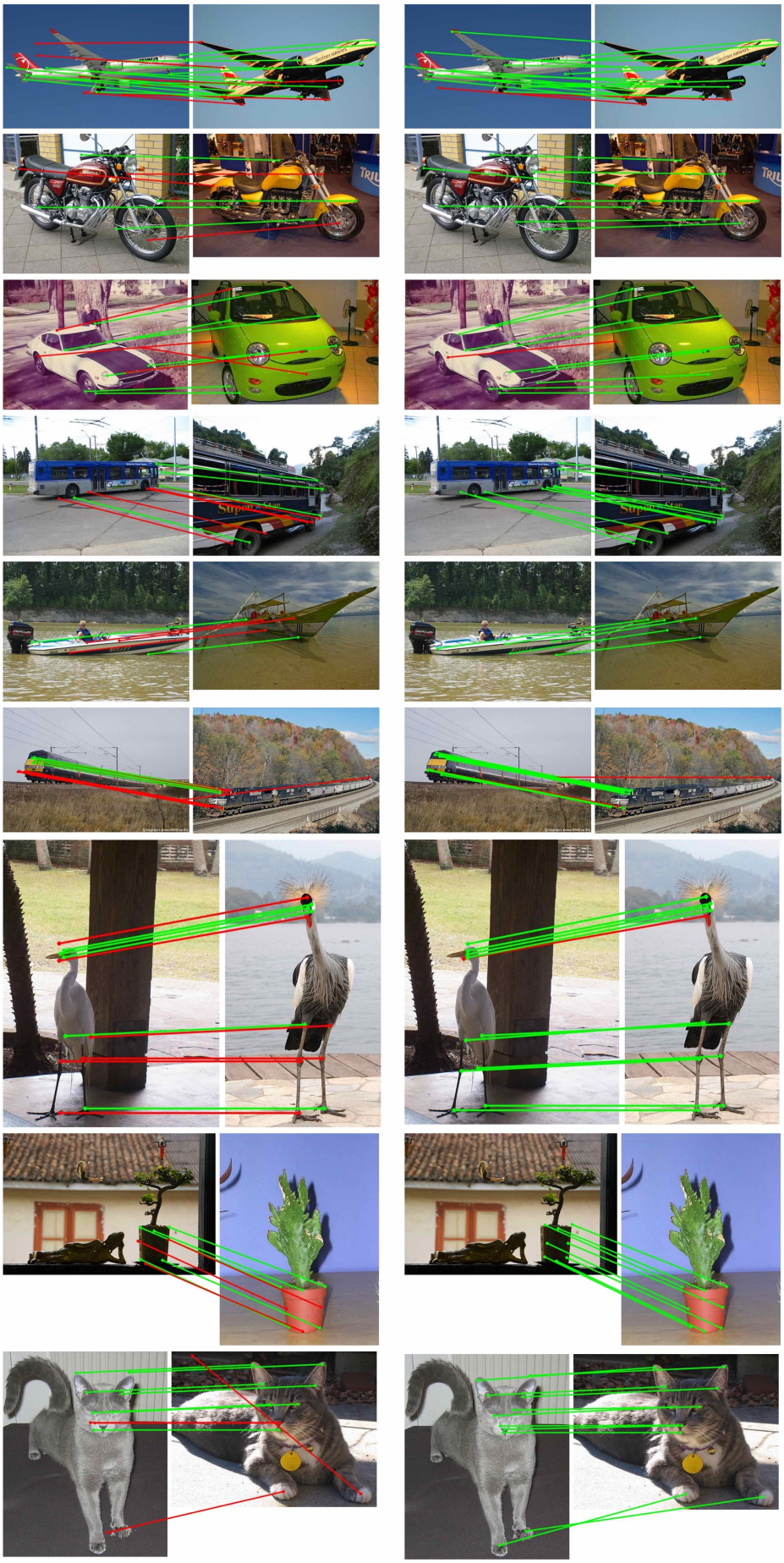


Figure 9. Example predictions on SPair-71K, of baseline NC-Net [22] (**left**) compared to our weakly-supervised PWarpC-NC-Net (**right**). Green and red line denotes correct and wrong predictions, respectively, with respect to the ground-truth.

References

[1] Oisin Mac Aodha, Ahmad Humayun, M. Pollefeys, and G. Brostow. Learning a confidence measure for optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1107–1120, 2013. 12

[2] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Semantic correspondence with transformers. *NeurIPS*, 2021. 6, 10, 15

[3] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 4, 5, 7, 8, 10

[4] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(7):1711–1725, 2018. 1, 2, 4, 5, 7, 8, 9, 10

[5] Kai Han, Rafael S. Rezende, Bumsub Ham, Kwan-Yee K. Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1849–1858, 2017. 11, 15

[6] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1647–1655. IEEE Computer Society, 2017. 12

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3, 6

[8] R. Fergus L. Fei-Fei and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Recognition and Machine Intelligence. In press.* 1

[9] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2278–2287, 2019. 3, 4, 10, 12, 14, 15, 16, 17

[10] Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N. Sinha. Patchmatch-based neighborhood consensus for semantic correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 13153–13163. Computer Vision Foundation / IEEE, 2021. 10

[11] Shuda Li, Kai Han, Theo W. Costain, Henry Howard-Jenkins, and Victor Prisacariu. Correspondence networks with adaptive neighbourhood consensus. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10193–10202. Computer Vision Foundation / IEEE, 2020. 1

[12] Xin Li, Deng-Ping Fan, Fan Yang, Ao Luo, Hong Cheng, and Zicheng Liu. Probabilistic model distillation for semantic correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 7505–7514. Computer Vision Foundation / IEEE, 2021. 10

[13] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 4462–4471. Computer Vision Foundation / IEEE, 2020. 10

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 7

[15] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, New Orleans, Louisiana, Feb. 2018. 12

[16] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 2940–2950. Computer Vision Foundation / IEEE, 2021. 10

[17] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *ICCV*, 2019. 10, 15

[18] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Spair-71k: A large-scale benchmark for semantic correspondence. *CoRR*, abs/1908.10543, 2019. 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 14

[19] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*, pages 346–363, 2020. 4, 5, 6, 7, 10, 12, 15

[20] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 39–48, 2017. 10, 15

[21] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6917–6925, 2018. 10, 15

[22] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.*, pages 1658–1669, 2018. 4, 5, 6, 10, 12, 15, 18, 19

[23] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 8

[24] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, pages 367–383, 2018. 10, 15

[25] Tatsunori Taniai, Sudipta N. Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4246–4255, 2016. 1, 4, 5, 7, 8, 10

[26] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 2021. 12

[27] Prune Truong, Martin Danelljan, Radu Timofte, and Luc Van Gool. Pdc-net+: Enhanced probabilistic dense correspondence network. In *Preprint*, 2021. 12

[28] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Warp consistency for unsupervised learning of dense correspondences. In *IEEE/CVF International Conference on Computer Vision, ICCV*, 2021. 2, 4, 5, 10, 12, 13

[29] Anne S. Wannenwetsch, Margret Keuper, and Stefan Roth. Probflow: Joint optical flow and uncertainty estimation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 1182–1191, 2017. 12

[30] Dongyang Zhao, Ziyang Song, Zhenghao Ji, Gangming Zhao, Weifeng Ge, and Yizhou Yu. Multi-scale matching networks for semantic correspondence. 2021. 10