Pixel-level Semantic Correspondence through Layout-aware Representation Learning and Multi-scale Matching Integration

Yixuan Sun[1,2,*], Zhangyue Yin[3,*], Haibo Wang[3], Yan Wang[1,2], Xipeng Qiu[3], Weifeng Ge[3,†] and Wenqiang Zhang[1,2,3,†]
[1] Academy for Engineering & Technology, Fudan University, Shanghai, China
[2] Engineering Research Center of AI & Robotics, Ministry of Education, China
[3]School of Computer Science, Fudan University, Shanghai, China
{wfge, wqzhang}@fudan.edu.cn

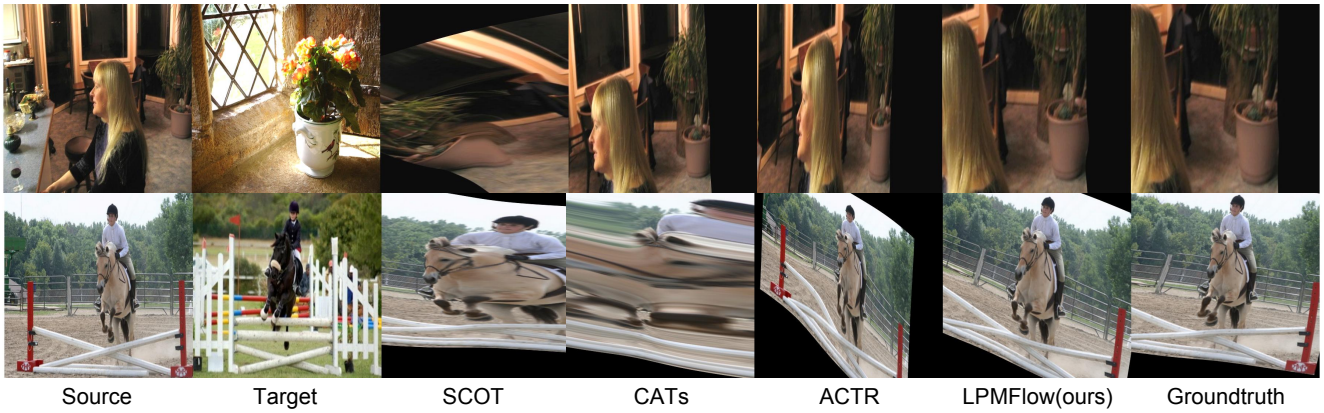

Figure 1. Visualization of dense correspondence for state-of-the-art methods namely SCOT [21], CATs [3], and ACTR [38] compared with our LPMFlow. Thin-plate splines algorithm [1] is used for image warping with instructed by predicted key points.

Abstract

Establishing precise semantic correspondence across object instances in different images is a fundamental and challenging task in computer vision. In this task, difficulty arises often due to three challenges: confusing regions with similar appearance, inconsistent object scale, and indistinguishable nearby pixels. Recognizing these challenges, our paper proposes a novel semantic matching pipeline named LPMFlow toward extracting fine-grained semantics and geometry layouts for building pixel-level semantic correspondences. LPMFlow consists of three modules, each addressing one of the aforementioned challenges. The layout-aware representation learning module uniformly encodes source and target tokens to distinguish pixels or regions with similar appearances but different geometry semantics. The progressive feature superresolution module outputs four sets of 4D correlation tensors to generate accurate semantic flow between objects in different scales. Finally, the matching flow integration and refinement module is exploited to fuse matching flow in different scales to give the final flow predictions. The whole pipeline can be trained end-to-end, with a balance of computational cost and correspondence details. Extensive experiments based on benchmarks such as SPair-71K, PF-PASCAL, and PF-WILLOW have proved that the proposed method can well tackle the three challenges and outperform the previous methods, especially in more stringent settings. Code is available at https://github.com/YXSUNMADMAX/LPMFlow.

1. Introduction

Semantic matching methods aim to establish precise visual correspondences between different objects of the same category [3, 32, 48], which needs a deep understanding of specific pixels, object parts, individual objects, and their spatial layouts. As a fundamental task in computer vision, semantic matching has been applied in object recognition [17, 37], co-segmentation [39], image editing [29, 30], 3D reconstruction [34], and etc. Different from other image under-

∗: Contribution Equally †: Corresponding Authors

standing tasks [36, 42], semantic matching models need to have strong abilities in understanding structural geometry layouts and fine-grained pixel-level semantics, which are critical to handle large intra-class variations in appearance, scale, orientation, and non-rigid deformations.

Current state-of-the-art methods, such as SCOT [21], CHM [25], MMNet [48], CATs [3], and ACTR [38], have designed geometric matching strategies, extracted discriminative features, enforced one-to-one matching, encouraged matching consistency to solve the semantic correspondence problem and achieved impressive performance on popular benchmarks such as PF-PASCAL, PF-WILLOW [8], and SPair-71k [26]. However, given image pairs with large appearance variations, the semantic matching models often generate unsatisfactory correspondences for the following reasons: 1) semantic regions that share similar appearances are often confused, making the matching results violate their geometric layout; 2) objects in different scales present a challenge in establishing correlations for details; 3) nearby pixels are hard to be distinguished.

In this paper, we propose a novel semantic correspondence pipeline that contains layout-aware representation learning (LARL), progressive feature super-resolution (PFSR), and multi-scale matching flow integration (MMFI) modules, and call it LPMFlow. Recent research [24] shows that the geometry layouts of semantic components and the difference of pixels are the keys to construct fine-grained matching. For the layout-aware representation learning module, to obtain the shared layout of regions in a pair of objects, we exploit a representation learning module assisted by a conditional semantics enhancement task that combines global semantics and association of local areas. Meanwhile, in the progressive feature superresolution module, we gradually generate feature maps at different scales to produce 4D matching tensors, which can tolerate large-scale variations of object instances. Finally, the multi-scale matching flow integration module is designed to discover fine-grained differences in the adjacent pixels, thus further improving the matching robustness in different scenarios.

