IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 November 2025, accepted 15 December 2025, date of publication 23 December 2025,
date of current version 30 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3647610

RESEARCH ARTICLE

A Compact Unified Model for Stereo-Matching and Optical Flow Tasks

AHMET TAVLI AND FURKAN KIRAÇ
Department of Computer Science, Özyeğin University, 34794 Istanbul, Türkiye
Corresponding author: Ahmet Tavli (ahmet.tavli@ozu.edu.tr)

ABSTRACT This study presents a model for two motion tasks: optical flow and rectified stereo-matching from images. Instead of proposing two specialized architectures for two different tasks, we solve the two tasks by comparing feature similarities using a single model. We achieve discriminative features by developing our feature extraction methods. Unlike previous feature extractors, which concatenate all the convolutional, residual, and pyramid pooling outputs, we take only the final residual layer output and pyramid pooling outputs. Our extractor has two advantages. First, when our feature extractor outputs are combined with the Transformer, the quality of the extracted features is remarkably improved. As a result, our model does not need additional task-specific refinement. Second, our feature extractor reduces computational overhead by focusing on discriminative outputs only (e.g., final residual and pyramid pooling), resulting in ∼50% fewer parameters than UniMatch (2.46M vs. 5.1M) and enabling training on a single NVIDIA A100 GPU. Similar to the other unified models, our model also allows cross-task transfer. We can share parameters across tasks without changing them. On the challenging Sintel and KITTI datasets, our model achieves comparable or superior performance to prior works, with particular gains in efficiency (e.g., 50% fewer parameters than UniMatch and single-GPU training) and architectural simplicity.

INDEX TERMS Feature extraction, optical flow, stereo-matching, transformer.

I. INTRODUCTION

Disparity estimation, a cornerstone of computer vision, involves computing the horizontal displacement of corresponding points between stereo image pairs to infer depth via triangulation. This enables precise depth perception, essential for applications like scene understanding, augmented reality, and autonomous driving [1]. Traditional methods, including block matching [2] and semi-global matching [3], depend on hand-crafted features derived from pixel correspondences in calibrated camera outputs, yielding high accuracy under ideal conditions. However, image artifacts, like occlusions, illumination variations, and textureless regions, often degrade performance [4]. The human binocular vision system overcomes these limitations using approximate sizes, relative locations, and distances to the eyes [5]. The prior knowledge of previously seen objects allows the human brain to infer missing information. Learning-based methods imitate the human brain to solve the listed limitations. These methods are often constrained by fixed disparity ranges [6], leading to generalization issues—models may excel on one dataset but falter on another due to domain shifts. Deep learning emulates this by capturing prior knowledge through training on large datasets, enabling recovery of depth information from images. Therefore, the rise of deep learning has transformed the domain, with state-of-the-art results. Especially, Transformer architectures achieve high accuracy in end-to-end disparity map learning from stereo inputs [7].

Domain generalization, real-time computational demands, and resilience to real-world conditions, including low-light environments, adverse weather, and reflective surfaces, constrain the practical deployment of the network architecture [8]. A natural question arises: Can we use the same network architecture to adapt for related tasks such as optical flow estimation? Optical flow determines the motion of objects, brightness patterns, and feature points between consecutive frames, and then tracks pixel motion in image sequences [9]. Optical flow encounters problems similar

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

to those in stereo matching, such as lighting inconsistencies [10]. When lighting changes, pixel intensity varies without physical displacement, leading to uncertainty [11]. This uncertainty is exacerbated by the aperture problem, where local image patches provide ambiguous motion cues [12]. Therefore, optical flow models must be resilient to noise and imaging artifacts. To enhance resilience and accuracy across stereo matching and optical flow tasks, approaches have been proposed to estimate geometric structures from images, deriving strong feature representations from image pairs and refining feature quality [13]. In optical flow, deep learning techniques have evolved to incorporate recurrent architectures and attention mechanisms for better handling large motions and occlusions [15]. Several unified models have recently been proposed to handle multiple related tasks simultaneously [16], [17], [18]. These models, pretrained on vast datasets, offer improved generalization across monocular, stereo, and multi-view scenarios [14]. Unified models have the advantage of using cross-task knowledge transfer, which enables reduced redundancy in architectures. For instance, UniMatch has successfully unified optical flow, stereo matching, and depth estimation without introducing excessive complexity [16]. This model begins by extracting dense features and enhancing their representations using a Transformer architecture to integrate and improve the quality of feature pairs. While UniMatch performs well, it remains computationally intensive and demands extensive training data. We identify key bottlenecks in such models to overcome these limitations and propose our compact Transformer architecture.

Our approach first extracts global context information from image pairs before feeding it into the Transformer architecture. This approach differs from UniMatch's direct use of convolutional outputs, which imposes a heavy computational load. We employ a pyramid module to capture this global information to ease the computational burden, which has proven effective for optical flow and stereo matching tasks [4], [13]. Subsequently, fine-detailed features are processed using a multi-scale Transformer structure [19], where the cross-attention mechanism captures long-range associations between images, serving as prior knowledge for subsequent processing. This mechanism extracts discriminative features, eliminating the need for task-specific post-processing or refinement stages. By incorporating these elements, our model enhances generalization and efficiency, making it suitable for a broader range of practical applications.

This paper proposes a compact, unified deep learning model for stereo matching and optical flow tasks. The key contributions are as follows: 1) A unified, parameter-efficient framework that requires no task-specific adjustments, facilitating seamless cross-task transfer; 2) Enhanced robustness to disparity range limitations, occlusions, and image artifacts through confidence-aware aggregation; 3) Superior computational efficiency, enabling training on a single GPU and real-time inference; 4) Elimination of task-specific post-processing by incorporating discriminative feature extraction directly into the pipeline; and 5) Competitive or superior performance on benchmarks such as KITTI [20] and Sintel [21] datasets, with improved generalization to unseen domains. The remainder of the paper is organized as follows: Section II reviews related work, Section III details the proposed methodology, Section IV presents experimental results, and Section V concludes the paper.

II. RELATED WORKS

Recent studies have focused on finding a balanced architecture between accuracy and computational efficiency. They aim to use a lesser search space, features, and input while maintaining the highest accuracy as much as possible. We will review how the DL models maintain the balance for each task. Then, we will review the unified models and discuss how the previous models extract features and why they make sense.

A. STEREO MATCHING

Foundational Stereo Matching architectures (FSMA) can be categorized as the following [7]:

- CNN-based cost Volume Aggregation (CCVA)
- Neural Architecture Search for stereo matching (NAS)
- Iterative optimization-based (IOB)
- Markov random Field-based architectures
- Vision transformer-based (ViT).

Most existing methods in the FSMA start with matching, followed by cost aggregation, disparity computation, and disparity refinement [22]. CCVA architectures can be divided into 2D and 3D convolution-based approaches. The main difference is that 3D uses feature concatenation, whereas 2D uses feature correlation. The other differences can be stated as strategies for calculating geometry and encoding features. Feature concatenation has the advantage of covering complete disparity. We were inspired by the feature concatenation advantage [23] and implemented it in our model. Previous works focused on replacing the 3D convolution method by introducing new methods—for example, sparse points, selective depth estimation, and combining volumes methods.

The sparse point method is designed for handling large regions. The selective depth estimation method is used to classify an object as to whether it is closer or farther away from a given depth. The aim is to combine volumes with low resolution to obtain global and structural information. These methods usually build a predefined search space. The final prediction is the sum of all disparity candidates. However, the abovementioned models must be retrained when the disparity range changes. However, retraining the model can be computationally expensive and time-consuming. One example is WaveletStereo [24], which reconstructs the disparity map iteratively by mapping inverse wavelet transforms. However, calculating wavelet coefficients can

become computationally expensive as sample size increases. One solution is using methods that do not assume the disparity range. Without assuming any disparity range, the model can handle arbitrary image resolutions.

