Correspondence Transformers with Asymmetric Feature Learning and Matching Flow Super-Resolution

Yixuan Sun[1], Dongyang Zhao[2], Zhangyue Yin[2], Yiwen Huang[2],
Tao Gui[2], Wenqiang Zhang[1,2] and Weifeng Ge[2,†]
[1]Academy of Engineering & Technology, Fudan University, Shanghai, China
[2]School of Computer Science, Fudan University, Shanghai, China
wfge@fudan.edu.cn

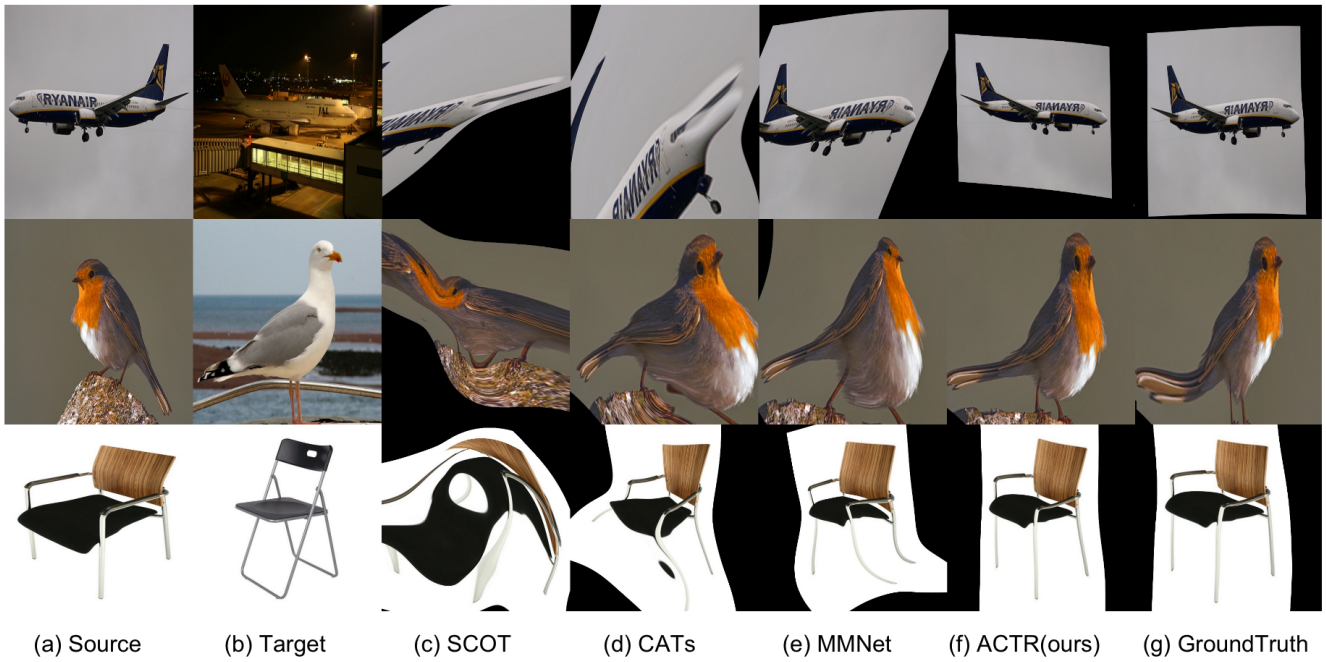

Figure 1. Dense visual correspondence generated by state-of-the-art algorithms, including SCOT [30], CATs [8], MMNet [49] and our asymmetric correspondence transformer. Images are warped with predicted key points using thin-plate splines algorithm [4].

Abstract

This paper solves the problem of learning dense visual correspondences between different object instances of the same category with only sparse annotations. We decompose this pixel-level semantic matching problem into two easier ones: (i) First, local feature descriptors of source and target images need to be mapped into shared semantic spaces to get coarse matching flows. (ii) Second, matching flows in low resolution should be refined to generate accurate point-to-point matching results. We propose asymmetric feature learning and matching flow super-resolution based on vision transformers to solve the above problems. The asymmetric feature learning module exploits a biased cross-attention mechanism to encode token features of source images with their target counterparts. Then matching flow in low resolutions is enhanced by a super-resolution network to get accurate correspondences. Our pipeline is built upon vision transformers and can be trained in an end-to-end manner. Extensive experimental results on several popular benchmarks, such as PF-PASCAL, PF-WILLOW, and SPair-71K, demonstrate that the proposed method can catch subtle semantic differences in pixels efficiently. Code is available on https://github.com/YXSUNMADMAX/ACTR.

†: Corresponding Authors

1. Introduction

Robust semantic matching methods aim to build dense visual correspondences between different objects or scenes from the same category regardless of the large variations in appearances and layouts. These algorithms have been widely exploited in various computer vision tasks, such as object recognition [12,48], cosegmentation [1,40], few-shot learning [20, 29], image editing [15, 18] and etc. Different from classical dense matching tasks, such as stereo matching [6,41] and image registration [2,19], semantic matching aims to find the visual consistency in image pairs with large intra-class appearance and layout variations.

State-of-the-art methods, such as Proposal Flow [16], NCNet [37], Hyperpixel Flow [34], CATs [8] and etc, typically extract features from backbones to measure point-to-point similarity in 4-D correlation tensors, and then refine these 4-D tensors to enforce neighborhood matching consistency. Despite that these algorithms have achieved impressive results, there are still two key issues that haven't been discussed thoroughly. First, how to learn feature representations appropriate for semantic correspondence? Second, how to enforce neighborhood consensus when the 4-D correlation tensors that are in high resolution? For the first problem, Hyperpixel Flow [34] selects feature maps in convolutional neural networks with Beam search [32], and MMNet [49] aggregates feature maps at different resolutions in a top-down manner to get feature maps in high resolution. For the second problem, VAT [20] firstly reduces the resolutions of 4-D correlation tensors with a 4-D convolution operation and then conducts shifted window attention [31] to further reduce the computational cost. Although these exploratory works brought a lot of new ideas, questions listed above still deserve to be discussed seriously.