For detailed designs, cascaded transformer blocks with region-based position embedding [35, 45] are used to enhance representation with shared geometry layouts in the LARL module. And inspired by works [5, 14], a referenced patch token correction (RPTC) task is also proposed to guide this learning stage. For the PFSR module, we progressively up-scale features of both sides with feature super-resolution block with the internal relationship of own patches and interaction of opposing features fused. Afterward, multi-scaled features from PFSR are summarized to generate cross-scaled 4D matching tensors. For the MMFI module, we convert these 4D matching tensors into 2D matching flows via soft-argmax and integrate them with a coarse-to-fine structure based on 2D swin-attention blocks [23]. This stage takes advantage of matching relationships among different neighborhoods to generate specific matching details. To our knowledge, our proposed LPMFlow is the first end-to-end method to establish semantic correspondences in units of 2×2 pixels. Extensive experiments on popular matching benchmarks such as SPair-71K [26], PF-PASCAL, and PF-WILLOW [8] demonstrate that LPMFlow can accurately identify semantic relevancy, capturing finer matching between images (shown in Figure 1). We summarize our contributions as follows:

• We propose a novel LPMFlow framework that combines LARL, PFSR, and MMFI modules. It focuses on shared geometries and pixel-level semantics for reliable correspondence, capable of handling large-scale object variations and improving robustness across various scenarios.

• Novel RPTC task is designed for the LARL module to guide shared layout representation learning. For PFSR module, we design a schema that progressively upgrades feature maps, summarizing multi-scale features to create cross-scaled matching tensors. For MMFI module, we design a coarse-to-fine refinement structure for fused cross-scale 2D flows and acquire pixel-to-pixel correspondence.

• Experiment on popular benchmarks indicates that LPMFlow significantly outperforms previous state-of-the-arts, especially in more strict metrics. Qualitative results demonstrate the effectiveness of LPMFlow in addressing three important challenges and generating precise correspondence.

2. Related Work

Representation Learning Cross Elements. As a semantic comprehension task [13], constructing joint representation is vital for understanding complex relationships across the elements. Early methods [15, 22, 28] often use interactive structures to capture the relation among separated embeddings. The advent of transformer-based encoding structure [14] like VisualBert [19] has revolutionized the way to jointly learn representations with implicit relations among the elements adopted in various applications [2, 45]. Recently, several innovative structures such as OSTrack [45] and SEEM [50] are proposed. While other methods extend from the idea of mask-language modeling [4, 14] and design several enhancement tasks for pretraining [43]. In order to extract the shared relations for semantic matching, we inherit the transformer-based encoding structure and craft a novel replace-recovery-based token enhancement technique. This method can augment visual representation by leveraging the correlation between global and local semantic tokens to incorporate shared object layouts.

Multi-scale Correspondence Construction. In matching tasks [31, 33], various methods [3, 25, 48] tried to construct multi-scale correspondence to overcome perspective

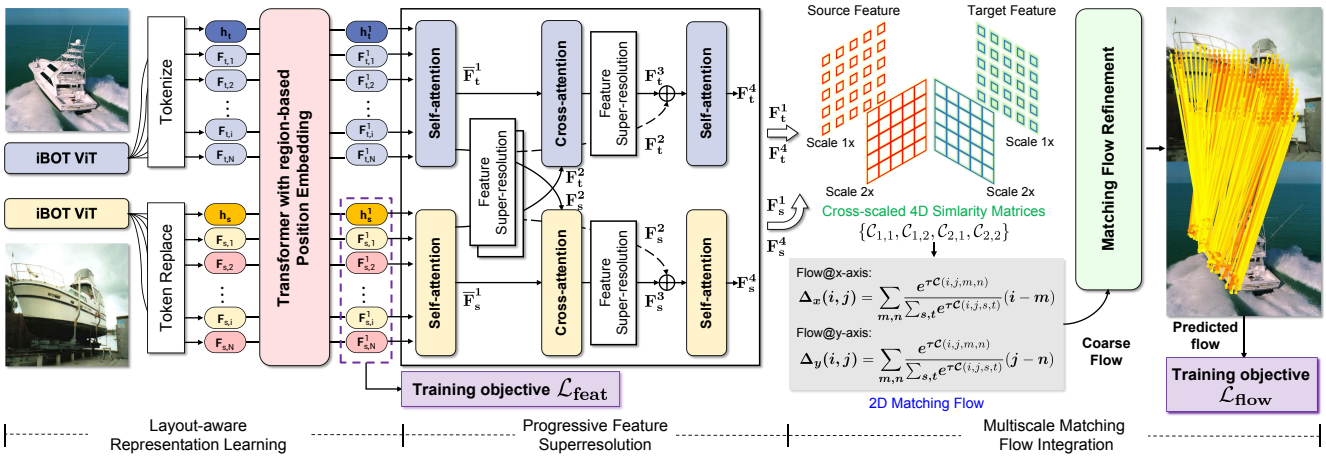

Figure 2. **Illustration of LPMFlow structure.** The full pipeline is composed by a vision transformer backbone, layout-aware representation learning (LARL), progressive feature super-resolution (PFSR), and multi-scale matching flow integration (MMFI) modules.

distortion. For example, MMNet [48] and VAT [11] build up a stair-wised pipeline to fuse lower-scaled matching results into a larger scale. DHPF [27] and CATs [3] tend to align multi-scaled features at first and construct correspondences uniformly. These methods help to the construction of matching details. To handle the problem of inconsistent scales in an image pair, several methods [25, 47] upscale features to different scales to construct and refine cross-scaled matching. However, obtaining semantic detail representations while increasing feature resolution remains a critical challenge. Our research counters this with a progressive pipeline for high-resolution feature construction.

Refinement of Matching Flow. Matching flow is an efficient format to represent matching compared with 4D tensors which can be refined in higher resolution with less computation cost [38, 44]. Several methods are designed for matching flow refinement. PWarpC-SF-Net [41] refines estimated matching with warping consistency. GM-Flow [44] and STTR [20] estimate the offset of patches according to fixed neighborhoods to refine the single-scaled flow. ACTR [38] fuses the multi-path coarse flows for refinement to benefit from different matching tensors. In this paper, we follow previous works and design a coarse-to-fine structure to integrate and refine multi-scaled flow, which can incorporate correspondence in different ranges to distinguish subtle differences in narrowed pixel regions.