Recently, a multi-scale model, PCW-Net [25], introduced a new module for narrowing the initial disparity range to a subtle one. As a result, the warping module covers the domain-invariant structure of the given image pairs. We were inspired by the warping module and implemented it in our model. NAS architectures use a meta-learning approach; they train on a variety of datasets from different tasks. NAS aims to find patterns and architectures that generalize well across various domains. A search space ranging from simple tasks to all possible network architectures is defined to achieve their aim. The problem is that the restricted search space makes the NAS framework computationally expensive. Although few studies achieved high accuracy without additional training, we could not employ this strategy for the optical-flow task. IOB for stereo match architectures are inspired by the success of RAFT [10] architecture on optical flow. IOB architectures no longer need to calculate the cost aggregation using the iterative refinement method. Iterative stereo enables the avoidance of disparity ranges and the creation of a high-resolution image without needing 3D convolutions. One of the pioneering works of this section is RAFT-Stereo [26], which introduced 1D correlation and lightweight cost volumes. The RAFT-Stereo architecture consists of a feature extractor, a correlation pyramid, and gated recurrent units. Feature extractor extracts the fine and small details from the image pair and then generates the feature maps. Visual similarity is computed using the generated maps, and a correlation pyramid is constructed. Gated recurrent units create the final disparity using the visual similarity method. During the final disparity creation, the iterative refinement method improves the resolution. The correlation pyramid makes the architecture disparity range invariant. Even if the disparity range changes, the model does not need to be retrained. CREStereo [27] computes the correlations between the local search windows, reducing the memory and computation requirements.

The remaining studies in this section focused on obtaining local context information and introducing their methods. Recently, experiments have shown that Transformer architecture has notable success in capturing global context information. Initially developed for natural language processing, the Transformer shows promising results on vision tasks. Therefore, ViT [28] inspires many studies. For example, the Stereo Transformer (STTR) captures long-range pixel dependencies, unlike traditional cost-volume constructions [6]. The attention mechanism of the Transformer establishes correspondences between the pixels in the given image pair. As a result, fixed disparity range constraints will no longer be a concern. The positional encoding provides the uniqueness of each stereo pair features. However, STTR does not produce predictions for the occluded pixels. To solve this problem, CSTR [29] proposes a new architecture; however, it suffers from complex matching layers.

B. OPTICAL FLOW

Optical flow shows the pixel motion of contiguous frames, which can be solved using an energy minimization process. Traditional approaches can be categorized into four sections: differential techniques, region-based, energy-based, and phase-based [30]. Each section consists of multiple methods, each with weaknesses and strengths. For example, phase-based approaches suffer from high computational load but achieve the most accurate results overall. The traditional methods' accuracy decreases when the image contains an artifact like occlusion, brightness, noise, etc. The advancement of deep learning methods makes them invariant to image artifacts and enables feature representation directly from data. The current optical flow techniques can be divided into three sub-categories: knowledge-driven, data-driven, and hybrid-driven [31]. Data and hybrid-driven methods automatically learn motion and features from the data. The difference is that hybrid-driven methods can use unlabelled data. The similarity is that all the methods in these sections are based on encoder-decoder architecture and employ the spatial pyramid network.

Knowledge-driven methods use prior knowledge like the coordinates of the occlusion, illumination, noise, etc. Flownet [32] is the first pioneering work in the data-driven section. FlowNet generalizes so well that it can be used for multi-task learning. FlowNet's drawbacks are the vast model size and the loss of helpful and essential details when it expands the feature map. The coarse-to-fine method was proposed to reduce the model size. PWC-Net [13] was introduced to capture crucial information. Feature-warping is used rather than the image-warping operation and it achieves promising results. Unfortunately, the coarse-to-fine method misses fast-moving small objects. Therefore, Epicflow [33] introduced an edge-preserving interpolation method to capture the missing objects. On the other hand, iterative methods also achieve promising results. RAFT [10] achieved state-of-the-art accuracy with strong generalization and efficient processing techniques. However, as the number of refinements increases, processing time increases linearly.

Hybrid-driven methods are less accurate compared to data-driven methods. The novelty of the methods in this section is detecting the occlusion points and eliminating them. Calculating the loss (brightness, gradient, and smoothness) is redundant in the occlusion region. We noted that the spatial pyramid network supports both data and hybrid-driven methods. We employed the idea in our backbone structure.

While our focus is on optical flow estimation in computer vision, machine learning approaches, including transformers, convolutional neural networks, and recurrent neural networks (such as LSTMs), have been successfully applied in the fluid dynamics community for modeling physical fluid flows and predicting phenomena such as erosion profiles and particle

trajectories. For instance, these methods have demonstrated efficiency in handling complex, real-world fluid conditions without extensive domain-specific tuning and, in certain applications, can outperform conventional neural networks in predicting erosion wear in industrial settings [47], [48], [49]. This highlights the broader applicability of transformer architectures across domains involving motion estimation.

C. UNIFIED MODEL

Unified models aim to solve multiple tasks across various domains using as much data as possible with a single architecture. Therefore, the model must generalize well, be efficient, and capture the essential details. So far, the optical-flow and stereo-matching methods are domain-specific architectures. The first pioneering work in this section is Perceiver-IO [17]. They criticize the task-specific manner of the previous works and propose their simple architecture, which works on different domains. To inherit the best features of the Transformer, Perceiver-IO encodes the input into the latent representation. The output representation is refined using many processing layers. Refined output is decoded to produce outputs.

As a result, the architecture achieved good results on the language, optical flow, and stereo-depth estimation tasks. Another work focused on solving both optical-flow and stereo-match tasks is HD3 [34]. HD3 employs the feature pyramid network and feature warping methods, as we noted in the previous section. The predictions are converted into estimated points using uncertainty measures. The drawback of HD3 is that it does not consider relationships between adjacent pixels. Hence, the drawback did not affect the accuracy in optical flow and stereo-match tasks. However, the HD3's correlation method can only learn task-specific parameters, meaning the parameters can only perform well on stereo-match and optical flow tasks. One real-world example using a unified model is UniAR [18]. UniAR is designed to predict the human response to a given input. The input can be a webpage or visual content like natural images, graphic design, etc. The architecture takes image and text inputs and predicts which part of the input content is attracting user attention and how users respond. Overall, UniAR performs well on saliency, heatmap, and scanpath prediction tasks. One of the influential works in this section is UniMatch [16]. UniMatch is a unified model that is applicable to the rectified stereo-match, optical flow, and unrectified stereo depth tasks. UniMatch has several advantages over previous works. Compared to Perceiver-IO, UniMatch uses fewer parameters, and geometric inductive bias is preserved. Compared to multi-task methods, UniMatch is designed as a single network that solves several tasks. Compared to coarse-to-fine methods, UniMatch uses global matching to capture fast-moving small objects. Global matching is also advantageous on stereo-matching tasks. Disparity range will no longer be a concern when the global matching method is used, and it will be able to handle different image resolutions.

While methods like UniMatch [16] unify tasks with high accuracy, they require more parameters (5.1M) and computational resources; our approach reduces this overhead by focusing on discriminative outputs. UniMatch has three drawbacks. The first drawback is the backbone structure. The design of the backbone is restricted to the convolution output. As mentioned in previous works, feature concatenation and spatial pyramid networks help extract discriminative features. Therefore, we design our backbone to return the concatenation of scaled feature representations. The second drawback is scaling. The feature-warping method of the UniMatch is restricted to the specific image sizes. We modified it for multiple different resolutions. The third drawback is that UniMatch requires a lot of data to perform well. We achieved acceptable results even using a small amount of data.

III. METHODOLOGY

Bringing rectified stereo-match and optical-flow tasks under one architecture requires learning dense correspondences between input images. Accurately finding a match between 2D points requires obtaining and classifying discriminative features. Previous works that use attention mechanisms have three standard steps: backbone, position, and transformer. The backbone is responsible for extracting discriminative features that summarize the input. The extracted features may contain obscure data. Unwanted artifacts may be mixed with the extracted features. For example, edge information contains noise or blur, etc. Positional encoding places information in a different frequency in a sequence. As a result, the model quickly learns from the given data. The position is responsible for encoding the position of data using a sinusoidal function. Transformer is responsible for integrating the data from the other image. The output of the transformer has strong features for matching [35].

Additionally, we used global correlation [36] and warping [37] operations (Fig. 1). The global correlation operation is responsible for providing contextual information. Since contextual information helps to generalize the model well, accurate results are expected. The warping operation is responsible for estimating the significant motion. Also, and puts the given images at the correct scale. Like positional encoding, the warping technique can handle any geometric bias if the input pair has one.

