Dense Flow from Adaptive Correspondence

Cheng-Lin Hsieh **Stella X. Yu**
University of Michigan
{clhsieh,stellayu}@umich.edu

Abstract

Optical flow requires reliable matching from sufficient visual context and precise motion at every pixel. Existing methods such as WAFT match and estimate flow on the same coarse grids, then recover dense flow through learned upsampling, without explicitly linking visual support for matching to dense flow reconstruction. *Our key insight is that dense pixel-level flow does not require equally dense matching. We decouple correspondence scale from flow resolution by adapting the size and shape of spatial supports to image structure for reliable matching, while reusing these supports for precise dense flow reconstruction.* We realize this in Flow from Adaptive Token Estimation (FATE), whose token supports expand within coherent regions and contract near motion boundaries. Shared pixel-to-token associations pool features for matching and reconstruct pixel flow from token motions. Flow supervision jointly trains pixel grouping, token matching, and flow estimation without grouping labels, allowing FATE to anticipate which pixels are likely to move together from appearance alone. Trained on FlyingChairs/FlyingThings3D, FATE achieves the highest *zero-shot* accuracy on Sintel Final and KITTI, a critical setting given scarce real-world flow labels. For visible pixels, FATE improves accuracy within coherent regions, and more near motion boundaries. Its largest gains occur where direct matches are unavailable: Occluded or out-of-frame pixels account for roughly 10% of valid pixels but 50% of WAFT's total error. Experiments show that adaptive correspondence through pixel-to-token grouping enables reliable matching and precise pixel-level flow, improving generalization to unseen real scenes.

1 Introduction

Visual correspondence asks which elements in two images refer to the same scene content; finding these correspondences is called matching. We study two-image optical flow, which represents image motion as displacement from each pixel in the first image to *its corresponding location* in the second, even if it is occluded or outside the image [1, 2]. This close relationship between matching and flow estimation has led to a common design where both operate on the same spatial grid.

Yet matching and flow estimation have different demands: Reliable matching needs sufficient visual context, while flow must be estimated precisely at every pixel. A distinctive corner is easy to match across images, making its displacement easy to estimate. A pixel on a white wall, however, could match many white pixels in the second image, making its displacement hard to pinpoint. Yet the wall itself may be easier to identify. Similarly, when part of an object is occluded or leaves the image, its motion may be inferred from the visible parts. In both cases, matching coarser structures can help estimate precise pixel flow where direct matches are unreliable or unavailable. The challenge is that broader spatial support can resolve ambiguous matches but risks mixing pixels with distinct motions, while precise flow must distinguish pixels on opposite sides of motion boundaries [3, 4].

Modern flow methods address these demands in two stages. They aggregate information from larger regions for reliable matching, but matching and flow estimation remain tied to the same fixed grid, often at reduced resolution for efficiency [5–8]. Learned upsampling then recovers precise dense flow

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

from coarse estimates using pixel-specific weights [5, 8]. Ideally, the pixels contributing to a match should receive flow estimates from that match. Yet despite end-to-end training, these methods do not explicitly link these pixels to the weights used for dense flow reconstruction. This raises the question of whether their implicit relationship generalizes from synthetic training data to unseen real scenes.

Our key insight is that dense pixel-level flow does not require equally dense matching. We decouple correspondence scale from flow resolution by adapting the size and shape of spatial supports to image structure for reliable matching, while reusing these supports for precise dense flow reconstruction. **1) Correspondence should adapt** to image structure, matching distinctive visual entities through spatial supports that make them reliably identifiable. These matches provide flow estimates for dense reconstruction. **2) The same grouping should govern** which pixels contribute to each match and how each match contributes to their flow estimates. This links matching and reconstruction without requiring pixels in a group to move identically. **3) Grouping should anticipate common fate from visual appearance.** The Gestalt principle of *common fate* holds that elements moving together tend to group together [9, 10]. Yet matching requires deciding what belongs together *before* knowing how it moves. Prior work has used motion to learn appearance-based object segmentation [11, 12]. We instead use flow supervision to *learn appearance-based grouping for correspondence*, allowing the model to predict groups in unseen images for more reliable and precise flow estimation.

We realize these ideas in FATE, short for *Flow from Adaptive Token Estimation*. **1) Adaptive correspondence tokens** enable coarse-scale matching while retaining pixel-level flow prediction. Each token is anchored to a coarse grid, but its spatial support adapts in size and shape to image structure. As Fig. 1 shows, token supports expand within coherent regions and contract near motion boundaries, following regions of common motion rather than fixed grid cells. Local attention further enriches token descriptors with broader context, while iterative matching refines token flow [8]. **2) Pixel-to-token grouping** connects matching and dense flow reconstruction through a shared differentiable association matrix: Normalized over pixels, it pools features into token descriptors; normalized over tokens, it combines token flows to reconstruct precise pixel flow. Each token predicts one motion vector, but each pixel combines motions from multiple tokens, allowing distinct pixel flow estimates within the same token's spatial support. **3) Flow supervision teaches appearance-based grouping.** Pixel-to-token associations, token matching, and dense flow estimation are learned jointly, without grouping labels. At test time, the model predicts these associations from appearance alone, enabling more reliable and precise flow estimation in unseen images.

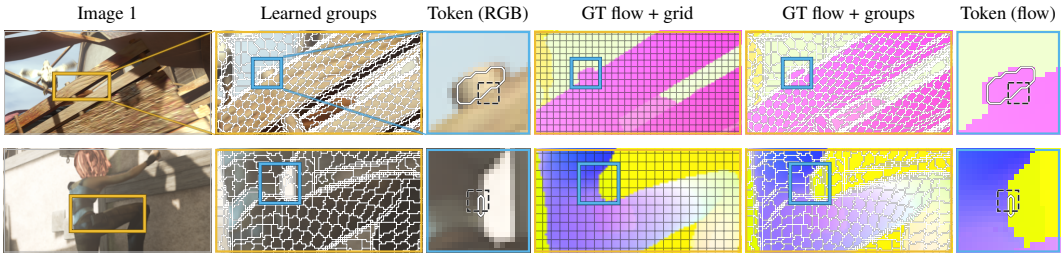


Figure 1: **FATE learns correspondence tokens whose spatial support adapts to image structure and motion boundaries.** Token supports spread within coherent regions and sharpen at motion boundaries, following common motion rather than fixed grid cells. Learned groups trace slat edges and a bent body outline, while fixed cells can straddle distinct motions. The same soft pixel-to-token associations pool features for matching and combine token motions into dense flow. White contours show dominant assignments and selected token supports; dashed squares mark nominal grid cells. Selected supports appear against RGB images and ground-truth (GT) flow in the token panels. Gold and blue boxes mark successive enlargements, with matching RGB and GT flow crops.

These ideas deliver stronger zero-shot generalization, a critical need in optical flow estimation. Dense flow labels are difficult to obtain in real scenes, so supervised models are often trained on synthetic images and must transfer to unseen domains without fine-tuning. While visual appearance may vary across domains, cues to which pixels are likely to move together may still transfer. Trained on FlyingChairs and FlyingThings3D [13, 14], FATE achieves endpoint errors of 2.10 on Sintel Final and 3.11 on KITTI, the lowest among published methods in our zero-shot comparison [15, 16]. Its largest gains occur where direct matches are unavailable: Occluded or out-of-frame pixels account for only 6.4–13.9% of valid pixels, yet contribute 41.9–52.7% of the total endpoint error of WAFT[8].

At equal token counts, FATE outperforms fixed-grid alternatives, benefits more from local attention, and better preserves pixel motion through pooling and reconstruction of ground-truth flow. With four refinement steps, it also outperforms five-step WAFT at lower computational cost.

In summary, we make three contributions. **1) Adaptive correspondence tokens** improve flow accuracy with fewer matching locations through learned spatial supports. **2) Shared pixel-to-token associations** provide consistent visual grounding for matching and dense flow reconstruction. **3) Strong zero-shot generalization** is demonstrated across benchmarks and supported by controlled analyses. These advances simultaneously enable reliable matching and precise pixel-level motion estimation, while generalizing effectively to unseen real-world scenes. Project website: FATE.

2 Related work

Spatial grouping in classical optical flow. Classical optical flow uses spatial support to infer motion from ambiguous local measurements. Lucas–Kanade [2] pools motion constraints within neighborhoods, while Horn–Schunck [1] couples neighboring estimates through smoothness. Robust estimation [3] accommodates competing motions. Ren [4] uses contour-based soft affinities and figure-ground organization to define spatially adaptive support for local affine motion estimation near boundaries. FATE learns spatial support jointly with matching features from flow supervision, using the resulting groups for correspondence and dense flow estimation.