In this paper, we aim to answer the above questions and propose a novel pipeline for semantic correspondence. Our pipeline consists of feature extraction based on pre-trained feature backbone [5, 17, 50], asymmetric feature learning, and matching flow super-resolution. Different from state-of-the-art methods [8, 34, 49] which directly calculate 4-D matching scores with refined generic backbone features, our core idea focuses on finding a shared semantic space where local feature descriptors of images can be aligned with their to-match counterpart. The asymmetric feature learning module reconstructs source image features with target image features to reduce domain discrepancy between the two thus avoiding reconstructing source and target image features synchronously as in [9, 28]. Meanwhile, to highlight important image regions in target images, we also use source images to identify discriminative parts of foreground objects. In this way, a specific feature space is found for every image pair to conduct semantic matching.

To avoid huge computational cost during neighborhood consensus enhancement, we map the matching information hidden in 4-D correlation matrices to 2-D matching flow maps through the soft argmax [25]. By reducing the dimension of the optimization goal from 4-D to 2-D, computational cost is alleviated drastically. To achieve pixel-level correspondence, we conduct matching flow super-resolution to enhance neighborhood consensus and improve matching accuracy at the same time. We find that the proposed method works quite well in conjunction with transformer feature backbones, such as MAE [17], DINO [5], and iBOT [50], so we call the proposed method asymmetric correspondence transformer, written as ACTRansformer, and train it end-to-end. We summarize our contributions as follows.

• We introduce a novel pipeline for semantic correspondence which contains generic feature extraction, asymmetric feature learning, and matching flow super-resolution. By conducting asymmetric feature learning, we extract specific features for every image pair and thus get more accurate correspondences. Besides, we replace the 4-D correlation refinement with the 2-D matching flow super-resolution, which saves computational cost greatly.

• We propose asymmetric feature learning that can project features of image pairs into a shared feature space easily and reduce the feature ambiguity at the same time. For matching flow super-resolution, we conduct multi-path super-resolution to benefit from different matching tensors and acquire significant improvements.

• Experiments on several popular benchmarks indicate that the proposed ACTRansformer outperforms previous state-of-the-art methods by clear margins. Qualitative results are presenred in Figure 1.

2. Related Work

Symmetric Cross Attention. Symmetric cross attention can capture dependencies between image pairs efficiently and has been widely used in various computer vision tasks, such as image matching [38], object tracking [7], video segmentation [22] and style transfer [39]. MixFormer [9] utilized cross attention structure to conduct mutual interaction of target template and search area for object tracking. Works in [28, 46] introduced bi-directional cross attention to map images in the left and right views into a shared feature space, and thus generate a much more accurate matching flow. And in video style transfer, the cross attention mechanism was exploited to learn a transfer matrix between image pairs for makeup transfer and removal [39]. While in this paper, we just adopt an asymmetric feature learning scheme that avoids mapping image pairs into shared semantic spaces symmetrically and just focuses on reconstructing source features with target references, which makes semantic matching relatively easier to learn.

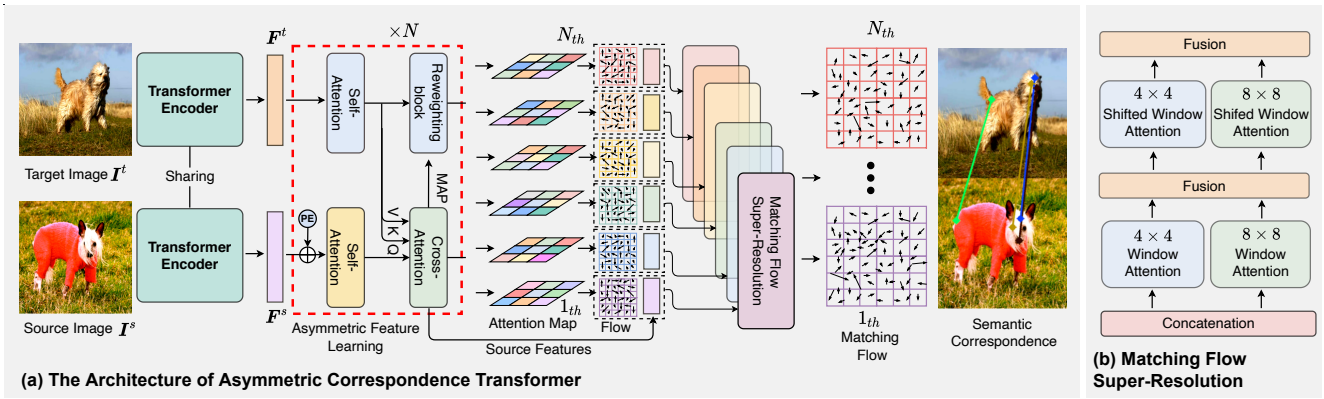

Figure 2. **Illustration of Asymmetric Correspondence TRansformer (ACTR).** (a) The proposed asymmetric correspondence transformer contains a pre-trained vision transformer backbone, an asymmetric semantics alignment module, and a multi-path matching flow super-resolution module. (b) The matching flow super-resolution module concatenates coarse matching flows and its features and feeds them through a set of windowed self-attention to get refined matching flows.

