Prototypical Transformer as Unified Motion Learners

Cheng Han [*] [1] [2] **Yawen Lu** [*] [3] **Guohao Sun** [2] **James C. Liang** [2] **Zhiwen Cao** [3] **Qifan Wang** [4] **Qiang Guan** [5]
Sohail A. Dianat [2] **Raghuveer M. Rao** [6] **Tong Geng** [7] **Zhiqiang Tao** [2] **Dongfang Liu** [2]

Abstract

In this work, we introduce the Prototypical Transformer (ProtoFormer), a general and unified framework that approaches various motion tasks from a prototype perspective. ProtoFormer seamlessly integrates prototype learning with Transformer by thoughtfully considering motion dynamics, introducing two innovative designs. First, *Cross-Attention Prototyping* discovers prototypes based on signature motion patterns, providing transparency in understanding motion scenes. Second, *Latent Synchronization* guides feature representation learning via prototypes, effectively mitigating the problem of motion uncertainty. Empirical results demonstrate that our approach achieves competitive performance on popular motion tasks such as optical flow and scene depth. Furthermore, it exhibits generality across various downstream tasks, including object tracking and video stabilization. Our code is available here.

1. Introduction

> *"All is flux, nothing is stationary."*
>
> – Heraclitus (Plato, 402 BC)

The aphorism attributed to Heraclitus underscores the foundation of physics in the natural world. The quest to understand motion holds the potential to unveil the intricate secrets of intelligence (Hawkins & Blakeslee, 2004), shedding light on the systematic construction of artificial entities. However, the proliferation of excessively granular motion tasks has catalyzed a fervor for *specialized* models in the realm of deep learning, standing in stark contrast to the enduring scientific tradition — a *generic* solution to elegantly describe physical phenomena in the universe. The following question naturally arises: ① Can we discover a *unified* model that serves as a comprehensive motion learner?

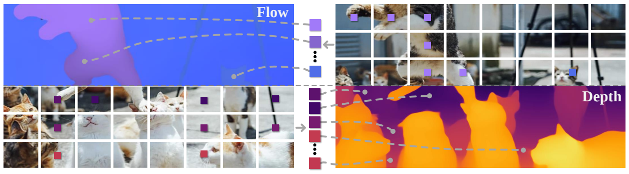

Figure 1: **ProtoFormer as a unified framework** considers motion as different levels of dynamics granularity (*e.g.*, instance-driven flow, pixel-anchored depth, *etc*). 🟪🟥🟥 are prototypes.

Motion learning tasks essentially encompass pixel-level dynamics and correspondence (*e.g.*, optical flow and depth scene estimation). A prevalent challenge in these tasks is the presence of photometric and geometric inconsistencies (*e.g.*, shadow and occlusion), which introduce significant uncertainty during the matching process (Xiong et al., 2021; Zhao et al., 2020). Consequently, the accuracy of pixel-wise feature matching is compromised, detrimentally impacting the learning of the underlying motion representation. To address this challenge, a promising solution lies in prototype learning (Smith & Minda, 2002; Jiang et al., 2012), where motion measurements are categorized into discrete exemplars. In each exemplar, a prototype functions as a central archetype, capturing the essential attributes of its associated motion patterns observed in the data. The clustering of similar patterns around prototypes can effectively minimize the impact of noise and outlier pixels in feature matching, thereby significantly mitigating the issue of uncertainty. In this context, we can approach question ① by exploring: ② How can we design a model that incorporates the principles of prototype learning in motion tasks?

Recently, the Transformer architecture has attained ubiquitous adoption, enjoying unambiguous acclaim in both the domains of vision and language (Zhu et al., 2021; Yang et al., 2023; Kim et al., 2021). Its accomplishments are underpinned by the attention mechanism, endowing models with the capability to selectively attend to salient entities within input data. The capacity to generate context-aware feature representations represents a substantial enhancement of the model's effectiveness, enabling it to apply as a general solution to diverse vision tasks. Inspired by its encouraging

[*]Equal contribution [1]University of Missouri – Kansas City [2]Rochester Institute of Technology [3]Purdue University [4]META AI [5]Kent State University [6]DEVCOM Army Research Laboratory [7]University of Rochester. Correspondence to: Zhiqiang Tao <zxtics@rit.edu>, Dongfang Liu <dongfang.liu@rit.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

success, our inquiry naturally delves into a more specific dimension: ③ How to incorporate the prototype learning capacity into the architecture of Transformer?

To this demand, we employ Prototypical TransFormer (ProtoFormer) as a unified solution on various motion tasks. Specifically, ProtoFormer incorporates prototype learning with Transformer. The method begins by tokenizing images features into patches, where the features are initialized into distinct prototypes. These prototypes are then recursively updated via *Cross-Attention Prototyping* (§3.2.1) to capture representative motion characteristics via clustering. After assignments and updates, *Latent Synchronization* (§3.2.2) builds up prototype-feature association, aiming at denoising and mitigating motion ambiguity. The refined features are later fed into the decoder for task-specific predictions.

Taking the innovations together, ProtoFormer exhibits several compelling attributes. ❶ **Architectural elegance**: Leveraging a prototype-guided Transformer architecture, ProtoFormer can handle heterogeneous motion tasks at different levels of dynamics granularity within the unified fashion (see Fig. 1). ❷ **Predictive robustness**: Prototype learning can inherently diminish the noisy outliers through its density criterion (§2). Anchored by the recursively refined prototypes, feature learning can be further guided towards more robust representations (§3.2.2). Consequently, it offers a viable solution to the challenge of motion ambiguity (see Fig. 3 and Fig. 4). ❸ **Systemic explainability:** The density-based nature from recursive prototyping offers intuitive visual demonstration of motion prototypes (see Fig. 5 and Fig. S1 in Appendix), enabling direct interpretation of various dynamic patterns sketched by the system.

We conduct a set of comprehensive experiments to evaluate the effectiveness of our approach. In §4.1, ProtoFormer presents compelling results on optical flow. For example, our approach distinctly outperforms CRAFT, achieving 0.48 and 0.69 on the clean and final pass of Sintel, respectively. In §4.2, we further show the superior performance on depth scene estimation (*e.g.*, 18.6% improvement in Sintel compared to AdaBins). Also, visual evidence in §4.3 demonstrates the systemic explainability, which displays direct prototype-pixel correlations. Results on various downstream tasks including object tracking (§S5) and video stabilization (§S6) are detailed in the Appendix. We hope our research could provide foundational insights into related fields.

2. Related Work

Motion Task. Motion tasks involve intricate processes, encompassing the identification, modeling, and prediction of motion patterns in objects and scenes. These tasks are foundational to diverse computer vision applications, including vehicle and pedestrian motion detection (Shen et al., 2023; Khalifa et al., 2020; Marathe et al., 2021; Liang et al., 2022b; Xu et al., 2022a; Cui et al., 2024), abnormal activity detection (Li et al., 2021b; Tudor Ionescu et al., 2017; Zhou et al., 2019), and video compression (Gao et al., 2022; Hu et al., 2021; Lu et al., 2019). In the domain of motion tasks, optical flow (Ranjan & Black, 2017; Sun et al., 2018; Teed & Deng, 2020; Huang et al., 2022; Shi et al., 2023; Lu et al., 2024) and depth estimation (Bhat et al., 2021; Patil et al., 2022), stand out as particularly representative, significantly influencing downstream tasks like object tracking and video stabilization. Current endeavors predominantly focus on task-specific solutions, resulting in duplicated research efforts and suboptimal hardware utilization. In contrast, ProtoFormer stands as a distinctive exploration, aiming to integrate motion tasks under a unified paradigm. This endeavor conceptually differentiates us from existing arts in the field.

Prototype Learning. Traditionally, prototype learning in machine learning establishes a metric space where features are distinguished by computing their distances/densities to prototypical representations (Lee et al., 2023). Early methods include classical approaches like support vector machines (Cortes & Vapnik, 1995), random forest (Breiman, 2001), logistic regression (Hastie et al., 2009), etc. With the advent of deep neural networks (DNNs), prototype-based deep learning models find broad applications in few-shot learning (Dong & Xing, 2018; Wang et al., 2019; Liu et al., 2020b; Yang et al., 2020; Li et al., 2021a; Wang et al., 2022), zero-shot learning (Jetley et al., 2015; Xu et al., 2020), text classification (Farhangi et al., 2022), and explainable classifiers (Wang et al., 2023; Zhou et al., 2022; Qin et al., 2023). In the context, we argue: movements within the same object or proximate regions exhibit noteworthy similarities, forming a collective of prototypes. Integrating prototype learning into the model design facilitates the natural encapsulation of diverse dynamic characteristics, enhancing the model's ability to comprehend motion in various contexts.

Moreover, human vision deploys a sophisticated prototyping ability, skillfully focusing on relevant parts of the visual tableau while filtering out extraneous elements (Simon & Newell, 1971; Rudin et al., 2022; Giese & Poggio, 2003). This feat is realized through Region-of-Interest (RoI) clustering (Meyer et al., 2004; Mantini et al., 2012), disassembling discrete pixel entities into salient conceptual groupings. This hierarchical process synthesizes elementary visual elements, like lines, forms, and hues, into complex abstractions representing objects, vistas, and individuals (Kepes, 1995). In an effort to mimic the human visual system, we conceptualize prototype learning from a clustering perspective, iteratively updating and exploring representative prototypes to capture nuanced motion characteristics.

Transformer Architecture. The transformative impact of Transformers in natural language processing (NLP) (Brown

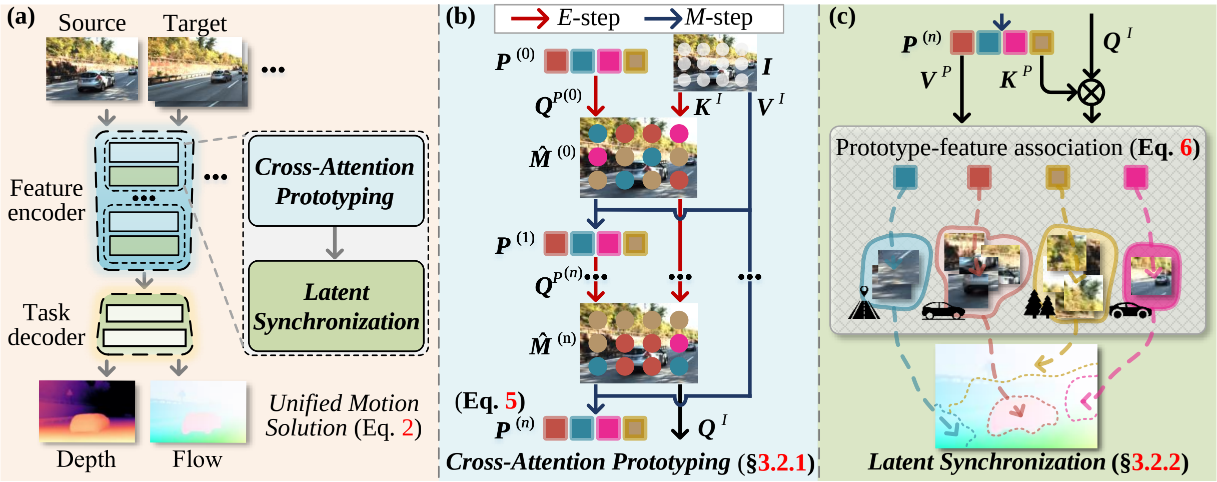

Figure 2: (a) **Overall pipeline of ProtoFormer** (§3.2). Movement of a small part of an object within an image is being considered as a rigid motion. In our approach, we use prototypes to understand or predict this kind of motion pattern. (b) In each layer of the ***Cross-Attention Prototyping*** (see §3.2.1), there are N sequential iterations encompassing the assignment of feature-prototypes (*i.e.*, E-step) and the subsequent updating of these prototypes (*i.e.*, M-step) via Eq. 5. (c) Concurrently, the ***Latent Synchronization*** process (see §3.2.2) associates the feature representations via the freshly updated motion prototypes, (see Eq. 7). For (b) and (c), we apply optical flow for illustration, which demonstrates straightforward systemic explainability. More visualization results are shown in §4.3.

et al., 2020; Devlin et al., 2018; Liu et al., 2020a; Raffel et al., 2020; Vaswani et al., 2017) has spurred their extensive application in vision-related tasks, including image classification (Dosovitskiy et al., 2021; Liu et al., 2022; 2021; Wang et al., 2023), and image segmentation (Strudel et al., 2021; Wang et al., 2021a;d; Zheng et al., 2021). Transformers excel in visual applications, outperforming convolutional neural networks (CNNs) (Han et al., 2023a; 2024). This superiority arises from their ability to capture extensive token dependencies in a global context, a limitation of concurrent CNN-based methods that focus on local interactions within convolutional layers (Han et al., 2022; Khan et al., 2022; Lin et al., 2022; Han et al., 2023a; Cai et al., 2022). The unique attention design in Transformers enables the understanding of global spatial relationships, making them ideal for motion-related tasks where extensive spatial interconnections are pivotal. By amalgamating the attention mechanism with prototype learning, we aim to harness Transformers' representational prowess to unravel intricate patterns in motion tasks, offering a unified, Transformer-based solution.