Common fate and motion-guided grouping. The Gestalt principle of common fate holds that elements moving together are perceived as a group [9, 10]. AMD [11] jointly learns appearance-based segmentation and segment motion through video reconstruction, pooling motion features to predict segment flow. RCF [12] relaxes common fate by fitting flow with segment motion and within-segment residuals, then refines grouping from appearance. These methods learn object segmentation from motion, whereas FATE learns grouping for correspondence and dense flow. Its soft associations need not follow object boundaries or impose uniform flow within a token.

Correspondence in deep flow networks. Modern two-frame networks largely match features on regular grids. Learned pyramids and cost volumes [5, 17], global matching and cost-memory encoding [6, 7], and pretrained representations [18, 19] strengthen features for matching. Recurrent estimation and learned motion priors [20, 21] improve flow estimation. Source-image affinities aggregate motion features [22], while feature similarity guides flow propagation [6]. Multiscale processing and learned upsampling recover spatial detail [5, 8, 23, 24]. In these methods, pixel support for matching and the weights that distribute motion to pixels are linked only implicitly. Building on iterative matching [8], FATE learns adaptive correspondence tokens whose shared pixel-to-token associations pool features for matching and combine token motions into dense flow.

Superpixels and differentiable grouping. Superpixel methods partition images into compact regions following appearance boundaries [25, 26]. Learned variants make grouping differentiable and adaptable to downstream tasks [27, 28]. Unlike image segmentation, grouping in FATE serves correspondence and dense motion estimation. Flow supervision jointly learns the associations and matching features, without requiring pixel-group labels.

3 FATE: Flow from Adaptive Token Estimation

Overview. Given two images, FATE estimates dense optical flow by matching adaptive correspondence tokens at a coarser resolution, then combining their motions into pixel-level flow (Fig. 2). The model has three components. **1) Adaptive correspondence tokens** are anchored to points on a coarse grid, but their spatial supports are learned from image features. **2) Shared pixel-to-token associations** pool each image's dense features into token descriptors, enriched by local attention for broader matching context. Iterative matching estimates token motion, which the same associations combine into pixel flow in the first image. Since each pixel combines motions from several tokens, pixels within a token's support need not share identical flow. **3) Flow-supervised grouping** jointly learns associations, matching, and flow estimation from ground-truth pixel flow, without pixel grouping labels. At test time, the model anticipates which pixels are likely to move together from appearance alone, enabling reliable matching and precise pixel flow in unseen scenes.

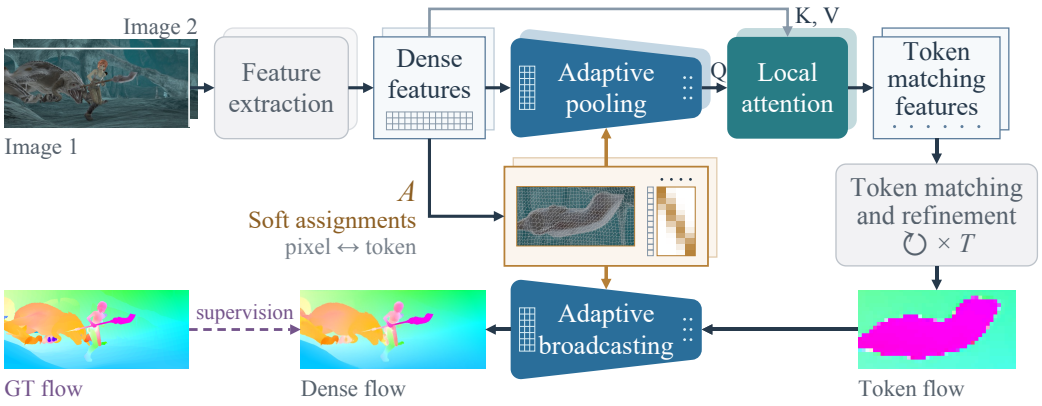


Figure 2: **Shared pixel-to-token associations connect token matching and dense flow estimation.** Learned associations A pool dense image features into adaptive correspondence tokens (center), enriched with nearby context through local attention before iterative matching across images (right). The same associations combine estimated token motions into dense pixel flow (bottom), allowing each pixel to draw on multiple tokens. Image 2 (lighter branch) has its own associations but shares network weights with Image 1. Dashed arrows indicate ground-truth flow supervision (Fig. 4).

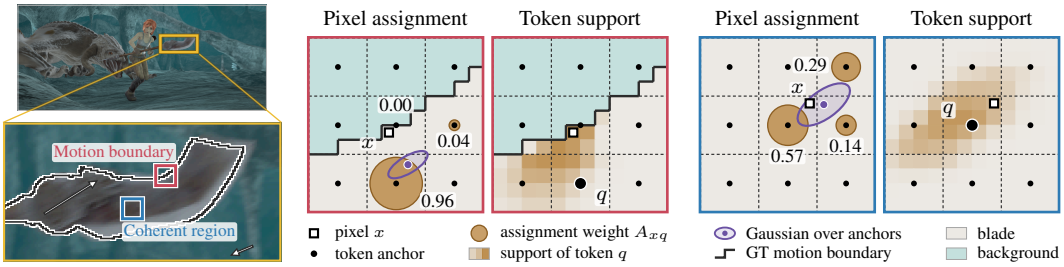


Figure 3: **Learned pixel-to-token associations adapt token support to motion structure.** Left: the original image and an enlarged view of the moving blade, with two marked regions: a motion boundary and a coherent region. Arrows indicate ground-truth motion. Center and right: Each region is shown from two perspectives. Pixel assignment shows how pixel x associates with nearby tokens (a row of A); token support shows the spatial support of token q (a column of A). Dashed grid lines mark the coarse token lattice, black dots mark token anchors, and squares mark pixels. Brown disks represent individual assignment weights A_{xq}, with size indicating weight; purple ellipses show predicted Gaussians, and brown shading shows token support. At the motion boundary, x favors a blade token across grid-cell borders, allowing its support to align with the boundary. Within the coherent region, x distributes its associations across several nearby tokens.

Adaptive correspondence tokens. For an image with N pixels indexed by x and M tokens indexed by q, we learn a pixel-to-token association matrix $A \in \mathbb{R}^{N \times M}$ of soft assignments, where $A_{xq} \geq 0$ measures how strongly pixel x is associated with token q. Each row describes a pixel's associations with tokens, while each column describes a token's spatial support.

Predicting associations from appearance. Each pixel associates with the 3×3 token anchors around its grid cell. Rather than predicting nine independent scores, a compact head predicts a two-dimensional Gaussian over these anchors from dense image features. Its center determines which nearby tokens the pixel favors, while its shape controls how widely the associations spread. Evaluating the Gaussian at the candidate anchors and normalizing over them gives the corresponding entries of A (Appendix F). As Fig. 3 illustrates, the resulting token supports can follow motion boundaries rather than fixed cell borders, while spreading more broadly within coherent regions.

Shared pixel-to-token associations. The same learned associations govern how pixels support their token matching and how token motions are combined into dense pixel flow (Fig. 4).

1) Pooling and broadcasting. The association matrix A connects pixel and token spaces in two ways: Normalized over pixels (columns of A), it pools features into token descriptors; normalized over tokens (rows of A), it combines token motions into pixel flow. Let $D(B) = \mathrm{diag}(B\mathbf{1})$ denote the diagonal degree matrix of row sums. Omitting the dependence on A, we have two operators:

$$\text{pixel-to-token feature pooling:} \quad P_q = D(A^\top)^{-1}A^\top, \tag{1}$$

$$\text{token-to-pixel flow broadcasting:} \quad P_x = D(A)^{-1}A. \tag{2}$$

The shared associations provide consistent visual grounding for matching and pixel flow.

2) Token features with local attention. Reliable matching requires both features pooled from each token's support and broader context to resolve ambiguity. Local attention contextualizes each token's pooled features with surrounding dense features, sharing network weights for both images.

3) Iterative flow refinement and broadcasting. Let F_q and F_x denote token and pixel flow from the first image to the second. Following the iterative matching framework of WAFT [8], each refinement step samples the second image's token features at current match locations and updates the token flow. Both images' associations are computed once and reused throughout refinement. At step t, the first image's broadcasting operator produces dense pixel flow:

$$F_x^{(t)} = P_x F_q^{(t)}. \tag{3}$$

Each token predicts one motion vector, but each pixel combines motions from multiple tokens, allowing distinct pixel flow estimates within the same token's spatial support.

Flow-supervised grouping. We jointly learn pixel-to-token associations and token-flow prediction from ground-truth pixel flow, since no annotation specifies which pixels should be grouped for correspondence (Fig. 4). Three complementary losses supervise token matching, pixel-to-token associations, and preservation of ground-truth pixel motion through pooling and reconstruction. Each has the form $\mathcal{L}(X \mid Y)$, training X with Y given.

