# Video Instance Segmentation by Weighted Structure Inference

Anonymous Author(s)
Submission Id: 1642

## Abstract

Video instance segmentation presents significant challenges in complex and dynamic environments, where instances experience progressive occlusion, either from objects obstructing each other or due to changes in the camera's viewpoint. Current state-of-the-art methods rely on memory bank mechanisms, but we still look forward to new paradigms that have the ability to capture and utilize structural information, the ability to model complex relationships, and the flexibility to adapt to dynamic scenarios. To this end, we propose the Weighted Structure Inference method for Video Instance Segmentation. We build on high-order structural relationships by constructing hypergraphs for each video frame, enabling the capture of complex interactions that go beyond traditional pairwise methods. To model intricate dynamics, we introduce Weighted Sheaf Hypergraph Convolution, which enhances the hierarchical and structural information embedded in the hypergraph. Furthermore, we ensure spatio-temporal consistency by employing a dynamic inference mechanism based on Weighted Sliced Wasserstein distance to compare structural features across adjacent frames. Our method preserves the topological characteristics of occlusion instances and improves the reliability of instance tracking across frames. Experimental results demonstrate that our method outperforms existing parallel frameworks in both Video Instance and Panoptic Segmentation tasks. The Code will be released.

## CCS Concepts

• **Computing methodologies** → **Activity recognition and understanding**.

## Keywords

Video Instance Segmentation, Weighted Structure Inference, Progressive Occlusion, Sheaf Hypergraph, Sliced Wasserstein

## 1 Introduction

In the realm of computer vision, spatial intelligence and embodied perception are gaining increasing attention due to their potential to enhance a system's ability to understand and interact with complex environments [5]. This study focuses on a fundamental aspect of this field: Video Instance Segmentation (VIS). Video instance segmentation aims to accurately segment all target instances within a video, which is crucial for applications such as autonomous driving, video editing, and surveillance [45]. Accurate instance segmentation enables systems to distinguish between individual objects, track their movements, and understand their interactions, making it indispensable in dynamic environments. Particularly in complex, dynamic open scenes, the background plays a significant role in segmentation tasks, as it is highly variable and can significantly contribute to the overall scene complexity [43].

Video instance segmentation faces numerous challenges, particularly due to **Progressive Occlusion**. As shown in Fig.1, progressive occlusion can be caused by both instance occlusion, where instances block each other, and camera occlusion, where instances move in and out of the camera's view. These types of occlusion complicate segmentation by making parts of the instances invisible. Maintaining spatio-temporal consistency across frames is crucial for accurate instance tracking in such cases. Current state-of-the-art methodologies predominantly rely on memory banks or attention-based mechanisms to address these challenges. For instance, approaches such as VISAGE [21] leverage appearance-object relationships, while TCOVIS [22] employs dual local-global matching strategies. Another notable method, CTVIS [40], constructs identity matrices to explicitly model the gradual appearance and disappearance of objects over time. Though effective and mainstream, these methods leave room for new paradigms. We still look forward to the emergence of new paradigms that can provide new perspectives for analyzing the motion patterns of instances and their spatiotemporal relationships, which may improve the performance and robustness of video instance segmentation tasks.

**Figure 1: Progressive occlusion can be further subdivided into two categories: object occlusion, which is formed by objects occluding each other; and camera occlusion, which is caused by changes in the camera's viewpoint or by objects moving in and out of the frame. Masking of externally exposed visual parts (blue boxes) is not sufficient to characterize the whole object (yellow boxes).**

With this in mind, we draw inspiration from the inherent structural richness of videos [30]. Our goal is to develop a novel paradigm that embodies three key properties at the same time: the ability to capture and utilize structural information, the ability to model complex relationships, and the flexibility to adapt to dynamic scenes. In the task of video instance segmentation, capturing and utilizing structural information can accurately locate the position and form of instances in space and time for *camera occlusion*, modeling complex relationships can effectively handle *interactions and occlusions between instances*, and the flexibility to adapt to dynamic scenes ensures that the model remains stable and efficient in complex and changing environments, thereby comprehensively improving the performance and robustness of video instance segmentation.

In this paper, we propose the Weighted Structure Inference (WSInfer) method to address these challenges by leveraging high-order

structural relationships between instances in adjacent frames. WSInfer employs a decoupling strategy, whereby a hypergraph is constructed from instance queries generated by a segmenter. To capture complex dynamics, we introduce weighted sheaf hypergraph convolution, which leverages the concept of cellular sheaves [26] to model hierarchical and structural information, allowing for the extraction of enriched structural features. Additionally, we achieve instance spatio-temporal consistency and alleviate the challenge of progressive occlusion by using a dynamic inference mechanism based on Weighted Sliced Wasserstein distance to compare structural features across adjacent frames.

Our contributions are organized into three main aspects: **(I) Weighted Temporal Consistency for Occlusion:** To handle progressive occlusion, WSInfer introduces an importance-weighted strategy that emphasizes key structural contributions to enhance feature representation. First, we apply hypergraph convolution with weighted hyperedges to extract enriched structural features, highlighting the impact of significant interactions. Next, we use a weighted Sliced Wasserstein Distance metric to ensure spatio-temporal consistency across adjacent frames. This dual weighting mechanism enhances the framework's ability to address complex occlusions, resulting in improved segmentation performance in dynamic and occlusion-heavy scenes. **(II) Hypergraph Convolution for Modeling Complex Dynamics:** We introduce weighted hypergraph convolution based on cellular sheaves to capture local higher-order subtleties. The cellular sheaves enrich the hypergraph nodes and edges with richer hierarchical and structural information, enabling a more accurate representation of the intricate relationships between instances. By integrating these subtleties into the hypergraph Laplacian operator, we ensure the convolution process captures hidden higher-order structures, facilitating effective feature propagation and aggregation within the hypergraph. **(III) Preserving Spatio-Temporal Consistency through Dynamic Inference:** To preserve spatio-temporal consistency across frames, we employ a dynamic inference mechanism that uses weighted Sliced Wasserstein distance to compare structural features across adjacent frames. Our method maintains the hypergraph's structural features while accurately capturing associative differences between instances. By preserving these structural invariants, instances undergoing occlusion are accurately associated across frames, thereby ensuring reliable instance tracking.

Incorporating these innovations, we present a flexible and efficient framework in §4, achieving remarkable results on widely recognized VIS and VPS tasks in §5.3. Accompanied by a series of comprehensive ablation studies in §5.4, our extensive experiments confirm the effectiveness and robustness of our proposed weighted structure inference method in addressing the challenges of video instance segmentation and panoptic segmentation.

## 2 Related Works

### 2.1 Video Instance Segmentation

Video instance segmentation, a cornerstone of advanced computer vision applications, has made significant strides in addressing challenges in dynamic scenes. Among these challenges, **progressive occlusion** remains a persistent issue, requiring sophisticated methods for reliable and continuous instance association and tracking.

Early approaches, such as MaskTrack R-CNN [38], laid the foundation for detection and tracking by incorporating temporal information, providing a starting point for tackling occlusion. Later, models like CrossVIS [39] and SipMask [4] advanced the field by improving instance association and real-time efficiency through cross-frame feature utilization and one-stage segmentation backbones. Transformer-based models marked a further leap forward. VisTR [32] introduced transformers but faced challenges with complex motion, while SeqFormer [34] leveraged deformable attention to better capture spatial and temporal interactions. IFC [18] introduced inter-frame communication to align instance.