3. Methodology

By going through existing literature (§2), the integration of prototype learning with the Transformer architecture presents a promising avenue to various motion tasks. In this section, we first revisit Transformer architecture and reformulate its attention mechanism as prototype learning (§3.1). Based on this insight, we introduce ProtoFormer (§3.2), including two pivotal contributions: *Cross-Attention Prototyping* (§3.2.1) and *Latent Synchronization* (§3.2.2), answering question ③. We elaborate our method below.

3.1. Preliminary

In our study, we re-conceptualize the Transformer's attention mechanism through the lens of classical clustering; while the traditional attention map is obtained by computing the similarity between all query-key pairs (Zhou et al., 2021a; Han et al., 2023b), our approach introduces a density-based cross-attention estimation, specifically designed to accommodate motion characteristics by aggregating local rigid motion patterns into distinct prototype clusters.

Classic clustering, a widely embraced paradigm, entails segregating m observations into k distinct groups. It ensures that each observation is aligned with only one cluster that it most closely associates with, based on the highest likelihood or minimal distance (*e.g.*, proximity to the mean). Formally, clustering can be optimized iteratively between two phases:

- *Assignment Phase* allocates each observation to the cluster where it exhibits the maximal probability of belonging or the least spatial separation.
- *Centroid Recalculation Phase* recalculates the centroids of the clusters to reflect the new configuration of the observations within each cluster.

These two phases persist until a point of convergence is reached, indicated either by a cessation in the modification of assignments or by changes that fall beneath a predetermined threshold, thus implying cluster stabilization.

From a mathematical perspective, assume θ as the centroids (Wang et al., 2023) of the clusters, with $x \in \mathcal{X}$ representing an individual observation:

$$\theta^{(n+1)} = \arg\max_{\theta} \mathbb{E}(p_{x\sim\mathcal{X}}(x|\theta^{(n)})). \tag{1}$$

Here, $\theta^{(n)}$ denotes the n-th iteration deduced centroid; $p(\cdot)$ represents the posterior probability of the data assignments.

3.2. ProtoFormer

The primary objective for ProtoFormer is to optimize the expected likelihood function in the context of clustering within a *unified motion solution* as:

$$\hat{\theta}_k = \sum_{j=1}^{K} p(\mathcal{X}|\theta_j) \cdot P(\theta_k, \theta_j), \tag{2}$$

where θ symbolizes the centroid representations. We recognize the centroid θ as *Prototype*, which is our optimization target. K represents the total cluster count. The probability $p(\mathcal{X}|\theta_k) \in (0,1)$ are the mixing coefficients for each cluster $k \in \mathcal{K}$, adhering to the constraint $\sum_k p(x|\theta_k) = 1$. The projected prototype representation $P(\cdot)$ describes the learnable dense vector from a shared parametric family from k-th prototype. This function aggregates across all clusters $\mathcal{K}$, with each projected prototype $P(\theta_k, \theta_j)$ denoting the new projected representations on its respective cluster. $\hat{\theta}_k$ demonstrates the updated prototype considering the prototype representation of θ, and the posterior probability $p(\mathcal{X}|\theta_k)$ (*i.e.*, the conditional likelihood of grouping the data $\mathcal{X}$ given prototype parameterization θ_k).

3.2.1. Cross-Attention Prototyping via EM clustering

To realize prototyping as a unified motion solution on Transformer, we reformulate the conventional Transformer's self-attention (Vaswani et al., 2017) into our novel prototypical cross-attention, optimized via Expectation-Maximization (EM) clustering. The designed optimization provides a density-based estimation to compute the maximum likelihood for p_k and θ_k, utilizing posterior probabilities.

E-Step: For each observation $x_i \in \mathcal{X}$, E-Step computes the n-th iteration posterior probabilities $p_k(x_i)$, indicating its affiliation to center θ_k with the logit vector $s_{x_i,k}$ iteratively as:

$$p_k^{(n)}(x_i) = \frac{s_{x_i,k}^{(n)} \cdot P(x_i, \theta_k^{(n)})}{\sum_{j=1}^{K} s_{x_i,j}^{(n)} \cdot P(x_i, \theta_j^{(n)})}. \tag{3}$$

$s_{x_i,k}^{(n)}$, $\theta_k^{(n)}$ are the parameters estimated at the n-th iteration.

M-Step: Each cluster θ_k obtains its maximum likelihood estimations $p_k^{(n)}$ and $\theta_k^{(n)}$ from projected sub-sample representations P', updated as:

$$\theta_k^{(n+1)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} p_k^{(n)}(x_i) \cdot P'(\theta_k^{(n)}, \theta_j^{(n)}). \tag{4}$$

In practice, given feature embeddings $\boldsymbol{I} \in \mathbb{R}^{HW\times D}$ and set initial cluster centers $\boldsymbol{P}^{(0)}$ as K prototypes, we encapsulate the discussed EM clustering process within a *Cross-Attention Prototyping* layer (see Fig. 2(b)) with N iterations:

$$\begin{aligned} &E\text{-step:} \quad \hat{\boldsymbol{M}}^{(n)} = \mathrm{softmax}_K(\boldsymbol{Q}^{P^{(n)}}(\boldsymbol{K}^I)^\top), \\ &M\text{-step:} \quad \boldsymbol{P}^{(n+1)} = \hat{\boldsymbol{M}}^{(n)}\boldsymbol{V}^I \in \mathbb{R}^{K\times D}, \end{aligned} \tag{5}$$

where $n \in \{1, \cdots, N\}$. $\hat{\boldsymbol{M}} \in [0,1]^{K\times HW}$ is the "soft" pixel-prototype assignment matrix, representing probability maps of prototypes. $\boldsymbol{Q}^P \in \mathbb{R}^{K\times D}$ is the query vector projected from the prototype representation $\boldsymbol{P}$, and $\boldsymbol{V}^I, \boldsymbol{K}^I \in \mathbb{R}^{HW\times D}$ are the value and key vectors projected from the image features $\boldsymbol{I}$, respectively. Our proposed layer can thus update the prototyping membership $\hat{\boldsymbol{M}}$ (*i.e.*, E-step) and the prototypes $\boldsymbol{P}$ (*i.e.*, M-step) iteratively. The key characteristic of this approach is its assurance of an incremental convergence in the likelihood function with each iteration (see Eq. 4). In essence, the E-step evaluates the current membership of the data representations based on existing prototypes, while the M-step refines the prototypes to align with pixels, ensuring a steady progression towards optimal clustering. By performing cross-attention prototyping on the source and target images separately, it addresses the complexities associated with motion uncertainty and photometric inconsistency. We also modify the default *softmax* operator from HW to K, mimicking the EM clustering.

The proposed layer enjoys several compelling features:

- *Convergence:* EM clustering monotonically improves the marginal likelihood and is empirically validated to converge towards a local optimum (Vattani, 2009; Ikotun et al., 2023; Balakrishnan et al., 2017), given a sufficient number of iterations (*Proof* is provided in Appendix §S4).

  ***Proposition 1**.* Suppose that the EM operator is contractive with parameter $\kappa \in (0,1)$ on the ball $\mathcal{B}_2(r;\theta)$, and the initial vector $\theta^{(0)}$ belongs to $\mathcal{B}_2(r;\theta)$. For a given iteration N, when the sample size m is large enough to ensure $\epsilon_M(\frac{m}{N}, \frac{\delta}{N}) \leq (1-\kappa)r$. Then the EM iterates $\{\theta^{(n)}\}_{n=0}^{N}$ based on $\frac{m}{N}$ samples per round satisfy the bound that:

  $$||\theta^{(n)} - \hat{\theta}||_2 \leq \kappa^n ||\theta^{(0)} - \hat{\theta}||_2 + \frac{1}{1-\kappa}\epsilon_M(\frac{m}{N}, \frac{\delta}{N}). \tag{6}$$

  In this context, our proposed *Cross-Attention Prototyping* leverages the strengths of recursive clustering, iterated over N steps. This approach significantly enhances the likelihood of converging to an optimal configuration for motion partitioning (see §4.3).
- *Transparency:* Prototyping emerges as an indispensable mechanism for contextual understanding from motion scenes, recognizing and grouping similar patterns and movements. It clusters pixels as prototypes that display homogeneity in characteristics such as flow or depth. By aggregating entities that exhibit shared attributes, the prototypes are able to describe the intrinsic dynamics. Furthermore, prototyping provides a foundational schema for

motion comprehension, wherein each prototype embodies a microcosm of the objects within the scene, encapsulating unique elements and their interrelations.

- *Efficiency: Cross-Attention Prototyping* operates with a time complexity of $\mathcal{O}(NKHWD)$, showing a significant improvement over the self-attention mechanism with $\mathcal{O}(H^2W^2D)$ (see §4.3). The foundation lies in the relationship that $NK \ll HW$ (*e.g.*, 60 vs 25920 in the first stage with image of 960 × 432 resolution). This difference becomes especially pronounced in pyramid architectures (Wang et al., 2021c; Liu et al., 2021; Liang et al., 2023; 2024), where the total number of NK tends to be substantially smaller than HW, particularly in the early stages of the network. In each iteration, only the query matrix $\boldsymbol{Q}$ requires an update; the key $\boldsymbol{K}$ and value $\boldsymbol{V}$ matrices are computed just once. This selective updating significantly reduces the computational load particularly beneficial in handling large-scale data in high-dimensional feature spaces.

3.2.2. Prototype-Feature Corresponding by Latent Synchronization

We further refine the feature representations, synchronizing the projection of K prototypes into a $H \times W$ feature representation. This approach aligns the prototype representations with motion features (see Fig. 2(c)).

The main technique lies in the Feed-Forward Network (FFN) that incorporates with a masked cross-attention mechanism:

$$\hat{I} = \text{FFN}(\text{Cross-Attention}(Q^I, K^P, V^P, \mathcal{M}_P)), \tag{7}$$

where $\mathcal{M}_P$ stands for the feature assignment mask maps based on the similarity of corresponding prototypes P. $\hat{I}$ represents the refined feature, Q^I denotes the query projection derived from the input image feature, K^P and V^P represent the key and value projections sourced from the learning prototypes, respectively. *Latent Synchronization* primarily aims at augmenting the feature learning for prototype-feature association, reducing motion ambiguity. It facilitates the extraction and encapsulation of each feature representation's latent distribution within its own prototype.