3. Method

In this section, we introduce the LPMFlow framework (shown in Figure 2) for generating semantic correspondence in high quality. Given an image pair $\{\boldsymbol{I_s}, \boldsymbol{I_t}\}$, the proposed method: 1) enhances the representation $\boldsymbol{F}_s^1, \boldsymbol{F}_t^1$ with layout consistency; 2) up-scales the enhanced feature of both sides progressively to build up cross-scaled 4D matching tensors $\{\mathcal{C}\}$; 3) calculates the multi-scale matching flows from $\{\mathcal{C}\}$, integrates and refines them into a fine-grained flow $\Delta_{d|s\mapsto t}$. Following this pipeline, our LPMFlow can generate matching flow $\Delta_{d|s\mapsto t}$ in $\frac{1}{2}$ scale using the initial features from ViT [7] backbone only in the scale of $\frac{1}{16}$ which outperforms the previous state-of-the-art methods.

3.1. Layout-aware Representation Learning

We design the Layout-aware Representation Learning (LARL) module to build up enhanced representation from $\{\boldsymbol{F}_s, \boldsymbol{F}_t\}$. Our proposed module is composed of cascaded transformer blocks with 2D relative position embedding introduced. To handle the problem of confusion among semantically irrelevant tokens with similar appearances, we design a learning task to restrict the representation of a token closer to its neighborhoods. Besides, the learning task also guides the model to learn the layout consistency of correspondences via token replacement & correction with reference. Unlike previous works [5, 19, 43], we set source tokens for replacement and recovery supervision ($\boldsymbol{T}_s$), while target tokens are defined as references ($\boldsymbol{T}_t$). We name this task referenced patch token correction (RPTC).

Structure of LARL Module: Structure of the LARL module is based on cascaded transformer blocks. In this module, isolated features $\boldsymbol{F}_s, \boldsymbol{F}_t \in \mathbb{R}^{(N+1)\times c}$ are first concentrated as $\{\boldsymbol{F}_s, \boldsymbol{F}_t\}$. And embeddings for the segment and position are superimposed on tokens to indicate the images to which the token belongs and provide spatial context. We also separately introduce the region-based position embedding (PE) [35, 45] in column and row to enhance the relative position modeling. The process is performed as follows:

$$\begin{aligned} \boldsymbol{q}^l, \boldsymbol{k}^l, \boldsymbol{v}^l &= \Psi_{\{q,k,v\}}(\{\mathbf{F}_s, \mathbf{F}_t\}'_{l-1} + \mathbf{p}_l), \\ \widehat{\mathbf{E}}'_l &= \text{MultiHead}(\text{LN}(\boldsymbol{q}^l), \text{LN}(\boldsymbol{k}^l), \text{LN}(\boldsymbol{v}^l)), \\ \{\mathbf{F}'_s, \mathbf{F}'_t\}_l &= \text{FFN}(\text{LN}(\widehat{\mathbf{E}}'_l)) + \widehat{\mathbf{E}}'_l, \end{aligned} \tag{1}$$

where $\boldsymbol{\Psi}_{\{q,k,v\}}$ stands for the function to generate $\boldsymbol{q}^l, \boldsymbol{k}^l, \boldsymbol{v}^l$, $\boldsymbol{p}_l$ for region based PE, LN for layer normalization, Multi-

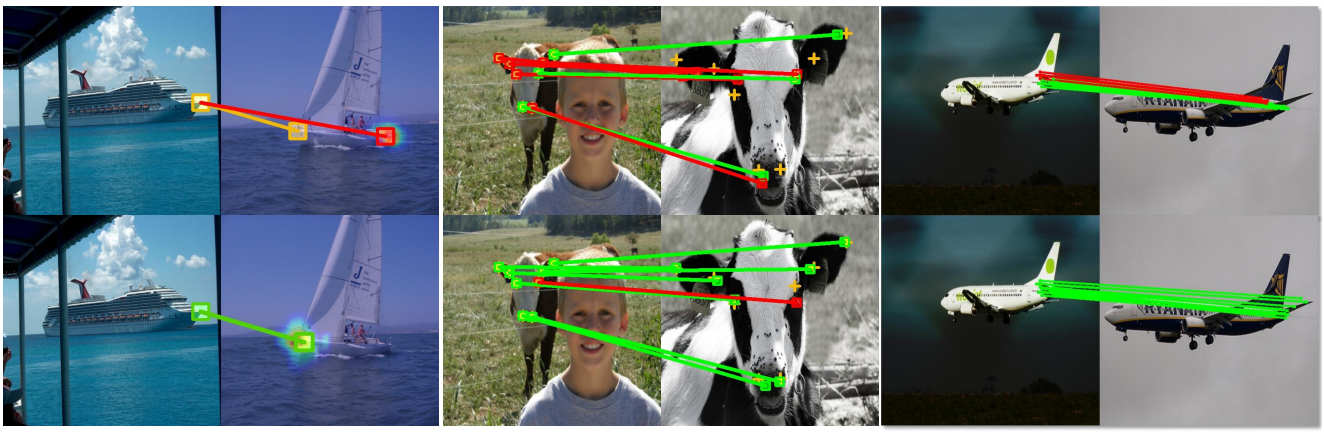

Figure 3. Visualization for the effectiveness of three designed modules on three challenges. The first row presents results without the modules, and the second row visualizes our method. Ground truth is indicated in yellow, successes in green, and failures in red.

Head for multi-head self attention [7] and FFN for feed-forward network. We note $\{\boldsymbol{F}_s^1, \boldsymbol{F}_t^1\}$ as output $\{\mathbf{F}'_s, \mathbf{F}'_t\}_l$ of the last layer. Through this structure with relation modeling, the representation of a token can be reconstructed and refined according to the tokens from both $\boldsymbol{I_s}$ and $\boldsymbol{I_t}$.