Matching Flow/Correspondence Refinement. Matching flow or correspondence refinement is vital for image matching. STTR [28] proposed a context adjustment layer that used raw images as additional features to guide disparity refinement. HITNet [41] relies on a U-Net-like structure to refine estimated disparity through differentiable 2D geometric propagation and warping mechanisms. For semantic correspondence, NCNet [37] enforced neighborhood consensus with 4-D convolution to refine source-target 4-D correlation tensors. Recent works [49] find that a 4-D matching score in higher resolutions can improve the accuracy of correspondence efficiently. However, higher resolutions will lead to rapid growth in computational cost and reach a bottleneck in matching performance. In this paper, we turn the 4-D correlation optimization into 2-D matching flow super-resolution, thus reducing the computational cost and making our framework much more flexible.

3. Asymmetric Correspondence Transformer

In this section, we introduce our end-to-end ACTRansformer for learning dense visual correspondences from image pairs with sparse annotations. Given a pair of images, our model: (1) aligns local patch features of the source image with its target counterpart, which can reduce local semantics discrepancy between source and target images to get a much more reliable matching flow; (2) estimates high-resolution matching flows from low-resolution inputs to distinguish subtle differences in neighborhood pixels and keep spatial consistency of correspondences. An overview of the ACTRansformer architecture is presented in Figure 2 (a). For an image pair $\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right)$ and the ground truth of their matched key points $\mathcal{M}_{gt} = \{m_i = (\boldsymbol{p}_i^s, \boldsymbol{p}_i^t) \,|i = 1, ..., K\}$, images are divided into patches as in ViT [11] and sent into ACTRansformer Φ to produce the matching flow Δ.

3.1. Feature Extraction with ViT

Vision transformer (ViT [11]) and its variants [31,44,47] have achieved impressive results on various computer vision tasks [27, 31, 45]. We adopt ViT with iBOT pre-training [50] as our feature backbone. For an input image $\boldsymbol{I}$ with the resolution $H \times W$, ViT reshapes $\boldsymbol{I}$ into a set of flattened patch tokens $\boldsymbol{T}_p \in \mathbb{R}^{N \times D}$. To get the global representation of $\boldsymbol{I}$, $\boldsymbol{T}_p$ is usually augmented with a learnable $[cls]$ token. So there are $N + 1$ tokens to describe one image. We partition an image into 16×16 patches. Then, there are $N = HW/16^2$ patch tokens, each of which represents parts of objects or scenes. The overall $N + 1$ tokens are passed through a consecutive set of multi-head self-attention blocks and get their semantic representations. Given an image pair $\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right)$, their features are denoted as $\boldsymbol{F}^s \in \mathbb{R}^{(h_s \times w_s + 1) \times c}$ and $\boldsymbol{F}^t \in \mathbb{R}^{(h_t \times w_t + 1) \times c}$. With those token embeddings, we conduct the subsequent semantic correspondence learning.

3.2. Asymmetric Feature Learning

Figure 2 (a) shows that source and target images are processed with different modules in an asymmetric feature learning block: (1) the source branch contains a robust self-attention module and a cross-attention module; (2) the target branch conducts salient target reweighting to enhance foreground patches. The source branch needs to identify the subtle differences between patches with high semantic similarities. Hence a learnable positional encoding (PE) [14] is added to source features to stress the location uniqueness. Then these tokens are sent into a standard transformer block [43]. The target branch aims to distinguish foreground patches and remove background clutters. We use all source tokens to reweight target tokens by projecting all source tokens into queries and target tokens into keys and

conduct attention matrix calculation (MAP). Attention values of different heads are summed up to produce the overall attention $\boldsymbol{A}_l^t \in \mathbb{R}^{(h_t w_t+1)\times(h_s w_s+1)}$. For every target token, attention values across all source tokens are added together and we get the attention vector $\boldsymbol{v}_l^t \in \mathbb{R}^{(h_t w_t+1)}$. After broadcasting $\boldsymbol{v}_l^t$ to c channels,the attention matrix as $\boldsymbol{v}_l^t \in \mathbb{R}^{(h_t w_t+1)\times c}$. For target features $\boldsymbol{z}_{l-1}$ at the $(l-1)$-th layer, the reweighting process are performed as below:

$$\begin{aligned}
\boldsymbol{z}_l' &= \mathrm{MultiHead}(\mathrm{LN}(\boldsymbol{z}_{l-1}), \mathrm{LN}(\boldsymbol{z}_{l-1}), \mathrm{LN}(\boldsymbol{z}_{l-1})), \\
\boldsymbol{z}_l'' &= \boldsymbol{z}_l' + \boldsymbol{z}_{l-1}, \\
\boldsymbol{z}_l^* &= \boldsymbol{z}_l'' \otimes \boldsymbol{v}_l^t, \\
\boldsymbol{z}_l &= \mathrm{FFN}(\mathrm{LN}(\boldsymbol{z}_l^*)) + \boldsymbol{z}_l^*,
\end{aligned} \tag{1}$$

where $\otimes$ stands for element-wise multiplication, $\boldsymbol{v}_l^t$ is used to align with $h_t w_t + 1$ tokens of $\boldsymbol{z}$, LN stands for layer normalization [3], MultiHead stands for multi-head self attention [43] and FFN stands for feed-forward network [43]. With foreground target tokens enhanced, we reconstruct source tokens with target tokens. This is implemented with an attention function $\mathrm{Attention}(Q, K, V)$ as in [43], where source tokens provide Q, and target tokens provide K and V (Figure 2 (a)). Several (usually $M = 6$) asymmetric feature learning blocks are concatenated to increase the discriminative ability of local feature descriptors.