Latent Synchronization enjoys appealing characteristics:

- *Blended Paradigm: Latent Synchronization* blends unsupervised prototype mining (§3.2.1) and supervised feature representation learning (§3.2.2) in a synergy − local significant motion patterns are automatically explored to facilitate density-based prototyping; the supervisory signal from task-specific heads directly optimizes the representation, which in turn boosts meaningful prototyping.
- *Prototype-Anchored Learning:* Density-based prototype learning computes reliable probabilities in prototype assignment recursively (Eq. 5). Anchored by the updated prototypes, the feature is further guided towards more robust representations via prototype-feature association (Eq. 7). Consequently, the motion patterns are more likely to center around areas of high data density, which in turn boosts robustness towards motion ambiguity.

3.3. Implementation Details

ProtoFormer is built upon Twins architecture (Chu et al., 2021). Detailed training and testing configurations are provided in §S1. The key components (see Fig. 2) are:

- *Feature Encoder* contains two stages with window sizes of 4 and 8, respectively, which convert input images into features. We follow the common practice (Chu et al., 2021) and utilize two blocks within each stage. In addition, we reformulate the vanilla self-attention into our cross-attention prototyping layers (§3.2.1) to recursively update the initialized prototypes. Once these prototypes have been updated, the latent synchronization layer (§3.2.2) augments the feature learning via prototype-feature association, reducing motion ambiguity.
- *Task Decoder* is designed for task-specific motion predictions. We follow the design (Huang et al., 2022; Zhou et al., 2021b) for flow and depth.
- *Cross-Attention Prototyping* reformulates the vanilla self-attention layers by the EM cross-attention clustering process for prototyping learning (§3.2.1). Within each Cross-Attention Prototyping layer, twenty prototypes and three iterations are conducted as default (§4.3).
- *Latent Synchronization* updates the feature map in accordance with the prototypes present in the latent feature space (§3.2.2) by applying Feed-Forward Networks and incorporating a masked cross-attention mechanism.

4. Experiments

We comprehensively examine the performance and unity of our proposed ProtoFormer on two representative motion tasks, including optical flow (see §4.1) and scene depth (see §4.2). In our pursuit of a unified solution for motion-related tasks, we not only underscore the merit of such integration but also exhibit superior performance, and to further enhance the *paradigmatic generalization*, we broaden its application to object tracking (§S5) and video stabilization (§S6) in Appendix, reaching *competitive* performance.

4.1. Experiments on Optical Flow

Datasets. Following the previous works (Huang et al., 2022; Dong et al., 2023), we first train the proposed method on FlyingChair (Dosovitskiy et al., 2015) and FlyingThings (Mayer et al., 2016), and then fine-tune it on a large combination of datasets (C+T+S+K+H) to allow evaluation on the Sintel and KITTI-2015 benchmarks.

Metrics. To facilitate a fair comparison, we adopt the commonly used metric, the average end-point-error (F1-epe),

Training	Method	Sintel (train)		KITTI-15 (train)		Sintel (test)		KITTI-15 (test)
		Clean ↓	Final ↓	F1-epe ↓	F1-all ↓	Clean ↓	Final ↓	F1-all ↓
A	Perceiver IO (Jaegle et al., 2022)	1.81	2.42	4.98	-	-	-	-
	RAFT-A (Sun et al., 2021)	1.95	2.57	4.23	-	-	-	-
C+T	RAFT (Teed & Deng, 2020)	1.43	2.71	5.04	17.4	-	-	-
	Separable Flow (Zhang et al., 2021)	1.30	2.59	4.60	15.9	-	-	-
	GMA (Jiang et al., 2021)	1.30	2.74	4.69	17.1	-	-	-
	AGFlow (Luo et al., 2022b)	1.31	2.69	4.82	17.0	-	-	-
	KPA-Flow (Luo et al., 2022a)	1.28	2.68	4.46	15.9	-	-	-
	DIP (Zheng et al., 2022)	1.30	2.82	4.29	13.7	-	-	-
	GMFlowNet (Zhao et al., 2022)	1.14	2.71	4.24	15.4	-	-	-
	GMFlow (Xu et al., 2022b)	1.08	2.48	7.77	23.4	-	-	-
	CRAFT (Sui et al., 2022)	1.27	2.79	4.88	17.5	-	-	-
	FlowFormer (Huang et al., 2022)	1.01	2.40	4.09	14.7	-	-	-
	SKFlow (Zhai et al., 2022)	1.22	2.46	4.27	15.5	-	-	-
	MatchFlow (Dong et al., 2023)	1.14	2.71	4.19	13.6	-	-	-
	ProtoFormer (Ours)	1.04	2.43	4.08	14.6	-	-	-
C+T+S+K+H	RAFT (Teed & Deng, 2020)	0.76	1.22	0.63	1.5	1.61	2.86	5.10
	RAFT-A (Sun et al., 2021)	-	-	-	-	2.01	3.14	4.78
	Separable Flow (Zhang et al., 2021)	0.69	1.10	0.69	1.60	1.50	2.67	4.64
	GMA (Jiang et al., 2021)	0.62	1.06	0.57	1.2	1.39	2.47	5.15
	AGFlow (Luo et al., 2022b)	0.65	1.07	0.58	1.2	1.43	2.47	4.89
	KPA-Flow (Luo et al., 2022a)	0.60	1.02	0.52	1.1	1.35	2.36	4.60
	DIP (Zheng et al., 2022)	-	-	-	-	1.44	2.83	4.21
	GMFlowNet (Zhao et al., 2022)	0.59	0.91	0.64	1.51	1.39	2.65	4.79
	GMFlow (Xu et al., 2022b)	-	-	-	-	1.74	2.90	9.32
	CRAFT (Sui et al., 2022)	0.60	1.06	0.58	1.34	1.45	2.42	4.79
	Flowformer (Huang et al., 2022)	0.48	0.74	0.53	1.11	1.20	2.12	4.68
	SKFlow (Zhai et al., 2022)	0.52	0.78	0.51	0.94	1.28	2.23	4.87
	MatchFlow (Dong et al., 2023)	0.51	0.81	0.59	1.3	1.33	2.64	4.72
	ProtoFormer (Ours)	0.48	0.69	0.50	1.09	1.06	2.07	4.35

Table 1: **Quantitative results on standard Sintel and KITTI flow benchmarks.** ‘A’ denotes the Autoflow dataset; ‘C + T’ denotes training on the FlyingChairs and FlyingThings datasets only; ‘C + T + S + K + H’ fine-tunes on a combination of Sintel, KITTI, and HD1K training sets. Error metrics are lower is better with "↓", and accuracy metrics are higher is better with "↑". Same for Table 2.

measuring the average l_2 distance between the prediction and ground truth, and the percentage of outliers over all pixels (F1-all), which describes the error exceeding 3 pixels or 5% $w.r.t.$ the ground truth, for optical flow estimation.

Quantitative Results. Table 1 reports the evaluation results of our model on the Sintel and KITTI datasets. The results under the ‘C+T’ setting reflect the generalization capability of our ProtoFormer, where it achieves 1.04 and 2.43 on the clean and final pass of Sintel, improving the recently popular CRAFT (Sui et al., 2022) by 0.23 and 0.36. After training under the mixed setting of ‘C+T+S+K+H’, the proposed prototype-based model achieves 1.06 and 2.07 on the clean and final pass of Sintel and 4.35 F1-epe on KITTI.

Qualitative Results. Fig. 3 shows the qualitative results on the Sintel flow dataset. As seen, ProtoFormer shows more global and finer details on both object and motion boundaries, without being affected by the shadows and textureless surfaces. In the first and fourth examples, our model recovers full shape and fine details remarkably well, *e.g.*, on bamboo and weapons. This is in stark contrast to other methods, which struggle with producing clear predictions, primarily due to challenges posed by occlusions and variations in illumination. In the second and third examples, we show the significantly consistent estimation of the occluded and textureless regions, *e.g.*, the backpack and birds in the sky. The highlighted regions with red boxes prove the efficacy of ProtoFormer in object clustering and avoiding motion ambiguity. More results are shown in Fig. S2.

4.2. Experiments on Scene Depth

Datasets. Similar to optical flow, we evaluate ProtoFormer on both real and synthetic datasets, using KITTI (Geiger et al., 2013) and MPI Sintel (Butler et al., 2012a) for evaluation. As an autonomous driving dataset consisting of 61 outdoor scenes of various modalities, we use the KITTI Eigen depth split, which contains a standard depth estimation split proposed by Eigen et al. (Eigen et al., 2014) consisting of 32 scenes for training and 29 scenes for testing. The MPI Sintel is a long synthetic stereo sequence with a large motion and depth range and contains a total of 35 rendered sequences.

Metrics. We follow the standard metrics of absolute relative error (Abs Rel), root mean square error (RMSE), and the percentage of inlier pixels with $\delta_1 < \tau$ $(\tau = 1.25)$.

Quantitative Results. Table 2 shows the test results on Sintel and KITTI. Our model achieved the best performance on most of the error and accuracy metrics, proving its powerful feature representation and generalization ability to different scenarios. Compared with recent directly supervised depth estimation methods (Eigen et al., 2014; Fu et al., 2018; Bhat et al., 2021), ProtoFormer demonstrates significantly superior capability, benefited from the incorporation of prototypical learning and cross-attention architecture. Even

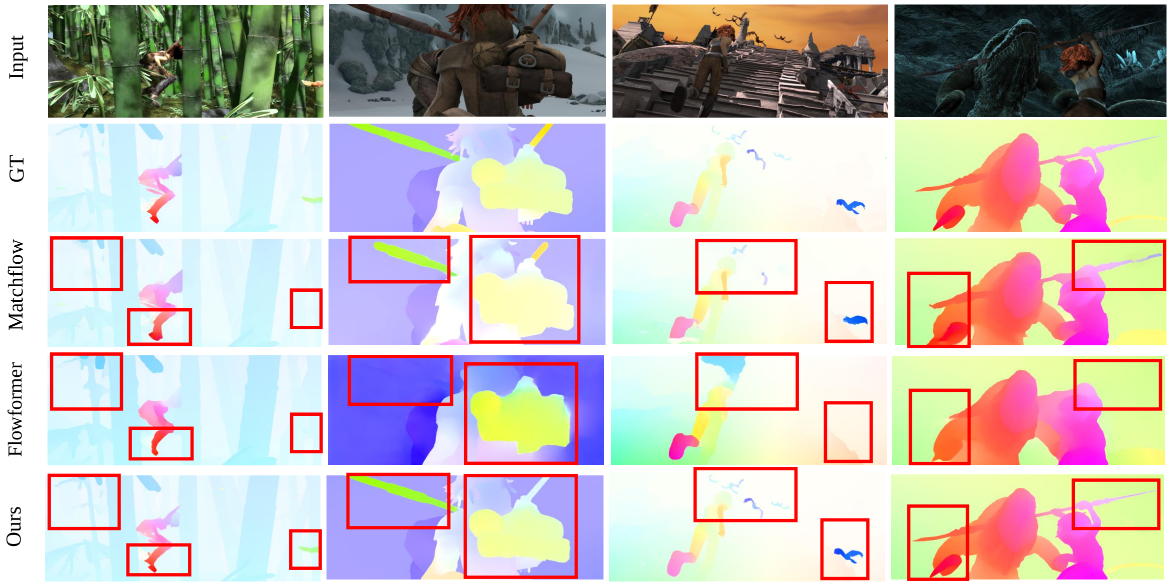

Figure 3: **Qualitative results on the Sintel.** The red boxes highlight the regions compared. Matchflow (Dong et al., 2023) appears blurry and ambiguous on textureless and occluded objects, while Flowformer (Huang et al., 2022) fails to recover complete and detailed information. Ours can estimate clear and complete flow motion, which is closer to ground truth.

without using additional constraints and priors such as surface normal and piecewise planarity in (Yin et al., 2019; Patil et al., 2022), our model improves the performance of concurrent P3Depth in error by a large margin on KITTI. Furthermore, our approach demonstrates superior performance compared to methods that implement self-supervised consistency and strategies (Godard et al., 2017; Zhang et al., 2023). This underscores the efficacy of our model, which exhibits exceptional adaptability and learning capacity for fine-tuning across more general motion-related tasks.

