EMatch: A Unified Framework for Event-based Optical Flow and Stereo Matching

Supplementary Material

6. More Details of EMatch

In this section, we provide more details about the forward process of EMatch, including the Temporal Recurrent Network (TRN), Spatial Contextual Attention (SCA), and Correspondence Matching.

6.1. Temporal Recurrent Network (TRN)

We design Temporal Recurrent Network (TRN) to iteratively aggregate asynchronous events, aligning temporal information retained by event voxels to the features at the last time step. Specially, we split event voxels into K groups chronologically, denoted as $\{V_{T_i}|i=0,...,K\}$, which are processed as shown in Fig. S1.

For each time step k, we use stacked resBlocks to extract intermediate features $C^{l=i}_{T_k}$ from V_{T_k} layer by layer, and use convGRUs to fuse the intermediate features $C^{l=i}_{T_k}$ with the historical features $F^{l=i}_{T_{k-1}}$, obtaining the current features $F^{l=i}_{T_k}$. To fully utilize historical information, we additionally introduce features $F^{l=i-1}_{T_k}$ as input when extracting intermediate features $C^{l=i}_{T_k}$. This process is represented as:

$$C^{l=i}_{T_k} = \text{resBlock}(\text{concat}(C^{l=i-1}_{T_k}, F^{l=i-1}_{T_k})), \tag{S1}$$

$$F^{l=i}_{T_k} = \text{convGRU}(F^{l=i}_{T_{k-1}}, C^{l=i}_{T_k}). \tag{S2}$$

The operation of each convGRU can be represented as:

$$\begin{aligned}
r_t &= \sigma(W_r \cdot [F_{T_{k-1}}, C_{T_k}] + b_r),\\
z_t &= \sigma(W_z \cdot [F_{T_{k-1}}, C_{T_k}] + b_z),\\
\tilde{F}_{T_k} &= \tanh(W_h \cdot [r_t * F_{T_{k-1}}, C_{T_k}] + b_h),\\
F_{T_k} &= (1 - z_t) * F_{T_{k-1}} + z_t * \tilde{F}_{T_k}.
\end{aligned} \tag{S3}$$

In addition, we construct a top-to-bottom connection pathway for feature $\{F^{l=i}_{T_k}|i=0,1,...,n\}$ between iterations to facilitate the flow of information between layers. Typically, we take the feature $F^{l=n}_{T_K}$ from the highest layer as the result of feature extraction. If multi-scale optimization is used, features $\{F^{l=i}_{T_k}|i=0,1,...,n-1\}$ from lower layers can also be utilized, and the network depth can be increased as needed. In the paper, we constructed a four-layer architecture and selected the features from the last two layers to adopt optimization at two scales.

6.2. Spatial Contextual Attention (SCA)

Previously, we have extracted reference and target features F_1, F_2 from event voxels using TRN independently.

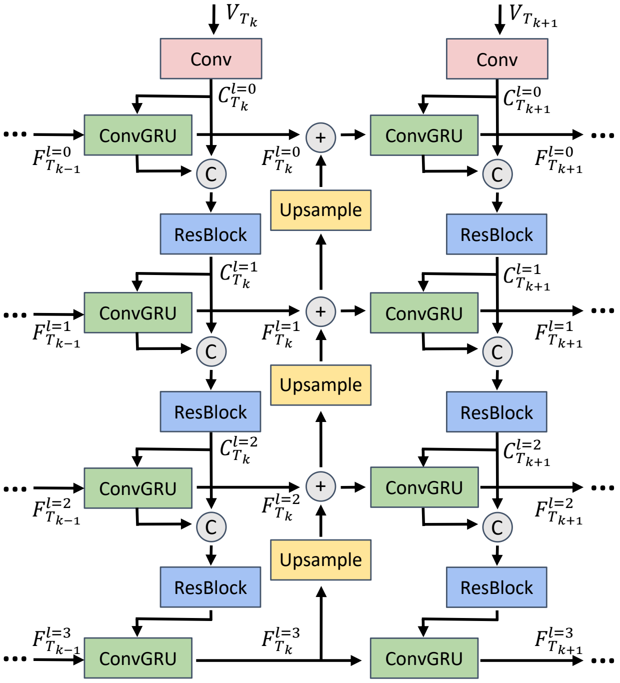


Figure S1. Operations of Temporal Recurrent Network (TRN). Event voxel is processed iteratively in a time-ordered manner to align asynchronous visual information to the target time. Finally, we obtain temporally aggregated multi-scale features $F^{l=i}_{T_K}$.