More recently, decoupling strategies, including Mask2Former-VIS [7], MinVIS [17], IDOL [35], and ROVIS [41], have adopted detect-then-associate approaches with explicit queries, significantly improving segmentation continuity. GenVIS [14] further advanced the field by eliminating heuristic matching and using refined instance representations for more effective inter-frame association. DVIS [43, 44] introduced a referring tracker to enhance frame association by denoising instance representations, while DVIS-DAQ [46] explicitly anchored new and disappearing objects to handle occlusions more effectively. [10] and [42] extend the boundaries of video instance segmentation methods from the perspective of SAM and lightweight design. VISAGE [21] focuses on leveraging appearance-object relationships to enhance segmentation. TCOVIS [22], on the other hand, employs dual local-global matching strategies to improve accuracy. Meanwhile, CTVIS [40] constructs identity matrices to explicitly track the gradual appearance and disappearance of objects over time. However, the prevailing methodologies in this field predominantly rely on memory/prototype mechanisms, thus underscoring the need for novel paradigms that can offer fresh perspectives and advancements in video instance segmentation.

Adhere to this intention, our proposed WSInfer method leverages hypergraph-based representations to model high-order relationships between instances. By incorporating hypergraph convolution and efficient temporal consistency inference, WSInfer effectively addresses the challenges of progressive occlusion, ensuring robust instance tracking and segmentation across dynamic video scenes.

### 2.2 Hypergraph Learning

Hypergraph neural networks (HGNNs)[11] extend traditional graph neural networks (GNNs) by modeling high-order interactions, which conventional GNNs, limited to pairwise relationships, cannot effectively capture. Unlike traditional graphs, hypergraphs use hyperedges to connect multiple nodes, enabling the representation of complex, multifaceted relationships. This transition from GNNs to HGNNs represents a significant advancement in capturing intricate data interactions, making HGNNs particularly valuable for applications that require the representation of complex relationships, such as action recognition and visual perception[6, 13, 19, 27]. For example, Hao et al.[11] used Hyper-GNN to model non-physical dependencies in action recognition, while An et al.[1] employed multi-hypergraph fusion to capture complex relationships in person re-identification. The ability of HGNNs to model higher-order relationships aligns well with the challenge of progressive occlusion in video instance segmentation, which requires robust tracking and

segmentation in dynamic environments. This capability motivates the adoption of hypergraphs in our proposed framework.

# 3 Preliminaries

## 3.1 Cellular Sheaves on Hypergraphs

A hypergraph is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V}$ represents the set of nodes, $\mathcal{E}$ is the set of hyperedges with $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$, and $\mathcal{W}$ defines the weights of the hyperedges. A hyperedge $e \in \mathcal{E}$ can connect any number of nodes. The number of nodes $\delta_e$ in each hyperedge, denoted as $|e|$, is called the degree of the hyperedge. A cellular sheaf $\mathcal{F}$ [9, 37] over a hypergraph $\mathcal{H}$ assigns vector spaces, referred to as stalks, to both nodes and hyperedges, and defines linear maps, known as restriction maps, between them:

- **Node/Hyperedge Stalks**: Each node $v \in \mathcal{V}$ and each hyperedge $e \in \mathcal{E}$ is assigned a vector space $\mathcal{F}(\cdot) = \mathbb{R}^d$, representing their respective $d$-dimensional feature.
- **Restriction Maps**: For every node $v$ in hyperedge $e$, a linear map $\mathcal{F}_{v \trianglelefteq e} = \text{MLP}(v, e) \in \mathbb{R}^{d \times d}$ is defined, linking the node and hyperedge feature spaces.

This sheaf structure allows the modeling of local consistency constraints and higher-order interactions within the hypergraph, providing a solid mathematical framework for complex video data understanding and analysis.

## 3.2 Sheaf Laplacian and Feature Propagation

The linear sheaf hypergraph Laplacian $\mathcal{L}^{\mathcal{F}} \in \mathbb{R}^{d \times d}$ for a hypergraph $\mathcal{H}$, normalized by $\delta_e$, is defined as:

$$\mathcal{L}^{\mathcal{F}}_{uv} = \begin{cases} -\sum\limits_{e;u,v \in e} \frac{1}{\delta_e} \mathcal{F}^T_{u \trianglelefteq e} \mathcal{F}_{v \trianglelefteq e}, & u \neq v, \\ \sum\limits_{e;v \in e} \frac{1}{\delta_e} \mathcal{F}^T_{v \trianglelefteq e} \mathcal{F}_{v \trianglelefteq e}, & u = v. \end{cases} \tag{1}$$

The linear sheaf Laplacian operator is described by applying $\mathcal{L}^{\mathcal{F}}$ to a feature $x \in \mathbb{R}^{n \times d}$ as follows:

$$\mathcal{L}^{\mathcal{F}}_{v} = \sum_{e;v \in e} \frac{1}{\delta_e} \mathcal{F}^T_{v \trianglelefteq e} \left( \sum_{u \in e, u \neq v} (\mathcal{F}_{v \trianglelefteq e} x_v - \mathcal{F}_{u \trianglelefteq e} x_u) \right). \tag{2}$$

This formulation enables the aggregation of information from neighboring nodes while preserving higher-order relationships and enforcing local consistency. As a result, sheaf-based hypergraph neural networks (SHNNs) are effective for modeling complex data interactions.

## 3.3 Sliced Wasserstein Distance

**One-dimensional Wasserstein Distance.** For one-dimensional probability measures $\mu$ and $\nu$ in $\mathcal{P}_p(\mathbb{R})$, the $p$-Wasserstein distance is defined as:

$$\mathrm{W}^p_p(\mu, \nu) = \int_0^1 |F^{-1}_\mu(z) - F^{-1}_\nu(z)|^p \, dz, \tag{3}$$

where $F_\mu$ and $F_\nu$ are the cumulative distribution functions (CDFs) of $\mu$ and $\nu$. This formulation provides a closed form for computing the Wasserstein distance in one-dimensional spaces, making it well-suited for projected measures.

**Sliced Wasserstein Distance.** To generalize the Wasserstein distance to higher-dimensional measures, the Sliced Wasserstein Distance (SWD) [3] projects the measures $\mu$ and $\nu$ in $\mathcal{P}_p(\mathbb{R}^d)$ onto one-dimensional subspaces, and then averages the one-dimensional Wasserstein distances from these projections. For $\mu, \nu \in \mathcal{P}_p(\mathbb{R}^d)$, the SWD is defined as:

$$\mathrm{SW}^p_p(\mu, \nu) = \mathbb{E}_{\theta \sim \mathcal{U}(\mathbb{S}^{d-1})} \left[ \mathrm{W}^p_p(\theta \sharp \mu, \theta \sharp \nu) \right], \tag{4}$$

where $\theta \sharp \mu$ and $\theta \sharp \nu$ represent the push-forward measures of $\mu$ and $\nu$ along direction $\theta \in \mathbb{S}^{d-1}$, the unit sphere in $\mathbb{R}^d$. This projection $f(x) = \theta^\top x$ maps points from $\mathbb{R}^d$ to $\mathbb{R}$, enabling the computation of Wasserstein distances in a one-dimensional space.