3.3. Generating Matching Flow from Features

With feature maps $\widetilde{\boldsymbol{F}}_s \in \mathbb{R}^{(h_s\times w_s+1)\times c}$ and $\widetilde{\boldsymbol{F}}_t \in \mathbb{R}^{(h_t\times w_t+1)\times c}$ from the previous section, we need to generate a dense matching flow $\Delta_{c|s\mapsto t}$ from the source to target. Here we remove the $[cls]$ token for the subsequent flow estimation. Each of these feature maps corresponds to $h_s \times w_s$ (or $h_t \times w_t$) grids of c-dimensional local features. Different from that in [25, 37], to build pairwise correspondence between $\widetilde{\boldsymbol{F}}_s$ and $\widetilde{\boldsymbol{F}}_t$, matching scores are computed through a multi-head attention scheme, resulting in a 4-dimensional correlation map of $h_s \times w_s \times h_t \times w_t$:

$$\mathcal{C}_{st} = \frac{1}{H}\sum_h \mathrm{softmax}\left(\frac{\left(\widetilde{\boldsymbol{F}}_s W_h^Q\right)\left(\widetilde{\boldsymbol{F}}_t W_h^K\right)^T}{\sqrt{c}}\right), \tag{2}$$

where H (=8) is the head number. For every point on the source feature map, directly applying the argmax function over a $h_t \times w_t$ correlation map can get the best matches and then generate matching flows. However, the argmax function is discrete and not differentiable. So we adopt the soft argmax in [25] to generate the raw matching flows.

$$\Delta_{c|s\mapsto t} = \mathrm{soft\_argmax}\left(\mathcal{C}_{st}; h_s, w_s, h_t, w_t\right). \tag{3}$$

3.4. Matching Flow Super-Resolution

Though the above matching flow $\Delta_{c|s\mapsto t}$ describes source-target semantic flow in high accuracy, it is not adequate to distinguish subtle appearance differences among neighborhood pixels or regions. Inspired by the idea of super-resolution [13, 42] which hallucinates high resolution details from low resolution inputs, we develop the matching flow super-resolution block to upscale the spatial resolution of $\Delta_{c|s\mapsto t}$ by 4 times, as shown in Figure 2 (b). There are two inputs: coarse matching flow $\Delta_{c|s\mapsto t} \in \mathbb{R}^{h_s\times w_s\times 2}$, source token features $\widetilde{\boldsymbol{F}}_s \in \mathbb{R}^{(h_s\times w_s+1)\times c}$ with $[cls]$ token removed. We use bilinear interpolation to upsample the coarse matching flow and source feature map to the resolution of $4h_s \times 4w_s$. These two tensors are concatenated along the channel dimension to generate the super-resolution input $\boldsymbol{F} \in \mathbb{R}^{4h_s\times 4w_s\times d}$.

The matching flow super-resolution block simply contains several transformer blocks with shifted window attention [31] to generate high resolution matching flow. To incorporate diversities in representation learning, we utilize two branches of windowed attention whose kernel size is 4×4 and 8×8 respectively. Then their output feature maps are concatenated and sent into a 3×3 convolution fusion layer. We then shift the window by stride 2 to conduct two shifted window attention in the same manner and fuse their features with a convolution layer again. Finally, after several rounds of window attention, we get the upscaled fine matching flow $\Delta_{f|s\mapsto t}$ in the resolution of $4h_s \times 4w_s$.

Multi-Path Super-Resolution. To enhance the matching flow during super-resolution, we generate matching flows after each cross attention block and conduct super-resolution respectively. We averaged all matching flows of different super-resolution branches to get the final flow output. Figure 3 visualizes matching flows generated by different cross-attention blocks in asymmetric feature learning. We find that the multi-path aggregation improves the matching accuracy significantly.

3.5. Training

For training, there are only sparse annotations on popular semantic correspondence benchmarks, such as PF-PASCAL [16] and SPair-71K [34]. To augment these sparse key point pairs, we followed work [8] and generating pseudo semantic flow [16] to supervise the training of ACTRansformer. For each key point, we estimate the matching flow of its 35×35 neighborhood for subsequent training and denote remaining regions as unvisited with a mask $\boldsymbol{S}^s$ as that in [8, 20]. We resize the matching flow to the resolution of $4h_s \times 4w_s$, and generate the matching flow $\Delta_{gt}\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right)$ for $\boldsymbol{I}^s$ and $\boldsymbol{I}^t$. Then our goal becomes finding the optimal parameters of $\Phi = (\Phi_b, \Phi_a, \Phi_s)$ to minimize