Now we further map them into a high-level representation space with Spatial Contextual Attention (SCA) as shown in Fig. S2.

Before SCA, we firstly add positional encoding to features F_1, F_2 for the supplement of missing position information during attention operations as follows:

$$\begin{aligned}
PE(pos, 2i) &= \sin(\frac{pos}{100000^{2i/d_{model}}}),\\
PE(pos, 2i+1) &= \cos(\frac{pos}{100000^{2i/d_{model}}}).
\end{aligned} \tag{S4}$$

Then, we use self-attention, cross-attention, and feed-forward networks to construct a SCA block, which is stacked into six layers with shared parameters. For each block, the features F_1 and F_2 are symmetrically processed. The formula of attention operation is represented as:

$$\text{attention}(q, k, v) = \text{softmax}(\frac{qk^T}{\sqrt{d}})v. \tag{S5}$$

First, we employ self-attention to enhance the quality of the features through global aggregation, which means the

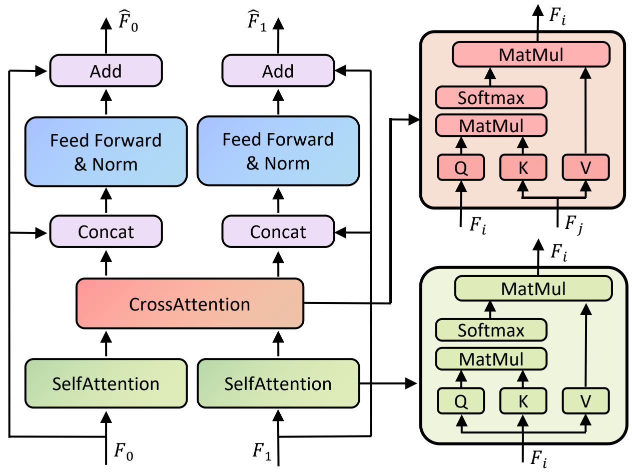


Figure S2. Operations of Spatial Contextual Attention (SCA). Using the attention mechanism to aggregate global information is beneficial for leverage spatial context to build more expressive feature representation, facilitating the fusion of optical flow and stereo matching in high-level representation space.

query, key and value come from the same feature:

$$f_i = \text{selfAttention}(q = f_i, k = f_i, v = f_i). \tag{S6}$$

Then, we use cross-attention to blend two different features together, which means the key and value come from the same feature but the query come from another feature:

$$f_i = \text{crossAttention}(q = f_i, k = f_j, v = f_j). \tag{S7}$$

Finally, a feed-forward network (FFN) achieved by MLP is adopted to generate feature residuals, which are added to the original features F_0 and F_1:

$$\hat{F}_i = F_i + \text{FFN}(\text{concat}(F_i, f_i)). \tag{S8}$$

Notably, to reduce the computational complexity of Transformer, we have adopted the shifted local window attention strategy proposed by Swin Transformer [8]. We split the features to a fixed number of local windows rather than a fixed window size, just like GMFlow [14]. Specifically, for the feature of size $H \times W$, if we set the total number of local windows as $K \times K$, the local window size will be $\frac{H}{K} \times \frac{W}{K}$. And the window partition will be shifted by $(\frac{H}{2K}, \frac{W}{2K})$ to introduce cross-window connections.

6.3. Correspondence Matching

In the following, we will provide a detailed description of how dense correspondence matching estimates optical flow and disparity. First of all, we should construct correlation volume C by taking the dot product between all pairs of feature vectors $\hat{F}_1, \hat{F}_2 \in R^{H \times W \times D}$. For optical flow and stereo matching, it can be respectively computed as follows:

$$C_{\text{flow}}(a, b, x, y) = \sum_i \hat{F}_1(a, b, i) \cdot \hat{F}_2(x, y, i), \tag{S9}$$

$$C_{\text{disparity}}(a, b, x) = \sum_i \hat{F}_1(a, b, i) \cdot \hat{F}_2(x, b, i). \tag{S10}$$

For optical flow estimation, we should find pixel-wise dense correspondences on the 2D plane. But for rectified stereo matching, we only need to find the per-pixel disparity along the horizontal scanline (i.e. 1D plane). Therefore, the correlation volume size of optical flow is $H \times W \times (H \times W)$, while the stereo matching is $H \times W \times W$.