Since the expectation in Eq. 4 is computationally expensive, SWD is typically approximated by averaging over $L$ independent directions, $\theta_1, \ldots, \theta_L$, sampled from $\mathcal{U}(\mathbb{S}^{d-1})$:

$$\widehat{\mathrm{SW}}^p_p(\mu, \nu; L) = \frac{1}{L} \sum_{l=1}^{L} \mathrm{W}^p_p(\theta_l \sharp \mu, \theta_l \sharp \nu), \tag{5}$$

where each $\theta_l \sharp \mu$ and $\theta_l \sharp \nu$ are projected representations of $\mu$ and $\nu$ along the direction $\theta_l$. The number of projections $L$ controls the accuracy of the Monte Carlo approximation.

# 4 Method

Our WSInfer method leverages Mask2Former [8] as the segmenter, following a decoupling strategy. We extract instance-level queries $\{\mathcal{Q}_t\}_{t=1}^T$, where $t \in [1, T]$ denotes the timestep. Each query $\mathcal{Q}_t = \{\mathbf{q}_{t,i}\}_{i=1}^{N_q} \in \mathbb{R}^d$ represents the $N_q$ queries in a frame, with $\mathbf{q}_{t,i} \in \mathbb{R}^d$ derived from Mask2Former's output embeddings, encapsulating appearance and spatial features. Using these queries, we construct a hypergraph, perform hypergraph convolution to enhance structural features (§4.1), and enforce spatio-temporal consistency (§4.2). The overall pipeline and objective function (§4.3) are illustrated in Fig. 2.

## 4.1 Building Higher-Order Relationships

**Hypergraph Construction.** To construct the hypergraph $\mathcal{H}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{W}_t)$ from $\mathcal{Q}_t$, we define the nodes set $\mathcal{V}_t$, where each node $v_{t,i}$ corresponds to a query $\mathbf{q}_{t,i}$. Hyperedges model spatial dependencies, connecting groups of nodes to capture complex spatial relationships beyond simple pairwise connections.

We construct hyperedges by computing pairwise distances between queries $\mathbf{q}_{t,i}$ and $\mathbf{q}_{t,j}$ for each pair of instances $i$ and $j$ ($i \neq j$). Specifically, the distance $d_{ij}$ between nodes $v_{t,i}$ and $v_{t,j}$, which are associated with feature vectors $\mathbf{q}_{t,i}$ and $\mathbf{q}_{t,j}$, is computed as:

$$d_{ij} = \|\mathbf{q}_{t,i} - \mathbf{q}_{t,j}\|_2, \tag{6}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. For each node $v_{t,i}$, we identify its $k$-nearest neighbors based on the smallest distances in feature space, forming a neighborhood set $\mathcal{N}_{t,i}$ that captures the local spatial context. A hyperedge $e_{t,i}$ is then defined to include $v_{t,i}$ and its neighbors $\mathcal{N}_{t,i}$, forming:

$$e_{t,i} = v_{t,i} \cup \mathcal{N}_{t,i}. \tag{7}$$

The strength of relationships within each hyperedge is represented by a weight $w_{t,i} \in \mathcal{W}t$, calculated as:

$$w_{t,i} = \frac{1}{|\mathcal{N}_{t,i}|} \sum v_{t,j} \in \mathcal{N}_{t,i} \frac{1}{d_{ij}}, \tag{8}$$

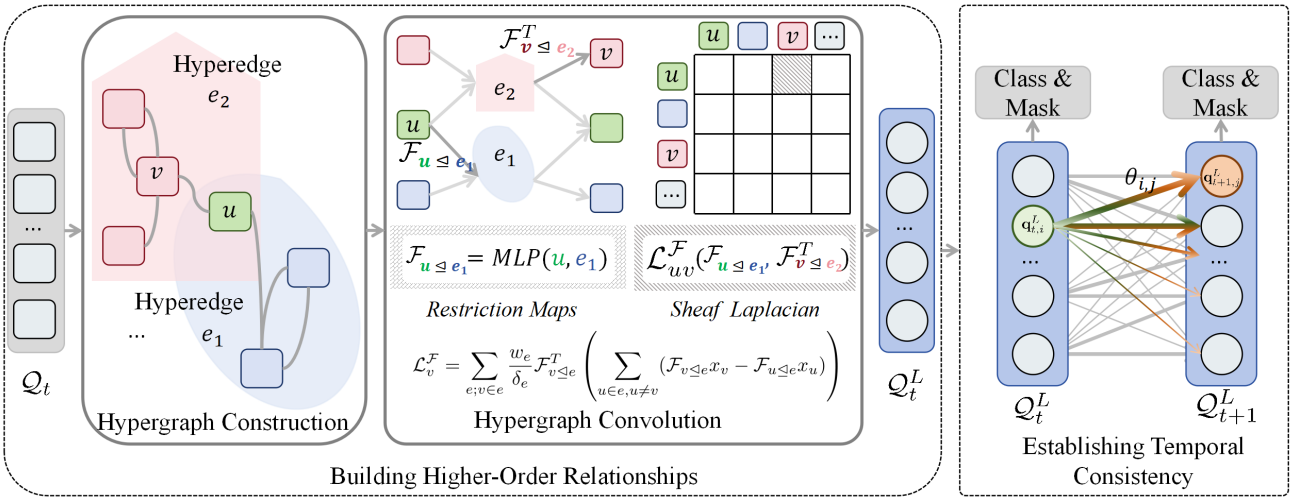


**Figure 2: The framework of our WSInfer. In the temporal consistency inference module, the direction of the arrow indicates the projection direction $\theta_{i,j}$, and the thickness of the arrow indicates the projection weights $w_{i,j}$.**

where $|\mathcal{N}_{t,i}|$ is the number of nodes in the neighborhood set. This weighting effectively encodes spatial information, preserving key characteristics of instance arrangements.

**Weighted Sheaf Hypergraph Convolution.** After constructing the hypergraph $\mathcal{H}_t$ for each frame, we use the sheaf Laplacian to perform hypergraph feature propagation to augment the structural feature representation. We fully consider the strength of the hyperedges on the node feature propagation, and when performing the node feature update, in addition to considering the size of the hyperedges, we also weight the differences between the nodes through the weights. Therefore, the formula is rewritten as:

$$\mathcal{L}_v^{\mathcal{F}} = \sum_{e;v\in e} \frac{w_e}{\delta_e} \mathcal{F}_{v\trianglelefteq e}^T \left( \sum_{u\in e, u\neq v} (\mathcal{F}_{v\trianglelefteq e} x_v - \mathcal{F}_{u\trianglelefteq e} x_u) \right). \tag{9}$$

The Weighted Sheaf Hypergraph Convolution (WSHC) layer operation is formulated as:

$$\mathcal{Q}_t^{l+1} = \sigma((\mathbb{I}_{nd} - \Delta^{\mathcal{F}})(\mathbb{I}_n \otimes W_1^l)\mathcal{Q}_t^l W_2^l), \tag{10}$$

$$\Delta^{\mathcal{F}} = D^{-1/2} \mathcal{L}_{\mathcal{F}} D^{-1/2}, \tag{11}$$

where $\mathbb{I}_{nd}$ and $\mathbb{I}_n$ are unit matrices of size $nd \times nd$ and $n \times n$ respectively. $n$ is the number of nodes, and $d$ is the sheaf dimension. $D = diag(D_1, D_2, \cdots, D_v) \in \mathbb{R}^{nd\times nd}$ is the corresponding block diagonal matrix, where each block $D_v = \sum_{e;v\in e} \mathcal{F}_{v\trianglelefteq e}^{\top} \mathcal{F}_{v\trianglelefteq e} \in \mathbb{R}^{d\times d}$ represents the degree matrix for each node. $l$ denotes the current layer index. The output of layer $l$ serves as the input for the next layer. We constructed a structural feature $\mathcal{Q}_t^L$ of the $L$-layer WSHC after it has been enhanced.