Method	Sintel			KITTI		
	Abs Rel ↓	RMSE ↓	Sq Rel ↓	Abs Rel ↓	RMSE ↓	δ_1 ↑
Eigen et al.	0.797	0.834	0.703	0.203	6.307	0.702
Godard et al.	-	-	-	0.114	4.935	0.861
Fu et al.	-	-	-	0.072	2.727	0.932
Yin et al.	0.746	0.611	0.652	0.072	3.258	0.938
AdaBins	0.730	0.572	0.647	0.067	2.960	0.949
P3Depth	0.653	0.396	0.571	0.071	2.842	0.953
Ours	0.594	0.486	0.538	0.062	2.716	0.949

Table 2: **Quantitative results on Sintel and KITTI depth datasets.** With both test data unseen by the model, we can achieve leading performance over state-of-the-art methods (Eigen et al., 2014; Godard et al., 2017; Fu et al., 2018; Yin et al., 2019; Bhat et al., 2021; Patil et al., 2022).

Qualitative Results. Fig. 4 shows the qualitative depth comparison on the KITTI Eigen depth datasets. Our ProtoFormer demonstrates superior capability in delineating object surface contours, particularly in scenarios involving dynamic entities such as pedestrians and vehicles, as well as in capturing the finer details of objects like traffic signs and light poles. For example, for the moving pedestrians and vehicles in Sample 1 and Sample 3, ProtoFormer estimates a more consistent and complete depth on object surfaces and provides a clearer boundary than P3Depth (Patil et al., 2022) and AdaBins (Bhat et al., 2021). For the plant stand and traffic poles in Sample 2, Sample 3 and Sample 4, our methods separates other methods evidently from the noisy and complex scene backgrounds. These demonstrated the superiority of incorporating prototype learning into depth training to perceive geometric consistent and mitigate motion ambiguity. More visual evidences are provided in Fig. S3.

4.3. Diagnostic Experiments

This section ablates ProtoFormer's key components and configurations. More ablations are included in Appendix §S3.

Key Components Analysis. We study the major components of ProtoFormer: *Cross-Attention Prototyping* (§3.2.1) and *Latent Synchronization* (§3.2.2). A `Base` model is designed without considering prototype updating and prototype-feature assignment. Shown in Table 3a, `Base` reaches 0.55 and 0.81 in average EPE. After adding *Cross-Attention Prototyping*, substantial improvements are observed (*i.e.*, 0.55 → 0.51 in clean pass), suggesting the efficacy of prototyping updating even without explicit prototype-feature assignment. Incorporating *Latent Synchronization* into `Base` can observe a noticeable performance gain (*i.e.*, 0.81 → 0.77 in final pass). Finally, the integration of the two techniques culminates in peak performance.

Cross-Attention Prototyping. We then study the efficacy of *Cross-Attention Prototyping* by comparing to different updating methods, including cosine similarity, conventional cross-attention (Vaswani et al., 2017), Criss cross-attention (Huang et al., 2019) and K-Means (Yu et al., 2022). From the efficient and effective perspectives, *Cross-Attention Prototyping* outperforms competitive methods (see Table 3b). We further study the iteration step N in Table 3c,

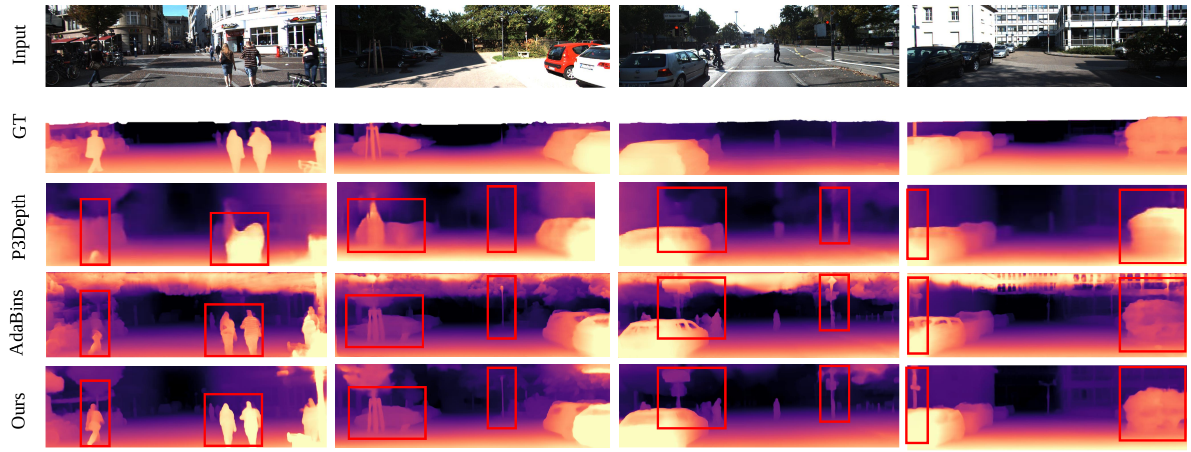

Figure 4: **Qualitative depth comparison results on the KITTI.** The red boxes indicate the highlighted regions. P3Depth (Patil et al., 2022) and AdaBins (Bhat et al., 2021) have limited receptive fields and do not consider conceptual object-level groupings, thus producing discontinuous and ambiguous predictions. While ours can estimate consistent and sharp depths, which is closer to ground truth.

Table 3: A set of **ablative studies** on optical flow (see §4.3). The best performances are marked in **bold**.

Algorithm Component	#Params	Sintel clean	Sintel final
Base	9.63M	0.55	0.81
+ Cross-Attention Prototyping	11.57M	0.51	0.74
+ Latent Synchronization	10.26M	0.53	0.77
ProtoFormer (All included)	11.90M	0.48	0.69

(a) Key Component Analysis

Variant Prototype Updating Strategy	#Params	Sintel clean	Sintel final
Cosine Similarity	10.28M	0.51	0.75
Vanilla Cross-Attention (Vaswani et al., 2017)	14.88M	0.50	0.73
Criss Cross-Attention (Huang et al., 2019)	14.56M	0.50	0.72
K-Means (Yu et al., 2022)	11.81M	0.49	0.71
Cross-Attention Prototyping (Eq. 5)	11.90M	0.48	0.69

(b) *Cross-Attention Prototyping*

#Iterations (N)	#Params	Sintel clean	Sintel final
1	11.90M	0.52	0.75
2		0.49	0.71
3		0.48	0.69
4		0.48	0.68

(c) Number of Iterations

#Prototypes (K)	#Params	Sintel clean	Sintel final
10	8.95M	0.53	0.78
50	9.78M	0.51	0.73
100	11.90M	0.48	0.69
200	14.21M	0.49	0.71

(d) Number of Prototypes

Latent Synchronization	#Params	Sintel clean	Sintel final
None	11.27M	0.51	0.74
Vanilla FC Layer	11.64M	0.50	0.73
FC w/ Similarity (Ma et al., 2023)	11.76M	0.49	0.71
Ours (Eq. 7)	11.90M	0.48	0.69

(e) *Latent Synchronization*

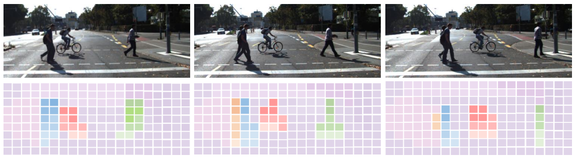

Figure 5: **Visualization of proto-feature mapping,** which demonstrates distinct prototypes with similar representations.

suggesting that the error progressively decreases from 0.52 to 0.48 when increasing N from 1 to 4, and saturates at 4. Considering the computation time, we set $N = 3$ to strike the balance between performance and computational cost. The number of prototypes K plays a pivotal role in defining the central grouping points for motion features. We therefore investigate the variant of K in Table 3d.

Latent Synchronization. Next, we study our *Latent Synchronization* in Table 3e. With a standard setting without any prototype-feature corresponding (*i.e.*, None), the model reports 0.74 in final pass. After applying a vanilla fully-connected layer to update the feature, the error decreases to 0.73. Though inspiring, our proposed *Latent Synchronization* with carefully anchored prototypes yields advanced performance across all ablative methods (*i.e.*, 0.69).

Systemic Explainability. Finally, we investigate the prototype-feature corresponding map on optical flow in Fig. 5. The systemic explainability hinges on the **Prototypes**, which emerge through the integration of probability density estimation within our cross-attention prototyping layer. The recursively optimized prototypes encapsulate the most characteristic features of the motion patterns within their respective density centers. Through the visualization of feature correspondence estimation derived from the updated prototypes, we enhance the interpretability of the network and transparency of the model's decision-making process.

5. Conclusion

We propose Prototypical TransFormer (ProtoFormer), a unified solution for motion tasks. The motivation of integrating Transformer and prototype learning leads us to innovate conventional self-attention, and propose *Cross-Attention Prototyping*. Our *Latent Synchronization* further refines the feature representations via prototype-feature association. Comprehensive empirical results show that ProtoFormer enjoys elegant architectural design, superior performance and systemic explainability. As a whole, we conclude that the findings from this work impart essential understandings and necessitate further exploration within this realm.

Impact Statement

Ethical Aspects. We provide asset licenses and consents for the datasets we applied in our paper in supplementary material. All the datasets are publicly available for academic usage. Since our work does not involve data augmentation or the creation of new datasets, ethical concerns or biases within our proposed ProtoFormer are significantly minimized. We should also highlight that the introduction of prototypical learning offers a significant advantages in addressing photometric inconsistency, which in turn reduces the possible biases during training when lighting condition is restricted.

Future Societal Consequences. ProtoFormer introduces a universal understanding for motion tasks via prototypical learning, possessing strong performance gains over several state-of-the-art baselines. On positive side, our approach is valuable in various real-world applications (*e.g.*, autonomous driving (Cheng et al., 2023c; Prakash et al., 2021; Cheng et al., 2023b; Shao et al., 2023; Cheng et al., 2022; 2023a), robotics navigation (Dev et al., 1997; Lookingbill et al., 2007; Song et al., 2022)), benefited from its transparency and efficiency. Regarding potential negative social impacts, it is noteworthy that our ProtoFormer, akin to other discriminative classifiers, encounters challenges in addressing out-of-distribution/open-set problems (Liang et al., 2022a). Its utility in open-world scenarios should be further examined.

Acknowledgement

We thank Prof. Majid Rabbani for detailed feedback on drafts of the paper. This research was supported by the National Science Foundation under Grant No. 2242243 and the DEVCOM Army Research Laboratory under Contract W911QX-21-D-0001. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. DEVCOM Army Research Laboratory (ARL), U.S. Naval Research Laboratory (NRL) or the U.S. Government.

References

Balakrishnan, S., Wainwright, M. J., and Yu, B. Statistical guarantees for the em algorithm: From population to sample-based analysis. *Annals of Statistics*, 2017.

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016.

Bhat, G., Danelljan, M., Van Gool, L., and Timofte, R. Know your surroundings: Exploiting scene information for object tracking. In *ECCV*, 2020.