Procedure of RPTC Task: This task is defined as recovering the replaced key tokens according to their neighborhood and reference image tokens. For the replacement procedure, we select tokens most relevant to the shared objects between two images as key tokens. Thus, we first generate a foreground weight map $\boldsymbol{W}$ for $\boldsymbol{T}_s$ based on global semantic ([cls] tokens) relevance. Afterward, top K source patch tokens $\boldsymbol{T}_s^p$ are randomly selected with the rate of θ according to $\boldsymbol{W}$ and replaced with averaged [cls] tokens of both sides. We note the $\boldsymbol{T}_s$ after token replacement as $\boldsymbol{F}_s$ and $\boldsymbol{T}_t$ after tokenization as $\boldsymbol{F}_t$. For the recovery procedure, we feed the $\{\boldsymbol{F}_s, \boldsymbol{F}_t\}$ into LARL and acquire the reconstructed representation $\{\boldsymbol{F}_s^1, \boldsymbol{F}_t^1\}$. To complete the learning process as *recovery*, two priors are used to design the training objective. First, the original local semantics of the replaced tokens are different from each other. Besides, as an image, a patch is usually more similar to the patches in its neighborhood. We design a self-contrastive loss based on InfoNCE function [10, 46] according to these priors as:

$$\mathcal{L}_{feat} = -\frac{1}{R}\sum_{\beta=0}^{R}\frac{\exp(\mathrm{q}_\beta \otimes \mathrm{k}^+/\tau)}{\sum_{k^* \in \{k^+, k^-\}} \exp(\mathrm{q}_\beta \otimes \mathrm{k}^*)/\tau}. \quad (2)$$

For each of R replaced tokens in $\boldsymbol{F}_s$ (set as q of InfoNCE), we set the negative embeddings k^- as other $R-1$ replaced tokens and we set the positive embedding k^+ as the averaged 8 tokens of 3×3 neighborhood of q in $\boldsymbol{F}_s^1$. We note that several k^- might appear in neighborhood of q. We explain this as k^- within the neighborhood of a query embedding should be more similar to the query embedding than other replaced tokens. We also design a training process that gradually weakens the intensity of the RPTC guidance λ. This design aims to guide the model focusing more on the finer-grained task of correspondence based on the refined representation as the learning progresses.

3.2. Progressive Feature Super-Resolution

After acquiring the high-quality representation $\{\boldsymbol{F}_s^1, \boldsymbol{F}_t^1\}$, we attempt to establish matching relationships at the highest possible resolution. However, limited by the currently used data structures: similarity matrix [32, 48] and matching flow [3], obtaining high-resolution matching from features that undergo significant down-scaling is a difficult task. Thus, we try to first upscale the resolution of features before calculating correspondence in a progressive way. Afterward, we use multiple scaled features to generate cross-scale similarity matrices. This design brings two benefits: 1) Compared with directly up-scaling the similarity matrix, the feature preserved richer semantic information to guide resolution improvement; 2) Fusing matching relationships at different scales can enhance the adaptability to the problem of the inconsistent scale of objects (shown in Figure 3).

Structure of PFSR Module: In our PFSR module, we enhance and upscale $\boldsymbol{F}_s^1$ and $\boldsymbol{F}_t^1$ in three stages. The first stage employs a self-attention block followed by a feature super-resolution (FSR) block, upgrading the features to $\boldsymbol{F}_s^2$ and $\boldsymbol{F}_t^2$. In this process, $\overline{\boldsymbol{F}}^1$ represents the output of $\boldsymbol{F}^1$ post self-attention, offering refined feature representations. The FSR block utilizes cascaded window and shifted window attention to further enhance these representations after bilinear interpolation. Subsequently, scale-asymmetric cross-attention layers facilitate semantic interaction between the features, yielding the hybrid enhanced $\boldsymbol{F}_s^3$ and $\boldsymbol{F}_t^3$. Here, $\overline{\boldsymbol{F}}_s^1$ and $\overline{\boldsymbol{F}}_t^1$ are used for query embeddings, while $\boldsymbol{F}_t^2$

and $\boldsymbol{F}_s^2$ serve as key and value embeddings, incorporating both cross-scale information and opposite path semantics. The final stage integrates these self-feature-guided and interaction-enhanced super-resolution features into $\boldsymbol{F}_s^4$ and $\boldsymbol{F}_t^4$ through an additional self-attention block, ensuring thorough and effective feature integration.

Cross-scaled Similarity Matrices: After the PFSR, we acquire up-scaled features $\{\boldsymbol{F}_s^4, \boldsymbol{F}_t^4\}$ two times as the resolution of $\boldsymbol{F}_s^1$ and $\boldsymbol{F}_t^1$. We combine the two scale features $\boldsymbol{F}^1$ and $\boldsymbol{F}^4$ of source and target in pairs to construct a similarity matrix of four scale relationships. Here we use the multi-head attention scheme to generate the 4-dimensional matrix of $h_s \times w_s \times h_t \times w_t$ with weights Θ_h^Q and Θ_h^K as work [3, 11] using a pair of features with $[cls]$ token removed. We show the formula of similarity matrix generation as:

$$\mathcal{C}_{a,b} = \frac{1}{H} \sum_h \text{Softmax}\left(\frac{(\boldsymbol{F}_s^a \Theta_h^Q)(\boldsymbol{F}_t^b \Theta_h^K)^T}{\sqrt{c}} \right), \quad (3)$$

where H is a number of heads. The scale of $\boldsymbol{F}^1$ is $\frac{1}{16}$ of the original resolution, and the scale of $\boldsymbol{F}^4$ is $\frac{1}{8}$ of the original resolution, which makes $\boldsymbol{F}^4$ twice the resolution of $\boldsymbol{F}^1$. We record $\{a, b\}$ as 1 for the feature $\boldsymbol{F}^1$, and as 2 for $\boldsymbol{F}^4$. This resulted in corss-scaled similarity matrices at four resolutions: $\mathcal{C}_{1,1}$:$\frac{1}{16}\times\frac{1}{16}$, $\mathcal{C}_{1,2}$:$\frac{1}{16}\times\frac{1}{8}$, $\mathcal{C}_{2,1}$:$\frac{1}{8}\times\frac{1}{16}$, and $\mathcal{C}_{2,2}$:$\frac{1}{8}\times\frac{1}{8}$.

3.3. Multi-scale Matching Flow Integration

With the multi-scaled similarity matrices provided by the PFSR module, the next step of our LPMFlow is to fuse these matrices and generate correspondence in higher resolution such as $\frac{1}{2}$. To avoid the $O(n^2)$ space complexity associated with direct optimization using the similarity matrix, we follow work [38, 40] to represent and refine the correlation with semantic flow $\Delta_{s\mapsto t}$. For this purpose, we define the Multi-scale Matching Flow Integration (MMFI) module composed of the procedure of multi-scale semantic flow calculation and the matching flow refinement block.