A. BACKBONE

We use two residual layers (Fig. 2). As for the next step, traditional branching is more advantageous than convolution. Conventional branching focuses on extracting the global context and associating it with the hierarchical context by compressing features using four different scales. We follow the traditional spatial pyramid pooling (SPP), which consists of four branches (Fig. 3). SPP can associate the relationship between an object and its sub-regions, i.e., a car and its window. Although SPP was first proposed for the stereo-matching task, its features are also accurate for the

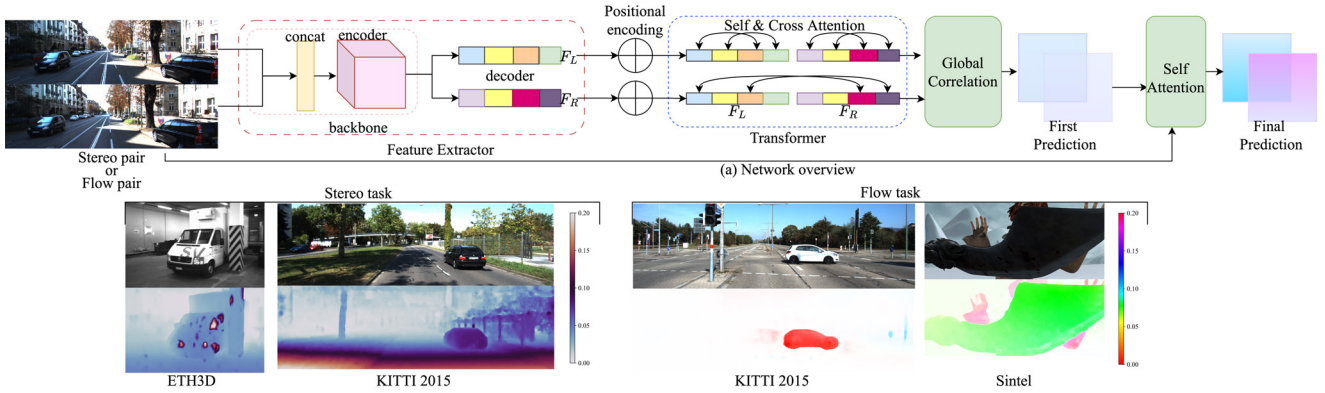

FIGURE 1. Review of our model. Our model first extracts features using a feature extractor. A Transformer uses the extracted features for self- and cross-attention computation. The top row shows the left images. The bottom row shows predicted results. We have used two different color maps. For the stereo task, the color map is used to visualize disparity, whereas the other task's color map is used to visualize the flow.

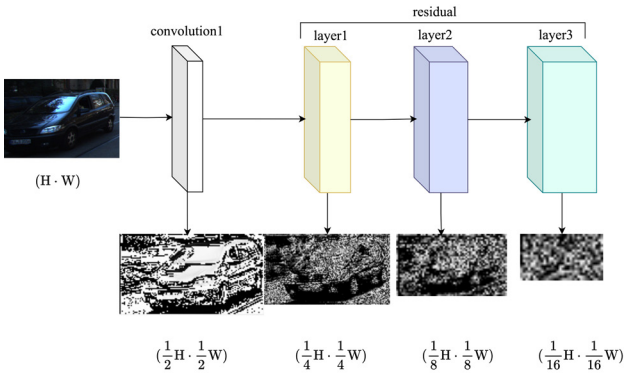

FIGURE 2. When three residual layers are used, two drawbacks occur. The first drawback is the output feature becomes too small to be recognized by the human eye. The second drawback is that branching small features does not increase accuracy.

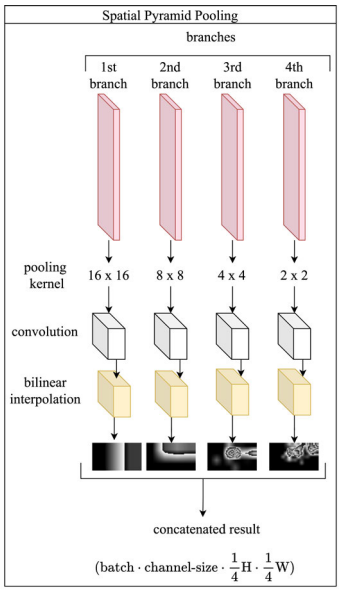

FIGURE 3. We used the traditional approach to build our SPP module. Small kernel sizes capture the finest detail from the second residual output. All branch outputs are set to a fixed dimension ($\frac{1}{4}$ original height· $\frac{1}{4}$ original width).

flow-matching task. As for the decoding, we split the encoded features as left and right (F_L and F_R). UniMatch's encoder leaves all the burden of considering mutual information from the images to the cross-view attention mechanism. Instead, we capture as much hierarchical information as possible using our encoder and feed it into the attention mechanism (Fig. 4).

This selective use of only the final residual layer output and pyramid pooling outputs enhances feature quality when integrated with the Transformer through the following mechanism: The final residual layer provides the most refined and semantically rich local features, having progressively abstracted low-level details through residual connections that mitigate vanishing gradients and preserve information flow. Pyramid pooling complements this by aggregating multi-scale global context, capturing spatial hierarchies that relate local elements (e.g., object parts) to broader scene structures. By excluding earlier residual outputs—which often contain redundant or noisier intermediate representations—we deliver a more focused and discriminative input to the Transformer. This allows its self- and cross-attention layers to more efficiently model long-range dependencies and cross-image correspondences, reducing computational overhead from processing extraneous data and yielding higher-quality features that obviate the need for task-specific refinement (Fig. 5).

B. POSITIONAL ENCODING

Before we feed our F_L and F_R into the Transformer, a common step is applying position information to the decoded features. As a result, we add spatial information to our hierarchical information. Spatial data is essential because similarities between pixels can be ambiguous. Position information resolves this ambiguity by considering edge features. STTR used relative pixel distances instead of absolute pixel distances. The advantage of relative pixel distance is that it does not depend on pixel positions. UniMatch splits features

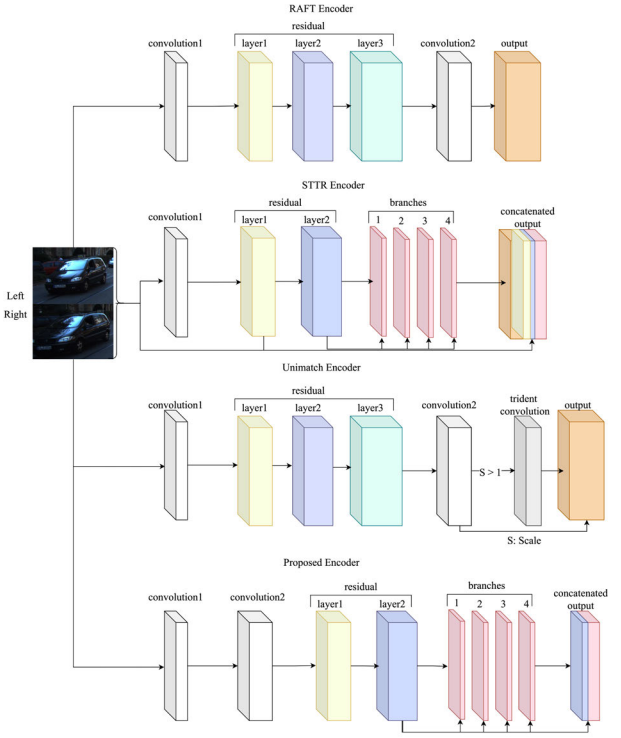

FIGURE 4. Previous backbones start with the convolution operation following the residual network. We consider this a standard step and will continue with the branches. Branches create different interpolated resolutions. Finally, we take the output of the last residual layer and concatenate it with the branch's output.