Bhat, S. F., Alhashim, I., and Wonka, P. Adabins: Depth estimation using adaptive bins. In *CVPR*, 2021.

Breiman, L. Random forests. *Machine learning*, 45:5–32, 2001.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012a.

Butler, D. J., Wulff, J., Stanley, G. B., and Black, M. J. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012b.

Cabon, Y., Murray, N., and Humenberger, M. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020.

Cai, Y., Lin, J., Hu, X., Wang, H., Yuan, X., Zhang, Y., Timofte, R., and Van Gool, L. Coarse-to-fine sparse transformer for hyperspectral image reconstruction. In *ECCV*, 2022.

Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., and Lu, H. Transformer tracking. In *CVPR*, 2021.

Chen, X., Peng, H., Wang, D., Lu, H., and Hu, H. Seqtrack: Sequence to sequence learning for visual object tracking. In *CVPR*, 2023.

Cheng, Z., Liang, J., Choi, H., Tao, G., Cao, Z., Liu, D., and Zhang, X. Physical attack on monocular depth estimation with optimal adversarial patches. In *ECCV*, 2022.

Cheng, Z., Choi, H., Feng, S., Liang, J. C., Tao, G., Liu, D., Zuzak, M., and Zhang, X. Fusion is not enough: Single modal attack on fusion models for 3d object detection. In *ICLR*, 2023a.

Cheng, Z., Choi, H., Liang, J., Feng, S., Tao, G., Liu, D., Zuzak, M., and Zhang, X. Fusion is not enough: single-modal attacks to compromise fusion models in autonomous driving. *arXiv preprint arXiv:2304.14614*, 2023b.

Cheng, Z., Liang, J., Tao, G., Liu, D., and Zhang, X. Adversarial training of self-supervised monocular depth estimation against physical-world attacks. *arXiv preprint arXiv:2301.13487*, 2023c.

Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., and Shen, C. Twins: Revisiting spatial attention design in vision transformers. *NeurIPS*, 2021.

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20:273–297, 1995.

Cui, Y., Han, C., and Liu, D. Collaborative multi-task learning for multi-object tracking and segmentation. *Journal on Autonomous Transportation Systems*, 1(2):1–23, 2024.

Danelljan, M., Bhat, G., Shahbaz Khan, F., and Felsberg, M. Eco: Efficient convolution operators for tracking. In *CVPR*, 2017.

Dev, A., Krose, B., and Groen, F. Navigation of a mobile robot on the temporal development of the optic flow. In *IROS*, 1997.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2018.

Dong, N. and Xing, E. P. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018.

Dong, Q., Cao, C., and Fu, Y. Rethinking optical flow from geometric matching consistent perspective. In *CVPR*, 2023.

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., and Brox, T. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *NeurIPS*, 2014.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., and Ling, H. Lasot: A high-quality benchmark for large-scale single object tracking. In *CVPR*, 2019.

Farhangi, A., Sui, N., Hua, N., Bai, H., Huang, A., and Guo, Z. Protoformer: Embedding prototypes for transformers. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 447–458. Springer, 2022.

Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018.

Gao, H., Cui, J., Ye, M., Li, S., Zhao, Y., and Zhu, X. Structure-preserving motion estimation for learned video compression. In *ACMMM*, 2022.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013.

Giese, M. A. and Poggio, T. Neural mechanisms for the recognition of biological movements. *Nature Reviews Neuroscience*, 2003.

Godard, C., Mac Aodha, O., and Brostow, G. J. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

Han, C., Wang, Q., Cui, Y., Cao, Z., Wang, W., Qi, S., and Liu, D. E^2vpt: An effective and efficient approach for visual prompt tuning. In *ICCV*, 2023a.

Han, C., Wang, Q., Cui, Y., Wang, W., Huang, L., Qi, S., and Liu, D. Facing the elephant in the room: Visual prompt tuning or full finetuning? In *ICLR*, 2024.

Han, D., Pan, X., Han, Y., Song, S., and Huang, G. Flatten transformer: Vision transformer using focused linear attention. In *ICCV*, 2023b.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al. A survey on vision transformer. *IEEE TPAMI*, 2022.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Hawkins, J. and Blakeslee, S. *On intelligence*. Macmillan, 2004.

Hu, Z., Lu, G., and Xu, D. Fvc: A new framework towards deep video compression in feature space. In *CVPR*, 2021.

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., and Liu, W. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, 2019.

Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K. C., Qin, H., Dai, J., and Li, H. Flowformer: A transformer architecture for optical flow. In *ECCV*, 2022.

Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., and Heming, J. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 2023.

Jaegle, A., Borgeaud, S., Alayrac, J.-B., Doersch, C., Ionescu, C., Ding, D., Koppula, S., Zoran, D., Brock, A., Shelhamer, E., et al. Perceiver io: A general architecture for structured inputs & outputs. *ICLR*, 2022.

Jetley, S., Romera-Paredes, B., Jayasumana, S., and Torr, P. Prototypical priors: From improving classification to zero-shot learning. *arXiv preprint arXiv:1512.01192*, 2015.

Jiang, S., Campbell, D., Lu, Y., Li, H., and Hartley, R. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, 2021.

Jiang, Z., Lin, Z., and Davis, L. Recognizing human actions by learning and matching shape-motion prototype trees. *IEEE TPAMI*, 2012.

Kepes, G. *Language of vision*. Courier Corporation, 1995.

Khalifa, A. B., Alouani, I., Mahjoub, M. A., and Amara, N. E. B. Pedestrian detection using a moving camera: A novel framework for foreground detection. *Cognitive Systems Research*, 2020.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., and Shah, M. Transformers in vision: A survey. *ACM Computing Surveys*, 54(10s):1–41, 2022.

Kim, W., Son, B., and Kim, I. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021.

Kondermann, D., Nair, R., Honauer, K., Krispin, K., Andrulis, J., Brock, A., Gussefeld, B., Rahimimoghaddam, M., Hofmann, S., Brenner, C., et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *CVPR Workshops*, 2016.

Lee, M., Cho, S., Lee, S., Park, C., and Lee, S. Unsupervised video object segmentation via prototype memory network. In *WACV*, 2023.

Li, G., Jampani, V., Sevilla-Lara, L., Sun, D., Kim, J., and Kim, J. Adaptive prototype learning and allocation for few-shot segmentation. In *CVPR*, 2021a.

Li, J., Huang, Q., Du, Y., Zhen, X., Chen, S., and Shao, L. Variational abnormal behavior detection with motion consistency. *IEEE TIP*, 2021b.

Liang, C., Wang, W., Miao, J., and Yang, Y. Gmmseg: Gaussian mixture based generative semantic segmentation models. *NeurIPS*, 2022a.

Liang, J., Wang, Y., Chen, Y., Yang, B., and Liu, D. A triangulation-based visual localization for field robots. *IEEE/CAA Journal of Automatica Sinica*, 9(6):1083–1086, 2022b.

Liang, J., Zhou, T., and Liu, D. Clustseg: Clustering for universal segmentation. In *ICML*, 2023.

Liang, J., Cui, Y., Wang, Q., Geng, T., Wang, W., and Liu, D. Clusterfomer: Clustering as a universal visual learner. *NeurIPS*, 2024.

Lin, T., Wang, Y., Liu, X., and Qiu, X. A survey of transformers. *AI Open*, 2022.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. In *ICLR*, 2020a.

Liu, Y., Zhang, X., Zhang, S., and He, X. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 2020b.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022.

Lookingbill, A., Rogers, J., Lieb, D., Curry, J., and Thrun, S. Reverse optical flow for self-supervised adaptive autonomous robot navigation. *IJCV*, 74:287–302, 2007.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *ICLR*, 2019.

Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., and Gao, Z. Dvc: An end-to-end deep video compression framework. In *CVPR*, 2019.

Lu, Y., Wang, Q., Ma, S., Geng, T., Chen, Y. V., Chen, H., and Liu, D. Transflow: Transformer as flow learner. In *CVPR*, 2023.

Lu, Y., Liu, D., Wang, Q., Han, C., Cui, Y., Cao, Z., Zhang, X., Chen, Y. V., and Fan, H. Promotion: Prototypes as motion learners. In *CVPR*, 2024.

Luo, A., Yang, F., Li, X., and Liu, S. Learning optical flow with kernel patch attention. In *CVPR*, 2022a.

Luo, A., Yang, F., Luo, K., Li, X., Fan, H., and Liu, S. Learning optical flow with adaptive graph reasoning. In *AAAI*, 2022b.

Ma, F., Shou, M. Z., Zhu, L., Fan, H., Xu, Y., Yang, Y., and Yan, Z. Unified transformer tracker for object tracking. In *CVPR*, 2022.

Ma, X., Zhou, Y., Wang, H., Qin, C., Sun, B., Liu, C., and Fu, Y. Image as set of points. In *ICLR*, 2023. URL https://openreview.net/forum?id=awnvqZja69.

Mantini, D., Corbetta, M., Romani, G. L., Orban, G. A., and Vanduffel, W. Data-driven analysis of analogous brain networks in monkeys and humans during natural vision. *Neuroimage*, 2012.

Marathe, A., Walambe, R., and Kotecha, K. Evaluating the performance of ensemble methods and voting strategies for dense 2d pedestrian detection in the wild. In *ICCV*, 2021.

Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

Meyer, G. E., Neto, J. C., Jones, D. D., and Hindman, T. W. Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images. *Computers and Electronics in Agriculture*, 2004.

Muller, M., Bibi, A., Giancola, S., Alsubaihi, S., and Ghanem, B. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. In *ECCV*, 2018.

Nam, H. and Han, B. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, 2016.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

Patil, V., Sakaridis, C., Liniger, A., and Van Gool, L. P3depth: Monocular depth estimation with a piecewise planarity prior. In *CVPR*, 2022.

Plato. *Cratylus*. Plato, 402 BC.

Prakash, A., Chitta, K., and Geiger, A. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, 2021.

Qin, Z., Han, C., Wang, Q., Nie, X., Yin, Y., and Xiankai, L. Unified 3d segmenter as prototypical classifiers. *NeurIPS*, 2023.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 2020.

Ranjan, A. and Black, M. J. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Shao, H., Wang, L., Chen, R., Li, H., and Liu, Y. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, 2023.

Shen, S., Kerofsky, L., and Yogamani, S. Optical flow for autonomous driving: Applications, challenges and improvements. *arXiv preprint arXiv:2301.04422*, 2023.

Shi, X., Huang, Z., Li, D., Zhang, M., Cheung, K. C., See, S., Qin, H., Dai, J., and Li, H. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *CVPR*, 2023.

Simon, H. A. and Newell, A. Human problem solving: The state of the theory in 1970. *American psychologist*, 26(2): 145, 1971.

Smith, J. D. and Minda, J. P. Distinguishing prototype-based and exemplar-based processes in dot-pattern category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 2002.

Song, Y., Wen, J., Liu, D., and Yu, C. Deep robotic grasping prediction with hierarchical rgb-d fusion. *IJCAS*, 20(1): 243–254, 2022.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021.

Sui, X., Li, S., Geng, X., Wu, Y., Xu, X., Liu, Y., Goh, R., and Zhu, H. Craft: Cross-attentional flow transformer for robust optical flow. In *CVPR*, 2022.

Sun, D., Yang, X., Liu, M.-Y., and Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

Sun, D., Vlasic, D., Herrmann, C., Jampani, V., Krainin, M., Chang, H., Zabih, R., Freeman, W. T., and Liu, C. Autoflow: Learning a better training set for optical flow. In *CVPR*, 2021.

Teed, Z. and Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

Tudor Ionescu, R., Smeureanu, S., Alexe, B., and Popescu, M. Unmasking the abnormal events in video. In *ICCV*, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *NeurIPS*, 2017.