Calculation of Multi-scaled Flows: We first up-scale $\{\mathcal{C}_{a,b}\}$ to the scale as $\mathcal{C}_{2,2}$. For each source patch, index of the best corresponding target patch can be found by applying argmax function over $h_t \times w_t$ dimension of $\mathcal{C}_{a,b}$. However, this non-differentiable operation prohibited us from adopting it for our end-to-end pipeline. Thus, we adopt soft-argmax [18] to generate flows from $\mathcal{C}$ as follows:

$$\begin{aligned} \boldsymbol{\Delta}_{\boldsymbol{x}}(i,j) &= \sum_{m,n} \frac{e^{\tau \boldsymbol{\mathcal{C}}(i,j,m,n)}}{\sum_{s,t} e^{\tau \boldsymbol{\mathcal{C}}(i,j,s,t)}} (i-m) \\ \boldsymbol{\Delta}_{\boldsymbol{y}}(i,j) &= \sum_{m,n} \frac{e^{\tau \boldsymbol{\mathcal{C}}(i,j,m,n)}}{\sum_{s,t} e^{\tau \boldsymbol{\mathcal{C}}(i,j,s,t)}} (j-n) \end{aligned} \quad (4)$$

where Δ_x and Δ_y are the offset value of $h_s \mapsto h_t$ and $w_s \mapsto w_t$ mappings. We concatenate the $\{\Delta_x, \Delta_y\}$ and generate the initial semantic flow $\Delta_{c|s\mapsto t}$.

Matching Flow Refinement Block: In order to generate the fine-grained correspondence in $\frac{1}{2}$ scale for further refinement, we propose the Matching Flow Refinement Block. In this structure, cross-scale flows $\{\Delta_{c|s\mapsto t}^{a,b}\}$ and source token features after channel projection with $[cls]$ token removed are used for this block. For all the inputs, we use bilinear interpolation to up-sample them to $\frac{1}{2}$ scale and concatenate them as $\mathbf{Z}^1$. In order to make full use of the matching consistency within different neighborhood ranges, we design a coarse-to-fine fusion structure based on swin-attention block [23] for flow refinement as follows:

$$\begin{aligned} \mathbf{Z}^{i+1} &= \text{SwinAttn}^i(\text{LN}(\mathbf{Z}^i)) + \mathbf{Z}^i \mid i \in \{1,2,3\} \\ \Upsilon_{d|s\mapsto t} &= \text{Conv}_{3\times3}(\mathbf{Z}^1 + \mathbf{Z}^2 + \mathbf{Z}^3 + \mathbf{Z}^4), \end{aligned} \quad (5)$$

where $\text{SwinAttn}^i(.)$ stands for three swin-attention blocks with window size as 16×16, 8×8 and 4×4, $\text{Conv}_{3\times3}(.)$ stands for 3×3 convolution layer to fuse embeddings from all the grains and provide dense offset $\Upsilon_{d|s\mapsto t}$. We use the above offset $\Upsilon_{d|s\mapsto t}$ to refine upscaled flow and obtain the dense correspondence $\Delta_{d|s\mapsto t}$ in scale of $\frac{1}{2}\times\frac{1}{2}$.

3.4. Training

We follow the work [3, 11, 38] to convert the sparse ground truth key point pairs $\mathcal{M}_{gt}$ into pseudo semantic flow to supervise the training of our method. For the visible regions controlled by a mask $\boldsymbol{S}^s$ as that in [38], except for the patches corresponding to the ground truth keypoints, flow values of other patches are impacted by all the ground truth key points in a 35×35 receptive field. We resize the matching flow to the scale of $\frac{1}{2}\times\frac{1}{2}$ identical to the output of LPMFlow and generate $\Delta_{gt|s\mapsto t}$. Then our learning goal is to refine the parameter Φ to minimize the endpoint error [40] loss function $\mathcal{L}_{flow}$ as follows:

$$\mathcal{L}_{flow} = \frac{1}{N} \sum_{(\boldsymbol{I}^s, \boldsymbol{I}^t)}^{\mathcal{D}} \frac{\boldsymbol{\vartheta}^s \left\| \Phi\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right) - \Delta_{gt}\left(\boldsymbol{I}^s, \boldsymbol{I}^t\right) \right\|^2}{|\boldsymbol{\vartheta}^s|}, \quad (6)$$

where $|\boldsymbol{\vartheta}^s|$ is the visible patches, N is the number of samples in training set $\mathcal{D}$, $\boldsymbol{\vartheta}^s$ and $\Delta_{gt}(.)$ are the matching flows generated by $\mathcal{M}_{gt}$. Combined with the loss $\mathcal{L}_{feat}$ for representation learning, we set our training loss $\mathcal{L}$ as the sum of $\mathcal{L}_{feat}$ and $\lambda\mathcal{L}_{flow}$.

4. Experiment

Datasets. Several semantic matching datasets, namely SPair-71K [26], PF-PASCAL, and PF-WILLOW [8], are selected for our experiment. We use SPair-71K and PF-PASCAL for the evaluation of performance. The model

Table 1. Quantitative results on benchmarks. Higher PCK is better. The best results are in bold, and second-best results are underlined. *: Method with iBOT backbone (others using ResNet-101 as backbone). Multi Scale: whether to employ multi-scale refinement. Corr Format: Format of output correspondence. 4D Mtrx is 4D similarity matrix and TransMatcher is TransforMatcher [16] for short.