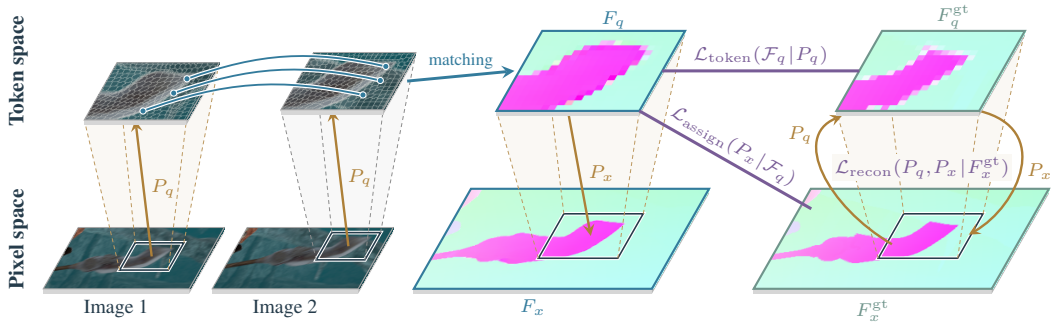


Figure 4: **Matching in token space estimates flow in pixel space through shared associations and flow supervision.** Matching operates in coarse token space (top), while dense flow is formed in pixel space (bottom), across both the image features (left) and flow predictions (right). Features from Image 1 and Image 2 are pooled into adaptive tokens and matched across images (blue links). On the right, predicted token flow F_q is combined into pixel flow F_x, while ground-truth pixel flow F_x^{gt} is pooled into pseudo-ground-truth token flow F_q^{gt}. The probabilistic predictions $\mathcal{F}_q$ include token flow F_q and its uncertainty. We learn token matching and pixel-to-token associations using only ground-truth pixel flow, through three complementary losses: $\mathcal{L}_{\mathrm{token}}(\mathcal{F}_q \mid P_q)$ trains token-flow prediction against pooled ground truth, holding P_q fixed in constructing the pseudo-ground truth; $\mathcal{L}_{\mathrm{assign}}(P_x \mid \mathcal{F}_q)$ trains associations to favor tokens that confidently explain pixel motion, holding token-flow predictions fixed; and $\mathcal{L}_{\mathrm{recon}}(P_q, P_x \mid F_x^{\mathrm{gt}})$ trains flow-preserving grouping by pooling and reconstructing fixed ground-truth flow, independently of matching. Upper views enlarge the boxed regions, token colors show pooled RGB, and flow maps share one color scale.

1) Token-flow prediction. Since matching operates at the coarse token level, we construct pseudo-ground-truth token flow by pooling ground-truth pixel flow, provisionally assuming the current associations are correct. Following SEA-RAFT [20], the network predicts token flow F_q and its uncertainty.

We denote these probabilistic predictions by $\mathcal{F}_q$ and define $\mathrm{NLL}(\mathcal{P}; Y) = -\sum_i \log p_{\mathcal{P}_i}(Y_i)$ as the negative log-likelihood of targets Y under predicted distributions $\mathcal{P}$:

$$F_q^{\mathrm{gt}} = P_q F_x^{\mathrm{gt}}, \tag{4}$$

$$\mathcal{L}_{\mathrm{token}}(\mathcal{F}_q \mid P_q) = \mathrm{NLL}(\mathcal{F}_q; F_q^{\mathrm{gt}}). \tag{5}$$

This loss primarily trains token matching, updating token features, local attention, iterative flow refinement, and uncertainty prediction. It also trains pixel-to-token associations through gradients from feature pooling. However, gradients are stopped when constructing pseudo-ground-truth token flow, preventing the associations from shifting this supervision to reduce the loss.

2) Pixel-to-token associations. Near motion boundaries, neighboring tokens may predict different flows. Averaging them with spread-out weights in Eq. (3) can produce a spurious intermediate flow. To preserve distinct motions, we learn pixel-to-token associations based on how well each token's predicted flow explains the pixel's ground truth, accounting for prediction uncertainty. Provisionally holding token-flow predictions and uncertainties fixed, we evaluate the likelihood of each pixel's ground-truth flow under each neighboring token's predicted distribution. The association weights P_x combine these likelihoods into a mixture, denoted by $\mathrm{Mix}_{P_x}[\mathcal{F}_q]$. We minimize

$$\mathcal{L}_{\mathrm{assign}}(P_x \mid \mathcal{F}_q) = \mathrm{NLL}\big(\mathrm{Mix}_{P_x}[\mathcal{F}_q]; F_x^{\mathrm{gt}}\big)\,. \tag{6}$$

Unlike averaging token flows, this mixture retains distinct predictions and uncertainties, encouraging pixels to associate with tokens that confidently explain their motion. With token predictions fixed, this loss updates the assignment head and its input features, not token matching or refinement.

3) Flow-preserving reconstruction. Ground-truth pixel flow should also be preserved when pooled into coarse token flow and reconstructed back at pixel resolution. We therefore minimize the reconstruction error, with an aggregated robust flow penalty $\mathcal{D}$:

$$\mathcal{L}_{\mathrm{recon}}(P_q, P_x \mid F_x^{\mathrm{gt}}) = \mathcal{D}\big(P_x P_q F_x^{\mathrm{gt}}, F_x^{\mathrm{gt}}\big)\,. \tag{7}$$

As in superpixel learning [27], this loss updates the associations through both P_q and P_x, encouraging the shared pixel-to-token associations to preserve distinct pixel motions. Since matching is bypassed, the loss trains grouping independently of token-flow prediction.

4) Joint training. We optimize all three losses jointly, applying token-flow and association losses at each refinement step and reconstruction loss once (Appendix F). Together, they learn token flow and pixel-to-token associations to achieve both reliable matching and precise pixel-level flow.

4 Experiments

Our experiments test whether consistent visual grounding improves dense flow estimation and its transfer to unseen domains. We first compare zero-shot accuracy with published methods. We then examine where errors arise and where FATE reduces them, focusing on two regions: occluded or out-of-frame pixels, which lack a direct match and depend on shared evidence, and pixels near motion boundaries, where distinct motions must be preserved. A comparison with a fixed-grid model at the same token count isolates the role of adaptive grouping. Ablations test design choices, and computation profiles show what fewer tokens make affordable.

Experimental setup. All compared methods are trained on FlyingChairs [13] followed by FlyingThings3D (Things) [14] and evaluated on two frames.

1) Models and training. We build FATE on WAFT's iterative matching and refinement framework [8] with a DINOv3 ViT-S backbone [29]. For an $H \times W$ image, both extract dense features on an $(H/2) \times (W/2)$ grid, abbreviated $H/2$. Released WAFT estimates flow on this grid, whereas FATE estimates it for adaptive tokens anchored on an $H/4$ lattice, with a quarter as many tokens. FATE and WAFT use five refinement steps in training and inference unless stated otherwise (Appendix D).

2) Evaluation protocol. Things test results are in distribution, while results on the Sintel [15] and KITTI 2015 [16] training splits are zero-shot, without target-domain flow training, as in real scenes where dense flow labels are difficult to acquire. Things and Sintel each have Clean and Final versions.

3) Metrics. We report endpoint error (EPE), the Euclidean flow error in pixels averaged per image and then over images, and KITTI Fl, the percentage of valid pixels whose error exceeds both 3 pixels and 5% of the ground-truth motion. Regional analyses pool endpoint errors over pixels (Appendix B). Computation is given in GMACs, billions of multiply–accumulates, per 540×960 image pair.

Table 1: **FATE achieves the lowest zero-shot Sintel Final and KITTI EPE among the compared methods.** All methods train on FlyingChairs and FlyingThings3D. Errors are EPE except KITTI Fl. Bold: lowest error in each accuracy column; –: not reported.

Method	Sintel Clean	Sintel Final	KITTI EPE	KITTI Fl (%)	GMACs
FlowNet2 [31]	2.02	3.14	10.10	30.4	230
PWC-Net [17]	2.55	3.93	10.40	33.7	101
RAFT [5]	1.43	2.71	5.04	17.4	938
GMA [22]	1.30	2.74	4.69	17.1	1352
SKFlow [32]	1.22	2.46	4.27	15.5	751
DIP [33]	1.30	2.82	4.29	13.7	3068
CRAFT [34]	1.27	2.79	4.88	17.5	2274
GMFlowNet [35]	1.14	2.71	4.24	15.4	1094
FlowFormer [7]	1.01	2.40	4.09	14.7	1715
EMD-L [36]	0.88	2.55	4.12	13.5	1755
FlowFormer++ [37]	0.90	2.30	3.93	14.1	1713
MatchFlow(G) [18]	1.03	2.45	4.08	15.6	1669
AnyFlow [24]	1.10	2.52	3.76	12.4	–
RPKNet [38]	1.12	2.45	–	13.0	137
CCMR+ [39]	0.98	2.36	–	12.9	12653
SEA-RAFT(L) [20]	1.19	4.11	3.62	12.9	655
SAMFlow [19]	0.87	2.11	3.44	12.3	9717
FlowDiffuser [21]	**0.86**	2.19	3.61	11.8	2466
DPFlow [23]	1.02	2.26	3.37	**11.1**	414
WAFT [8]	1.28	2.56	3.49	12.9	732
FATE	1.03	**2.10**	**3.11**	11.9	796

Table 2: **FATE's gain over WAFT is larger at occluded or out-of-frame pixels and near motion boundaries than at coherent pixels.** The categories partition valid pixels, with visible pixels split by their distance to ground-truth motion boundaries (Appendix B). Pixel and Error give percentages of the pixels and of WAFT's summed endpoint error; WAFT and FATE give pooled EPE, and Gain is WAFT minus FATE. Bold marks the lower EPE in each pair.