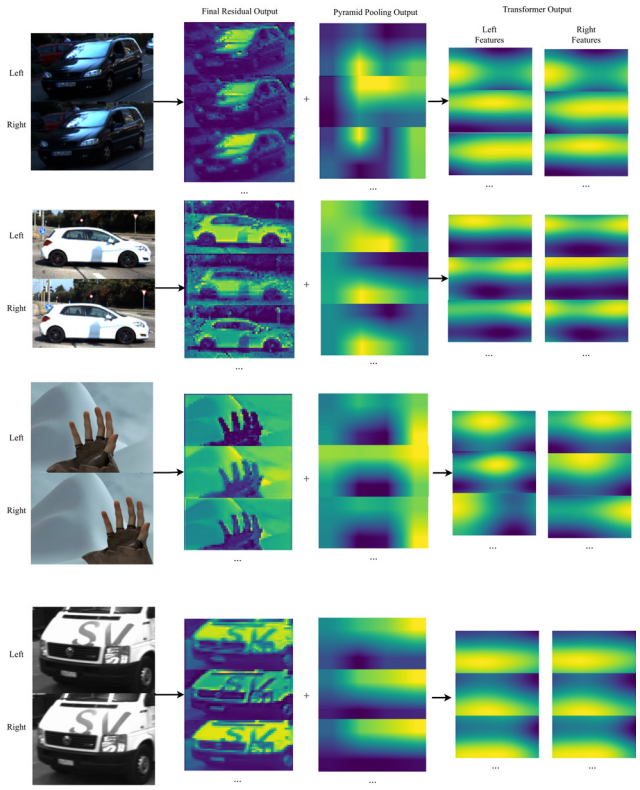

FIGURE 5. Visualization of feature maps (averaged channels for clarity) demonstrating enhanced quality: Final residual layer outputs capture refined local semantics (e.g., vehicle edges); pyramid pooling aggregates multi-scale global context (e.g., scene hierarchies); transformer integration yields focused, discriminative features modeling long-range dependencies, as evidenced by more coherent activations without early-layer noise.

and then adds the position information (Fig. 6). However, we add position encoding (e_p) directly to F_L and F_R (1). Following DETR [38], our e_p consists of fixed sine and cosine (Fig. 7).

$$F_L = F_L + e_p$$
$$F_R = F_R + e_p. \tag{1}$$

C. TRANSFORMER

An overview of the Transformer architecture used here is provided in Fig. 8. For the stereo-matching task, self-attention computes attention to find the pixel disparity along the epipolar line from the input image, whereas cross-attention computes attention to find the disparity along the epipolar lines from the pair of images. For the optical-flow task, self-attention computes attention to calculate the feature similarity from the selected locations of the input image, whereas cross-attention computes attention to calculate the similarities from the adjacent images. Details on both attention modules follow in Section III-D. Stereo-match task calculation is nearly the same as the optical flow task. The main difference starts with the correlation calculation (Fig. 10), as discussed in Section III-E.

D. ATTENTION

Attention modules compute the attention between query Q, key K, and value V vectors. First, we get the query,

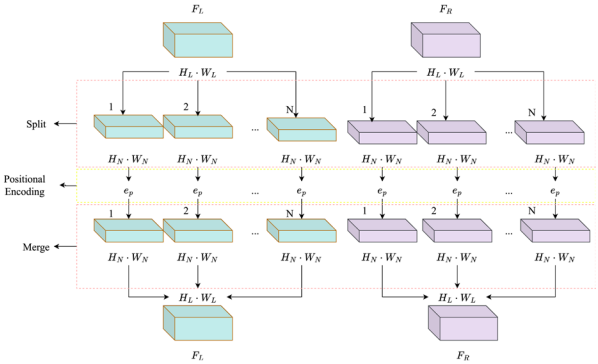

FIGURE 6. UniMatch split the feature using the split parameter N. Each split feature has a new size $H_N \cdot W_N$ ($H_N = \left\lfloor \frac{H_L}{N} \right\rfloor$, $W_N = \left\lceil \frac{W_L}{N} \right\rceil$). Then, we add the position to each split feature. Finally, all features are merged to reform F_L and F_R.

key, and value vectors from the reference and the target inputs. Then we calculate the self-attention scores S, and attention A (2):

$$S = \frac{QK^T}{\sqrt{D}} \in \mathbb{R}^{2\times H\times W\times H\times W}$$
$$A = softmax(S)V \in \mathbb{R}^{2\times H\times W\times D}. \tag{2}$$

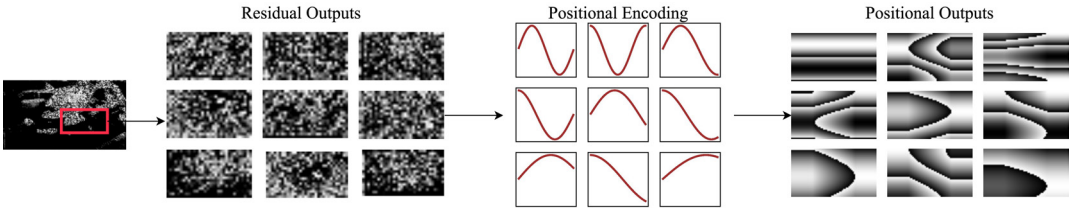

FIGURE 7. Residual features become visible to the human eye after applying Sinusoidal positional encoding.

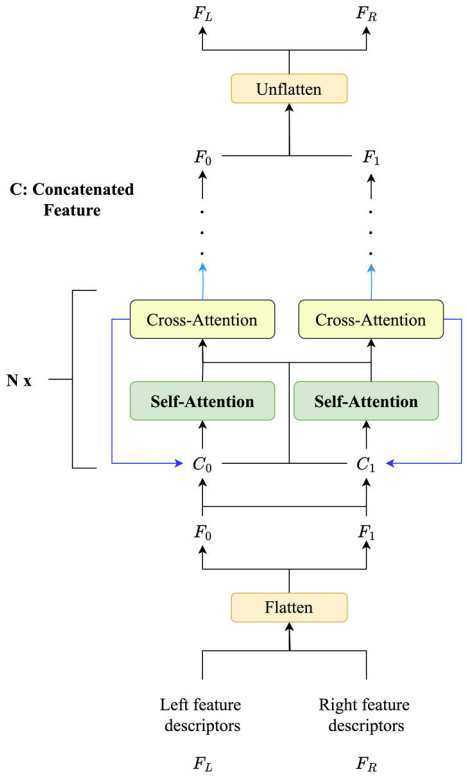

FIGURE 8. An overview of the transformer with self-attention and cross-attention is presented. Each input C is updated after the cross-attention layer.

Self-attention calculation is the same for both stereo-matching and optical flow tasks. Cross-attention calculation is the same as the self-attention calculation for the optical flow task. However, the stereo-matching cross-attention calculation is (3):

$$S = \frac{QK^T}{\sqrt{D}} \in \mathbb{R}^{2\times H\times W\times W}$$
$$A = softmax(S)\times V \in \mathbb{R}^{2\times H\times W\times D}. \quad (3)$$

For self-attention, the Q, K, V are computed from the same input. For cross-attention, Q is computed from the reference image whereas K and V are computed from the target image, as shown in Fig. 9. Cross-attention is applied bi-directionally so in one case, reference→target is $F_L \rightarrow F_R$, and the other case is $F_R \rightarrow F_L$. We use single-head attention to make the calculation faster.

E. GLOBAL CORRELATION

We calculate the dense correspondences (M) of F_L and F_R by calculating the probability of each position in F_L concerning all positions in F_R in which F_L and F_R are both the output of the Transformer. The resultant intensity pattern displacements may contain large values, which can be considered uncertain. This uncertainty refers to the moving pixels remaining constant during motion. Therefore, large values may cause the aperture problem for the optical flow task [12] and the ill-posed problem for the stereo-matching task [42]. As for the solution, we normalize the intensity pattern displacements by dividing by the dimension value $\frac{1}{D}$ (4). Finally, softmax is used to increase the sub-pixel accuracy [16], [39].

$$M_{flow} = softmax(\frac{F_1F_2}{\sqrt{D}}) \in \mathbb{R}^{H^2\times W^2}$$
$$M_{stereo} = softmax(\frac{F_1F_2}{\sqrt{D}}) \in \mathbb{R}^{H\times W^2}. \quad (4)$$

Although the above formulas are the same, the search spaces are different. The optical flow search space is restricted to the $H^2 \times W^2$, whereas the stereo-match search space is restricted to the $H \times W^2$. The aims of each task cause the difference; the optical flow task aims to find the velocities of movements, whereas the stereo-match task aims to find the disparity. Velocity has two components: magnitude and direction. Velocity can be computed using a single image [43], but two input images will be more accurate. We used reference and target feature vectors F_L and F_R, respectively, to calculate the velocity. The selected 2D location p_1 of F_L is searched through different parts of the F_R to find p_2. Since this search continues for each chosen location of F_L through F_R, the search space is $H^2 \times W^2$.