Next, for each reference pixel, we will select the target pixel with the highest correlation (i.e. the highest feature similarity) to construct a corresponding relation. Specifically, we use softmax operation to normalize the correlation volume C to obtain matching distribution M:

$$M_{\text{flow}}(a, b, x, y) = \frac{\exp[C_{\text{flow}}(a, b, x, y)]}{\sum_{i,j} \exp[C_{\text{flow}}(a, b, i, j)]}, \tag{S11}$$

$$M_{\text{disparity}}(a, b, x) = \frac{\exp[C_{\text{disparity}}(a, b, x)]}{\sum_i \exp[C_{\text{disparity}}(a, b, i)]}. \tag{S12}$$

Then, with the matching distribution M, the corresponding relation grid G can be determined by taking a weighted average of all the candidate coordinates. For optical flow, we use 2D grid $U_{2d} \in R^{H \times W \times 2}$, which stores arranged 2D coordinates on the 2D plane. For stereo matching, we use 1D horizontal position $U_{1d} \in R^W$, which only stores arranged 1D coordinates on the scanline.

$$G_{\text{flow}}(a, b) = \sum_{i,j} M_{\text{flow}}(a, b, i, j) U_{2D}(i, j), \tag{S13}$$

$$G_{\text{disparity}}(a, b) = \sum_i M_{\text{disparity}}(a, b, i) U_{1d}(i). \tag{S14}$$

Finally, the displacement D can be obtained by computing the difference between the corresponding and initial coordinate grid. For the optical flow and stereo matching, we respectively use 2D gird $G_{2D} \in R^{H \times W \times 2}$ and 1D grid $G_{1D} \in R^{H \times W}$ as initial coordinate grid.

$$D_{\text{flow}} = G_{\text{flow}} - G_{2d} \in R^{H \times W \times 2}, \tag{S15}$$

$$D_{\text{disparity}} = G_{\text{disparity}} - G_{1d} \in R^{H \times W}. \tag{S16}$$

7. Previous Task-specific Frameworks

In this section, we summarize previous task-specific frameworks of optical flow and stereo matching to explain their incompatible architectures hindering unification, and highlight the advantages of EMatch in eliminating barriers between temporal and spatial domains.