	Coherent					Near boundary					Occluded/out-of-frame				
Dataset	Pixel	Error	WAFT	FATE	Gain	Pixel	Error	WAFT	FATE	Gain	Pixel	Error	WAFT	FATE	Gain
Sintel Clean	87.9	36.4	0.52	**0.38**	0.14	3.6	10.8	3.79	**3.24**	0.54	8.4	52.7	7.96	**6.78**	1.18
Sintel Final	87.9	46.7	1.35	**1.05**	0.29	3.6	10.7	7.51	**6.63**	0.87	8.4	42.5	12.88	**10.93**	1.95
KITTI	77.8	36.8	**1.72**	1.76	−0.04	8.1	10.8	4.83	**4.81**	0.01	13.9	52.3	13.63	**11.05**	2.57
Things Clean	84.8	35.4	1.45	**0.92**	0.53	8.7	22.0	8.82	**5.76**	3.06	6.4	42.5	23.04	**16.56**	6.47
Things Final	84.8	35.1	1.39	**0.93**	0.46	8.7	22.9	8.87	**6.03**	2.83	6.4	41.9	21.96	**15.88**	6.07

Zero-shot transfer and regional accuracy. We compare overall accuracy with published results for prior methods and regional accuracy with the released WAFT checkpoint.

1) FATE achieves the lowest zero-shot EPE on Sintel Final and KITTI. In Table 1, FATE has lower Sintel Final and KITTI EPE than every compared method and improves on WAFT in every accuracy column. With a Depth Anything V2 backbone [30] instead of DINOv3, FATE again has lower EPE than WAFT on both Sintel and KITTI (Appendix A).

2) Pixels without direct matches or near motion boundaries carry most of WAFT's error. In Table 2, occluded or out-of-frame pixels make up only 6.4–13.9% of the pixels yet account for 41.9–52.7% of WAFT's summed endpoint error. Near-boundary pixels likewise account for 10.7–22.9% of the error from 3.6–8.7% of the pixels. Coherent pixels, the remaining 77.8–87.9%, account for only 35.1–46.7% of the error, so overall EPE reveals little about accuracy in the other two regions. Fig. 5 illustrates three cases that make flow hard: a thin structure near motion boundaries, an occluded background and misleading lighting. Fig. 6 shows more such cases over wider views.

3) FATE's largest gains occur where direct matches are missing. Coherent pixels gain at most 0.53 pixels, versus 1.18–2.57 pixels for occluded or out-of-frame pixels on Sintel and KITTI and over 6 on Things, the largest gain in each row of Table 2. Without direct matches, these pixels' motion must be inferred from shared evidence, such as visible parts of the same moving region. In Fig. 5(b), FATE recovers the motion of leaves hidden in Image 2, unlike WAFT; Fig. 6 shows similar cases

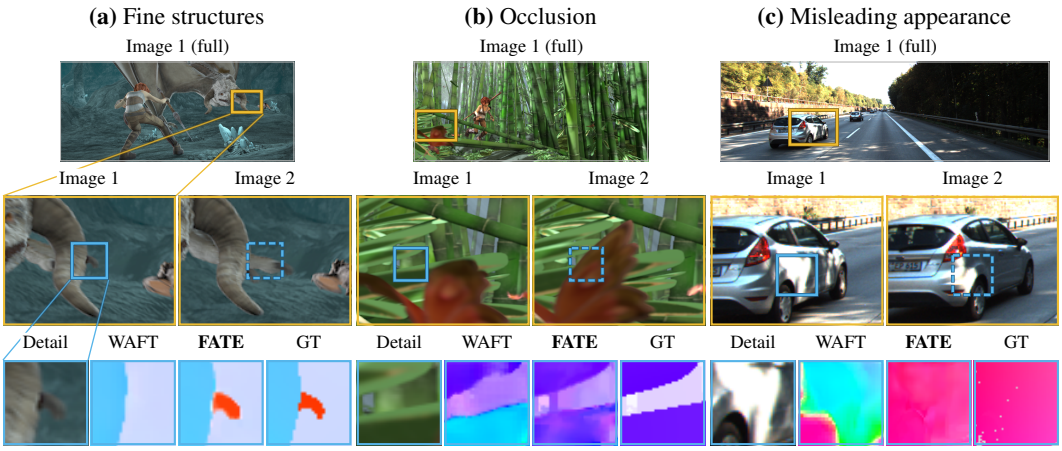


Figure 5: **FATE better handles fine structures, occlusion and misleading appearance.** (a) A thin protrusion moves unlike the surfaces on both sides; per-pixel assignments keep its motion distinct. (b) Leaves hidden by a wing in Image 2 have no match; their tokens infer motion from surrounding context. (c) Sunlight and shadows on a car change between frames; WAFT's flow follows them, while FATE's tokens keep the whole surface's motion coherent. Gold and blue boxes locate context and detail crops; dashed boxes repeat Image 1 coordinates in Image 2. All predictions are zero-shot; colors share a scale per example, and gray marks unlabeled GT.

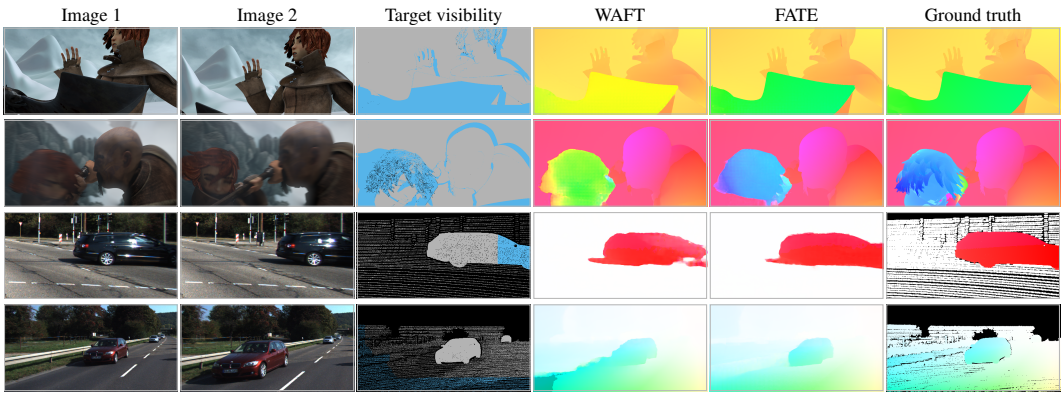


Figure 6: **FATE recovers motion where matches are missing or ambiguous and preserves motion boundaries.** Matches are missing for an occluded cape (row 1) and a car leaving the frame (row 3), and ambiguous for blurred hair (row 2); an approaching car's motion must stay separate from the road's (row 4). Target visibility marks occluded or out-of-frame pixels in blue, other valid pixels in gray, and excluded pixels in black. All predictions are zero-shot on Sintel (rows 1–2) and KITTI (rows 3–4); each row shares a crop and color scale, and sparse KITTI labels are enlarged for display.

in rows 1 and 3. Each token pools evidence over its learned support, draws on local context, and contributes its motion estimate to associated pixels; Table 3(b) tests the benefit of context at equal token counts. Since these pixels are few, overall EPE dilutes their gain to 0.38 pixels on KITTI (Table 1), suggesting greater benefit where missing matches are more common.

4) FATE is more precise near motion boundaries with a quarter of WAFT's tokens. Mixing motions from different surfaces can give boundary pixels incorrect flow, a risk finer tokens should reduce. Yet FATE's near-boundary EPE is 0.54–3.06 pixels lower than WAFT's on Sintel and Things, three to six times its coherent gain, and matches WAFT on KITTI. In Fig. 5(a), WAFT assigns a thin protrusion the background's motion, whereas FATE keeps it distinct. Including occluded or out-of-frame pixels, FATE lowers boundary EPE on every dataset at all three thresholds (Appendix B.2; Fig. 6, row 4). As Figs. 1 and 3 show, learned support follows moving surfaces across grid-cell borders, separating pixels on opposite sides of motion boundaries.