Method	Description		Performance					Generalizability		Efficiency			
			Spair-71K		PF-PASCAL			PF-WILLOW		TITAN RTX: 24GB			
	Multi Scale	Corr Format	α: bbox		α: img			α: bbox	α: bkp	Params(M)		Mem	Time
			0.05	0.1	0.05	0.1	0.15	0.1	0.1	Head	Total	(GB)	(ms)
NC-Net[32]	✗	4D Mtrx	-	20.1	54.3	78.9	86.0	-	67.0	0.2	27.6	1.2	222.9
SCOT[21]	✗	4D Mtrx	20.0	35.6	63.1	85.4	92.7	-	76.0	-	44.5	4.6	133.5
DHPF[27]	✓	4D Mtrx	-	37.3	75.7	90.7	95.0	77.6	71.0	5.8	50.3	1.6	58.2
CHM[25]	✗	4D Mtrx	22.7	46.3	80.1	91.6	94.9	79.4	69.6	7.1	94.1	1.7	55.3
CATs[3]	✓	2D Flow	27.7	49.9	75.4	92.6	96.4	79.2	69.0	4.7	49.2	2.0	45.4
MMNet-FCN[48]	✓	4D Mtrx	33.3	50.4	81.1	91.6	95.9	-	-	10.3	64.7	5.4	258.6
TransMatcher[16]	✓	4D Mtrx	-	53.7	80.8	91.8	-	65.3	76.0	0.9	87.9	2.7	54.2
CATs* [3]	✓	2D Flow	30.7	55.2	77.8	93.1	96.8	86.3	79.5	5.7	90.7	2.8	54.2
TransMatcher* [16]	✓	4D Mtrx	33.1	57.9	77.3	93.3	96.6	84.3	78.3	1.6	86.6	2.4	48.5
ACTR* [38]	✗	2D Flow	42.0	62.1	81.2	94.0	97.0	87.2	79.9	87.8	172.8	3.9	84.1
LPMFlow*	✓	2D Flow	**46.7**	**65.6**	**82.4**	**94.3**	**97.2**	**87.6**	**81.0**	93.9	178.9	3.8	85.7

trained on PF-PASCAL is evaluated with PF-WILLOW to assess the generalizability of our method. Regarding quantitative statistics, SPair-71K contains 53,340 training, 5,384 validation, and 12,234 testing image pairs in 18 categories. This dataset often includes challenging cases such as scale differences, occlusion, and truncation. According to work [48], we partition PF-PASCAL [8] into splits of 700, 300, and 300 pairs as training, validation, and testing sets, respectively. All 900 image pairs from 10 classes of PF-WILLOW [8] are used to test generalizability.

Evaluation Metric. We use the percentage of correct key points within an acceptance threshold α (PCK@α) [3, 21] to evaluate performance. In this metric, a maximum matching range d is introduced for the circular acceptance area with a radius of $\alpha \times d$. A predicted key point falling into this area is considered *correctly* matched. For PF-PASCAL, d is the longer side of an image, and we denote the metric as α_{img}. For SPair-71K and PF-WILLOW, d is the longer side of the object bounding box, denoted as α_{bbox}. In PF-WILLOW, the maximum distance of annotated key points is also used as d, with the metric noted as α_{bkp}.

Implementation Details. Our model employs ViT-B/16, pretrained using the iBOT [49] method on ImageNet-1K [6] as the backbone. The LARL module contains 6 transformer blocks, randomly utilizing 25% foreground tokens (K=64) for the RPTC task, with an initial guidance intensity λ of 0.2 and decreasing gradually at a rate of 10% until λ=0. In the PFSR module, we set the FSR block window size to 4×4 with two cascaded attention layers in each block. For the MMFI module, swin attention is dimensioned at 96 with 8 heads. Training is conducted with a 256×256 input resolution, utilizing the AdamW optimizer with a weight decay of 0.05. The learning rates for the backbone and subsequent structures are set at 4e-5 and 3e-6 for both SPair-71K [26] and PF-PASCAL [8], to ensure a balance between detailed feature learning and optimizing non-pretrained structures. We implement the model with PyTorch [12] and train on one NVidia TITAN RTX GPU. The batch size is set to 8, and training converged within 12 and 50 epochs for SPair-71K [26] and PF-PASCAL [8].

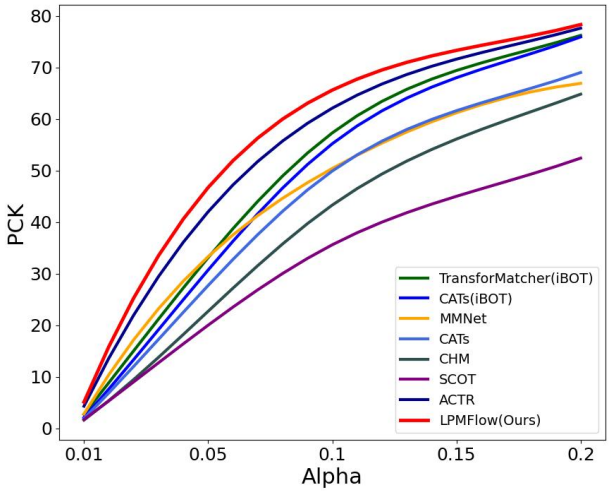

Figure 4. The polyline of PCK@α for our method and previous works on SPair-71K [26] in multiple α. Our method outperforms other methods especially with smaller error thresholds (smaller α).

4.1. Comparison with State-of-The-Art

Performance Comparison. This study presents a comparison between our approach and the latest state-of-the-art (SOTA) on the SPair-71K [26] and PF-PASCAL [8]

Table 2. Ablations on designed modules.

LARL	PFSR	MMFI	SPair-71K $\alpha_{bbox} = 0.1$
✓	✓	✓	65.6
✗	✓	✓	63.2 (2.4↓)
✓	✗	✓	62.0 (3.6↓)
✓	✓	✗	63.9 (1.7↓)

Table 3. Ablations on components in LARL module.

Methods	SPair-71K $\alpha_{bbox} = 0.1$
LPMFlow	65.6
w/o Gradual Guidance of RPTC	64.5 (1.1↓)
w/o Self Contrastive Loss	63.9 (1.7↓)
w/o Region-based PE	64.8 (0.8↓)

Table 4. Ablations on components in PFSR module.

Methods	SPair-71K $\alpha_{bbox} = 0.1$
LPMFlow	65.6
w/o Interactive Super-Resolution	64.1 (1.5↓)
w/o Internal Super-Resolution	63.8 (1.8↓)
w/o Feature Super-Resolution block	63.4 (2.2↓)

Table 5. Ablations on components in MMFI module.