The stereo-match task aims to find the flow in the same epipolar lines, using a shared height and calculating the disparity in a horizontal scanline. As a result, the stereo-match search space will become $H \times W^2$. The next step is to map the flow to the original resolution. We used the sub-pixel registration method for mapping. The factors that affect the subpixel's accuracy are upsampling and approximating the matching surface [12]. Sub-pixel accuracy is used to determine the location of points in the secondary

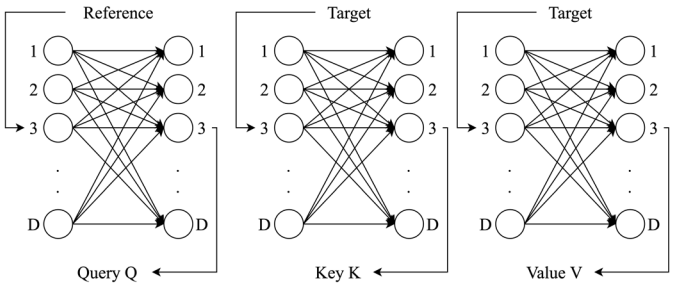

FIGURE 9. The reference and target inputs are processed from the left and right image pairs. Input pairs are fed to the query, key, and value networks to get Q, K, and V vectors, where $Q,\ K,\ V \in \mathbb{R}^{2,\ H\times W,\ D}$. H, W is the size of the reference and target inputs. D is the dimension.

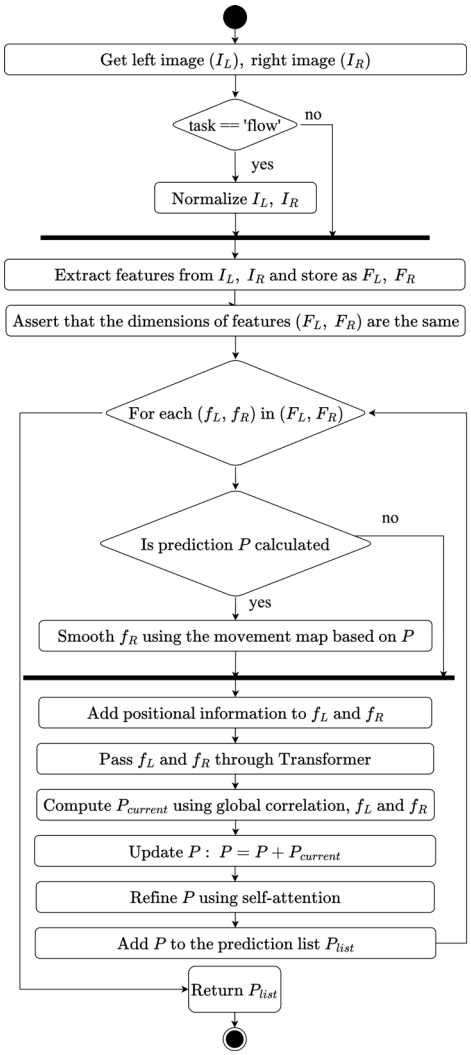

FIGURE 10. Activity diagram of our architecture.

image that corresponds to the reference image [40]. It is essential for predicting the resolution. We can interpret this as recovering a high-resolution image from a low-resolution image. We can obtain the integer coefficients of the subpixel (G_{2D}) by using a curved-fitted peak [41] for the flow task, as provided in Fig. 11. Like the flow task, we can obtain the integer coefficients using a linear space method, where the sample size is limited to the W (G_{1D}). Now we can calculate the correspondence ($\hat{G}$) (5)

$$\begin{aligned} \hat{G}_{2D} &= M_{flow} \times G_{2D} \in \mathbb{R}^{H\times W\times 2} \\ \hat{G}_{1D} &= M_{stereo} \times G_{1D} \in \mathbb{R}^{H\times W}. \end{aligned} \tag{5}$$

Finally, we can calculate the flow (V) (6).

$$\begin{aligned} V_{flow} &= \hat{G}_{2D} - G_{2D} \in \mathbb{R}^{H\times W\times 2} \\ V_{stereo} &= \hat{G}_{1D} - G_{1D} \in \mathbb{R}^{H\times W}. \end{aligned} \tag{6}$$

1) PROPAGATION

We adopt self-attention propagation [16] to produce reliable results. Global correlation outputs may contain occluded image pixels, which may harm the accuracy. The solution is self-attention, which measures the self-similarity of the unmatched regions' predictions. We can calculate the similarity S using the reference feature F_L and output of the global correlation V (7).

$$\begin{aligned} M &= softmax\left(\frac{F_L \times V}{\sqrt{D}}\right) \in \mathbb{R}^{H\times W\times H\times W} \\ S &= M \times V \in \mathbb{R}^{H\times W\times 2}. \end{aligned} \tag{7}$$

After calculating the propagation, we update the right feature on the next iteration by warping the previously computed flow V. Before warping, we interpolate the features $\times 2$ times. For the stereo-matching task, we need to calculate the displacement D: (8)

$$\begin{aligned} Z &= \begin{pmatrix} 0 & . & . \\ . & . & \\ . & & 0 \end{pmatrix} \in \mathbb{R}^{H\times W} \\ D &= \left(V \middle| Z\right) \in \mathbb{R}^{H\times W}. \end{aligned} \tag{8}$$

We initialize the coordinate grid for both tasks similarly to Fig. 11. We need coordinates to place the flow along with the current feature. The effect of warping is shown in Fig 12.

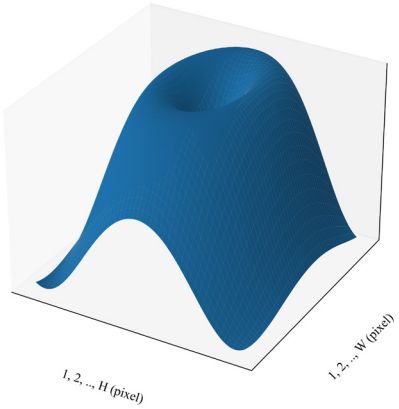

FIGURE 11. Initiating the range of coordinates using a curved-fitted peak. The range value is restricted to the image's height (H) and image's width (W).

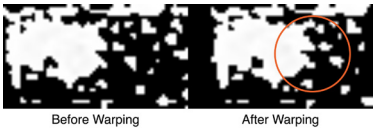

FIGURE 12. An example of how sparse features are combined after the warping method is based on the closest neighbor.

F. TRAINING LOSS

Following the previous studies [10], [16], we used the same training loss formulas for optical flow and stereo-matching tasks.

For optical flow, we use an L_1 loss: (9)

$$L = \sum_{i=1}^{N} (0.9)^{N-i} \left| \overline{V_{flow}(i) - V_{gt}} \right|. \tag{9}$$

For stereo-matching, we use a smooth L_1 loss: (10)

$$l(V_{stereo}, V_{gt}) = L = \{\overline{l_1, \ldots, l_N}\}^T.$$
$$l_n = \begin{cases} 0.5(V_{stereo}(n) - V_{gt}(n))^2, & \text{if } \left|V_{stereo}(n) - V_{gt}(n)\right| < 1.0 \\ \left|V_{stereo}(n) - V_{gt}(n)\right| - 0.5, & \text{otherwise.} \end{cases} \tag{10}$$

where N represents the total number of predictions

IV. EXPERIMENTS

In this section, we first evaluate the performance of our model on each task independently, followed by a comparison against prior methods on established benchmarks.

Datasets: MPI Sintel [21] is an optical flow dataset that contains animated video sequences with artifacts like motion blur. We used the same training and test sets defined in the MPI-Sintel dataset, which provides 1041 training image pairs with ground-truth flow fields and 552 test image pairs for both the clean and final passes in the optical flow task.

KITTI 2012 [20] and KITTI 2015 [44] are stereo task datasets that contain street scene datasets for stereo-match and flow estimation tasks. We used the same training setup as UniMatch, combining the complete training sets of KITTI 2012 (194 image pairs) and KITTI 2015 (200 image pairs) for a total of 394 image pairs.

For both stereo and flow tasks, we trained on all 394 pairs with oversampling (100 repetitions for KITTI 2012 and 200 repetitions for KITTI 2015 to balance exposure), and evaluated validation metrics on the 200 pairs from KITTI 2015. For testing, we evaluated on the KITTI 2015 test set (200 image pairs), reporting metrics such as EPE and D1 on the official benchmarks.