Table 3: **At the same token count, adaptive tokens improve accuracy, gain more from local context and preserve more pixel motion.** All systems estimate flow for $H/4$ tokens. (a) Overall accuracy. (b) Occluded/out-of-frame EPE; attention gain is EPE without minus with local attention. (c) EPE of ground-truth flow pooled into tokens and returned to pixels with each model's image-predicted weights, without retraining. Bold marks the best in (a) and (c) and the larger gain in (b).

(a) Overall accuracy		Things		Sintel		KITTI	
System	Attn.	Clean	Final	Clean	Final	EPE	Fl (%)
WAFT-H/4	–	2.93	2.72	1.08	2.49	3.96	13.3
WAFT-H/4	✓	2.73	2.62	1.07	2.57	3.75	12.6
FATE	–	2.56	2.48	1.12	2.14	3.54	13.0
FATE	✓	**2.39**	**2.38**	**1.03**	**2.10**	**3.11**	**11.9**
(c) Supplied-motion reconstruction EPE							
WAFT-H/4	✓	0.50	0.53	0.21	0.39	1.52	–
FATE	✓	**0.19**	**0.21**	**0.09**	**0.19**	**0.39**	–

(b) Occluded/out-of-frame EPE		Things		Sintel		KITTI
System	Attn.	Clean	Final	Clean	Final	EPE
WAFT-H/4	–	19.65	17.79	7.15	12.91	14.87
WAFT-H/4	✓	18.98	17.64	7.19	12.90	14.09
Attention gain		0.67	0.15	−0.03	0.01	0.78
FATE	–	17.49	16.55	6.89	11.40	12.96
FATE	✓	16.56	15.88	6.78	10.93	11.05
Attention gain		**0.93**	**0.66**	**0.11**	**0.46**	**1.90**

Adaptive versus fixed-grid correspondence. FATE differs from released WAFT in the number of tokens and how they are formed. To remove the first difference, we train WAFT-H/4, a fixed-grid baseline that estimates flow on an $H/4$ grid with WAFT's code and loss. It matches FATE's token count, backbone, dense-feature resolution, iterative framework, refinement steps and training recipe, but its tokens are fixed grid cells. We train both with and without local attention and compare accuracy, the use of context for pixels without direct matches and the preservation of pixel motion.

1) Adaptive tokens improve every reported accuracy result. In Table 3(a), FATE outperforms both WAFT-H/4 variants on every reported accuracy result. With attention in both systems, it lowers Sintel Final EPE from 2.57 to 2.10 and KITTI EPE from 3.75 to 3.11. Without attention in either, FATE still improves every result except Sintel Clean EPE. FATE's advantage over WAFT therefore does not come from using coarser tokens or from local attention alone.

2) Adaptive tokens benefit more from local context. In Table 3(b), FATE has lower occluded/out-of-frame EPE than WAFT-H/4 even without local attention, and adding attention widens this gap on every dataset. Attention lowers FATE's EPE by 1.90 pixels on KITTI versus 0.78 for WAFT-H/4, and changes WAFT-H/4's EPE on Sintel by less than 0.05 pixels. This supports our account of the gains in Table 2: learned support defines which pixels share an estimate, so a token can draw on broader context while its estimate remains tied to the pixels it represents.

3) Learned support preserves more pixel motion. Grouping pixels to share evidence should not erase the motion differences among them. To measure how much pixel motion survives grouping, independent of matching accuracy, we replace predicted token flow with pooled ground truth and return it to pixels with each model's image-predicted weights (Appendix C). In Table 3(c), FATE has lower reconstruction EPE than WAFT-H/4 on every dataset: 0.19 versus 0.39 on Sintel Final and 0.39 versus 1.52 on KITTI. At the same token count, learned support thus preserves motion that fixed cells average away. Consistent with this, FATE also has lower EPE than WAFT-H/4 near motion boundaries on every dataset at all three tested thresholds (Appendix B.2).

Learning effective correspondence tokens. Table 4 tests how supervision, local context and the assignment parameterization help tokens learn, changing one choice at a time and training each with the same recipe. Each design choice improves at least five of the six reported results.

1) Training the grouping to preserve motion improves every result. Compared with training without it, the reconstruction loss (Eq. (7)) improves every result, most on Things, where Clean EPE falls from 2.68 to 2.39. Unlike the token and assignment losses, which act through predicted token flow, it trains pooling and broadcasting directly on ground-truth flow, so the grouping learns which motion differences to preserve independently of matching accuracy.

2) Local attention improves every result. Compared with a learned projection of pooled features, local attention improves every result, most on KITTI, where EPE falls from 3.54 to 3.11. It lets each token draw on nearby dense features, while the assignments still decide which pixels receive its flow.

Table 4: **Each design choice improves at least five of the six results.** Each variant changes one choice of FATE and is trained with the same recipe. *No reconstruction loss* trains without Eq. (7). *No local attention* forms token features by a learned projection of pooled features. *End-to-end assignment supervision* lets the assignment loss (Eq. (6)) also update token flow and uncertainty; pseudo-ground-truth token flow stays fixed. With *independent assignment scores*, the assignment head predicts the nine candidate scores directly instead of through a Gaussian. Errors are EPE except KITTI Fl. Bold marks the lowest error in each column.

Variant	Things Clean	Things Final	Sintel Clean	Sintel Final	KITTI EPE	KITTI Fl (%)
FATE	**2.39**	**2.38**	1.03	**2.10**	**3.11**	**11.9**
No reconstruction loss	2.68	2.61	1.05	2.17	3.14	12.4
No local attention	2.56	2.48	1.12	2.14	3.54	13.0
End-to-end assignment supervision	2.49	2.46	**1.00**	2.18	3.64	12.3
Independent assignment scores	2.58	2.55	1.02	2.24	3.16	12.1

3) Supervising assignments directly, rather than end to end, improves accuracy. FATE supervises each module for its role (Fig. 4). The token loss trains matching to produce each token's correspondence, and the assignment loss trains the assignments to decide which token should carry each pixel, given the current probabilistic token-flow predictions. Applied end to end, the assignment loss also updates these predictions, so it can be lowered by shifting a token's flow or widening its uncertainty toward a disagreeing pixel instead of reassigning that pixel. Direct assignment supervision improves every result except Sintel Clean EPE, most on KITTI, where EPE falls from 3.64 to 3.11. Features, token predictions and assignments are still learned jointly.

4) Gaussian assignment scores outperform independently predicted scores. Compared with predicting the nine candidate scores independently from the same head, Gaussian scores improve every result except Sintel Clean EPE, including Sintel Final EPE, which falls from 2.24 to 2.10. The Gaussian couples each pixel's candidate scores through the anchors' relative positions (Appendix F), yet a token's support can still take an irregular shape (Fig. 3).

Accuracy and computation. FATE predicts assignments at image resolution but matches and refines fewer tokens. Appendix D profiles how this allocation affects accuracy and cost.

1) Fewer tokens make wider refinement and local attention affordable. In Table 7, with a quarter as many tokens, FATE refines with twice WAFT's recurrent-state and refinement width at similar refinement cost, 453.9 versus 425.8 GMACs, whereas widening WAFT the same way would more than double its refinement cost. Local attention of the same width likewise costs 20.6 GMACs over FATE's $H/4$ tokens versus 57.0 GMACs over WAFT's $H/2$ grid.

2) With four refinement steps, FATE outperforms five-step WAFT at lower cost. In Table 8, four-step FATE has lower error than five-step WAFT on every reported result while using 706.1 versus 732.4 GMACs. Our five-step default costs more than WAFT, 796.8 GMACs, and further lowers error on five of the six reported results. A narrower FATE with WAFT's width costs 636.3 GMACs and has lower error than WAFT on every reported result except KITTI EPE.

5 Conclusion

Optical flow requires reliable matching from sufficient visual context and precise motion at every pixel. FATE reconciles these demands by decoupling correspondence scale from flow resolution while maintaining consistent visual grounding. Learned pixel-to-token associations adapt the size and shape of correspondence support to image structure, pooling features for matching and combining token motions into dense pixel flow. Flow supervision learns these associations without grouping labels, allowing the model to predict them from appearance alone.

FATE achieves the lowest zero-shot Sintel Final and KITTI EPE among compared published methods. Its largest gains occur at occluded or out-of-frame pixels, with further gains near motion boundaries. At equal token counts, adaptive grouping outperforms fixed-grid alternatives, benefits more from local attention, and better preserves pixel motion. With four refinement steps, FATE also outperforms five-step WAFT at lower cost. These results show that adaptive correspondence support enables reliable matching and precise pixel-level flow, with strong generalization to unseen real scenes.