The WSHC layer propagates and refines these features by incorporating higher-order relationships captured by the hypergraph. The sheaf Laplacian $\Delta^{\mathcal{F}}$ enforces local consistency among node features based on the constraints defined by each hyperedge $e_{t,j}$. The term $\mathbb{I}_{nd} - \Delta^{\mathcal{F}}$ incorporates the sheaf Laplacian for feature diffusion while respecting the sheaf constraints. The learnable weight matrices $W_1 \in \mathbb{R}^{d\times d}$ and $W_2 \in \mathbb{R}^{d\times d}$ dynamically adjust feature transformations and aggregations. $\mathbb{I}_n \otimes W_1$ uniformly scales feature vectors, while the ReLU activation $\sigma$ introduces non-linearity to model complex patterns.

## 4.2 Establishing Temporal Consistency

Once the higher-order hypergraph-based relationships have been refined, the correspondence between instances in successive frames $t$ and $t+1$ needs to be established. As objects undergo occlusion, movement, and appearance changes, accurate instance tracking over time is crucial. We use the Weighted Sliced Wasserstein (WSW) metric to establish instances correspondences between instances in consecutive frames.

For each query $i$ in frame $t$ and instance $j$ in frame $t+1$, we define the random path $Z_{i,j} = \mathbf{q}_{t,i}^L - \mathbf{q}_{t+1,j}^L$, capturing the directional difference between the two enhanced query. $\mathbf{q}_{t,i}^L \in \mathcal{Q}_t^L$ and $\mathbf{q}_{t+1,i}^L \in \mathcal{Q}_{t+1}^L$ are the output of the WSHC. This path is then normalized to obtain the unit direction $\theta_{i,j} = \frac{Z_{i,j}}{\|Z_{i,j}\|_2}$, which serves as the random-path projecting direction. In cases of near-zero $Z_{i,j}$ (i.e., nearly identical features), a predefined small constant is added to ensure stability. The set of such directions, enriched by a small random perturbation if needed, forms a distribution of projection directions that highlights spatial and appearance differences.

For each projecting direction $\theta_{i,j}$, we project $\mathcal{Q}_t^L$ and $\mathcal{Q}_{t+1}^L$ along $\theta_{i,j}$ to obtain one-dimensional representations, and calculate the Wasserstein distance $W_p(\theta_{i,j}\sharp\mathcal{Q}_t^L, \theta_{i,j}\sharp\mathcal{Q}_{t+1}^L)$ between the projected distributions. This distance measures how well the instances align along that particular direction. Larger Wasserstein distances indicate greater dissimilarity between instances in the direction of $\theta_{i,j}$, thus capturing more distinctive variations. To emphasize informative directions, we weight these distances that applies higher

weights to projections with larger differences, enhancing the discriminative capability of the WSW measure. The final WSW distance between $Q_t^L$ and $Q_{t+1}^L$ (see Eq.5) is an aggregate of these weighted distances:

$$\mathrm{WSW}_p^p(Q_t^L, Q_{t+1}^L) = \sum_{i,j} w_{i,j} W_p(\theta_{i,j}\sharp Q_t^L, \theta_{i,j}\sharp Q_{t+1}^L), \tag{12}$$

where $w_{i,j}$ represents the importance weight for each projected distance between instances in consecutive frames. The weight $w_{i,j}$ determines the contribution of each projecting direction to the overall WSW distance.

The weight $w_{i,j}$ is typically calculated based on the projected Wasserstein distance $W_p(\theta_{i,j}\sharp Q_t^L, \theta_{i,j}\sharp Q_{t+1}^L)$ between the query in frames $t$ and $t+1$. This distance indicates how dissimilar the projected distributions are along the direction $\theta_{i,j}$, which is based on the difference between query $\mathbf{q}_{t,i}^{(L)}$ and $\mathbf{q}_{t+1,j}^{(L)}$. The weight $w_{i,j}$ for a direction $\theta_{i,j}$ is:

$$w_{i,j} = \frac{f(W_p(\theta_{i,j}\sharp Q_t^L, \theta_{i,j}\sharp Q_{t+1}^L))}{\sum_{k,l} f(W_p(\theta_{k,l}\sharp Q_t^L, \theta_{k,l}\sharp Q_{t+1}^L))}, \tag{13}$$

where $f(x) = e^x$ is typically a monotonic function, places higher importance on larger differences, which may highlight instances with more substantial temporal changes. This normalization ensures that the weights sum up to 1 across all pairs, providing a proportionate contribution of each pair to the final WSW distance. This normalization ensures that all projection pairs contribute proportionately, preventing any single pair from disproportionately affecting the WSW.

To establish instance correspondences, we compute the WSW values between all pairs of instances across consecutive frames $t$ and $t+1$, denoted as $\mathrm{WSW}_p^p(Q_t^L, Q_{t+1}^L)$. These values quantify the difference between instances along the direction of projection, with smaller WSW values indicating stronger spatio-temporal consistency. By selecting the instance pairs with the smallest WSW values, we identify those with the highest temporal alignment, ensuring robust and meaningful associations for instance tracking.

Formally, the instance matching objective is defined as:

$$\mathcal{L}_{Match} = \arg\min_{i,j} \mathrm{WSW}_p^p(Q_t^L, Q_{t+1}^L), \tag{14}$$

where the selected pairs correspond to instances with the strongest spatio-temporal consistency, facilitating accurate tracking and correlation across frames.

In this way, our method emphasizes directions and features that capture significant inter-frame changes, effectively prioritizing instances that reflect spatial and appearance consistency. The adaptive weighting in WSW further enhances tracking accuracy by highlighting subtle variations, ensuring robust temporal consistency.

**Computational Complexity:** In Eq. 12, when $Q_t^L$ and $Q_{t+1}^L$ are discrete measures with at most $n$ supports, sampling their random-path projections incurs the following complexities: **Random Path Sampling:** The cost for $L$ projections is $O(Ldn)$ in both time and memory. **Projection Sampling:** Sampling from von Mises-Fisher (vMF) and Power Spherical (PS) distributions costs $O(Ld)$. **One-Dimensional Wasserstein Distance:** Computing $W_p^p$ for all projections adds $O(Ln\log n)$. The overall time complexity is therefore $O(Ln\log n + Ldn)$, and the space complexity is $O(Ld + Ln)$. These complexities are manageable, ensuring the method is scalable for large datasets.

### 4.3 Objective Function.

Finally, the enhanced instance query, denoted as $Q_t^L$, is used as input for both the class head and the mask head, which generate the category and mask coefficient outputs, $\hat{y}_t$.

The total loss function for our WSInfer is defined as:

$$\mathcal{L}_{Total} = \mathcal{L}_{Mask}(y, \hat{y}) + \mathcal{L}_{Match}, \tag{15}$$

where $\hat{y}$ is label, $\mathcal{L}_{Mask}$ [7] refers to the mask loss related to both class and mask predictions.