Methods	SPair-71K $\alpha_{bbox} = 0.1$
LPMFlow	65.6
w/o Multi-Scale Flow Integration	64.3 (1.3↓)
w/o C2F Refinement (16×16)	64.6 (1.0↓)
w/o C2F Refinement (4×4)	64.0 (1.6↓)

datasets, detailed in Table 1. The evaluation first categorizes the methods based on the usage of multi-scale refinement and the format to represent correspondence. To ensure an equitable comparison, we further differentiate the methods based on their backbone frameworks, such as ResNet-101 [9] and iBOT [49]. Our approach demonstrates significant improvements over previous state-of-the-art methods, achieving an accuracy of 65.6% using the PCK metric on SPair-71K at the α:0.1 threshold, which is 3.5% higher than the second best method. This advantage extends to 4.7% under the more stringent α:0.05 threshold. A similar improvement trend is observed on the PF-PASCAL dataset at α:0.05, with a 1.2% enhancement.

These results underscore the enhanced precision and detail of our proposed LPMFlow in matching, particularly in more challenging scenarios. To further substantiate this claim, we present additional comparisons across a range of diverse reception thresholds in Figure 4. Moreover, when comparing methods that originally used the ResNet-101 backbone and switched to the iBOT-B backbone, our approach still shows superior performance, leading by at least 7.7% at the α:0.1 threshold. This demonstrates the effectiveness of our method in achieving accurate correspondences with equivalent image feature quality.

Analysis of Efficiency and Generalizability. We also provide a comparison of generalizability and efficiency in Table 1. For generalization ability, we evaluate LPMFlow and state-of-the-arts on PF-WILLOW trained on PF-PASCAL [8]. The results for both metric PCK@α_{bbox}:0.1 and PCK@α_{bbkp}:0.1 have proved the better generalizability of LPMFlow. For the more strict PCK@$\alpha_{bbkp} = 0.1$ metric, our method also shows more significant improvement as 1.1%. We also compare the efficiency in Table 1 which shows our method has comparable computation cost with previous full transformer pipelines such as ACTR [38].

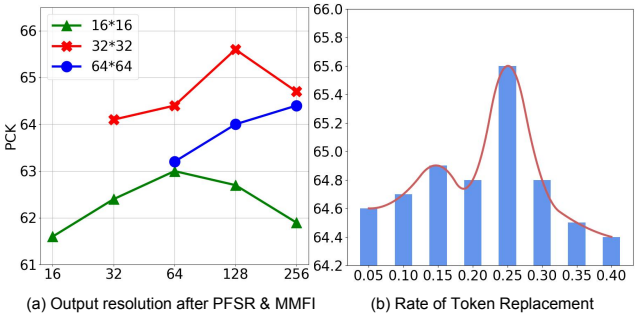

Figure 5. Ablations on the hyper-parameters:(a) performance of our method with output resolution of PFSR from 16 to 64 and resolution of MMFI from 16 to 256; (b) performance of our method with different rates of token selected for replacement and recovery.

Besides, Our LPMFlow can provide more fine-grained correspondence with obvious performance improvement.

4.2. Ablation Study and Analysis

Ablation on Designed Modules. We conduct several experiments to validate the effectiveness of the three modules in Table 2. We set the ablation models as the models replacing the designed components with basic structures preserved. The performance of the method without LARL declined by 2.4%, without PFSR declined by 3.6%, and without MMFI declined by 1.7%. This result is mutually confirmed with the visualization comparison in Figure 3 as the designed modules can handle three proposed challenges for semantic correspondence and contribute to the detailed optimization of matching results in higher resolution.

Ablation on Designed Components. We also prove the effectiveness of inside components of three modules in Table 3, 4 and 5. For three components of LARL, when we remove the design of gradual guidance of RPTC (using the consistent λ=0.2), designed self-contrastive loss and the

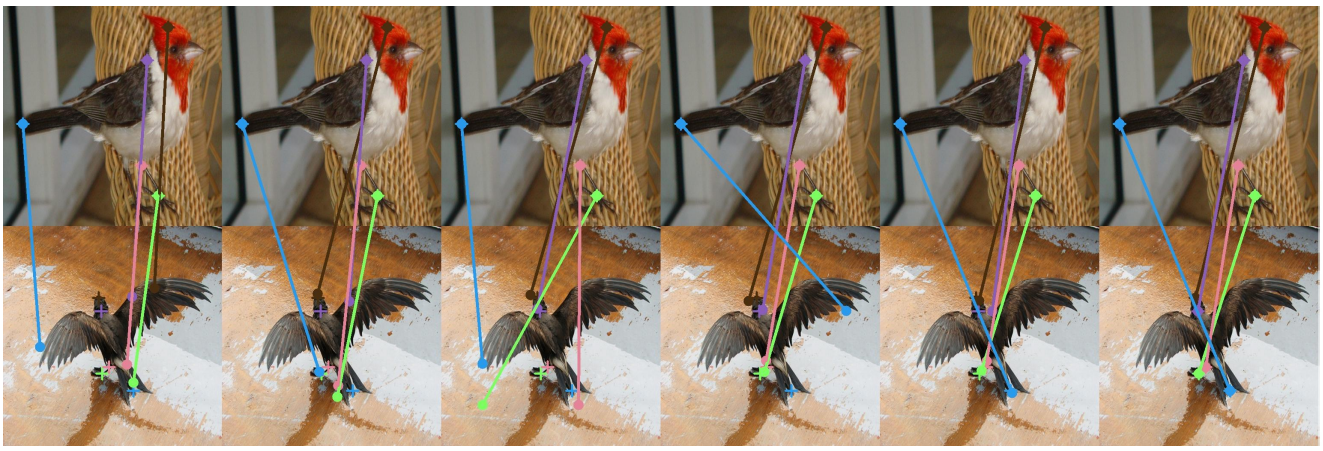

Figure 6. Visualization of matching result. The upper image is the source image and below image is the target image, and crosses are the ground truth labels. We compared the results of SCOT [21], CATs [11], MMNet [48], ACTR [38] and our LPMFlow.

Region-based PE performance dropped for 1.1%, 1.7% and 0.8%. We also evaluate the performance when removing the interactive/internal super-resolution, design of feature super-resolution block (replaced with bilinear interpolation) for PFSR and multi-scale flow fusion, coarse-to-fine flow refinement structure (using constant 16×16 and 4×4 window size) for MMFI. Performance of all these components also dropped by 1.0%-2.2%. This proves all these designs contribute to the performance of LPMFlow.