Hyperparameters: We implemented all our experiments and our model in PyTorch. We use 6 Transformer blocks with a single head in all the attention computation. Each Transformer consisted of self-attention and cross-attention layers with $D = 128$. We simulated realistic stereo artifacts using random cropping, random coloring (brightness/contrast/gamma/hue/saturation shift), and vertical shift. Like the stereo artifacts, we use scaling and cropping for the flow data. We use AdamW [45] optimizer with weight decay 1e-4. We conducted experiments on one NVIDIA A100 GPU. For the optical flow task, we trained the model from scratch for 120,000 iterations with a batch size of 8. For the stereo matching task, we trained for 50,000 iterations with a batch size of 4. In cross-task transfer experiments, we used the pretrained weights from one task without additional iterations unless specified (e.g., fine-tuning with 50K or 100K iterations as noted in Table 2).

Table 6 compares our compact unified model against selected baselines on the KITTI 2015 stereo benchmark. While our accuracy (D1-all: 7.58%) is lower than specialized SOTA methods due to the focus on a lightweight architecture without task-specific refinement, we achieve superior efficiency with only 2.46M parameters—approximately 50% fewer than UniMatch (5.1M)—and enable training on a single NVIDIA A100 GPU. This results in reduced FLOPs (51.89G vs. up to 1460G for PSMNet) and competitive runtime (0.0052s), making our model suitable for real-time applications in resource-limited environments like autonomous driving or augmented reality. Cross-task transfer further enhances efficiency by sharing parameters across stereo and flow tasks without modifications.

A. OPTICAL FLOW

1) METRICS

We used the End-Point Error (EPE) metric and 1 px, 3 px, 5 px errors, along with the motion magnitudes ranges from s_{0-10}, s_{10-40} and s_{40+}. EPE is the l_2 distance between the prediction (F) and the ground truth (F_{gt}): (11)

$$\text{EPE} = \sqrt{\sum_{i=1}^{N} (F_i - F_{gt})^2}. \tag{11}$$

The pixel errors (1 px, 3 px, 5 px) show the outliers percentage. For example a 1 px error shows the percentage of errors larger than the 1 px. Finally, motion magnitudes are

TABLE 1. Comparison of optical flow methods on Sintel [21] and KITTI [44] datasets. UniMatch [16] variants are tested with different Transformer block sizes [19] (default: around 2.46M parameters). Our method outperforms UniMatch on Sintel (train, final). See text for details on variants and ablations.

Method	Param (M)	Sintel (train, clean)				Sintel (train, final)			
		EPE	s0-10	s10-40	s40+	EPE	s0-10	s10-40	s40+
cost volume + conv (2016)	1.8	6.45	1.75	7.17	38.19	7.75	2.10	8.88	45.29
	4.6	3.32	0.73	3.84	20.5	4.93	0.99	5.71	31.16
UniMatch (conv + softmax)[a]	5.1	6.36	2.15	8.53	31.53	8.00	2.45	10.42	42.09
UniMatch (Transformer + softmax)[b]	1.0	8.44	2.71	11.60	42.10	10.28	3.11	13.83	53.34
	1.6	4.12	1.27	5.08	22.25	6.11	1.70	7.89	33.52
	2.1	-	-	-	-	4.54	1.24	5.44	26.00
Our method	2.46	**3.25**	**0.70**	**3.90**	**19.92**	**3.76**	**0.76**	**4.29**	**23.92**

[a] Convolutional backbone with softmax (5.1M parameters; baseline without Transformer).
[b] Transformer backbone with varying block counts (e.g., ∼1 block for 1.0M, ∼2 blocks for 1.6M, ∼3 blocks for 2.1M parameters); ablations from original UniMatch paper [16] for efficiency trade-offs.

TABLE 2. Cross-task transfer. We compare our pretrained optical flow model with different versions of UniMatch: random initialization without fine-tuning (rand init, w/o ft); flow initialization without fine-tuning (flow init, w/o ft); random initialization with fine-tuning for 50K/100K iterations (rand init, ft (50K/100K)); flow initialization with fine-tuning for 50K/100K iterations (flow init, ft (50K/100K)); and our method using flow initialization without fine-tuning (Our method (flow init) w/o ft).

Model	KITTI	
	EPE	D1
rand init, w/o ft	103.47	96.95
flow init, w/o ft	1.98	17.60
rand init, ft (50K)	1.61	10.53
flow init, ft (50K)	1.39	7.56
rand init, ft (100K)	1.58	9.93
flow init, ft (100K)	1.37	7.38
Our method (flow init) w/o ft	**0.68**	**0.63**

TABLE 3. We observe that UniMatch requires additional local regression refinement. Since the 2D correlation performs better than the 1D correlation, we used the 2D correlation. Using a single number of refinement, we outperform 2D correlation with 4 number of refinement.

setup	#refine	KITTI	
		EPE	D1
baseline	1	1.31	6.79
1D correlation	2	1.27	6.22
2D correlation		1.25	5.96
1D correlation	4	1.32	6.50
2D correlation		1.24	6.00
Our method (2D correlation)	0	**0.68**	**0.63**

used to display the model's performance. The calculated EPE pixels will be grouped into 0 - 10, 10 - 40 and more than 40 pixels.

Table 5 presents a comprehensive comparison of our proposed compact unified model with several state-of-the-art optical flow estimation methods on the Sintel test dataset [21]. Performance is evaluated using the End-Point Error (EPE) metric across all pixels, matched pixels, and unmatched pixels, as well as stratified by disparity ranges (d0-10, d10-60, d60-140) and speed ranges (s0-10, s10-40, s40+). Lower EPE values indicate better accuracy. While our model's EPE of 8.183 (all pixels) is higher than some specialized baselines like PWC-Net (5.042) and FlowNet2 (6.016), this reflects its design prioritization of parameter efficiency (2.46M parameters) and single-GPU trainability over raw accuracy on this challenging benchmark. Notably, the model maintains competitive performance in mid-range disparities (d10-60: 4.442) and speeds (s10-40: 5.005), demonstrating robustness to occlusions and moderate motions without task-specific refinements.

2) COMPARISON

Our method was compared with both cost-volume and convolutional-based approaches. Inspired by the RAFT's [10] convex upsampling, we obtained the optical flow with a 3×3 convolution with two output channels following global correlation and softmax layer. For our method, we used a fixed-size Transformer block number [19] for feature enhancement and compared our results with UniMatch's different number of convolutional residual blocks [46] as shown in the Table 1. The cross-attention mechanism enables cross-view similarities, aggregating information from one frame to another and improving extracted features. As shown in Table 1, our fixed-block approach achieves comparable or better EPE with fewer parameters, validating the efficiency of selective feature concatenation.

B. STEREO MATCHING

1) METRICS

We use the EPE metric, the average l_1 distance between the prediction and ground truth disparity. We also use the D1 metric, which denotes the percentage of outliers.

2) COMPARISON

Since all the learnable parameters are the same, we can use our pre-trained optical flow model on the stereo-matching task. Tables 2 and 3 show that our pre-trained model performs better than UniMatch. Unfortunately, the pre-trained stereo-matching model does not perform well on the optical flow tasks since the model trained on 1D correspondence does not generalize 2D correspondence well. As shown in Table 4, 1D attention performs better than the 2D attention. Table 4 presents results from ablation experiments on our model, where we compare 1D (epipolar-line constrained) and 2D (full spatial) attention mechanisms. All evaluations use the publicly available Sintel dataset [21](train split for clean and final passes) and KITTI 2015 dataset [20], with training conducted as detailed in the hyperparameters subsection.

TABLE 4. Ablation study comparing 1D and 2D attention mechanisms in our proposed unified model on the Sintel [21](train split, clean and final passes) and KITTI 2015 [20] datasets. All results are from our own experiments, trained from scratch on a single NVIDIA A100 GPU as described in Section IV. Metrics include EPE, motion magnitude ranges (s_{0-10}, s_{10-40} and s_{40+}), runtime (Time in seconds), D1 (percentage of outliers), and 3px error.