## 5 Experiments

### 5.1 Implementation Details

For the Restriction Maps $\mathcal{F}_{v\trianglelefteq e} = \mathrm{MLP}(v, e) \in \mathbb{R}^{d\times d}$, the inputs to the MLP are the node feature $v$ and the hyperedge feature $e$. The MLP architecture consists of two fully - connected layers with ReLU activation in between. The first layer has an input dimension of the sum of the dimensions of $v$ and $e$, and an output dimension of $d$. The second layer outputs $d$ elements. The final restriction block is obtained by creating a diagonal matrix with these $d$ elements. In our setup, $d = 8$. We conducted a hyperparameter search for $K$ in the k-nearest neighbors strategy, testing values from 3 to 10. Given that nodes are constructed using 100 queries from Mask2Former, excessively large $K$ values risk over-smoothing. Results show the model is insensitive to $K$, leading us to select $K = 5$ for its balance between local and global context modeling. To ensure the robustness of our framework, we carefully fine-tuned several hyperparameters through validation experiments. The learning rate was set to $1 \times 10^{-4}$, providing a stable optimization process while avoiding overfitting. We used three layers of Weighted Sheaf Hypergraph Convolution (WSHC), striking a balance between computational efficiency and the depth of structural feature learning. For hypergraph construction, we employed a $k$-nearest neighbor strategy with $k = 5$, which effectively captured local spatial relationships while avoiding over-clustering.

In addition, we chose 100 random-path projections ($L = 100$) for the Weighted Sliced Wasserstein (WSW) computation, ensuring sufficient diversity in the projection directions for temporal consistency. A regularization weight of $\alpha = 0.1$ was applied to the sheaf Laplacian to ensure smooth feature propagation while maintaining local structural consistency. For training, we set the batch size to 8 and trained the model over 50 epochs, which provided sufficient iterations to capture both spatial and temporal relationships within the datasets. All experiments were conducted on a high-performance machine equipped with 8 NVIDIA A800 GPUs.

### 5.2 Datasets and Evaluation Metrics

We evaluate the performance of WSInfer for VIS on the YouTube VIS 2019, 2021, and 2022 [38] datasets, OVIS [28], and for VPS on the VIPSeg [25] dataset.

**YouTube-VIS** datasets from 2019, 2021, and 2022 are established as standard benchmarks for Video Instance Segmentation (VIS). These datasets encompass a diverse collection of video sequences,

**Table 1: Results on the validation sets of YouTube-VIS 2019 & 2021 and OVIS. † denotes offline methods.**

| Method | Backbone | Youtube-VIS 2019 | | | Youtube-VIS 2021 | | | OVIS | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | AP | $AP_{75}$ | $AR_{10}$ | AP | $AP_{75}$ | $AR_{10}$ | AP | $AP_{75}$ | $AR_{10}$ |
| MaskTrack R-CNN [38] | ResNet-50 | 30.3 | 32.6 | 35.5 | 28.6 | 29.6 | 33.8 | 10.8 | 8.5 | 14.9 |
| SipMask [4] | ResNet-50 | 33.7 | 35.8 | 40.1 | 31.7 | 34.0 | 37.8 | - | - | - |
| CrossVIS [39] | ResNet-50 | 36.3 | 38.9 | 40.7 | 34.2 | 37.9 | 38.2 | 14.9 | 12.1 | 19.8 |
| EfficientVIS [36]† | ResNet-50 | 37.9 | 43.0 | 46.6 | 34.0 | 37.3 | 42.5 | - | - | - |
| IFC [18]† | ResNet-50 | 41.2 | 44.6 | 49.6 | 35.2 | 37.7 | 42.9 | 13.1 | 11.6 | 23.9 |
| Mask2Former-VIS [7]† | ResNet-50 | 46.4 | 50.0 | - | 40.6 | 41.8 | - | 17.3 | 15.1 | 23.5 |
| SeqFormer [34]† | ResNet-50 | 47.4 | 51.8 | 54.8 | 40.5 | 43.7 | 48.1 | 15.1 | 13.8 | 27.1 |
| VISOLO [12] | ResNet-50 | 38.6 | 43.7 | 42.5 | 36.9 | 40.2 | 40.9 | 15.3 | 13.8 | 20.0 |
| MinVIS [17] | ResNet-50 | 47.4 | 52.1 | 55.7 | 44.2 | 48.1 | 51.7 | 25.0 | 24.0 | 29.7 |
| IDOL [35] | ResNet-50 | 49.5 | 52.9 | 58.7 | 43.9 | 49.6 | 50.9 | 28.2 | 28.0 | 38.6 |
| VITA [16]† | ResNet-50 | 49.8 | 54.5 | 61.0 | 45.7 | 49.5 | 53.6 | 19.6 | 17.4 | 26.0 |
| GenVIS[15] | ResNet-50 | 50.0 | 54.6 | 59.7 | 47.1 | 51.5 | 54.7 | 35.8 | 36.2 | 39.6 |
| DVIS[43] | ResNet-50 | 51.2 | 57.1 | 59.3 | 46.4 | 49.6 | 53.5 | 31.0 | 31.9 | 37.6 |
| TCOVIS [22] | ResNet-50 | 52.3 | 57.6 | 60.2 | 49.5 | 53.8 | 55.9 | 35.3 | 36.6 | 39.5 |
| CTVIS [40] | ResNet-50 | 55.1 | 59.1 | **63.2** | 50.1 | 54.7 | **59.5** | 35.5 | 34.9 | **41.9** |
| VISAGE [21] | ResNet-50 | 55.1 | 60.6 | 62.3 | 51.6 | 56.1 | 59.3 | 36.2 | 35.3 | 40.3 |
| Our | ResNet-50 | **55.3** | **61.2** | 62.8 | **51.7** | **56.3** | 59.1 | **36.8** | **36.9** | 41.7 |
| SeqFormer [34]† | Swin-L | 59.3 | 66.4 | 64.4 | 51.8 | 58.2 | 58.1 | - | - | - |
| Mask2Former-VIS [7]† | Swin-L | 60.4 | 67.0 | - | 52.6 | 57.2 | - | 25.8 | 24.4 | 32.2 |
| MinVIS [17] | Swin-L | 61.6 | 68.6 | 66.6 | 55.3 | 62.0 | 60.8 | 39.4 | 41.3 | 43.3 |
| VITA [16]† | Swin-L | 63.0 | 67.9 | 68.1 | 57.5 | 61.0 | 62.6 | 27.7 | 24.9 | 33.0 |
| GenVIS[15] | Swin-L | 64.0 | 68.3 | 69.4 | 59.6 | 65.8 | 65.0 | 45.2 | 48.4 | 48.6 |
| IDOL [35] | Swin-L | 64.3 | 71.0 | 69.1 | 56.1 | 63.5 | 60.1 | 40.0 | 40.5 | 46.4 |
| DVIS[43] | Swin-L | 63.9 | 70.4 | 69.0 | 58.7 | 66.6 | 64.6 | 45.9 | 48.3 | 51.5 |
| TCOVIS [22] | Swin-L | 64.1 | 69.5 | 69.0 | 61.3 | 68.0 | 65.1 | 35.3 | 36.6 | 39.5 |
| CTVIS [40] | Swin-L | 65.6 | 72.2 | **70.4** | 61.2 | 68.8 | **65.8** | **46.9** | 47.5 | 52.1 |
| Our | Swin-L | **66.2** | **72.5** | 70.3 | **61.6** | **69.5** | **65.8** | **46.9** | **49.0** | **52.4** |

featuring multiple interacting object instances with substantial category variation and complex spatiotemporal dynamics.

**OVIS** dataset is specifically curated for Occluded Video Instance Segmentation, containing highly occluded and intricate scenes that present significant challenges for model evaluation under adverse visual conditions. It serves as a rigorous benchmark for assessing the robustness of models in handling severe occlusions.