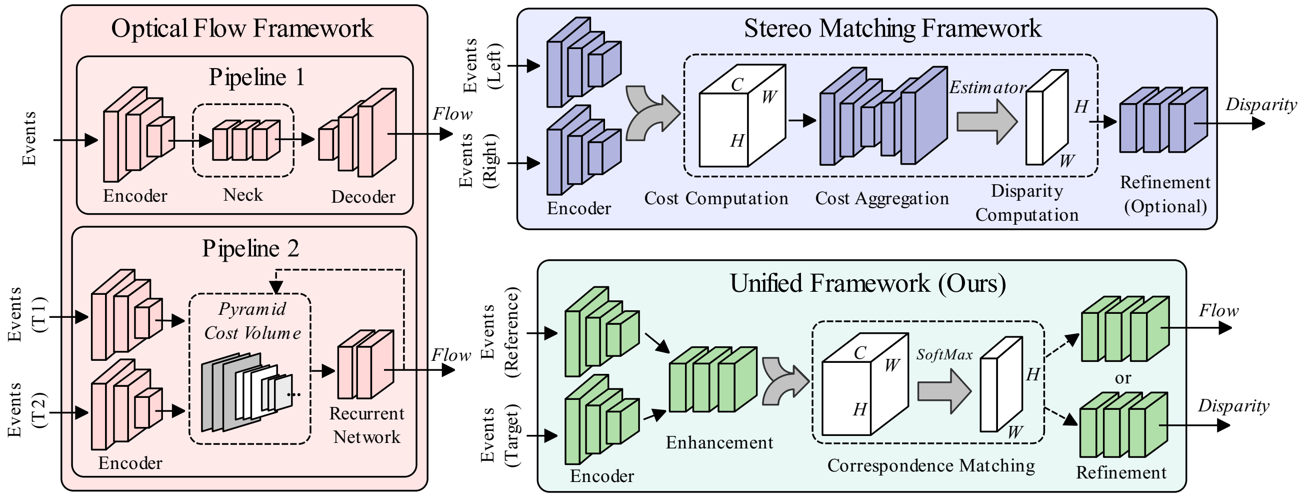


Figure S3. Comparison of our unified framework with other task-specific frameworks. Previous works designed frameworks within their respective domains, introducing many additional designs that are incompatible across tasks. Instead, our framework unifies optical flow and stereo matching (within temporal and spatial domains) into a model with a shared representation space through dense correspondence matching, which consists of feature encoder, feature enhancement, correspondence matching, and refinement.

As shown in Fig. S3, previous works for optical flow estimation and stereo matching have developed many specialized frameworks to efficiently extract task-specific features. For optical flow estimation, most frameworks derive temporal motion features either from accumulated events [3, 13, 16] or event-based cost volume [5–7, 9]. For stereo matching, many frameworks [1, 10, 12, 15] employ a classic approach to compute spatial matching costs from events.

For event-based optical flow estimation, previous works generally follow two pipelines. Firstly, EV-Flownet [16] was proposed as the first deep learning framework following Flownet [4], which extracted motion features directly from event frames. Secondly, E-RAFT [5] introduced another framework inspired by RAFT [11] to retrieve motion features iteratively from event-based cost volume. Both of them regress flow from motion features, rather than calculating flow by dense correspondence matching as us.

For event-based stereo matching, most deep-learning works follow a traditional stereo matching pipeline, consisting of feature extraction, cost computation, cost aggregation, disparity computation and refinement. By contrast, they focus more on the representation of event features. For example, DDES [12] proposed the first learning-based network with event sequence embedding, and se-cff [10] proposed another network with concentrated event stacks. Obviously, their frameworks are all built around constructing spatial matching costs to optimize the disparity results, which introduce a lot of task-specific redundant designs.

In conclusion, existing works are confined to these task-specific frameworks, resulting in a significant domain gap between optical flow and stereo matching, which greatly hinders their unification. Unlike them, we innovatively propose to utilize dense correspondence matching to unify optical flow and stereo matching within the same domain. By sharing the representation space during feature extraction, we facilitate knowledge transfer and integration between temporal domain and spatial domain, effectively achieving the unification of optical flow and stereo matching.