Vattani, A. K-means requires exponentially many iterations even in the plane. In *Annual Symposium on Computational Geometry*, 2009.

Wang, H., Zhu, Y., Adam, H., Yuille, A., and Chen, L.-C. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021a.

Wang, K., Liew, J. H., Zou, Y., Zhou, D., and Feng, J. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019.

Wang, M., Yang, G.-Y., Lin, J.-K., Zhang, S.-H., Shamir, A., Lu, S.-P., and Hu, S.-M. Deep online video stabilization with multi-grid warping transformation learning. *IEEE TIP*, 2018.

Wang, N., Zhou, W., Wang, J., and Li, H. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, 2021b.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021c.

Wang, W., Liang, J., and Liu, D. Learning equivariant segmentation with instance-unique querying. *NeurIPS*, 2022.

Wang, W., Han, C., Zhou, T., and Liu, D. Visual recognition with deep nearest centroids. In *ICLR*, 2023.

Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., and Xia, H. End-to-end video instance segmentation with transformers. In *CVPR*, 2021d.

Wang, Z., Zhao, H., Li, Y.-L., Wang, S., Torr, P., and Bertinetto, L. Do different tracking tasks require different appearance models? *NeurIPS*, 2021e.

Xiong, M., Zhang, Z., Zhong, W., Ji, J., Liu, J., and Xiong, H. Self-supervised monocular depth and visual odometry learning with scale-consistent geometric constraints. In *IJCAI*, 2021.

Xu, H., Guo, M., Nedjah, N., Zhang, J., and Li, P. Vehicle and pedestrian detection algorithm based on lightweight yolov3-promote and semi-precision acceleration. *IEEE Transactions on Intelligent Transportation Systems*, 2022a.

Xu, H., Zhang, J., Cai, J., Rezatofighi, H., and Tao, D. Gmflow: Learning optical flow via global matching. In *CVPR*, 2022b.

Xu, W., Xian, Y., Wang, J., Schiele, B., and Akata, Z. Attribute prototype network for zero-shot learning. *NeurIPS*, 2020.

Yang, B., Liu, C., Li, B., Jiao, J., and Ye, Q. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 2020.

Yang, J., Liu, J., Xu, N., and Huang, J. Tvt: Transferable vision transformer for unsupervised domain adaptation. In *WACV*, 2023.

Yin, W., Liu, Y., Shen, C., and Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In *ICCV*, 2019.

Yu, Q., Wang, H., Qiao, S., Collins, M., Zhu, Y., Adam, H., Yuille, A., and Chen, L.-C. k-means mask transformer. *ECCV*, 2022.

Zhai, M., Xiang, X., Lv, N., Ali, S. M., and Saddik, A. E. Skflow: Optical flow estimation using selective kernel networks. *NeurIPS*, 2022.

Zhang, F., Woodford, O. J., Prisacariu, V. A., and Torr, P. H. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021.

Zhang, N., Nex, F., Vosselman, G., and Kerle, N. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *CVPR*, 2023.

Zhang, Z., Peng, H., Fu, J., Li, B., and Hu, W. Ocean: Object-aware anchor-free tracking. In *ECCV*, 2020.

Zhao, M. and Ling, Q. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE TIP*, 2020.

Zhao, S., Zhao, L., Zhang, Z., Zhou, E., and Metaxas, D. Global matching with overlapping attention for optical flow estimation. In *CVPR*, 2022.

Zhao, W., Liu, S., Shu, Y., and Liu, Y.-J. Towards better generalization: Joint depth-pose learning without posenet. In *CVPR*, 2020.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021.

Zheng, Z., Nie, N., Ling, Z., Xiong, P., Liu, J., Wang, H., and Li, J. Dip: Deep inverse patchmatch for high-resolution optical flow. In *CVPR*, 2022.

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., and Feng, J. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021a.

Zhou, J. T., Du, J., Zhu, H., Peng, X., Liu, Y., and Goh, R. S. M. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 2019.

Zhou, T., Wang, W., Konukoglu, E., and Van Gool, L. Rethinking semantic segmentation: A prototype view. In *CVPR*, 2022.

Zhou, Z., Fan, X., Shi, P., and Xin, Y. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *ICCV*, 2021b.

Zhu, C., Ping, W., Xiao, C., Shoeybi, M., Goldstein, T., Anandkumar, A., and Catanzaro, B. Long-short transformer: Efficient transformers for language and vision. *NeurIPS*, 2021.

SUMMARY OF THE APPENDIX

This supplementary contains additional experimental results and discussions of our ICML 2024 submission: *Prototypical Transformer as Unified Motion Learners*, organized as follows:

- §S1 provides detailed training configuration and testing configuration on optical flow and scene depth estimation.
- §S2 provides more qualitative results and comparisons for optical flow and scene depth estimation.
- §S3 offers comprehensive ablation studies on scene depth estimation.
- §S4 provides detailed *Proof* on the guarantees of EM convergence.
- §S5 includes additional experiments on object tracking.
- §S6 includes additional experiments on video stabilization.
- §S7 discusses the Pseudo codes.

S1. Training and Testing Configuration

Our training methodology for ProtoFormer was adapted from established optical flow training protocols (Jiang et al., 2021; Huang et al., 2022). Initially, the model underwent a pre-training phase on the FlyingChairs dataset (Dosovitskiy et al., 2015), followed by an additional 120, 000 iterations on the FlyingThings dataset (Mayer et al., 2016), a procedure we denote as "C+T." Subsequently, the model underwent fine-tuning on a combined dataset encompassing FlyingThings (Mayer et al., 2016), Sintel (Butler et al., 2012b), KITTI-2015 (Geiger et al., 2013), and HD1K (Kondermann et al., 2016), referred to as "C+T+S+K+H". To optimize performance specifically for the KITTI-2015 benchmark (Geiger et al., 2013), we conducted a further fine-tuning phase on the KITTI-2015 dataset for 50, 000 iterations. The training employed AdamW (Loshchilov & Hutter, 2019) optimizer and a one-cycle learning rate scheduler, with the peak learning rate set at 2.5×10^{-4} for the FlyingChairs dataset and 1.25×10^{-4} for the other datasets. Recognizing the sensitivity of transformer positional encodings to variations in image size, we adopted an image processing approach akin to that used in Perceiver IO (Jaegle et al., 2022). This involved cropping image pairs for flow estimation and subsequently tiling them to reconstruct complete flows. For depth prediction, we adhere to the architecture and configurations analogous to those employed for optical flow, as delineated in the respective headers. We initially adopt the VKITTI (Cabon et al., 2020) as a pretraining, and subsequently canonical Eigen split (Eigen et al., 2014) and MPI Sintel dataset (Butler et al., 2012b) to refine the model through fine-tuning, noted for its distinct edges and diverse motion intensities. No additional data augmentation was used for the testing of all tasks.

Loss Function. For optical flow, A sequence loss is utilized for the training, which is defined over the sequence of flow predictions. For depth prediction, square root of the scale invariant logarithmic loss (SILog) is utilized for the training.

Method	LaSOT		TrackingNet	
	Success ↑	Precision ↑	Success ↑	Precision ↑
SiamFC (Bertinetto et al., 2016)	33.6	33.9	57.1	66.3
MDNet (Nam & Han, 2016)	39.7	37.3	60.6	56.6
ECO (Danelljan et al., 2017)	32.4	30.1	55.4	49.2
KYS (Bhat et al., 2020)	55.4	55.8	74.0	68.8
Ocean (Zhang et al., 2020)	52.6	52.6	70.3	68.8
TrDiMP (Wang et al., 2021b)	63.9	61.4	78.4	73.1
TransT (Chen et al., 2021)	64.9	69.0	81.4	80.3
UniTrack (Wang et al., 2021e)	35.1	32.6	59.1	51.2
UTT (Ma et al., 2022)	64.6	67.2	79.7	77.0
UTT + Ours	64.8	67.4	80.0	77.2

Table S1: Quantitative results on LaSOT and TrackingNet datasets. We are able to achieve competitive performance over state-of-the-art methods.

S2. More Qualitative Results

We show more qualitative results on the main tasks, optical flow and scene depth, in Fig. S2 and Fig. S3.

S3. Ablation Studies on Scene Depth Estimation

In §4.3, we ablate comprehensively under the optical flow setting. In this section, we further report ablation studies on scene depth estimation for completeness.

Key Components Analysis. We study the two major components of ProtoFormer: *Cross-Attention Prototyping* (§3.2.1) and *Latent Synchronization* (§3.2.2). Same to our paper, the `Base` model is designed without considering prototype updating and prototype-feature assignment. In Table S2a, `Base` reaches 0.074 in Abs Rel and 2.835 in RMSE. Adding *Cross-Attention Prototyping* gets substantial improvements (*i.e.*, 0.074 → 0.067 in Abs Rel). Considering *Latent Synchronization* brings a performance gain (*i.e.*, 0.074 → 0.071). Finally, the integration of these two techniques reaches peak performance as ProtoFormer, which is consistent to the tendency in our paper.

Cross-Attention Prototyping. We also study the efficacy of *Cross-Attention Prototyping* design by comparing to different updating methods. For efficient and effective perspectives, our *Cross-Attention Prototyping* outperforms competitive methods (see Table S2b). We further study the iteration step N in Table S2c, suggesting that when increasing N from 1 to 4, the error progressively decreases from 0.070 to 0.061, and almost saturates at 4. Considering the computa-

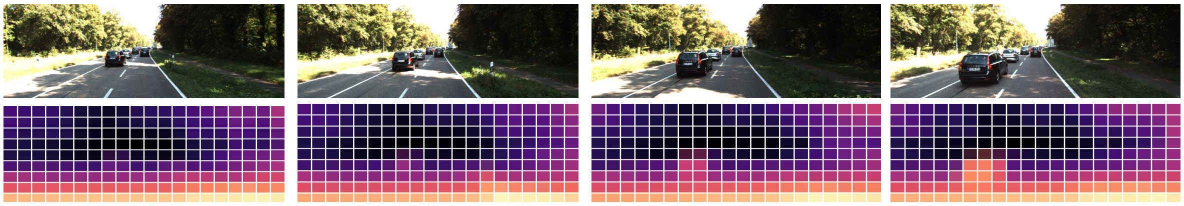

Figure S1: **Visualization of proto-feature mapping in depth.** The map shows distinct prototypes with similar representations, indicating straightforward explainability.

Table S2: **Ablative studies** on scene depth (see §S3).

Algorithm Component	#Params	Abs Rel	RMSE
`Base`	9.63M	0.074	2.835
+ Cross-Attention Prototyping	11.57M	0.067	2.742
+ Latent Synchronization	10.26M	0.071	2.819
ProtoFormer (**All included**)	11.90M	0.062	2.716

(a) Key Component Analysis

Variant Prototype Updating Strategy	#Params	Abs Rel	RMSE
Cosine Similarity	10.28M	0.066	2.775
Vanilla Cross-Attention (Vaswani et al., 2017)	14.88M	0.066	2.767
Criss Cross-Attention (Huang et al., 2019)	14.56M	0.065	2.750
K-Means (Yu et al., 2022)	11.81M	0.063	2.743
Cross-Attention Prototyping	11.90M	0.062	2.716

(b) *Cross-Attention Prototyping*

#Iterations (N)	#Params	Abs Rel	RMSE
1	11.90M	0.070	2.801
2		0.065	2.737
3		0.062	2.716
4		0.061	2.713

(c) Number of Iterations

#Prototypes (K)	#Params	Abs Rel	RMSE
10	8.95M	0.070	2.775
50	9.78M	0.067	2.734
100	11.90M	0.062	2.716
200	14.21M	0.064	2.720

(d) Number of Prototypes

Latent Synchronization	#Params	Abs Rel	RMSE
None	11.27M	0.067	2.732
Vanilla FC Layer	11.64M	0.064	2.724
FC w/ Similarity (Ma et al., 2023)	11.76M	0.063	2.718
Ours	11.90M	0.062	2.716

(e) *Latent Synchronization*

tion time in iterations, we set $N = 3$ to strike the optimal balance between performance and computation. Consistent to our paper, we investigate the variant of K (*i.e.*, number of prototypes) in Table S2d. We select the preferred setting at $K = 100$.