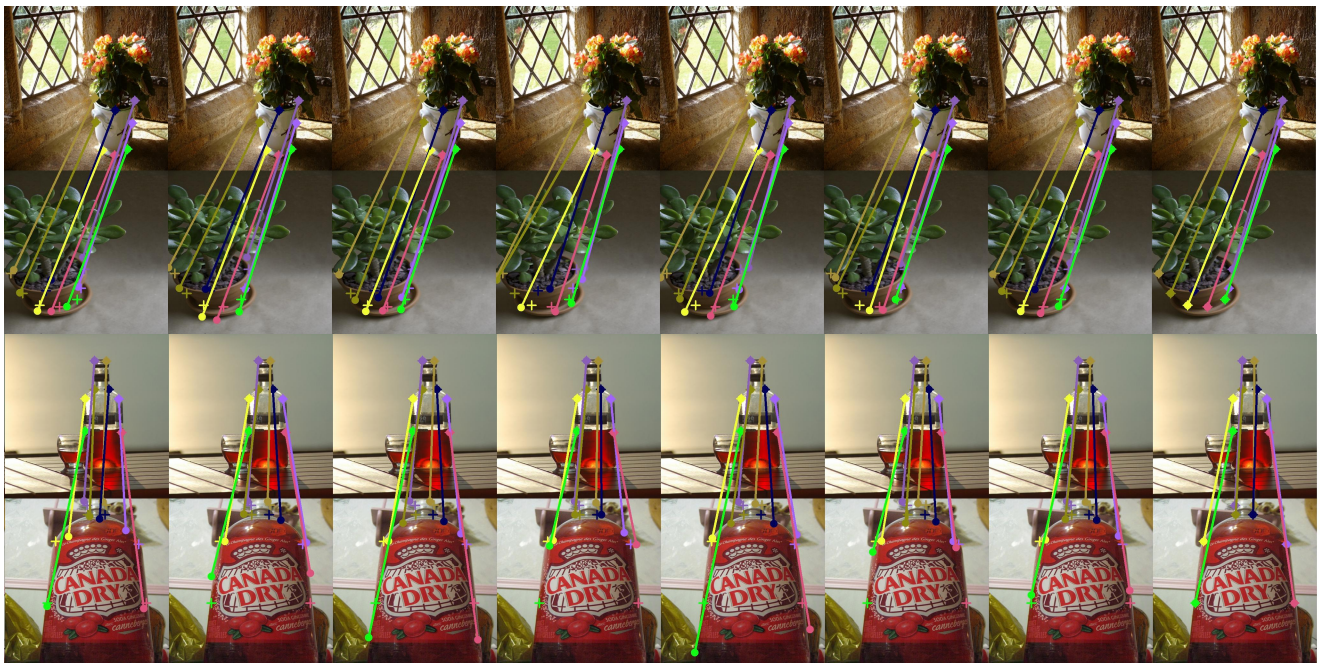

Figure 3. **Matching results of different paths.** Results show that the multi-path design can summarize complementary information from cross-attention blocks to refine flows. From left to right, matching results from path 1-6 (a-f), fused (g) and ground truth (h) are given.

the following objective function:

$$\mathcal{L} = \frac{1}{N} \sum_{(\boldsymbol{I}^s, \boldsymbol{I}^t) \in \mathcal{D}} \frac{\boldsymbol{S}^s \otimes \left\| \Phi\left(\boldsymbol{I}^s, \boldsymbol{I}^t; \theta\right) - \Delta_{gt}\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right) \right\|^2}{|\boldsymbol{S}^s|}, \quad (4)$$

where $|\boldsymbol{S}^s|$ is the number of non-zero elements in $\boldsymbol{S}^s$, θ is the learnable parameters in all modules of ACTR, N is the number of samples in training set $\mathcal{D}$, $\boldsymbol{S}^s$ and $\Delta_{gt}\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right)$ are the matching mask and flow generated by $\mathcal{M}_{gt}$.

4. Experiment

Datasets. We conducted experiments on several datasets: SPair-71K [34], PF-PASCAL [16], PF-Willow [16]. SPair-71K [34] is a challenging large-scale benchmark that contains 70,958 image pairs of 18 categories with large intra-class variations, scale differences, occlusion, and truncation. We used the same split as previous works [8, 49] in which for training, validation, and testing split, 53,340, 5,384, and 12,234 image pairs were used respectively. PF-PASCAL [16] contains 1,351 image pairs from 20 classes, we split them as approximately 700, 300, 300 for train, validation, and test process following the work [49] and PF-WILLOW [16] with 900 image pairs from 10 classes are used to test the generalization ability of models trained on PF-PASCAL [16].

Evaluation Metric. To evaluate the performance, we employ PCK@α (percentage of correct keypoints with threshold α) as in previous works [8, 21, 26, 30, 34, 49]. A predicted keypoint is considered *correct* when it falls into the circle of radius $\alpha \times d$ centering at its ground-truth counterpart, where d is the longer side of the image in PF-PASCAL (denoted as α_{img}) or object bounding box in SPair-71K and PF-WILLOW (denoted as α_{bbox}). For PF-WILLOW, metric α_{bkp} is used with d standing for the maximum distance of annotated key points.

Implementation Details. We use ViT-B/16 pre-trained with iBOT on ImageNet [10] 1K as our backbone. We test two input resolutions as 256×256 and 512×512. There are 6 asymmetric semantics alignment blocks whose hyper-parameters are the same as the feature backbone. For matching flow super-resolution, the fusion module is a cascaded two layers 3×3 convolution with 32 and 2 kernels. The feature dimension and the number of the attention head in the 4×4 and 8×8 shifted window attention block are 96 and 8 respectively. During training, we used an AdamW optimizer with 0.05 weight decay. The learning rates are set as 6e-6 and 6e-5 for the backbone and the following two modules respectively. Our model is implemented on PyTorch [36] and trained on two NVidia TITAN RTX GPUs. The batch size is set to 8 for 256×256 resolution and 6 for 512×512 for all the experiments. The training converged within 50 and 10 epochs for PF-PASCAL [16] and SPair-71K [34] dataset respectively. Note that we exploit the model with ViT-B/16 backbone pre-trained on ImageNet

Table 1. Quantitative results on standard benchmarks. Higher PCK is better. The best results are in bold, and the second-best results are underlined. *: *Method used dynamic resolution with the listed maximum threshold. Multi-Scale: whether to employ multi-scale features.*