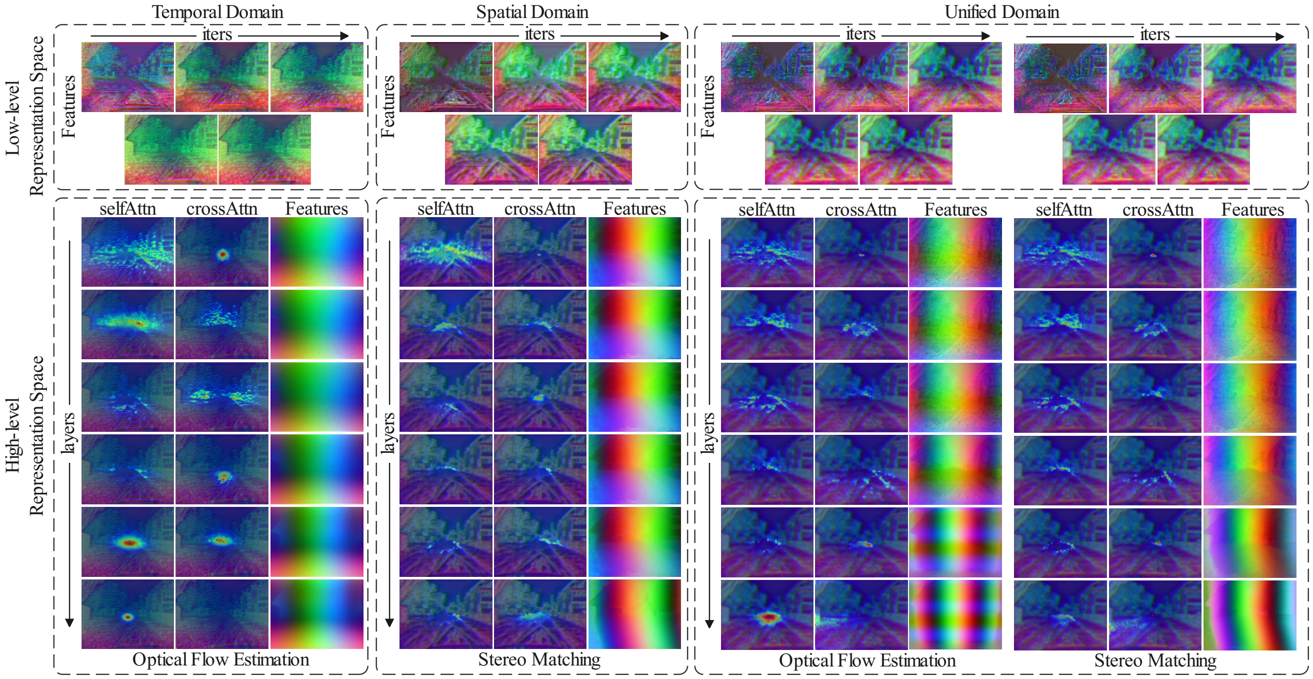
8. Supplementary Experiments

In this section, we first conduct an in-depth examination of the shared representation space for optical flow and stereo matching mentioned in the paper, by visualizing intermediate features of EMatch. Then, we summarize and analyze the complexity of existing models in terms of model size and running speed. Finally, we provide additional visualizing results to further supplement the qualitative comparisons between EMatch and other SOTA methods.

8.1. Analysis of Representation Space

EMatch can unify optical flow and stereo matching into the same domain within a shared representation space. When performing single-task training, EMatch (single) converges to domains of flow and disparity separately (i.e. temporal and spatial domains). When performing multi-task training, EMatch (unified) converges to a shared domain across flow and disparity. In Fig. S4, we visualize the intermediate features of these two models, showing how the shared representation space unifies the two tasks into a shared domain.

Firstly, we visualize the intermediate features of TRN. In the paper, we divide event voxels into $K = 5$ groups, so



(a) Single-task Training

(b) Multi-task Training

Figure S4. More visualizing features of EMatch within different domains. The figure shows the results of TRN over five iterations and the results of SCA over six layers. We apply PCA on intermediate features of TRN and SCA, and we also visualize attention maps from six stacked layers of SCA. The results show that, EMatch with multi-task training can unify optical flow and stereo matching into the same domain by leveraging a shared representation space.

Table S1. Analysis of model complexity. Our model has a similar complexity to other methods when deployed for single tasks, but achieves the highest performance with lower complexity when deployed for multiple tasks.

Task	Method	Params	FLOPs
Flow	ERAFT [5]	5.3M	251G
	TMA [7]	6.8M	344G
	IDNet [13]	2.5M	1202G
	EMatch (single)	6.7M	502G
Disparity	Se-cff [10]	5.9M	225G
	TESNet [2]	5.6M	2478G
	EMatch (single)	6.7M	502G
Flow&Disparity	TMA[7]+TESNet[2]	12.4M	2822G
	EMatch (unified)	6.7M	1004G