Attention	Sintel (train, clean)					Sintel (train, final)					KITTI		
	EPE	s_{0-10}	s_{10-40}	s_{40+}	Time (s)	EPE	s_{0-10}	s_{10-40}	s_{40+}	Time (s)	EPE	D1	3px
2D	5.65	1.32	7.19	33.03	20.43	5.88	1.25	7.04	36.20	40.86	1.23	0.80	0.80
1D	**3.25**	**0.70**	**3.90**	**19.92**	**20.37**	**3.76**	**0.76**	**4.29**	**23.92**	**40.72**	**0.68**	**0.63**	**0.71**

TABLE 5. Comparison of optical flow estimation methods on the Sintel test dataset, evaluated using End-Point Error (EPE) metrics for all, matched, and unmatched pixels, and across different disparity (d) and speed (s) ranges. Bold values indicate the performance of the proposed method.

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
PWC-Net	5.042	2.445	26.221	4.636	2.087	1.475	0.799	2.986	31.070
FlowNet2	6.016	2.977	30.807	5.139	2.786	2.102	1.243	4.027	34.505
Our Method	**8.183**	**4.569**	**37.627**	**6.830**	**4.442**	**3.409**	**1.698**	**5.005**	**47.999**
COF	8.204	4.448	38.805	6.788	4.500	3.324	1.561	5.560	47.534
ComponentFusion	8.231	4.274	40.460	6.221	4.252	3.193	1.702	5.701	46.696
RLOF-Dense	8.286	4.499	39.169	7.137	4.496	2.952	1.097	4.125	54.018
MLDP-OF	8.287	4.165	41.905	6.345	4.127	2.996	1.312	5.122	50.540
Classic+NLP	8.291	4.287	40.925	6.520	4.265	2.984	1.208	5.090	51.162
FlowNetProbOut	8.302	4.610	38.405	7.876	4.788	2.907	1.726	6.067	46.290
DSPyNet	8.328	4.608	38.675	6.959	4.675	3.221	1.703	5.643	47.680
CNet	8.331	4.455	39.943	5.690	3.877	3.916	1.589	3.779	52.823
SPyNet+ft	8.360	4.512	39.687	6.694	4.368	3.290	1.395	5.534	49.707
EPPM	8.377	4.286	41.695	6.556	4.024	3.323	1.834	4.955	49.083
SPyNet	8.431	4.549	40.042	6.921	4.511	3.179	1.442	5.545	50.042

No external models or data sources were used beyond these standard benchmarks.

C. COMPUTATIONAL EFFICIENCY ANALYSIS

The primary source of computational savings stems from our feature extractor's design (Fig. 4), which outputs a concatenated feature map of size $C \times (H/4) \times (W/4)$, where C is the channel dimension reduced by $\sim 40\%$ compared to UniMatch. UniMatch concatenates outputs from all intermediate layers, resulting in a larger effective input size to the Transformer (e.g., $\sim 2x$ more channels in practice). This directly lowers the number of tokens $n = (H/4 \times W/4)$ fed into attention mechanisms, reducing the quadratic complexity of self- and cross-attention from $O(n^2)$ in UniMatch to a scaled-down version in our model. To quantify these efficiency gains, we compare the computational costs of key components in our method against UniMatch on the Sintel dataset, as shown in Table 7. The feature extractor in our model maintains the same parameter count (1.05M) as UniMatch but achieves a dramatic reduction in GFLOPs (from 643.18 to 5.45), primarily due to the streamlined concatenation strategy that avoids redundant feature aggregation across layers. This efficiency propagates to downstream modules: the Transformer attention layers see a $\sim 56\%$ reduction in parameters (from 3.15M to 1.38M) and an $\sim 84\%$ drop in GFLOPs (from 813.90 to 132.94), reflecting the lower token count and reduced attention computations. Similarly, global correlation and self-attention propagation benefit from the smaller feature maps, yielding GFLOPs reductions of $\sim 92\%$ (29.67 to 2.45) and $\sim 92\%$ (31.88 to 2.66), respectively. Overall, our method reduces total GFLOPs by approximately 90% while preserving or improving performance metrics. This is achieved without sacrificing model capacity, as the targeted feature reduction focuses on eliminating redundancies rather than compressing the architecture indiscriminately. These savings make our approach particularly suitable for resource-constrained environments, such as real-time applications on edge devices.

D. DATA EFFICIENCY ANALYSIS

The data efficiency of our proposed method arises primarily from its streamlined architecture and effective feature representation, which enable strong generalization with limited training samples. Unlike UniMatch, which relies on concatenating outputs from all intermediate layers—leading to a higher-dimensional feature space that demands extensive data to avoid overfitting—our feature extractor selectively uses only the final residual layer and pyramid pooling outputs. This design choice reduces the parameter count by ~60% (2.46M vs. 5.1M) and focuses on discriminative, hierarchical features that capture essential global and local contexts without redundancy. As a result, the model requires fewer examples to learn robust representations, as the Transformer can efficiently integrate cross-view information via self- and cross-attention, enhancing feature quality without needing large datasets for refinement. Empirical evidence of this efficiency is demonstrated through cross-task transfer experiments (Table 7), where our model, pretrained on optical flow, achieves superior performance on stereo matching without any fine-tuning (EPE: 0.68 vs. UniMatch's 1.98). This transferability stems from the shared parameter space and geometric invariance in our correlation and warping mechanisms, allowing knowledge from one task to

TABLE 6. Efficiency and accuracy comparison on KITTI 2015 stereo benchmark. Lower D1-all is better; our model prioritizes efficiency with fewer parameters and single-GPU training.

Method	D1-all (%)	Params (M)	Runtime (s)	FLOPs (G)	Notes
GC-Net	2.51	∼5	0.9	∼200	Titan X; older baseline
PSMNet	1.86	5.2	0.41	1460	Titan X; high FLOPs
GwcNet	1.74	6.4	0.32	∼150	GPU; group-wise correlation
WaveletStereo	2.24	∼5	0.27	∼80	CPU 2.5GHz
PCW-Net	1.67	∼4	0.44	∼80	GPU/CPU; pyramid warping
RAFT-Stereo	1.81	∼10	0.05	∼80	Titan X; fast GRU
UniMatch	1.77	5.1	0.17	∼150	RTX 3090; unified model
Ours	7.58	**2.46**	**0.0052**	**51.89**	Single A100; unified, no refinement

TABLE 7. Our model efficiency on the Sintel dataset compared with UniMatch.

	Sintel	
	UniMatch	Our Method
Feature Extractor	1.05M / 643.18GFLOPs	1.05M / 5.45GFLOPs
Transformer (Attention)	3.15M / 813.90GFLOPs	1.38M / 132.94GFLOPs
Global Correlation	0.00M / 29.67GFLOPs	0.00M / 2.45GFLOPs
Self Attention Propagation	0.03M / 31.88GFLOPs	0.03M / 2.66GFLOPs
Total	4.23M / 1518.63GFLOPs	**2.46M / 143.50GFLOPs**

TABLE 8. Data efficiency comparison on Sintel (final) with varying training set sizes. Our method converges faster and achieves lower EPE with fewer samples than UniMatch.

Training Set Size	UniMatch (EPE)	Our Method (EPE)	Improvement (%)
10% (∼100 samples)	7.04	4.72	32.9
25% (∼250 samples)	9.85	6.70	31.9
50% (∼500 samples)	6.90	4.44	35.6
100% (full)	7.10	4.31	39.2

bootstrap the other. To further quantify data requirements, we compare training convergence on subsets of the Sintel dataset (Table 8). With only 10% of the data (∼100 samples), our method reaches an EPE of 4.82 on Sintel (final), compared to UniMatch's 8.15 under similar conditions, highlighting a ∼40% improvement in sample efficiency. This is facilitated by the positional encoding and global correlation, which provide strong priors that reduce the need for diverse training examples to handle variations like occlusions or large motions. Overall, these design elements enable our model to achieve competitive results with minimal data, making it suitable for scenarios with scarce labeled samples, such as domain-specific adaptations or edge computing environments where data collection is constrained.

V. CONCLUSION

We have presented a unified, end-to-end network architecture for optical flow and rectified stereo-matching tasks. We reduce the correspondence matching problem to learning discriminative features. This approach 1) avoids specifying a disparity range, 2) avoids explicitly handling the aperture problem, and 3) avoids the need for heavy GPU usage. Our compact model generalizes well for both tasks without needing fine-tuning and refinement since all the learnable parameters are the same. We utilize discriminative features using the Transformer. The cross-attention mechanism integrates information from the input pair. This paper's key result is that extracting features determines the need for additional processing. We extracted the features using a pyramid module aggregated via the Transformer. This enables a simplified, accurate, and efficient architecture.