**VIPSeg** dataset addresses the broader task of Video Panoptic Segmentation (VPS) by providing comprehensive pixel-level annotations that include both instance-level objects ("things") and semantic background regions ("stuff"). This richer annotation scheme allows for the simultaneous assessment of models' abilities to handle instance-level segmentation and semantic understanding, thereby requiring a more holistic interpretation of the video, including the interaction between instances and the environment.

We employ Average Precision (AP) and Average Recall (AR) as evaluation metrics for the VIS datasets, following the methodology in [38]. For the VPS datasets, we utilize Video Panoptic Quality (VPQ) and Segmentation and Tracking Quality (STQ) as evaluation metrics [20]. Supplemental Materials reports more results.

**Table 2: Results on the YouTube-VIS 2022 dataset with Swin-L backbone. The best metrics in each group are bolded.**

| Method | YouTube-VIS 2022 | | | | |
|---|---|---|---|---|---|
| | AP | $AP_{50}$ | $AP_{75}$ | $AR_1$ | $AR_{10}$ |
| MinVIS [17] | 33.1 | 54.8 | 33.7 | 29.5 | 36.6 |
| VITA [16] | 41.1 | 63.0 | 44.0 | 39.3 | 44.3 |
| GenVIS [14] | 44.3 | 69.9 | 44.9 | 39.9 | 48.4 |
| DVIS[43] | 45.9 | 69.0 | 48.8 | 37.2 | 51.8 |
| Our | **48.6** | **72.5** | **51.2** | **40.8** | **55.1** |

## 5.3 Results of Video Segmentation

**Performance on Youtube-VIS 2019&2021 datasets.** The experimental results in Table. 1 demonstrate the effectiveness of our proposed methodology in comparison to state-of-the-art models for Video Instance Segmentation (VIS) on the YouTube-VIS 2019 and 2021 datasets. By leveraging the Weighted Structure Modeling framework, our method consistently outperforms existing models across key metrics. Specifically, our model shows significant improvements in capturing higher-order interactions and maintaining robust tracking, particularly in complex and occlusion-heavy scenes.

**Table 3: Results on the VIPSeg dataset. R and S indicate ResNet50 and Swin-L backbone networks, respectively.**

| | Method | VIPSeg | | | |
|---|---|---|---|---|---|
| | | VPQ | $VPQ^{Th}$ | $VPQ^{St}$ | STQ |
| R | VPSNet [20] | 14.0 | 14.0 | 14.2 | 20.8 |
| | VPSNet-SiamTrack [33] | 17.2 | 17.3 | 17.3 | 21.1 |
| | VIP-Deeplab [29] | 16.0 | 12.3 | 18.2 | 22.0 |
| | Clip-PanoFCN [25] | 22.9 | 25.0 | 20.8 | 31.5 |
| | Video K-Net [24] | 26.1 | - | - | 31.5 |
| | TarVIS [2] | 33.5 | 39.2 | 28.5 | **43.1** |
| | Tube-Link [23] | 39.2 | - | - | 39.5 |
| | Video-kMax [31] | 38.2 | - | - | 39.9 |
| | DVIS[43] | 43.2 | 43.6 | 42.8 | 42.8 |
| | Our | **44.1** | **44.3** | **43.2** | 42.9 |
| S | TarVIS [2] | 48.0 | 58.2 | 39.0 | 52.9 |
| | DVIS[43] | 57.6 | 59.9 | 55.5 | 55.3 |
| | Our | **58.1** | **60.5** | **56.8** | **56.0** |

The use of hypergraph-based representations allows for more nuanced modeling of intricate relationships, surpassing traditional graph neural network approaches and outperforming advanced decoupling strategies. These results highlight the strength of hypergraph modeling in addressing progressive occlusion challenges, leading to marked improvements in both segmentation accuracy and temporal continuity across dynamic video environments.

**Performance on OVIS dataset.** The results in Table. 1 on the OVIS dataset further emphasize the strengths of our proposed method, especially in tackling the highly challenging occlusion scenarios prevalent in this dataset. By incorporating a weighted strategy for high-order relationship modeling, our method achieves leading performance in key metrics, demonstrating its ability to capture the complex dependencies between instances. This is crucial for managing progressive and overlapping occlusions, underscoring the robustness of our approach in difficult visual conditions.

**Performance on Youtube-VIS 2022 dataset.** On the YouTube-VIS 2022 dataset, our method excels in handling long and complex video sequences characterized by intricate object interactions and frequent occlusions. As shown in Table. 2, our model achieves the highest Average Precision (AP) of 48.6, $AP_{75}$ of 51.2, and Average Recall ($AR_{10}$) of 55.1, showcasing superior precision and robustness in tracking and segmenting objects over extended frames.

**Performance on VIPseg dataset.** The VIPSeg dataset presents a unique challenge due to its requirement for robust temporal and spatial understanding, given the dynamic object movements and scene transitions. In Table. 3, our model achieves the highest scores in Video Panoptic Quality (VPQ), $VPQ^{Th}$, $VPQ^{St}$, and Segmentation and Tracking Quality (STQ), demonstrating its superior ability to maintain temporal consistency while ensuring high segmentation quality for both moving objects and background elements. These results validate the effectiveness of hypergraph-based modeling in integrating both semantic and instance-level cues, which are crucial for excelling in complex video segmentation tasks such as those presented by VIPSeg.

Furthermore, we provide a few representative visual examples of video instance segmentation results on OVIS, YT-VIS, and VIPseg datasets, demonstrating robust performance in crowded scenes (Fig. 3), objects with significant shape variations (Fig. 4), and small-scale objects (Fig. 5).

**Figure 3: Qualitative segmentation results of OVIS dataset.**

**Figure 4: Qualitative segmentation results of YT-VIS datasets.**

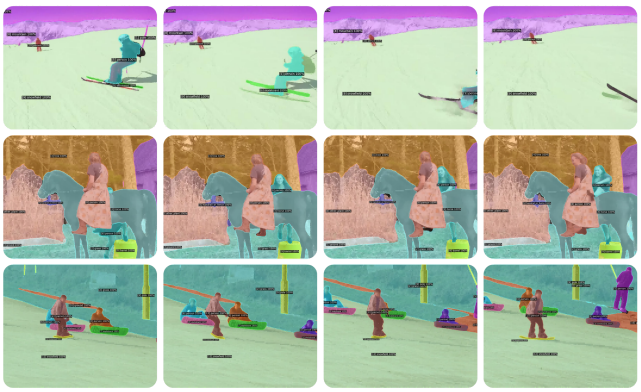

**Figure 5: Qualitative segmentation results of VIPSeg dataset.**

## 5.4 Ablation Experiments

Ablation experiments were conducted on the Youtube-VIS 2019 dataset, with WSInfer evaluated using ResNet50 and input resized to 360p unless otherwise specified.