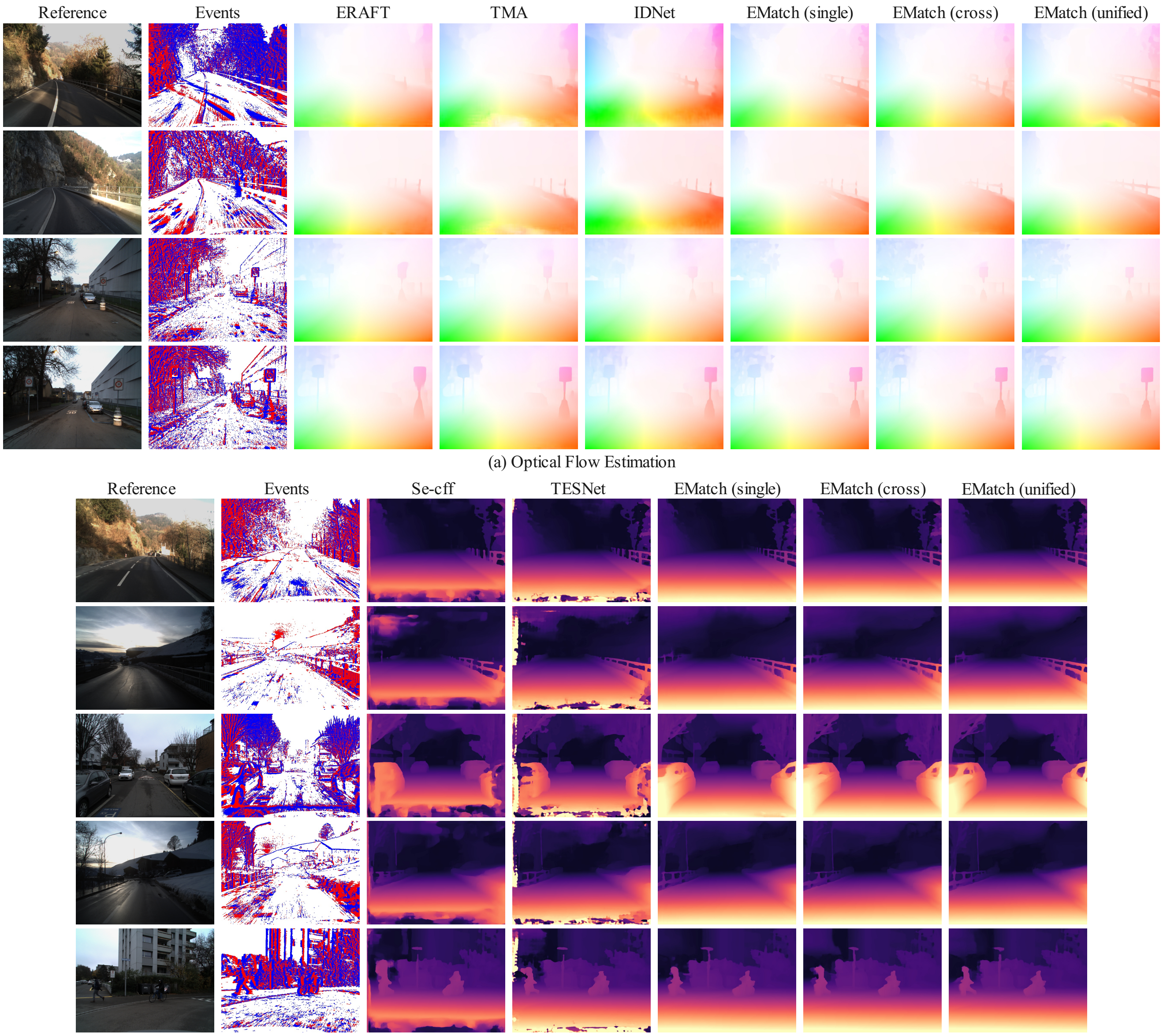
we visualize features during five iterations of TRN. The results show that TRN can gradually update the state of event features during iterations, aligning them to the target time. Through TRN, EMatch maps event voxels to a low-level representation space, where the features represent the visual state at a specific moment.

Secondly, we visualize the intermediate features of SCA. As mentioned earlier, we stack six layers of SCA blocks with shared parameters. Therefore, we visualize the features of these six layers separately. Additionally, we provide the attention maps of self-attention and cross-attention within each SCA block. Through SCA, EMatch can map features to a high-level representation space, where the features can be used for dense correspondence matching.

After multi-task training, optical flow and stereo matching can be unified within shared representation spaces. In our paper, we mainly focus on the high-level representation space, in which similar features are assigned to the same pixels while dissimilar features are assigned to different pixels. Overall, EMatch can learn priors from temporal and spatial domains within shared representation space, achieving the unification of optical flow and stereo matching.