Acknowledgements and disclosures

This project was supported in part by NSF grant 2313151, with additional computing resources provided through NAIRR Pilot CIS240421 and CIS240431.

The authors declare no competing interests.

References

[1] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 (1):185–203, August 1981. ISSN 0004-3702. doi: 10.1016/0004-3702(81)90024-2.

[2] Bruce D Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI'81: 7th International Joint Conference on Artificial Intelligence*, volume 2, pages 674–679, Vancouver, Canada, August 1981.

[3] Michael J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996. URL `https://www.csd.uoc.gr/~hy576/bibliography/RobustMotion.pdf`.

[4] Xiaofeng Ren. Local grouping for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. URL `https://iacl.ece.jhu.edu/proceedings/cvpr2008/papers/196.pdf`.

[5] Zachary Teed and Jia Deng. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 402–419, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58536-5. doi: 10.1007/978-3-030-58536-5_24.

[6] 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.

[7] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer: A Transformer Architecture for Optical Flow. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 668–685, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19790-1. doi: 10.1007/978-3-031-19790-1_40.

[8] Yihan Wang and Jia Deng. WAFT: Warping-Alone Field Transforms for Optical Flow, October 2025.

[9] Max Wertheimer. Laws of organization in perceptual forms. In *A Source Book of Gestalt Psychology*, pages 71–88. Kegan Paul, Trench, Trubner & Company, London, England, 1938. doi: 10.1037/11496-005.

[10] Alan L. Yuille and Norberto M. Grzywacz. A computational theory for the perception of coherent visual motion. *Nature*, 333(6168):71–74, May 1988. ISSN 1476-4687. doi: 10.1038/333071a0.

[11] Runtao Liu, Zhirong Wu, Stella X. Yu, and Stephen Lin. The emergence of objectness: Learning zero-shot segmentation from videos. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL `https://papers.neurips.cc/paper/2021/hash/6d9cb7de5e8ac30bd5e8734bc96a35c1-Abstract.html`.

[12] Long Lian, Zhirong Wu, and Stella X. Yu. Bootstrapping objectness from videos by relaxed common fate and visual grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14582–14591, 2023. doi: 10.1109/CVPR52729.2023.01401. URL `https://openaccess.thecvf.com/content/CVPR2023/html/Lian_Bootstrapping_Objectness_From_Videos_by_Relaxed_Common_Fate_and_Visual_CVPR_2023_paper.html`.

[13] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow With Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[14] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[15] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A Naturalistic Open Source Movie for Optical Flow Evaluation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 611–625, Berlin, Heidelberg, 2012. Springer. ISBN 978-3-642-33783-3. doi: 10.1007/978-3-642-33783-3_44.

[16] Moritz Menze and Andreas Geiger. Object Scene Flow for Autonomous Vehicles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015.

[17] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[18] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Rethinking Optical Flow from Geometric Matching Consistent Perspective. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1337–1347, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.00135.

[19] Shili Zhou, Ruian He, Weimin Tan, and Bo Yan. SAMFlow: Eliminating Any Fragmentation in Optical Flow with Segment Anything Model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):7695–7703, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i7.28603.

[20] Yihan Wang, Lahav Lipson, and Jia Deng. SEA-RAFT: Simple, Efficient, Accurate RAFT for Optical Flow. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 36–54, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-72667-5. doi: 10.1007/978-3-031-72667-5_3.

[21] Ao Luo, Xin Li, Fan Yang, Jiangyu Liu, Haoqiang Fan, and Shuaicheng Liu. FlowDiffuser: Advancing Optical Flow Estimation with Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19167–19176, 2024.

[22] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning To Estimate Hidden Motions With Global Motion Aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021.

[23] Henrique Morimitsu, Xiaobin Zhu, Roberto M. Cesar, Xiangyang Ji, and Xu-Cheng Yin. DPFlow: Adaptive Optical Flow Estimation with a Dual-Pyramid Framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17810–17820, 2025.

[24] Hyunyoung Jung, Zhuo Hui, Lei Luo, Haitao Yang, Feng Liu, Sungjoo Yoo, Rakesh Ranjan, and Denis Demandolx. AnyFlow: Arbitrary Scale Optical Flow With Implicit Neural Representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5455–5465, 2023.

[25] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, November 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2012.120.

[26] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 13–26, Berlin, Heidelberg, 2012. Springer. ISBN 978-3-642-33786-4. doi: 10.1007/978-3-642-33786-4_2.

[27] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel Sampling Networks, July 2018.

[28] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel Segmentation With Fully Convolutional Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13964–13973, 2020.

[29] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, August 2025.

[30] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth Anything V2. *Advances in Neural Information Processing Systems*, 37:21875–21911, December 2024. doi: 10.52202/079017-0688.

[31] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of Optical Flow Estimation With Deep Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2462–2470, 2017.

[32] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. SKFlow: Learning Optical Flow with Super Kernels. In *Advances in Neural Information Processing Systems*, 2022.

[33] Zihua Zheng, Ni Nie, Zhi Ling, Pengfei Xiong, Jiangyu Liu, Hao Wang, and Jiankun Li. DIP: Deep Inverse Patchmatch for High-Resolution Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8925–8934, 2022.

[34] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. CRAFT: Cross-Attentional Flow Transformer for Robust Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17602–17611, 2022.

[35] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global Matching with Overlapping Attention for Optical Flow Estimation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17571–17580, New Orleans, LA, USA, June 2022. IEEE. ISBN 978-1-6654-6946-3. doi: 10.1109/CVPR52688.2022.01707.

[36] Changxing Deng, Ao Luo, Haibin Huang, Shaodan Ma, Jiangyu Liu, and Shuaicheng Liu. Explicit Motion Disentangling for Efficient Optical Flow Estimation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9487–9496, Paris, France, October 2023. IEEE. ISBN 979-8-3503-0718-4. doi: 10.1109/ICCV51070.2023.00873.

[37] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. FlowFormer++: Masked Cost Volume Autoencoding for Pretraining Optical Flow Estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1599–1610, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.00160.

[38] Henrique Morimitsu, Xiaobin Zhu, Xiangyang Ji, and Xu-Cheng Yin. Recurrent Partial Kernel Network for Efficient Optical Flow Estimation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5):4278–4286, March 2024. ISSN 2374-3468. doi: 10.1609/aaai.v38i5.28224.

[39] Azin Jahedi, Maximilian Luz, Marc Rivinius, and Andrés Bruhn. CCMR: High Resolution Optical Flow Estimation via Coarse-To-Fine Context-Guided Motion Reasoning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6899–6908, 2024.

Common setting. Unless stated otherwise, we evaluate the DINOv3 systems from Section 4: FATE, released WAFT, and the fixed-grid WAFT-H/4 baseline with local attention, all trained on FlyingChairs and FlyingThings3D. The fine-tuned results in Appendix E are the exception.

A Zero-shot transfer across backbones

In Table 5, FATE lowers zero-shot EPE on Sintel and KITTI with both DINOv3 and Depth Anything V2 (DAv2) [30], compared with WAFT's published results [8]. KITTI Fl also improves with DINOv3, and with DAv2 the two models tie at the reported precision.

Table 5: **FATE lowers EPE with both backbones.** Zero-shot results on the Sintel and KITTI training splits. Bold: lowest error within each backbone, including ties at the reported precision.

		Sintel		KITTI	
Backbone	Method	Clean	Final	EPE	Fl (%)
DINOv3	WAFT [8]	1.28	2.56	3.49	12.9
	FATE	**1.03**	**2.10**	**3.11**	**11.9**
DAv2	WAFT [8]	1.01	2.49	3.28	**10.9**
	FATE	**0.97**	**2.30**	**2.93**	**10.9**

B Regional evaluation

The regional analyses test accuracy where direct matches are unavailable, at occluded or out-of-frame pixels, and where neighboring motions differ, near motion boundaries.

B.1 Occluded and out-of-frame pixels

Pixel sets. A pixel is out of frame if its ground-truth endpoint $x + F_x^{\mathrm{gt}}(x)$ falls outside Image 2. On Sintel and KITTI, we combine these pixels with those whose targets are marked occluded by the dataset annotations. On Things, we use endpoint bounds only, so in-frame occlusions are not identified and visible means in frame. All sets are restricted to the valid pixels used for standard EPE.

Sintel's official invalid mask further excludes pixels from visibility analysis, even when they remain valid for standard EPE. Table 2 partitions the remaining population: visible-boundary pixels lie in the boundary neighborhoods defined below, visible-coherent pixels are their complement among visible pixels (coherent does not imply constant flow), and the third group is non-visible.