Methods	Backbone	Input Resolution	Multi-Scale	SPair-71K α : bbox 0.1	PF-PASCAL α : img 0.05	PF-PASCAL α : img 0.1	PF-PASCAL α : img 0.15
SCOT [30]	ResNet-101	$300 \times 300^*$	✓	35.6	63.1	85.4	92.7
DHPF [35]	ResNet-101	240×240	✓	37.3	75.7	90.7	95
CHM [33]	ResNet-101	256×256	×	46.3	80.1	91.6	94.9
CATs [8]	ResNet-101	256×256	✓	49.9	75.4	92.6	96.4
MMNet-FCN [49]	ResNet-101	224×320	✓	50.4	<u>81.1</u>	91.6	95.9
TransforMatcher [24]	ResNet-101	240×240	✓	53.7	80.8	91.8	-
CATs [8]	iBOT-B	256×256	✓	55.2	77.8	93.1	<u>96.8</u>
TransforMatcher [24]	iBOT-B	240×240	✓	<u>57.9</u>	77.3	<u>93.3</u>	96.6
Baseline	iBOT-B	256×256	×	57.7	78.9	93.2	96.5
ACTR	iBOT-B	256×256	×	**62.1**	**81.2**	**94.0**	**97.0**
VAT [20]	ResNet-101	512×512	✓	54.2	-	92.3	-
VAT [20]	iBOT-B	512×512	✓	59.0	73.0	<u>92.6</u>	**96.7**
Baseline$_h$	iBOT-B	512×512	×	<u>61.6</u>	<u>79.3</u>	91.6	95.9
ACTR$_h$	iBOT-B	512×512	×	**65.4**	**82.0**	**93.5**	**96.7**

Table 2. Generalizability evaluation on PF-WILLOW dataset with PF-PASCAL trained model. ‡ stands for the method implemented with iBOT-B backbone same with ACTR.

Methods	PF-WILLOW α : bbox 0.05	PF-WILLOW α : bbox 0.1	PF-WILLOW α : bkp 0.05	PF-WILLOW α : bkp 0.1
DHPF [35]	49.5	77.6	-	71.0
CHM [33]	52.7	79.4	-	69.6
CATs [8]	50.3	79.2	40.7	69.0
SCOT [30]	-	-	47.8	76.0
TransforMatcher [24]	-	65.3	-	76.0
CATs‡	59.4	86.3	51.1	79.5
TransforMatcher‡	57.0	84.3	48.8	78.3
ACTR	**60.3**	**87.2**	**52.6**	**79.9**

1K with resolution 256×256 as our base model, since it has learnable parameters comparable with other methods.

Baseline Models. We add 6 transformer blocks with symmetric cross attention [23,46] after the iBOT-B backbone to conduct baseline experiments. We adopt the multi-head attention scheme for 4-D correlation computation and use the bilinear interpolation for upsampling the matching flow. We name this model Baseline which contains 171.13 million learnable parameters almost the same as ACTR. Except for the network structure, we use identical settings with ACTR to train and test the Baseline. We also implemented several state-of-the-art methods such as CATs [8] and TransforMatcher [24] for 256×256 input resolution as well as VAT [20] for 512×512 resolution using iBOT-B backbone like ACTR. This allows us to compare matching head designs with the same feature quality.

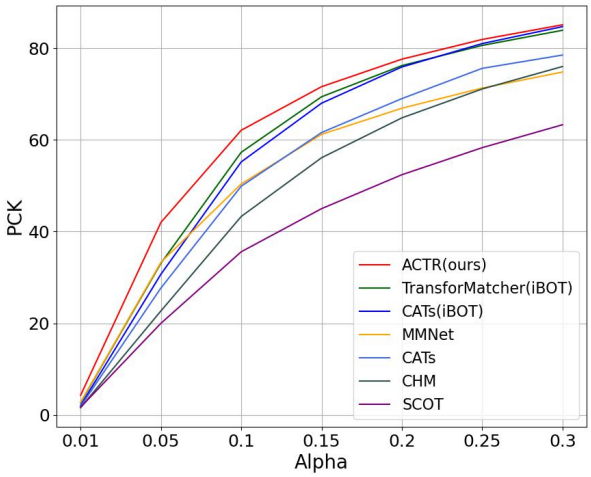

Figure 4. The PCK@α curves of our method and previous works on SPair-71K [34]. Our method performs better than other methods with small error thresholds (small α).

4.1. Comparison with State-of-The-Art

We report our quantitative results in comparison with other state-of-the-art methods on two popular benchmarks, SPair-71K [34] and PF-PASCAL [16] in Table 1. To ensure a fair comparison, we explicitly list backbone types, input image sizes, and whether to employ multi-scale features with each method. When in resolution 256×256, our base model ACTR outperforms other methods with 62.1% PCK@0.1 on SPair-71K, and 94.0% PCK@0.1 and 97.0% PCK@0.15 on PF-PASCAL, better than previous state-of-the-art algorithms. Especially, it is 4.2% higher than TransforMatcher [24] with the same iBOT-B backbone

Table 3. Comparison of efficiency between ACTR and state-of-the-art methods. We compared total learnable parameters, memory consumption as well as run time between our ACTR and state-of-the-arts. All models are tested on NVidia TITIAN RTX GPU with 24GB memory. ‡ stands for the method implemented with iBOT-B backbone same with ACTR.