8.2. Analysis of Model Complexity

We summarize the complexity of the existing models, including the number of parameters and computational cost, as shown in the Table S1. For every single task, EMatch (single) achieves the highest performance while having a similar complexity compared to other models, striking a balance between model size and running speed. For multiple tasks, EMatch (unified) can deploy optical flow and stereo matching with lower complexity by sharing feature representation spaces. What's more, when utilizing EMatch (unified), the representation of event-based inputs for optical flow and stereo matching are identical, which can reduce



(a) Optical Flow Estimation

(b) Stereo Matching

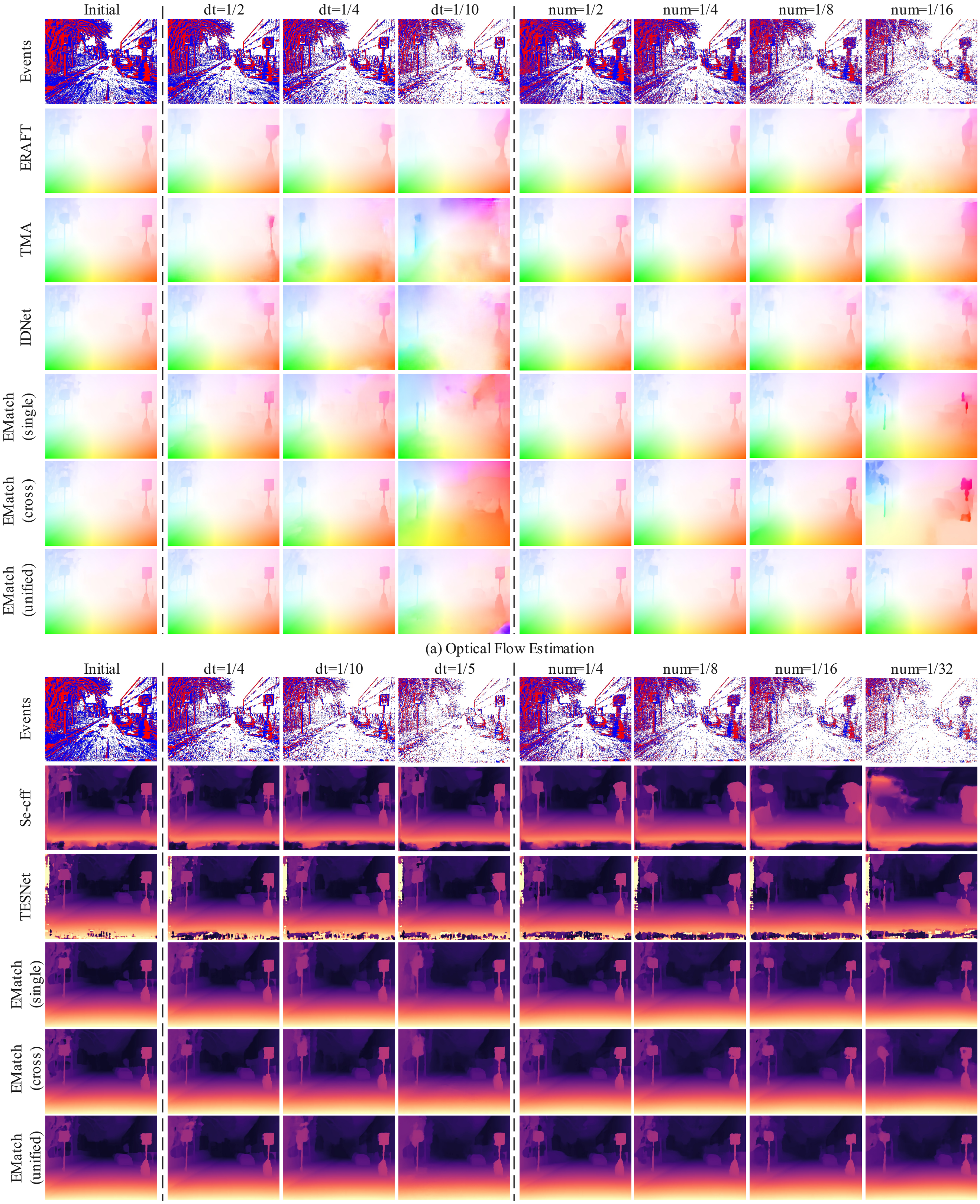
Figure S5. More qualitative comparison of EMatch with other methods. Compared to previous methods, our model can be trained using both optical flow and disparity simultaneously, alleviating overfitting on a single task in temporal or spatial domains.

the complexity of preprocessing event data.