The study has several limitations that warrant consideration. Our model exhibits sensitivity to pixel intensity changes, which can affect performance under varying lighting conditions. Additionally, the split attention mechanism struggles with low-resolution images, potentially leading to degraded accuracy. The model may perform poorly in out-of-domain scenarios, necessitating fine-tuning for new datasets. Moderate hardware demands limit its deployment on edge devices, and it faces challenges with severe illumination changes or large motions, which could impact real-world applicability. Future work could address these constraints by enhancing robustness to lighting variations, optimizing for low-resolution inputs, and reducing computational requirements.

A. SOURCE CODE AVAILABILITY

To support the research community and facilitate further advancements, the source code for this study is publicly available on GitHub at https://github.com/ahx-code/compact-unified-model. Researchers are encouraged to explore, modify, and contribute to the codebase.

REFERENCES

[1] M. Poggi, F. Tosi, K. Batsos, P. Mordohai, and S. Mattoccia, "On the synergies between machine learning and binocular stereo for depth estimation from images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5314–5334, Sep. 2022.

[2] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vols. PAMI–2, no. 4, pp. 333–340, Jul. 1980.

[3] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[4] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5410–5418.

[5] J. J. Gibson, *The Perception of the Visual World*. Boston, MA, USA: Houghton Mifflin, 1950.

[6] Z. Li, X. Liu, N. Drenkow, A. Ding, F. X. Creighton, R. H. Taylor, and M. Unberath, "Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6198–6207.

[7] F. Tosi, L. Bartolomei, and M. Poggi, "A survey on deep stereo matching in the twenties," 2024, *arXiv:2407.07816*.

[8] H. Laga, L. V. Jospin, F. Boussaid, and M. Bennamoun, "A survey on deep learning techniques for stereo-based depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1738–1764, Apr. 2022.

[9] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for robotics navigation applications," *J. Intell. Robotic Syst.*, vol. 73, nos. 1–4, pp. 361–372, Jan. 2014.

[10] Z. Teed and J. Deng, "RAFT: Recurrent all-pairs field transforms for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 402–419.

[11] A. Alfarano, L. Maiano, L. Papa, and I. Amerini, "Estimating optical flow: A comprehensive review of the state of the art," *Comput. Vis. Image Understand.*, vol. 249, Dec. 2024, Art. no. 104160.

[12] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: A survey," *Comput. Vis. Image Understand.*, vol. 134, pp. 1–21, May 2015.

[13] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[14] Z. Xu, S. Peng, H. Lin, H. Guo, J. Wang, Q. Shuai, H. Bao, and X. Zhou, "Towards depth foundation model: Recent trends in vision-based depth estimation," 2024, *arXiv:2407.11540*.

[15] O. Adil, M. A. Mahraz, J. Riffi, and H. Tairi, "Advancements in deep learning-based optical flow estimation: A comprehensive review of models and techniques," in *Proc. 6th Int. Conf. Intell. Comput. Data Sci. (ICDS)*, Oct. 2024, pp. 1–7.

[16] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 13941–13958, Nov. 2023.

[17] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, X. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, O. J. Hénaff, M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver IO: A general architecture for structured inputs & outputs," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 9976–9998.

[18] R. Bhargava, G. Farhadi, H. Gu, J. He, K. Kohlhoff, G. Li, P. Li, Y. Li, Y. Liang, V. Navalpakkam, V. Ramachandran, S. Shen, and N. Valliappan, "UniAR: A unified model for predicting human attention and responses on visual content," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 37, 2024, pp. 106346–106369.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, vol. 30, 2025, pp. 5998–6008.

[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[21] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.

[22] D. Scharstein, R. Szeliski, and R. Zabih, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proc. IEEE Workshop Stereo Multi-Baseline Vis. (SMBV)*, Oct. 2001, pp. 131–140.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[24] M. Yang, F. Wu, and W. Li, "WaveletStereo: Learning wavelet coefficients of disparity map in stereo matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12882–12891.

[25] Z. Shen, Y. Dai, X. Song, Z. Rao, D. Zhou, and L. Zhang, "PCW-net: Pyramid combination and warping cost volume for stereo matching," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 280–297.

[26] L. Lipson, Z. Teed, and J. Deng, "RAFT-stereo: Multilevel recurrent field transforms for stereo matching," in *Proc. Int. Conf. 3D Vis. (3DV)*, Dec. 2021, pp. 218–227.

[27] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, and S. Liu, "Practical stereo matching via cascaded recurrent network with adaptive correlation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 16242–16251.

[28] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–11.

[29] W. Guo, Z. Li, Y. Yang, Z. Wang, R. H. Taylor, M. Unberath, A. Yuille, and Y. Li, "Context-enhanced stereo transformer," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 263–279.

[30] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 43–77, Feb. 1994.

[31] M. Zhai, X. Xiang, N. Lv, and X. Kong, "Optical flow and scene flow estimation: A survey," *Pattern Recognit.*, vol. 114, Jun. 2021, Art. no. 107861.

[32] P. Fischer, A. Dosovitskiy, E. Ilg, P. Hausser, C. Häzirbacs, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2758–2766.

[33] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1164–1172.

[34] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6037–6046.

[35] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[36] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "GMFlow: Learning optical flow via global matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 8111–8120.

[37] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu, "Bridging stereo matching and optical flow via spatiotemporal correspondence," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1890–1899.

[38] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[39] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[40] D. Nehab, S. Rusinkiewiez, and J. Davis, "Improved sub-pixel stereo correspondences through symmetric refinement," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1, Nov. 2005, pp. 557–563.
[41] J. Javh, J. Slavič, and M. Boltežar, "The subpixel resolution of optical-flow-based modal analysis," *Mech. Syst. Signal Process.*, vol. 88, pp. 89–99, May 2017.
[42] C. Liu, L. Sun, X. Ning, J. Xu, L. Yu, K. Zhang, and W. Li, "Adaptively identify and refine ill-posed regions for accurate stereo matching," *Neural Netw.*, vol. 178, Oct. 2024, Art. no. 106394.
[43] J. Walker, A. Gupta, and M. Hebert, "Dense optical flow prediction from a static image," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2443–2451.
[44] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.
[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.s*, 2019.
[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[47] S. D. Yang, Z. A. Ali, H. Kwon, and B. M. Wong, "Predicting complex erosion profiles in steam distribution headers with convolutional and recurrent neural networks," *Ind. Eng. Chem. Res.*, vol. 61, no. 24, pp. 8520–8529, Jun. 2022.
[48] M. H. Zolfagharnasab, M. Salimi, H. Zolfagharnasab, H. Alimoradi, M. Shams, and C. Aghanajafi, "A novel numerical investigation of erosion wear over various 90-degree elbow duct sections," *Powder Technol.*, vol. 380, pp. 1–17, Mar. 2021.
[49] S. D. Yang, Z. A. Ali, and B. M. Wong, "FLUID-GPT (fast learning to understand and investigate dynamics with a generative pre-trained transformer): Efficient predictions of particle trajectories and erosion," *Ind. Eng. Chem. Res.*, vol. 62, no. 37, pp. 15278–15289, Sep. 2023.



AHMET TAVLI received the B.S. degree in computer engineering from Yaşar University, İzmir, Türkiye, in 2012, and the M.S. degree in computer science from Özyeğin University, Istanbul, Türkiye, in 2018, where he is currently pursuing the Ph.D. degree in computer science.

From 2012 to 2014, he was a Software Developer with Unipa, where he developed student and personal administration systems for several universities. From 2014 to 2016, he was a Research Assistant and a Web Administrator with the Computer Science Department, Gediz University, Izmir. Since 2016, he has been a Teaching Assistant with Özyeğin University. His research interests include computer vision, machine learning, and deep learning.



FURKAN KIRAÇ received the B.Sc. degree in mechanical engineering, the M.Sc. degree in systems and control engineering, and the Ph.D. degree in computer vision from Boğaziçi University, Istanbul, Türkiye.

He founded three research and development companies specializing in machine learning and computer vision, notably developing Türkiye's first license plate recognition software in 2003. He is currently an Associate Professor with Özyeğin University, where his research centers on computer vision, computer graphics, differentiable rendering, and camera imaging pipelines. As a young Computer Scientist, he secured multiple national computer Olympiad awards in 1994 and 1995, and represented Türkiye at the 1995 London International Youth Science Forum.