Methods	Total Parameters (M)			Memory (GB)	Run time (ms)	SPair-71K
	Backbone	Matching head	Total			$\alpha_{bbox} = 0.1$
SCOT [30]	44.5	-	44.5	4.6	133.5	35.6
CATs [8]	44.5	4.7	49.2	2.0	45.4	49.9
MMNet-FCN [49]	54.4	10.3	64.7	5.4	258.6	50.4
TransforMatcher [24]	87	0.9	87.9	2.7	54.0	53.7
CATs‡ [8]	85.0	5.7	90.7	2.8	54.2	55.2
TransforMatcher‡ [24]	85.0	1.6	86.6	2.4	48.5	57.9
ACTR-S	21.0	23.2	44.2	1.9	43.7	55.8
ACTR	85.0	87.8	172.8	3.9	84.1	62.1
VAT [20]	44.5	3.3	48.4	3.9	141.2	54.2
VAT‡ [20]	85.0	4.2	89.2	4.4	172.4	59.0
ACTR$_h$-S	21.0	25.2	46.2	2.3	64.8	59.7
ACTR$_h$	85.0	91.9	176.9	4.7	121.5	65.4

on PCK@0.1 for SPair-71K. Moreover, ACTR also shown an overall improvement than previous works at different evaluation standards reported on Figure 4. It indicates that ACTR has a strong ability in capturing complex appearance and layout variations.

When we compare with VAT [20] and VAT with iBOT backbone(VAT‡) in higher resolution 512×512, ACTR$_h$ get 65.4% PCK@0.1 on SPair-71K, which is 11.2% and 6.4% higher than VAT and VAT‡. We attribute these results to that VAT focus on 4-D matching correlation refinement in a symmetric manner which makes the backbone feature fine-tuning more difficult. Since ACTR and ACTR$_h$ are only different in input resolution, we find that the input resolution can directly impact the performance of models. Experiments on PF-PASCAL do not show obvious advantages as that in SPair-71K which may be attributed to insufficient training pairs (1,351). But transfer evaluation on PF-WILLOW (trained on PF-PASCAL) in Table 2 indicates that ACTR has a strong generalization ability.

Parameters, Memory, and Speed. We also conducted efficiency analysis in Table 3. In which we implemented ACTR-S and ACTR$_h$-S with a smaller ViT-S/16. Results show that our ACTR-S and ACTR$_h$-S have fewer learnable parameters than listed state-of-the-art methods but have impressive results. Especially, our ACTR$_h$-S outperforms the VAT‡ [20] which used iBOT feature, and our ACTR-S also has comparable performance to CATs‡ [8] and TransforMatcher‡ [24]. Meanwhile, they run faster than other models for comparison. If we adopt our ACTR model, the computational cost increases, but ACTR and ACTR$_h$ still have comparable performance to most of the previous methods in memory consumption and running time while getting impressive results.

Table 4. Ablations on components in ACTR.

Methods	SPair-71K $\alpha_{bbox} = 0.1$
ACTR	62.1
w/o source branch positional encoding	60.4 (1.7↓)
w/o target branch token reweighting	60.7 (1.4↓)
w/o asymmetric cross attention module	60.1 (2.0↓)
w/o multi-path super-resolution	61.0 (1.1↓)
w/o dual window flow refinement	60.6 (1.5↓)
w/o flow super-resolution module	59.0 (3.1↓)
Baseline	57.7 (4.4↓)

4.2. Ablation Studies

We designed several experiments of evaluating the design of asymmetric feature learning and flow super-resolution modules in ACTR and micro components (in Table 4). All the experiments were held on SPair-71K [34] dataset and using PCK metric and the α_{bbox} is set as 0.1.

Ablation on designed modules. To investigate whether asymmetric feature learning and matching flow super-resolution is necessary, we report performances in Table 4 by replacing each of them with the baseline transformer blocks. Performances decline 2.0%, 3.1% for each setting. It indicates that the asymmetric feature learning mechanism brings benefits while matching flow super-resolution can further improve accuracy.

In Table 4, we also zoom inside proposed modules and discuss how our design takes effect. For asymmetric feature learning, we conduct ablations on noise embedding learning in the source branch and token reweighting in the tar-

Figure 5. Visual comparison of matched key points. From left to right: (a) SCOT [30], (b) CATs [20], (c) MMNet [49], (d) ours ACTR and (e) the ground truth. Source and target images are in odd and even rows. Crosses denote destination key points on target images.

get branch. For the multi-path flow super-resolution module, we evaluated the effectiveness of fusing flows generated by different attention stages as well as the dual window design for each flow refinement layer. The performance dropped by 1.7% and 1.4% for two micro designs on the asymmetric feature learning stage. And the performance dropped by 1.1% and 1.5% for the design of multi-path super-resolution. We find that all these micro designs in ACTR can help to get better results, which validates our motivations of asymmetric feature learning and matching flow super-resolution.

4.3. Qualitative Results and Visual Analysis

We provide a visual comparison with point-level matches and image-level warping results using the model-predicted keypoint pairs. In Figure 5, the predicted key point pairs are linked with line segments while the ground-truth matching pairs are labeled with crosses. Compared with the state-of-the-art methods such as SCOT [30], CATs [8], and MMNet [49], our ACTR can clearly distinguish the subtle semantic differences which usually leads to mismatching for previous methods. In Figure 1, we warped the source image with the target one guide by predicted keypoint pairs. We also provide warp results using ground truth pairs as the reference. The result shows our ACTR can better build up global correspondence for an image pair. The three chosen image pairs have great variation in appearance and viewpoint, especially for the first-row objects that become difficult to observe in dark conditions. However, our method can well overcome these challenges and build up accurate matching for the whole instance in both image pairs.

5. Conclusions and Limitations

In this paper, we have proposed, for the first time, a fully Transformer-based pipeline for semantic matching which enables end-to-end training, dubbed ACTR. We have made two architectural designs: an asymmetric cross-attention mechanism to establish reliable flows and super-resolution flow refinement for more precise representation. Results have shown that our method surpasses the state-of-the-art in several benchmarks by a great margin and also made great improvement over our backbone baseline. Moreover, extensive studies are conducted to validate our choices and evaluate sources of performance gains.