8.3. More Qualitative Results

In Fig. S5, we present more qualitative comparison of different models, including the EMatch with different training strategies and other SOTA methods. By observing the foreground objects, it is clear that the predictions from EMatch models are more accurate, and the EMatch-unified demonstrates more significant advantages. Notably, we used data augmentation to compensate for the absence of original disparity labels at the edges, making the disparity predictions more natural. This does not affect the quantitative results, as there are no ground truths at the edges when testing.

In Fig. S6, we present more qualitative comparison for generalization performance of current models on different event data distributions. We simulate a sparser data distribution by changing the original event sampling settings (reducing sampling time dt, or deleting events at intervals) to qualitatively test the performance degradation of different models. It can be seen that our unified model has better generalization performance compared to other single-task models, because it learns priors from a wider range of data distributions (i.e. optical flow in temporal domain and disparity in spatial domain).



(a) Optical Flow Estimation

(b) Stereo Matching

Figure S6. More qualitative comparison for generalization performance. To simulate different data distributions, we reduce dt to 1/n of its original setting, or keep dt fixed but sample events at intervals to reduce its number by 1/n. Obviously, EMatch-unified performs the best.

References

[1] Wu Chen, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. Event-based stereo depth estimation by temporal-spatial context learning. *IEEE Signal Processing Letters*, 2024. 3

[2] Hoonhee Cho*, Jae-Young Kang*, and Kuk-Jin Yoon. Temporal event stereo via joint learning with stereoscopic flow. In *Proceedings of the European Conference on Computer Vision*, 2024. 4

[3] Ziluo Ding, Rui Zhao, Jiyuan Zhang, Tianxiao Gao, Ruiqin Xiong, Zhaofei Yu, and Tiejun Huang. Spatio-temporal recurrent networks for event-based optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 525–533, 2022. 3

[4] 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. 3

[5] Mathias Gehrig, Mario Millhäusler, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *Proceedings of the International Conference on 3D Vision*, pages 197–206, 2021. 3, 4

[6] Yijin Li, Zhaoyang Huang, Shuo Chen, Xiaoyu Shi, Hongsheng Li, Hujun Bao, Zhaopeng Cui, and Guofeng Zhang. Blinkflow: A dataset to push the limits of event-based optical flow estimation. In *Proceedings of the International Conference on Intelligent Robots and System*, pages 3881–3888. IEEE, 2023.

[7] Haotian Liu, Guang Chen, Sanqing Qu, Yanping Zhang, Zhijun Li, Alois Knoll, and Changjun Jiang. Tma: Temporal motion aggregation for event-based optical flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9685–9694, 2023. 3, 4

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

[9] Xinglong Luo, Kunming Luo, Ao Luo, Zhengning Wang, Ping Tan, and Shuaicheng Liu. Learning optical flow from event camera with rendered dataset. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9847–9857, 2023. 3

[10] Yeongwoo Nam, Mohammad Mostafavi, Kuk-Jin Yoon, and Jonghyun Choi. Stereo depth from events cameras: Concentrate and focus on the future. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6114–6123, 2022. 3, 4

[11] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419, 2020. 3

[12] Stepan Tulyakov, Francois Fleuret, Martin Kiefel, Peter Gehler, and Michael Hirsch. Learning an event sequence embedding for dense event-based deep stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1527–1537, 2019. 3

[13] Yilun Wu, Federico Paredes-Vallés, and Guido CHE De Croon. Lightweight event-based optical flow estimation via iterative deblurring. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 14708–14715. IEEE, 2024. 3, 4

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

[15] Kaixuan Zhang, Kaiwei Che, Jianguo Zhang, Jie Cheng, Ziyang Zhang, Qinghai Guo, and Luziwei Leng. Discrete time convolution for fast event-based stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8676–8686, 2022. 3

[16] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018. 3