**Table 4: Ablation study results on core components.**

| Modules | AP | $AP_{50}$ | $AP_{75}$ | $AR_1$ | $AR_{10}$ |
|---|---|---|---|---|---|
| WSInfer | **55.3** | **78.5** | **61.2** | **50.9** | **62.8** |
| w/o HGC | 48.6 | 72.7 | 53.5 | 45.0 | 57.8 |
| w/o WSHC | 50.1 | 74.2 | 55.0 | 46.6 | 59.9 |
| w/o WSW | 51.3 | 75.3 | 57.2 | 48.3 | 61.1 |

**Table 5: Ablation study results comparing different hypergraph construction modules for the WSInfer framework.**

| Modules | AP | $AP_{50}$ | $AP_{75}$ | $AR_1$ | $AR_{10}$ |
|---|---|---|---|---|---|
| K-Nearest Neighbors | **55.3** | **78.5** | **61.2** | **50.9** | **62.8** |
| Spectral Clustering | 52.1 | 75.3 | 57.4 | 48.2 | 60.2 |
| K-Means Clustering | 51.6 | 74.6 | 56.7 | 47.5 | 59.5 |
| Fuzzy C-Means | 52.9 | 76.4 | 58.3 | 49.1 | 61.1 |

**Table 6: Ablation study results comparing different hypergraph convolution modules for the WSInfer framework.**

| Modules | AP | $AP_{50}$ | $AP_{75}$ | $AR_1$ | $AR_{10}$ |
|---|---|---|---|---|---|
| WSHC (Eq.10) | **55.3** | **78.5** | **61.2** | **50.9** | **62.8** |
| Standard | 50.3 | 73.1 | 55.0 | 47.0 | 59.2 |
| Attention-based | 52.4 | 75.0 | 57.7 | 48.6 | 60.8 |
| Spectral | 51.7 | 74.4 | 56.3 | 48.1 | 60.0 |

**Core Component.** The ablation study in Table. 4 highlights the importance of each core component in the WSInfer framework. Removing hypergraph construction (HGC) caused the largest drop in AP (from 55.3 to 48.6), underscoring its critical role in modeling high-order relationships among instances. The absence of Weighted Sheaf Hypergraph Convolution (WSHC) reduced AP to 50.1, demonstrating its contribution to enriching feature representation. Similarly, removing Weighted Sliced Wasserstein (WSW) led to a decrease in AP to 51.3, indicating its importance in maintaining temporal coherence across frames.

**Hypergraph Construction.** The results in Table. 5 demonstrate the influence of different hypergraph construction modules on the WSInfer framework's performance. The k-Nearest Neighbors (k-NN) module achieves the highest AP (55.3) and overall best performance across all metrics, indicating that connecting nodes based on feature space proximity effectively captures local interactions critical for video instance segmentation. Spectral Clustering performs slightly worse, with an AP of 52.1, suggesting that while spectral properties capture broader community structures, they may not effectively represent fine-grained local relationships. K-Means Clustering shows a further drop in performance (AP of 51.6), implying that hard clustering may oversimplify the underlying relationships between nodes, leading to less effective hyperedges. Fuzzy C-Means (FCM) provides a balance, with an AP of 52.9, benefiting from overlapping clusters that capture more nuanced relationships, though still not outperforming k-NN.

**Hypergraph Convolution.** The results in Table. 6 of the ablation study underscore the superior performance of the Weighted Sheaf Hypergraph Convolution (WSHC) module, which leads with the highest AP score of 55.3. This can be attributed to WSHC's ability to enforce strong local consistency while simultaneously enriching feature representations, making it particularly well-suited to the demands of video instance segmentation. In comparison, the Standard Hypergraph Convolution [37] module lags behind, achieving a lower AP of 50.3. Its basic aggregation strategy is insufficient for capturing the complex relationships among nodes, leading to suboptimal performance. The Attention-based Hypergraph Convolution [47] module shows some improvement with an AP of 52.4, as it adaptively weights the contributions of individual nodes. However, it still fails to surpass WSHC, particularly in its ability to model intricate local interactions. Spectral Hypergraph Convolution is effective in capturing broad global structures but struggles with the local relationships necessary for more accurate segmentation.

**Table 7: Ablation study results comparing different temporal consistency modules for the WSInfer framework.**

| Modules | AP | $AP_{50}$ | $AP_{75}$ | $AR_1$ | $AR_{10}$ |
|---|---|---|---|---|---|
| WSW (Eq.12) | **55.3** | **78.5** | **61.2** | **50.9** | **62.8** |
| Refiner | 51.5 | 74.0 | 56.1 | 48.0 | 60.0 |
| 1D-W | 50.7 | 73.5 | 55.5 | 47.0 | 59.5 |
| RNNC | 52.1 | 75.5 | 57.0 | 48.7 | 61.0 |

**Temporal Consistency.** The results in Table. 7 of the ablation study highlight the superior performance of the Weighted Sliced Wasserstein (WSW) module in ensuring temporal consistency, where it achieves the highest AP score of 55.3. This success can be attributed to WSW's ability to capture complex high-order relationships and effectively model significant temporal changes across frames. In contrast, other modules, such as the Temporal Refiner [43], One-dimensional Wasserstein (1D-W) and Recurrent Neural Network-based Consistency (RNNC), fall short in modeling intricate temporal dependencies, which limits their ability to account in the video.

## 6 Conclusion

This paper introduces the Weighted Structure Modeling Hypergraph (WSInfer) framework to address the challenge of progressive occlusion in video instance segmentation. By leveraging weighted sheaf hypergraph convolution, WSInfer effectively captures higher-order relationships, overcoming the limitations of traditional graph neural networks (GNNs). Key contributions of this work include the enhancement of higher-order correspondences between local and global features across frames, facilitated by a weighted approach. Additionally, we introduce the weighted sheaf hypergraph convolution and weighted slice distance (WSD) measures, which further improve the model's performance. Experimental results demonstrate that WSInfer outperforms existing methods, leading to significant improvements in segmentation accuracy and continuity in complex scenes. For future work, we propose optimizing the WSD of all matched instances globally across multiple frames, which will enable offline handling of long-term progressive occlusions and further enhance spatio-temporal consistency.

## References

[1] Le An, Xiaojing Chen, Songfan Yang, and Xuelong Li. 2016. Person re-identification by multi-hypergraph fusion. *IEEE transactions on neural networks and learning systems* 28, 11 (2016), 2763–2774.

[2] Ali Athar, Alexander Hermans, Jonathon Luiten, Deva Ramanan, and Bastian Leibe. 2023. TarViS: A Unified Approach for Target-based Video Segmentation. *arXiv preprint arXiv:2301.02657* (2023).

[3] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. 2015. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision* 51 (2015), 22–45.

[4] Jiale Cao, Rao Muhammad Anwer, Hisham Cholakkal, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. 2020. Sipmask: Spatial information preservation for fast image and video instance segmentation. In *ECCV*.

[5] Keshigeyan Chandrasegaran, Agrim Gupta, Lea M Hadzic, Taran Kota, Jimming He, Cristóbal Eyzaguirre, Zane Durante, Manling Li, Jiajun Wu, and Li Fei-Fei. 2024. HourVideo: 1-Hour Video-Language Understanding. *arXiv preprint arXiv:2411.04998* (2024).

[6] Lu Chen, Qiangchang Wang, Zhaohui Li, and Yilong Yin. 2024. Hypergraph-guided Intra-and Inter-category Relation Modeling for Fine-grained Visual Recognition. In *ACM MM*.

[7] Bowen Cheng, Anwesa Choudhuri, Ishan Misra, Alexander Kirillov, Rohit Girdhar, and Alexander G Schwing. 2021. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764* (2021).

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. 2022. Masked-attention mask transformer for universal image segmentation. In *CVPR*.

[9] Iulia Duta, Giulia Cassarà, Fabrizio Silvestri, and Pietro Liò. 2023. Sheaf hypergraph networks. *NeurIPS* (2023).