Limitations. Several limitations are listed to our method. Firstly, our method is not versatile enough to make easy shifts to other mainstream backbones (ResNet, etc.), since too many tokens and feature channels are not affordable for fully attention blocks. Besides, the impacts of backbone pre-training strategies deserves thorough investigation in the future. Moreover, the extension of ACTR to multi-instance correspondence task requires further researches. This work does not have obvious negative societal impacts.

6. Acknowledgment

This work was supported by National Natural Science Foundation of China (NO. 62106051, No. 62072112), National Key R&D Program of China (No. 2020AAA0108301), Scientific and Technological Innovation Action Plan of Shanghai Science and Technology Committee (No. 22511101502, 22511102202), the Shanghai Pujiang Program Nos.21PJ1400600.

References

[1] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2(3):4, 2021. 2

[2] Moab Arar, Yiftach Ginger, Dov Danon, Amit H Bermano, and Daniel Cohen-Or. Unsupervised multi-modal image registration via geometry preserving image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13410–13419, 2020. 2

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization, 2016. 4

[4] Fred L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989. 1

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 2

[6] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5410–5418, 2018. 2

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8126–8135, 2021. 2

[8] Seokju Cho, Sunghwan Hong, Sangryul Jeon, Yunsung Lee, Kwanghoon Sohn, and Seungryong Kim. Cats: Cost aggregation transformers for visual correspondence. *Advances in Neural Information Processing Systems*, 34:9011–9023, 2021. 1, 2, 4, 5, 6, 7, 8

[9] Yutao Cui, Jiang Cheng, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention, 2022. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255, 2009. 5

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[12] Olivier Duchenne, Armand Joulin, and Jean Ponce. A graph-matching kernel for object categorization. In *2011 International Conference on Computer Vision*, pages 1792–1799, 2011. 2

[13] Weifeng Ge, Bingchen Gong, and Yizhou Yu. Image super-resolution via deterministic-stochastic synthesis and local statistical rectification. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 4

[14] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, pages 1243–1252. PMLR, 2017. 3

[15] Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. Non-rigid dense correspondence with applications for image enhancement. *ACM transactions on graphics (TOG)*, 30(4):1–10, 2011. 2

[16] Bumsub Ham, Minsu Cho, Cordelia Schmid, and Jean Ponce. Proposal flow: Semantic correspondences from object proposals. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1711–1725, 2017. 2, 4, 5, 6

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2

[18] Mingming He, Jing Liao, Dongdong Chen, Lu Yuan, and Pedro V Sander. Progressive color transfer with dense semantic correspondences. *ACM Transactions on Graphics (TOG)*, 38(2):1–18, 2019. 2

[19] Derek LG Hill, Philipp G Batchelor, Mark Holden, and David J Hawkes. Medical image registration. *Physics in medicine & biology*, 46(3):R1, 2001. 2

[20] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2022. 2, 4, 6, 7, 8

[21] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2010–2019, 2019. 5

[22] Ge-Peng Ji, Keren Fu, Zhe Wu, Deng-Ping Fan, Jianbing Shen, and Ling Shao. Full-duplex strategy for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4922–4933, 2021. 2

[23] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6207–6217, 2021. 6

[24] Seungwook Kim, Juhong Min, and Minsu Cho. Transformatcher: Match-to-match attention for semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8707, 2022. 6, 7

[25] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2278–2287, 2019. 2, 4

[26] Jae Yong Lee, Joseph DeGol, Victor Fragoso, and Sudipta N Sinha. Patchmatch-based neighborhood consensus for semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13153–13163, 2021. 5

[27] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection, 2022. 3

[28] Zhaoshuo Li, Xingtong Liu, Nathan Drenkow, Andy Ding, Francis X Creighton, Russell H Taylor, and Mathias Unberath. Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6197–6206, 2021. 2, 3

[29] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020. 2

[30] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4463–4472, 2020. 1, 5, 6, 7, 8

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 3, 4

[32] Mark F. Medress, Franklin S Cooper, Jim W. Forgie, CC Green, Dennis H. Klatt, Michael H. O'Malley, Edward P Neuburg, Allen Newell, DR Reddy, B Ritea, et al. Speech understanding systems: Report of a steering committee. *Artificial Intelligence*, 9(3):307–316, 1977. 2

[33] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2940–2950, 2021. 6

[34] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3395–3404, 2019. 2, 4, 5, 6, 7

[35] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Learning to compose hypercolumns for visual correspondence. In *European Conference on Computer Vision*, pages 346–363. Springer, 2020. 6

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[37] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelovic, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Ncnet: Neighbourhood consensus networks for estimating image correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2, 3, 4

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 2

[39] Zhaoyang Sun, Yaxiong Chen, and Shengwu Xiong. Ssat: A symmetric semantic-aware transformer network for makeup transfer and removal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2325–2334, 2022. 2

[40] Tatsunori Taniai, Sudipta N Sinha, and Yoichi Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4246–4255, 2016. 2

[41] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14372, 2021. 2, 3

[42] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 4

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3, 4

[44] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention, 2021. 3

[45] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021. 3

[46] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022. 2, 6

[47] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021. 3

[48] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12203–12213, 2020. 2

[49] Dongyang Zhao, Ziyang Song, Zhenghao Ji, Gangming Zhao, Weifeng Ge, and Yizhou Yu. Multi-scale matching networks for semantic correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3354–3364, 2021. 1, 2, 3, 5, 6, 7, 8

[50] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *ICLR*, 2022. 2, 3