Regional EPE and error share. Let V be the pixels eligible for visibility analysis, pooled over all evaluated image pairs, $R \subseteq V$ one of the three categories, and $e(x) = \|F_x(x) - F_x^{\mathrm{gt}}(x)\|_2$ the endpoint error at pixel x. Table 2 reports the regional EPE and, as a percentage, WAFT's error share

$$\mathrm{EPE}_R = \frac{1}{|R|} \sum_{x \in R} e(x), \qquad S_R = \frac{\sum_{x \in R} e(x)}{\sum_{x \in V} e(x)}. \tag{8}$$

Unlike standard EPE, which averages within each image first, both weight every pixel equally. The pixel and error shares of the three categories therefore each sum to 100% before truncation.

B.2 Accuracy near motion boundaries

Defining the evaluation region. To test accuracy where neighboring motions differ, we measure EPE over a region we define around ground-truth motion boundaries. We identify these boundaries from local changes in ground-truth flow F_x^{gt}, with strength

$$s(x) = \|\Delta_1 F_x^{\mathrm{gt}}(x)\|_2 + \|\Delta_2 F_x^{\mathrm{gt}}(x)\|_2, \tag{9}$$

where each Δ takes the difference to the next pixel along one image axis. A difference contributes only when both pixels have valid labels. For threshold τ, pixels with $s(x) > \tau$ seed the boundary

region, which contains valid pixels in a 5×5 window around any seed. We pool endpoint errors over this region. We evaluate $\tau = 2, 3, 4$ pixels; Table 2 uses $\tau = 2$ intersected with the visible set on Sintel and KITTI, or the in-frame set on Things. The threshold study in Table 6 includes both visible and non-visible pixels. At $\tau = 2$, this full boundary region covers 9.3% of valid pixels on Things, 6.2% on Sintel and 13.6% on KITTI.

Boundary precision improves across the tested thresholds. In Table 6, FATE has lower EPE in this region than both fixed-grid models on every dataset at all three thresholds.

Table 6: **FATE has lower EPE near motion boundaries at all tested thresholds.** EPE is pooled over the boundary region, including non-visible pixels; the threshold sets how large a local flow change must be to count as a boundary. WAFT estimates motion for four times as many tokens as FATE, and WAFT-H/4 with attention for the same number.

	Things		Sintel		KITTI
System	Clean	Final	Clean	Final	EPE
(a) Boundary threshold: 2 pixels					
WAFT	13.40	13.14	7.30	11.92	9.97
WAFT-H/4 + attention	10.77	10.65	6.97	11.40	11.75
FATE	**9.24**	**9.48**	**6.63**	**10.45**	**8.51**
(b) Boundary threshold: 3 pixels					
WAFT	12.50	12.37	7.38	12.52	12.84
WAFT-H/4 + attention	10.08	10.07	7.02	11.90	15.30
FATE	**8.56**	**8.77**	**6.64**	**11.03**	**10.65**
(c) Boundary threshold: 4 pixels					
WAFT	12.29	12.26	7.75	13.38	15.75
WAFT-H/4 + attention	10.02	9.99	7.34	12.73	18.87
FATE	**8.40**	**8.63**	**6.85**	**11.78**	**12.76**

C Supplied-motion reconstruction protocol

This diagnostic measures how much pixel motion each model preserves when flow is pooled into tokens and returned to pixels, independent of its token-flow accuracy. Table 3(c) reports the results.

Substituting pooled ground-truth flow. We first predict each model's weights from the input images and keep them fixed; neither model is retrained. For FATE, P_q pools ground-truth flow into tokens and P_x broadcasts it back through the same assignments A. This is the ground-truth round trip used by reconstruction supervision (Eq. (7)) and illustrated in Fig. 4. For WAFT-H/4 with local attention, we average ground-truth flow within each fixed cell and upsample it with the predicted convex weights. Pooling and evaluation use valid ground-truth pixels. We report reconstruction EPE, averaged within each image and then over images, rather than the robust training penalty $\mathcal{D}$ in Eq. (7).

What the reconstruction measures. Lower reconstruction EPE means that less pixel motion is lost in the round trip. The comparison evaluates pooling and the return to pixels together; it does not isolate their effects or decompose the end-to-end prediction error.

D Accuracy and computation

Computation is measured per 540×960 image pair with model-specific padding. Counts include feature extraction, token matching features, all refinement steps and the final per-pixel flow output.

Fewer tokens make wider refinement and local attention affordable. In Table 7, suffixes give recurrent-state and refinement widths: WAFT-64 is released WAFT, and FATE-128 is our main model. At WAFT's width, refinement over FATE's tokens costs 293.8 versus 425.8 GMACs. Widening WAFT to FATE-128's width, profiled without training, raises its refinement cost to 943.9 GMACs.

Table 7: **Fewer tokens make wider refinement and local attention affordable.** Resolution denotes the token lattice; suffixes give recurrent-state and refinement widths; Steps counts refinement steps. Costs are GMACs per 540×960 image pair; errors are EPE except KITTI Fl. Bold marks the lowest refinement cost, total cost and error in each column. †: profiled only, not trained.

			Cost (GMACs)			Things		Sintel		KITTI	
System	Resolution	Steps	Attention	Refinement	Total	Clean	Final	Clean	Final	EPE	Fl (%)
WAFT-64 [8]	$H/2$	5	–	425.8	732.4	3.54	3.44	1.28	2.56	3.49	12.9
WAFT-128 + attn.†	$H/2$	5	57.0	943.9	1313.2	–	–	–	–	–	–
FATE-64	$H/4$	5	20.6	**293.8**	**636.3**	2.55	2.48	**1.03**	2.24	3.52	12.5
FATE-128	$H/4$	4	20.6	363.1	706.1	2.44	2.43	**1.03**	2.15	3.18	12.2
FATE-128 (main)	$H/4$	5	20.6	453.9	796.8	**2.39**	**2.38**	**1.03**	**2.10**	**3.11**	**11.9**

Fewer refinement steps retain a favorable accuracy–computation trade-off. Table 8 evaluates FATE and WAFT, both trained with five refinement steps, with one to five steps at inference and no retraining. At the same number of steps, FATE has lower error than WAFT on every reported result but uses more computation. Three-step FATE improves on five-step WAFT on every reported result except KITTI Fl, at 16% lower computation (615.3 versus 732.4 GMACs).

Table 8: **FATE has lower error than WAFT at every number of refinement steps.** Both are trained with five steps and evaluated with one to five without retraining. FATE-64 is separately trained with 64-dimensional recurrent-state and refinement features. Errors are EPE except KITTI Fl.

		Things		Sintel		KITTI		
System	Steps	Clean	Final	Clean	Final	EPE	Fl (%)	GMACs
WAFT	1	6.70	6.44	2.20	3.50	8.47	37.1	391.8
FATE	1	5.01	4.84	1.54	2.67	8.30	36.9	433.8
WAFT	2	4.27	4.15	1.49	2.88	4.67	18.5	477.0
FATE	2	2.91	2.84	1.10	2.29	4.09	17.1	524.5
WAFT	3	3.78	3.66	1.34	2.70	3.81	14.0	562.1
FATE	3	2.55	2.49	1.06	2.22	3.38	13.2	615.3
WAFT	4	3.61	3.51	1.29	2.56	3.56	13.1	647.3
FATE	4	2.44	2.43	1.03	2.15	3.18	12.2	706.1
WAFT	5	3.54	3.44	1.28	2.56	3.49	12.9	732.4
FATE	5	2.39	2.38	1.03	2.10	3.11	11.9	796.8
FATE-64	5	2.55	2.48	1.03	2.24	3.52	12.5	636.3

E Target-specific fine-tuning

The zero-shot advantage does not carry over to the current fine-tuned models. Table 9 compares FATE, fine-tuned with WAFT's recipe and submitted to the benchmark servers, with WAFT's published DINOv3 results [8] on the hidden test sets of Sintel, KITTI 2015, and Spring. FATE matches WAFT's reported Sintel Final EPE and trails it on Sintel Clean, both KITTI Fl variants, and all four Spring metrics. Extending the transfer gains to target-specific fine-tuning remains open.

Benchmark metrics. Sintel reports EPE. KITTI reports Fl over all labeled pixels (Fl-all) and non-occluded pixels (Fl-noc). Spring provides four ground-truth values per image pixel and measures error against the closest one. Its 1px metric is the percentage of pixels with endpoint error above one pixel, and WAUC summarizes accuracy across error thresholds.

Table 9: **After fine-tuning, FATE matches WAFT on Sintel Final but trails it elsewhere.** Benchmark test results for FATE and DINOv3 WAFT. Lower is better except WAUC (↑).