Latent Synchronization. We further study our *Latent Synchronization* in Table S2e. With a standard setting without any feature-prototype corresponding, the model achieves 0.067 in Abs Rel. Applying a vanilla fully-connected layer increases the performance to 0.064. Though inspiring, our proposed *Latent Synchronization* with carefully anchored prototypes yields advanced performance across all ablative methods (*i.e.*, 0.062).

S4. Proof on the Guarantees of EM Convergence

We first introduce the regularity condition, including a Euclidean ball of radius r around the fixed point $\hat{\theta}$, set as:

$$\mathcal{B}_2(r;\theta) := \Big\{\theta \in \Omega \mid ||\theta - \hat{\theta}||_2 \leq r\Big\}. \tag{8}$$

For simplicity, we define $\hat{\theta} = \arg\max_{\theta} U(\cdot|\theta)$ while we have introduced our *unified motion solution* in Eq. 2.

For First-order Stability (FOS), the functions $U(\cdot|\theta)$ satisfy condition FOS(γ) over $\mathcal{B}_2(r;\theta)$ if:

$$||\nabla U(M(\cdot|\hat{\theta})) - \nabla U(M(\cdot|\theta))||_2 \leq \gamma ||\theta - \hat{\theta}||_2, \tag{9}$$

for all $\theta \in \mathcal{B}_2(r;\hat{\theta})$. With radius $\gamma = 0$, the condition of Eq. 9 is always held at the fixed point $\hat{\theta}$. Extend further, by allowing for an always positive γ, given fixed point $\hat{\theta}$, Eq. 9 would always hold in a local neighborhood $\mathcal{B}_2(r;\hat{\theta})$.

Under these conditions, we further guarantee the EM operator to be locally contraction.

Theorem 1. For $\gamma > 0$, and having $0 \leq \gamma \leq \lambda$, suppose the function $U(\cdot|\hat{\theta})$ is λ-strongly concave and FOS(γ) holds for $\mathcal{B}_2(r;\hat{\theta})$, we have the EM operator M is contractive over $\mathcal{B}_2(r;\hat{\theta})$ as:

$$||M(\theta) - \hat{\theta}||_2 \leq \frac{\gamma}{\lambda}||\theta - \hat{\theta}||_2 \tag{10}$$

for all $\theta \in \mathcal{B}_2(r;\hat{\theta})$. Intuitively, we can conduct that for any initial point $\theta^{(0)} \in \mathcal{B}_2(r;\hat{\theta})$, $\big\{\theta^{(n)}\big\}_{n=0}^{\infty}$ exhibits linear convergence. Formally, we have:

$$||\theta^{(n)} - \hat{\theta}||_2 \leq (\frac{\gamma}{\lambda})^n ||\theta^0 - \hat{\theta}||_2, \tag{11}$$

for all $n \in \{1, 2, ..., N\}$. Here we define n in a finite set for intuitive reference below.

Acknowledging the preliminary conditions in ***Theorem 1***, we further present the proof for ***Proposition 1*** below.

Proof. For any iteration $n \in \{1, 2, ..., N\}$, we have:

$$||M_{m/N}(\theta^{(n)}) - M(\theta^{(n)})||_2 \leq \epsilon_M(\frac{m}{N}, \frac{\delta}{N}), \tag{12}$$

with probability at least $1 - \frac{\delta}{N}$. Consequently, by a union bound over N, Eq. 12 holds uniformly with probability at

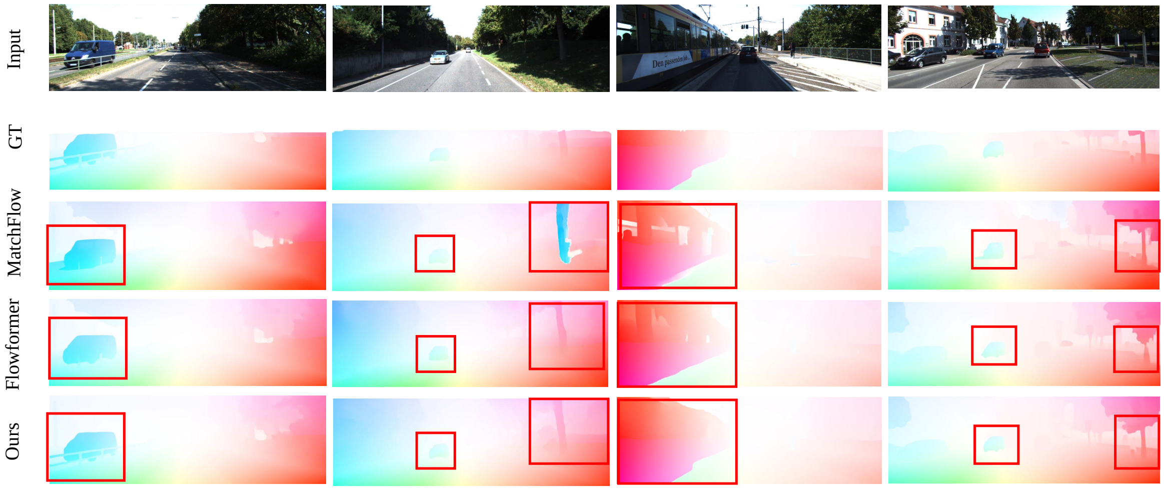

Figure S2: **More qualitative comparison on the KITTI test set.** The red boxes highlight the regions compared. Our method estimates more consistent and detailed flows.

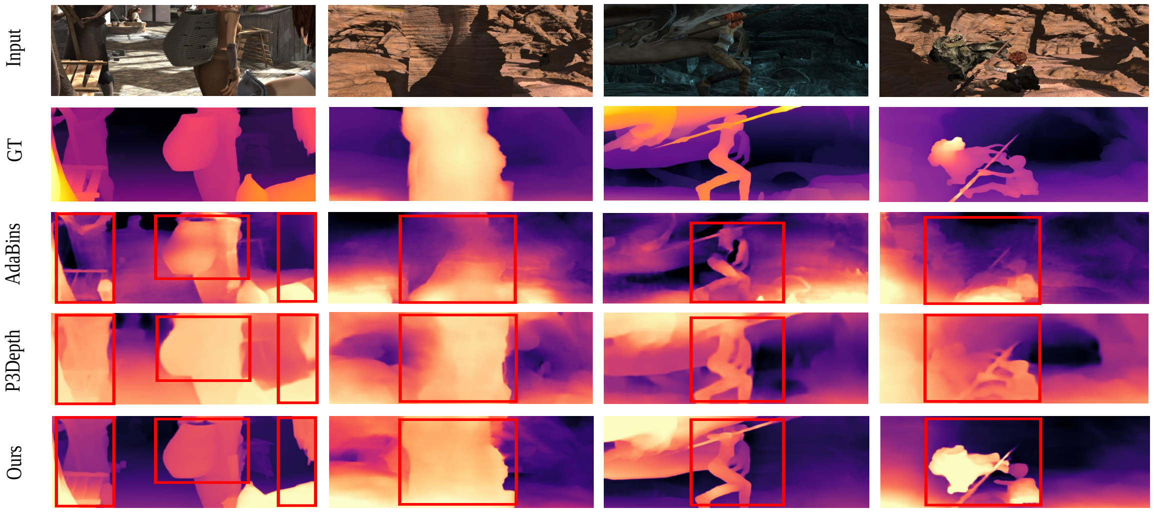

Figure S3: **More qualitative comparison on the Sintel test set.** The red boxes highlight the regions compared. Our method contributes to clearer depths without being affected by shadows or occlusions.

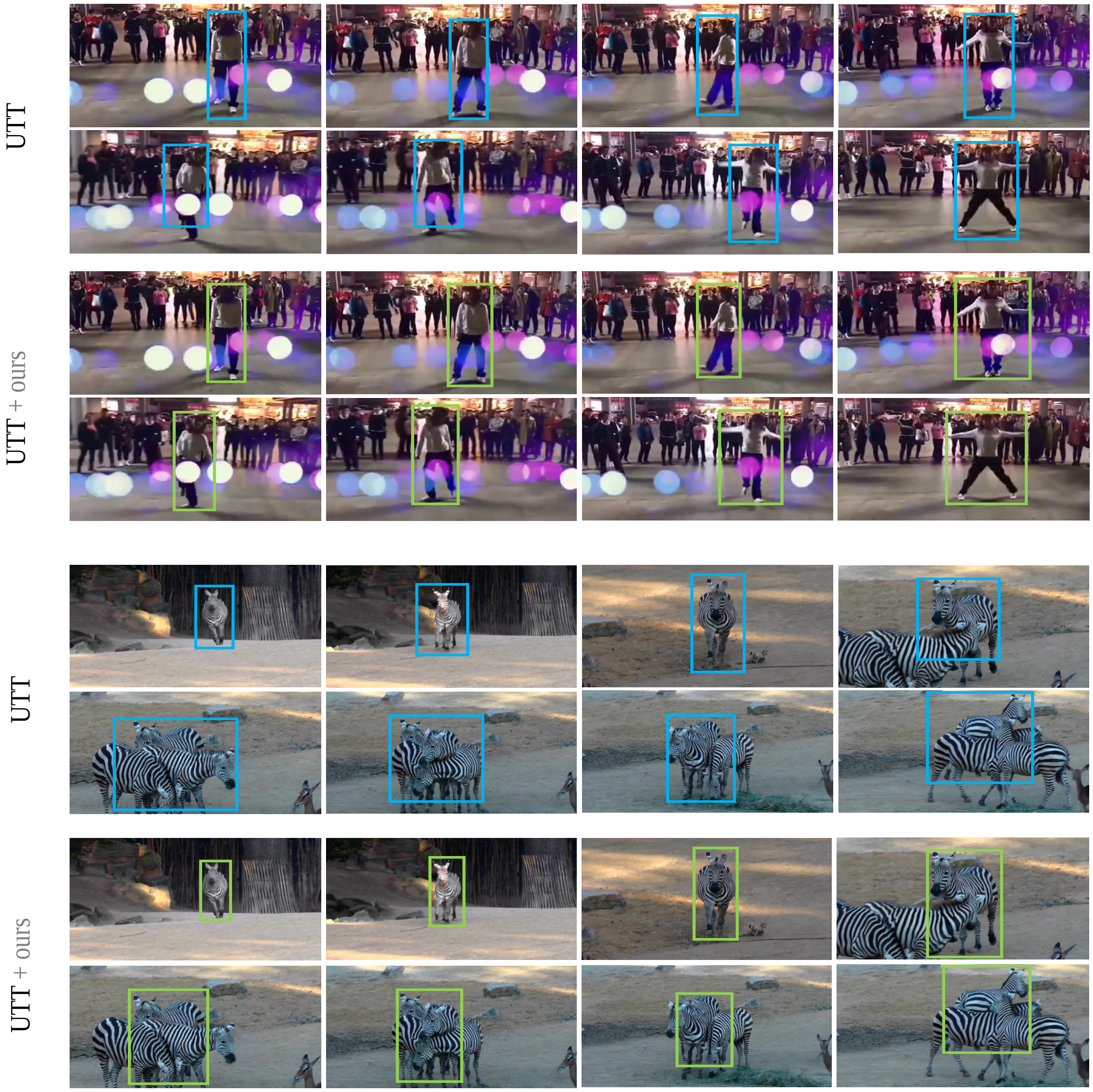

Figure S4: **Qualiative results on object tracking on LaSOT dataset.** With our enhanced method, the original ambiguous tracking due to illuminant changes and heavy occlusion becomes more accurate.