[10] Hao Fang, Tong Zhang, Xiaofei Zhou, and Xinxin Zhang. 2024. Learning better video query with sam for video instance segmentation. *IEEE Transactions on Circuits and Systems for Video Technology* (2024).

[11] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *AAAI*.

[12] Su Ho Han, Sukjun Hwang, Seoung Wug Oh, Yeonchool Park, Hyunwoo Kim, Min-Jung Kim, and Seon Joo Kim. 2022. Visolo: Grid-based space-time aggregation for efficient online video instance segmentation. In *CVPR*.

[13] Yan Han, Peihao Wang, Souvik Kundu, Ying Ding, and Zhangyang Wang. 2023. Vision HGNN: An Image is More than a Graph of Nodes. In *ICCV*.

[14] Miran Heo, Sukjun Hwang, Jeongseok Hyun, Hanjung Kim, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. 2022. A Generalized Framework for Video Instance Segmentation. *arXiv preprint arXiv:2211.08834* (2022).

[15] Miran Heo, Sukjun Hwang, Jeongseok Hyun, Hanjung Kim, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. 2023. A generalized framework for video instance segmentation. In *CVPR*.

[16] Miran Heo, Sukjun Hwang, Seoung Wug Oh, Joon-Young Lee, and Seon Joo Kim. 2022. Vita: Video instance segmentation via object token association. *arXiv preprint arXiv:2206.04403* (2022).

[17] De-An Huang, Zhiding Yu, and Anima Anandkumar. 2022. Minvis: A minimal video instance segmentation framework without video-based training. *arXiv preprint arXiv:2208.02245* (2022).

[18] Sukjun Hwang, Miran Heo, Seoung Wug Oh, and Seon Joo Kim. 2021. Video instance segmentation using inter-frame communication transformers. In *NeurIPS*.

[19] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. 2019. Dynamic hypergraph neural networks.. In *IJCAI*.

[20] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. 2020. Video panoptic segmentation. In *CVPR*.

[21] Hanjung Kim, Jaehyun Kang, Miran Heo, Sukjun Hwang, Seoung Wug Oh, and Seon Joo Kim. 2025. VISAGE: Video Instance Segmentation with Appearance-Guided Enhancement. In *ECCV*.

[22] Junlong Li, Bingyao Yu, Yongming Rao, Jie Zhou, and Jiwen Lu. 2023. Tcovis: Temporally consistent online video instance segmentation. In *ICCV*.

[23] Xiangtai Li, Haobo Yuan, Wenwei Zhang, Guangliang Cheng, Jiangmiao Pang, and Chen Change Loy. 2023. Tube-link: A flexible cross tube baseline for universal video segmentation. *arXiv preprint arXiv:2303.12782* (2023).

[24] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. 2022. Video k-net: A simple, strong, and unified baseline for video segmentation. In *CVPR*.

[25] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. 2022. Large-scale video panoptic segmentation in the wild: A benchmark. In *CVPR*.

[26] Khai Nguyen, Shujian Zhang, Tam Le, and Nhat Ho. 2024. Sliced Wasserstein with random-path projecting directions. *arXiv preprint arXiv:2401.15889* (2024).

[27] Liping Nong, Jie Peng, Wenhui Zhang, Jiming Lin, Hongbing Qiu, and Junyi Wang. 2022. Adaptive multi-hypergraph convolutional networks for 3d object classification. *IEEE Transactions on Multimedia* 25 (2022), 4842–4855.

[28] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. 2022. Occluded video instance segmentation: A benchmark. *International Journal of Computer Vision* 130, 8 (2022), 2022–2039.

[29] Siyuan Qiao, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. 2021. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. In *CVPR*.

[30] Zheyun Qin, Xiankai Lu, Dongfang Liu, Xiushan Nie, Yilong Yin, Jianbing Shen, and Alexander C Loui. 2023. Reformulating graph kernels for self-supervised space-time correspondence learning. *IEEE Transactions on Image Processing* (2023).

[31] Inkyu Shin, Dahun Kim, Qihang Yu, Jun Xie, Hong-Seok Kim, Bradley Green, In So Kweon, Kuk-Jin Yoon, and Liang-Chieh Chen. 2023. Video-kMaX: A simple unified approach for online and near-online video panoptic segmentation. *arXiv preprint arXiv:2304.04694* (2023).

[32] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. 2021. End-to-end video instance segmentation with transformers. In *CVPR*.

[33] Sanghyun Woo, Dahun Kim, Joon-Young Lee, and In So Kweon. 2021. Learning to associate every segment for video panoptic segmentation. In *CVPR*.

[34] Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. 2022. Seqformer: Sequential transformer for video instance segmentation. In *ECCV*.

[35] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. 2022. In defense of online models for video instance segmentation. In *ECCV*.

[36] Jialian Wu, Sudhir Yarram, Hui Liang, Tian Lan, Junsong Yuan, Jayan Eledath, and Gerard Medioni. 2022. Efficient video instance segmentation via tracklet query and proposal. In *CVPR*.

[37] Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. 2019. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *NeurIPS* (2019).

[38] Linjie Yang, Yuchen Fan, and Ning Xu. 2019. Video instance segmentation. In *ICCV*.

[39] Shusheng Yang, Yuxin Fang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. 2021. Crossover learning for fast online video instance segmentation. In *ICCV*.

[40] Kaining Ying, Qing Zhong, Weian Mao, Zhenhua Wang, Hao Chen, Lin Yuanbo Wu, Yifan Liu, Chengxiang Fan, Yunzhi Zhuge, and Chunhua Shen. 2023. Ctvis: Consistent training for online video instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 899–908.

[41] Zitong Zhan, Daniel McKee, and Svetlana Lazebnik. 2022. Robust Online Video Instance Segmentation with Track Queries. *arXiv preprint arXiv:2211.09108* (2022).

[42] Renhong Zhang, Tianheng Cheng, Shusheng Yang, Haoyi Jiang, Shuai Zhang, Jiancheng Lyu, Xin Li, Xiaowen Ying, Dashan Gao, Wenyu Liu, et al. 2024. Mobileinst: Video instance segmentation on the mobile. In *AAAI*.

[43] Tao Zhang, Xingye Tian, Yu Wu, Shunping Ji, Xuebo Wang, Yuan Zhang, and Pengfei Wan. 2023. Dvis: Decoupled video instance segmentation framework. In *ICCV*.

[44] Tao Zhang, Xingye Tian, Yikang Zhou, Shunping Ji, Xuebo Wang, Xin Tao, Yuan Zhang, Pengfei Wan, Zhongyuan Wang, and Yu Wu. 2025. Dvis++: Improved decoupled framework for universal video segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

[45] Tianfei Zhou, Fatih Porikli, David J Crandall, Luc Van Gool, and Wenguan Wang. 2022. A survey on deep learning technique for video segmentation. *IEEE transactions on pattern analysis and machine intelligence* 45, 6 (2022), 7099–7122.

[46] Yikang Zhou, Tao Zhang, Shunping Ji, Shuicheng Yan, and Xiangtai Li. 2024. DVIS-DAQ: Improving Video Segmentation via Dynamic Anchor Queries. In *ECCV*.

[47] Xiaolin Zhu, Dongli Wang, Jianxun Li, Rui Su, Qin Wan, and Yan Zhou. 2024. Dynamical Attention Hypergraph Convolutional Network for Group Activity Recognition. *IEEE Transactions on Neural Networks and Learning Systems* (2024).