Ablation on Hyper-Parameters. We also prove the effectiveness of hyper-parameters as resolution magnification in feature level (PFSR) and flow level (MMFI) as well as the rate of token replacement in Figure 5. For resolution magnification, 2 times up-scaling for PFSR module and 4 times for MMFI module can provide the best result. The performance decline for output flows in original 256×256 resolution is attributed to the sparse annotations for correspondence supervision. We obverse that a token replacement rate of 0.25 in the RPTC task is highly effective in promoting the learning of distinct representations among tokens.

Qualitative Results and Visual Analysis. In order to provide an intuitive comparison of our method and SOTAs, we visualize the matching result in both sparse and dense format. We visualize the linked predicted key point pairs compared with SCOT [21], CATs [3], and MMNet [48] and ACTR [38] in Figure 6. The labeled crosses are the ground-truth matching pairs. The given image pair contains several issues as perspective distortion, inconsistent directions, large differences in posture, and large differences in color and texture, making it challenging to semantic correspondence methods. In this condition, our LPMFlow can still provide accurate matching details. We attribute this to our better representation of consistent relations and the more accurate correspondence decoding. We also provide dense warping results in Figure 1 as an overview of matching. The result shows that our LPMFlow can better overcome the occlusion, appearance variation, and perspective distortion.

5. Conclusions and Limitations

In this work, we introduce LPMFlow, a novel pipeline that achieves high-resolution, fine-grained correspondence. This is accomplished by addressing three core challenges: 1) We mitigate confusion caused by similar object regions through a layout-aware representation learning module, bolstered by our designed global semantic replacement-restoration task. 2) We address the scale inconsistency for instances in image pairs with a progressive feature super-resolution module that enhances multi-scale correlation. 3) We tackle weak discriminability among neighborhoods with a multi-scale matching flow integration module that combines cross-scale flows, enabling us to refine the high-resolution flow with precise offsets. Our experimental results demonstrate that LPMFlow surpasses existing methods, particularly in fine-grained metrics. Ablation studies and detailed visualizations further validate the effectiveness of our approach in resolving key challenges and producing accurate correspondence across detailed regions. We list the limitations of our method as heavy usage of [CLS] token, sparse key points annotation, and not fit for multi-instance correspondence tasks as that defined in work [37].

6. Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.62072112, Nos.62106051), the Scientific and Technological Innovation Action Plan of Shanghai Science and Technology Committee (No.22511102202, No.22511101502, No.21DZ2203300) and National Key R&D Program of China (2022YFC3601405).

References

[1] 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

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[3] 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, 3, 4, 5, 6, 8

[4] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2019. 2

[5] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, 2020. 2, 3

[6] 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. 6

[7] 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, 4

[8] 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, 5, 6, 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 4

[11] 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. 3, 5, 8

[12] Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104, 2021. 6

[13] 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. 2

[14] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, page 2, 2019. 2

[15] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. *Advances in neural information processing systems*, 31, 2018. 2

[16] 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

[17] Shiyi Lan, Zhiding Yu, Christopher Choy, Subhashree Radhakrishnan, Guilin Liu, Yuke Zhu, Larry S Davis, and Anima Anandkumar. Discobox: Weakly supervised instance segmentation and semantic correspondence from box supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3406–3416, 2021. 1

[18] Junghyup Lee, Dohyung Kim, Jean Ponce, and Bumsub Ham. Sfnet: Learning object-aware semantic correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2278–2287, 2019. 5

[19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2, 3

[20] 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. 3

[21] 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, 2, 6, 8

[22] Zhilei Liu, Jiahui Dong, Cuicui Zhang, Longbiao Wang, and Jianwu Dang. Relation modeling with graph convolutional networks for facial action unit detection. In *MultiMedia Modeling: 26th International Conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, Proceedings, Part II 26*, pages 489–501. Springer, 2020. 2

[23] 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, 5

[24] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, 129(1):23–79, 2021. 2

[25] 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. 2, 3, 6

[26] 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, 5, 6

[27] 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. 3, 6

[28] Will Norcliffe-Brown, Stathis Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. *Advances in neural information processing systems*, 31, 2018. 2

[29] Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel. Neural congealing: Aligning images to a joint semantic atlas. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19403–19412, 2023. 1

[30] William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei A. Efros, and Eli Shechtman. Gan-supervised dense visual alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13470–13481, 2022. 1

[31] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007. 2

[32] 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. 1, 4, 6

[33] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47:7–42, 2002. 2

[34] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 1

[35] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 2, 3

[36] Oriane Siméoni, Chloé Sekkat, Gilles Puy, Antonin Vobecky, Éloi Zablocki, and Patrick Pérez. Unsupervised object localization: Observing the background to discover objects. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2023. 2

[37] Yixuan Sun, Yiwen Huang, Haijing Guo, Yuzhou Zhao, Runmin Wu, Yizhou Yu, Weifeng Ge, and Wenqiang Zhang. Misc210k: A large-scale dataset for multi-instance semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7121–7130, 2023. 1, 8

[38] Yixuan Sun, Dongyang Zhao, Zhangyue Yin, Yiwen Huang, Tao Gui, Wenqiang Zhang, and Weifeng Ge. Correspondence transformers with asymmetric feature learning and matching flow super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17787–17796, 2023. 1, 2, 3, 5, 6, 7, 8

[39] 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. 1

[40] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 5

[41] Prune Truong, Martin Danelljan, Fisher Yu, and Luc Van Gool. Probabilistic warp consistency for weakly-supervised semantic correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8708–8718, 2022. 3

[42] Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Conference on Computer Vision and Pattern Recognition*, 2022. 2

[43] Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. In *European Conference on Computer Vision*, pages 337–353. Springer, 2022. 2, 3

[44] 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. 3

[45] Botao Ye, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Joint feature learning and relation modeling for tracking: A one-stream framework. In *ECCV*, 2022. 2, 3

[46] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022. 4

[47] Kang Zhang, Yuqiang Fang, Dongbo Min, Lifeng Sun, Shiqiang Yang, Shuicheng Yan, and Qi Tian. Cross-scale cost aggregation for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1590–1597, 2014. 3

[48] 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, 4, 6, 8

[49] 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. 6, 7

[50] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *arXiv preprint arXiv:2304.06718*, 2023. 2