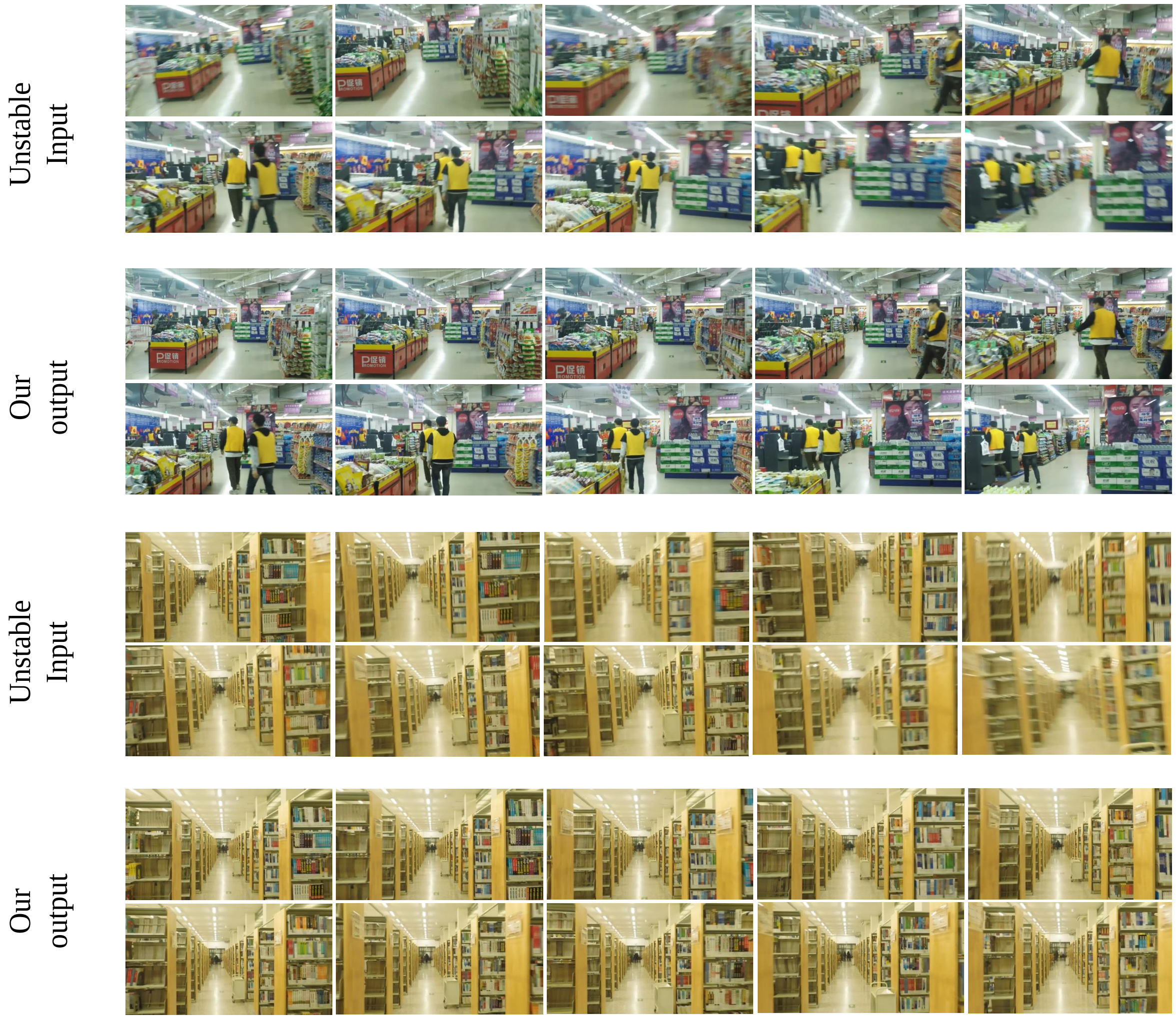

Figure S5: **Qualiative results on video stabilization.** With our method enhanced, the original unstable video frames become much smoother and clearer.

least $1 - \delta$. It suffices to show that

$$||\theta^{(n+1)} - \hat{\theta}||_2 \leq \kappa ||\theta^{(n)} - \hat{\theta}||_2 + \epsilon_M(\frac{m}{N}, \frac{\delta}{N}), \quad (13)$$

for each iteration $n \in \{1, 2, ..., N\}$. When Eq. 13 holds, we can iteratively show that:

$$\begin{aligned} ||\theta^{(n)} - \hat{\theta}||_2 &\leq \kappa ||\theta^{(n-1)} - \hat{\theta}||_2 + \epsilon_M(\frac{m}{N}, \frac{\delta}{N}) \\ &\leq \kappa \left\{ \kappa ||\theta^{(n-2)} - \hat{\theta}||_2 + \epsilon_M(\frac{m}{N}, \frac{\delta}{N}) \right\} + \epsilon_M(\frac{m}{N}, \frac{\delta}{N}) \\ &\leq \kappa^{(n)} ||\theta^{(0)} - \hat{\theta}||_2 + \left\{ \sum_{n=0}^{n-1} \kappa^n \right\} \epsilon_M(\frac{m}{N}, \frac{\delta}{N}) \\ &\leq \kappa^n ||\theta^{(0)} - \hat{\theta}||_2 + \frac{1}{1-\kappa} \epsilon_M(\frac{m}{N}, \frac{\delta}{N}). \end{aligned} \quad (14)$$

The final step follows by summing the geometric series.

Thus, we need to prove Eq. 13 via induction on the iteration number. Start with $n = 1$, we have:

$$\begin{aligned} &||\theta^{(1)} - \hat{\theta}||_2 = ||M_{m/N}(\theta^{(0)}) - M(\theta^{(n)})||_2 \\ &(\textbf{I}) \leq ||M(\theta^{(0)}) - \hat{\theta}||_2 + ||M_{m/N}(\theta^{(0)}) - M(\theta^{(0)})||_2 \\ &(\textbf{II}) \leq \kappa ||M(\theta^{(0)}) - \hat{\theta}||_2 + \epsilon_M(\frac{m}{N}, \frac{\delta}{N}), \end{aligned} \quad (15)$$

where step **(I)** follows by the triangle inequality, step **(II)** follows from Eq. 13, and the contractivity of the operator applied to $\theta^{(0)} \in \mathcal{B}_2(r; \hat{\theta})$. In the induction from $n \to n+1$, suppose that $||\theta^{(n)} - \hat{\theta}||_2 \leq r$, and the bound holds (*i.e.*, Eq. 13) for iteration n. The same augment then implies that the bound from Eq. 13 also holds for iteration $n + 1$, and that $||\theta^{(n+1)} - \hat{\theta}||_2 \leq r$, thus completing the proof. □

Method	Distortion Value ↑	Stability Score ↑
StabNet (Wang et al., 2018)	0.83	0.75
StabNet + Ours	0.85 (0.02 ↑)	0.80 (0.05 ↑)
PWStableNet (Zhao & Ling, 2020)	0.79	0.80
PWStableNet + Ours	0.82 (0.03 ↑)	0.83 (0.03 ↑)

Table S3: Quantitative results on DeepStab dataset. We are able to achieve competitive performance over current methods.

S5. Experiments on Object Tracking

To further support our proposed ProtoFormer as a general solution to various tasks, we extend our design to object tracking following (Ma et al., 2022). Intuitively, we follow (Ma et al., 2022) and integrate the encoder and decoder into one object transformer where we replace the self-attention into our proposed cross-attention prototyping.

We evaluate our method on the testing splits of LaSOT (Fan et al., 2019) and TrackingNet (Muller et al., 2018) following common practices (Ma et al., 2022; Chen et al., 2023). Specifically, LaSOT (Fan et al., 2019) includes $1,400$ sequences: $1,120$ for training and 280 for testing, respectively. TrackingNet (Muller et al., 2018) contains $30K$ sequences with 511 sequences for testing. Success and Precision metrics are applied for performance evaluation. We follow the same training schedule as (Ma et al., 2022) for fairness. As seen in Table S1, our approach achieves competitive results to current methods (*e.g.*, 0.2% and 0.2% higher than UTT on Success and Precision on LaSOT dataset, respectively). Qualitative results are shown in Fig S4.

Algorithm 1 Pseudo-code of *Cross-attention Prototyping* in a PyTorch-like style.

```
"""
feats: output feature embeddings from regular
    projection, shape: (batch_size, channels,
    height, width)
P_0: initial cluster centers by adaptive pooling
    from the features, shape: (batch_size,
    num_clusters, dimension)
P: cluster centers, shape: (batch_size,
    num_clusters, dimension)
N: iteration number for recursive prototyping
    layer
"""

# One-step cross-attention prototyping in Eq.5
def one_prototyping_layer(Q, K, V):

    # E-step
    output = torch.matmul(Q, K.transpose(-2, -1))
    M = torch.nn.functional.softmax(output, dim =
        -2)

    # M-step
    P = torch.matmul(M, V)

    return P

# Iteratively cross-attention prototyping layer
def Cross_Attention_Prototyping(feats, P_0, N):

    Q = nn.Linear(P_0)
    K = nn.Linear(feats)
    V = nn.Linear(feats)
    P = P_0 + one_prototyping_layer(Q, K, V)

    for _ in range(N - 1):
        Q = nn.Linear(P)
        P = P + one_prototyping_layer(Q, K, V)

    return P
```

S6. Experiments on Video Stabilization

We further evaluate our method on video-based downstream task – video stabilization, following common training configurations from (Wang et al., 2018; Zhao & Ling, 2020; Lu et al., 2023). Specifically, DeepStab (Wang et al., 2018) contains 61 pairs of synchronized videos with diverse camera movements. Distortion Value and Stability Score are applied for performance evaluation. In Table S3, we report the results comparing to competitive methods (*i.e.*, StabNet (Wang et al., 2018), PWStableNet (Zhao & Ling, 2020)). Specifically, we follow TransFlow (Lu et al., 2023), aggregating the learned features from TransFlow's encoder

Algorithm 2 Pseudo-code of *Latent Synchronization* in a PyTorch-like style.

```
"""
feats: output feature embeddings from regular
    projection, shape: (batch_size, channels,
    height, width)
P: prototypes, shape: (batch_size, num_prototypes,
     dimension)
"""

# Latent sychronization in Eq.6
def latent_sychronization(feats, P):

   max_value, max_index = similarity(feats, P).
       max(dim = 1, keepdim = True)
   mask = torch.zeros_like(similarity(feats, P))
   mask.scatter_(1, max_index, 1.)

   Q = nn.Linear(feats)
   K = nn.Linear(P)
   V = nn.Linear(P)

   feats += FFN(attention_layer(Q, K, V,
       attn_mask = mask))

   return feats
```

(*i.e.*, replacing the origin attention with our proposed cross-attention prototyping and latent synchronization) and the original encoder together for the later regressor. Significant performance boost can be observed in both Distortion Value and Stability Score. Qualitative results are shown in Fig S5.

S7. Pseudo-codes

ProtoFormer is implemented in Pytorch (Paszke et al., 2019). Experiments are conducted on eight NVIDIA A100-40GB GPUs. We provide the pseudo codes of our proposed ProtoFormer *Cross-Attention Prototyping* in Algorithm 1 and *Latent Synchronization* in Algorithm 2.