	Sintel		KITTI 2015		Spring			
System	Clean	Final	Fl-all (%)	Fl-noc (%)	1px (%)	EPE	Fl (%)	WAUC ↑
WAFT	0.94	2.02	3.56	2.13	3.18	0.33	1.25	95.05
FATE	1.06	2.02	3.67	2.35	3.43	0.35	1.30	94.86

F Assignment parameterization and training objectives

We specify the assignment scores and training objectives summarized in Section 3. The operators and flow fields retain the same pixel/token subscripts. Matrix entries select an individual pixel or token and flow component; for example, $[F_q]_{q,c}$ is component c of token q's flow. Both transformations in the loss signatures are determined by the shared assignments A.

Gaussian assignment scores. For pixel x, the anchor of its lattice cell defines the local origin. Its candidate anchors $q \in \mathcal{N}(x)$ have offsets $\delta_{xq} \in \{-1, 0, 1\}^2$. The five predicted parameters are a mean offset $\mu_x \in \mathbb{R}^2$ and three entries of a lower-triangular factor L_x with positive diagonal, which defines the inverse covariance $G_x = L_x L_x^\top$. The scores and normalized assignments are

$$a_{xq} = -\tfrac{1}{2}(\delta_{xq} - \mu_x)^\top G_x(\delta_{xq} - \mu_x), \qquad A_{xq} = \frac{\exp a_{xq}}{\sum_{r\in\mathcal{N}(x)} \exp a_{xr}}. \tag{10}$$

Entries outside $\mathcal{N}(x)$ are zero. Since a_{xq} is the log-density of a Gaussian with mean μ_x and covariance G_x^{-1} up to a constant that cancels in the softmax, A_{xq} is that density at δ_{xq}, normalized over $\mathcal{N}(x)$. The parameterization couples each pixel's candidate scores through their relative spatial positions; it does not constrain a token's collected support to a Gaussian shape.

Probabilistic token-flow predictions. The probabilistic flow field $\mathcal{F}_q^{(t)}$ comprises a distribution for each flow component of each token. We write its density as $p_{q,c}^{(t)}(f)$. Following SEA-RAFT [20], it is a two-component Laplace mixture centered on the corresponding flow estimate:

$$p_{q,c}^{(t)}(f) = \sum_{k=1}^{2} \pi_{q,k}^{(t)} \frac{1}{2b_{q,k}^{(t)}} \exp\left(-\frac{|f - [F_q^{(t)}]_{q,c}|}{b_{q,k}^{(t)}}\right). \tag{11}$$

Here $c \in \{1, 2\}$ indexes the flow components, and $\pi_{q,k}^{(t)}$ are softmax-normalized mixture weights. Both mixture components share the flow center. The implementation bounds the first log-scale to $[0, 10]$ and fixes the second to zero; scales and mixture weights are shared across the two flow components. These mixture weights are distinct from pixel-to-token assignments. Image inputs and model parameters are suppressed in the notation.

Valid labels and pseudo-ground truth. Let V denote the valid training pixels. Write $P_q^V(A)$ for weighted pooling restricted to valid labels. The pseudo-ground-truth token flow is $F_q^{\mathrm{gt}} = P_q^V(A)F_x^{\mathrm{gt}}$, and the validity weight $w_q = [P_q(A)\mathbf{1}_V]_q$ is the fraction of a token's support with valid labels. The pseudo-ground truth and validity weights are recomputed from the current assignments and held fixed for token supervision. Feature pooling and matching remain differentiable through the assignments.

Mixtures of neighboring flow predictions. The pixelwise mixture $\mathrm{Mix}_{P_x}[\mathcal{F}_q^{(t)}]$ in Eq. (6) has component density $p_{x,c}^{(t)}$. Since $[P_x]_{xq} = A_{xq}$, its weights come from the pixel assignments. For the assignment objective, let $\mathcal{S}(x)$ denote the candidate tokens included in the training mixture. Their assignment weights are renormalized within this set:

$$p_{x,c}^{(t)}(f) = \sum_{q\in\mathcal{S}(x)} \frac{A_{xq}}{\sum_{r\in\mathcal{S}(x)} A_{xr}} p_{q,c}^{(t)}(f). \tag{12}$$

Token-flow and uncertainty predictions are held fixed for this objective; the assignments and their shared features learn through the mixture weights. The end-to-end assignment supervision variant

in Table 4 removes this restriction while keeping the pseudo-ground-truth token flow fixed. Mixing is performed separately for the two flow components, rather than forming a likelihood around an averaged flow vector. Dense inference still uses all nine assignment weights in $F_x = P_x F_q$.

Loss reductions and joint training. The NLL operator in Eqs. (5) and (6) takes a predicted distribution first and a target second. It averages over flow components and aggregates over tokens or valid pixels. With pseudo-ground truth and validity weights fixed for token supervision, and token predictions fixed for assignment supervision, the two objectives are

$$\mathcal{L}_{\text{token}}^{(t)} = -\frac{1}{2\sum_q w_q} \sum_q w_q \sum_{c=1}^{2} \log p_{q,c}^{(t)}\big([F_q^{\text{gt}}]_{q,c}\big)\,, \tag{13}$$

$$\mathcal{L}_{\text{assign}}^{(t)} = -\frac{1}{2|V|} \sum_{x\in V} \sum_{c=1}^{2} \log p_{x,c}^{(t)}\big([F_x^{\text{gt}}]_{x,c}\big)\,. \tag{14}$$

For motion preservation, we recompute $\widetilde{F}_x^{\text{gt}} = P_x(A)P_q^V(A)F_x^{\text{gt}}$ with both transformations differentiable. With $\rho(z) = \sqrt{\|z\|_2^2 + 10^{-6}} - 10^{-3}$, the reconstruction objective, specifying $\mathcal{D}$ in Eq. (7), is

$$\mathcal{L}_{\text{recon}} = \frac{1}{|V|} \sum_{x\in V} \rho\Big([\widetilde{F}_x^{\text{gt}}]_{x,:} - [F_x^{\text{gt}}]_{x,:}\Big)\,. \tag{15}$$

This branch does not reuse the fixed pseudo-ground-truth token flow. Training numerators and denominators pool entries across the minibatch; these training reductions are distinct from benchmark evaluation conventions.

The first two objectives supervise every refinement step, while reconstruction is added once:

$$\mathcal{L} = \sum_{t=1}^{T} \gamma^{T-t} \left(\mathcal{L}_{\text{token}}^{(t)} + \mathcal{L}_{\text{assign}}^{(t)}\right) + \lambda_{\text{recon}}\mathcal{L}_{\text{recon}}. \tag{16}$$

We use $T = 5$, $\gamma = 0.85$, and $\lambda_{\text{recon}} = 1$. All objectives are optimized jointly. Holding the stated predictions fixed within one objective separates its direct gradient paths; it does not freeze the shared features or introduce alternating optimization.

G Code and model availability

The training code and the main checkpoint are available at github.com/chenglinh/FATE.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's central claim: learning adaptive correspondence tokens, whose shared pixel-to-token associations pool features for matching and combine token motions into dense flow, improves zero-shot optical flow. The method, ablations, zero-shot results, and benchmark comparisons are aligned with this scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix E discusses the limitation that FATE's zero-shot advantage does not consistently transfer to target-specific fine-tuning. With WAFT's fine-tuning recipe, FATE matches WAFT on Sintel Final but trails it on the other reported benchmark metrics; improving target-specific fine-tuning remains an open direction.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not present theoretical theorems or formal proofs. The mathematical content defines the model, assignment parameterization, reconstruction, and training objective.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes FATE's architecture, training objectives, experimental setup, and evaluation protocols. The training code and the checkpoint of our main model are available at `https://github.com/chenglinh/FATE`; see Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments use public datasets. The training code and the checkpoint of our main model are available at `https://github.com/chenglinh/FATE`; see Appendix G.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper describes the datasets, feature backbones, ablation configurations, metrics, and evaluation protocols. The remaining training settings, such as optimizer parameters, learning-rate schedules, batch sizes, and augmentations, are specified in the released training code; see Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The paper reports deterministic benchmark metrics and controlled ablations but does not report error bars or statistical significance tests, due to the high computational cost of training multiple optical-flow models for each setting.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper reports per-pair computation in GMACs (Appendix D) but not the hardware, memory, runtime, and total-compute information required by this question.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. It uses public optical-flow datasets and does not involve private data, human subjects, or data collection from users.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper focuses on optical-flow methodology and benchmark evaluation; it does not include a dedicated discussion of broader societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped datasets, or models with obvious dual-use risks requiring special safeguards. The work is an optical-flow estimation method evaluated on standard public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the datasets, pretrained backbones, and baseline methods used in the experiments. FATE code is released under the BSD 3-Clause license; pretrained DINOv3 weights and other third-party assets retain their respective licenses and terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new assets are the training code and the checkpoint of our main model, which are available with usage instructions at `https://github.com/chenglinh/FATE`; see Appendix G.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research and therefore does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the core method, experiments, or scientific contribution. Any use for writing, editing, or formatting does not affect the methodology or results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.