# (2.5+1)D Spatio-Temporal Scene Graphs for Video Question Answering





Anoop Cherian Chiori Hori Tim K. Marks Jonathan Le Roux

Mitsubishi Electric Research Labs (MERL), Cambridge, MA


**AAAI Virtual, 2022**

# Video Question Answering

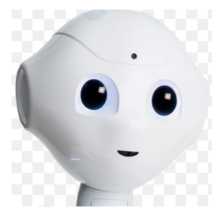
# Video Question Answering

**Question**:
Why did the book drop?

# Video Question Answering

Video from the NExT-QA dataset

**Question**:
Why did the book drop?

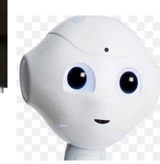


*Because the baby kicked it!*

NExTQA: Next Phase of Question-Answering to Explaining Temporal Actions, Xiao et al., CVPR, 2021

# Video Question Answering

Video from the NExT-QA dataset



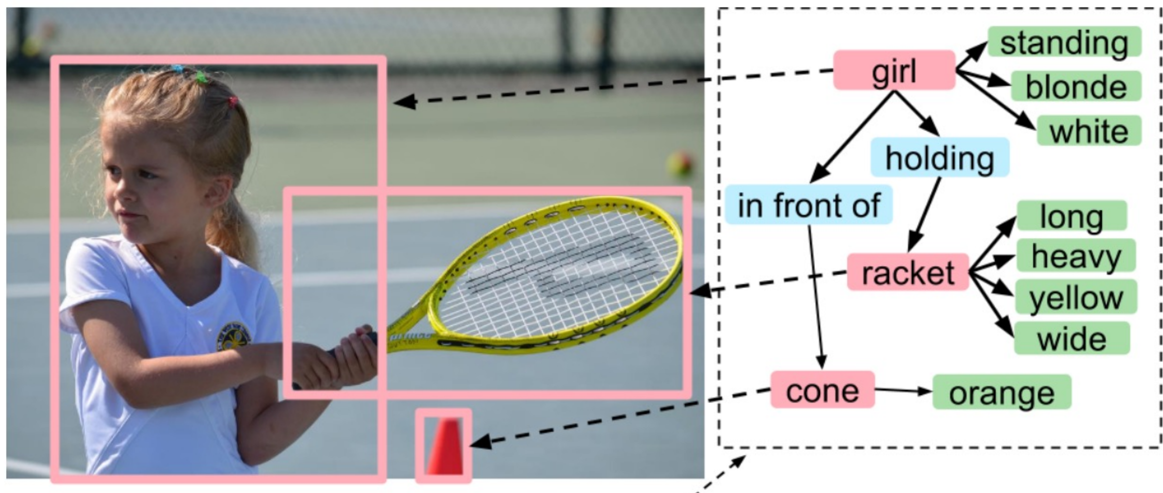
**Question**:
Why did the book drop?

Candidate answers

A1: open the cup
A2: baby kicked it
A3: the girl in pink slipped
A4: safety
A5: lady pushed it too hard

NExTQA: Next Phase of Question-Answering to Explaining Temporal Actions, Xiao et al., CVPR, 2021

# Visual Scene Graphs for Question Answering




Scene graphs hierarchically decompose a visual scene into objects, relations, and attributes, thus allowing efficient representation and inference.

It provides features as well as allows symbolic scene reasoning.

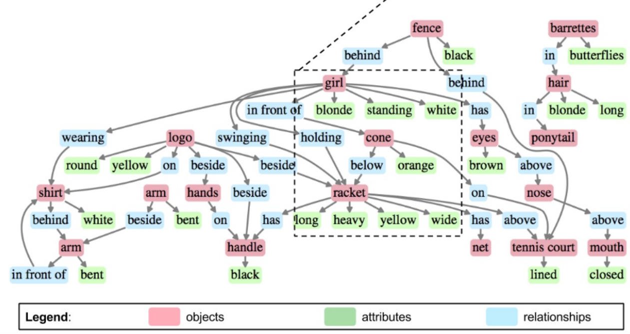
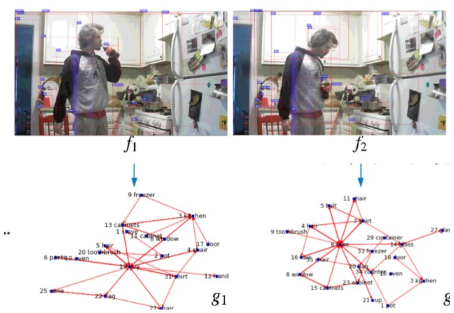
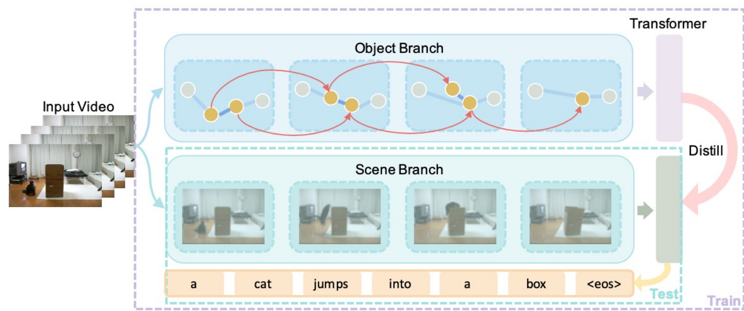



Image Generation from Scene Graphs, Johnson et al., CVPR'18

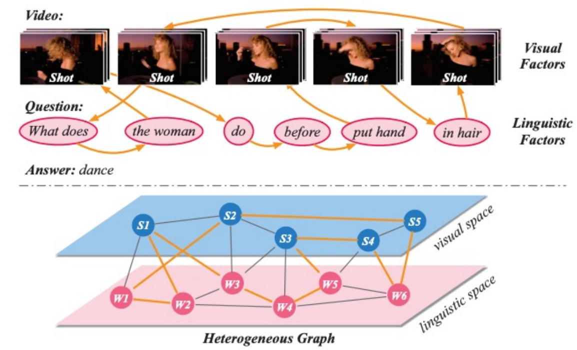
# Scene Graphs for Video Reasoning



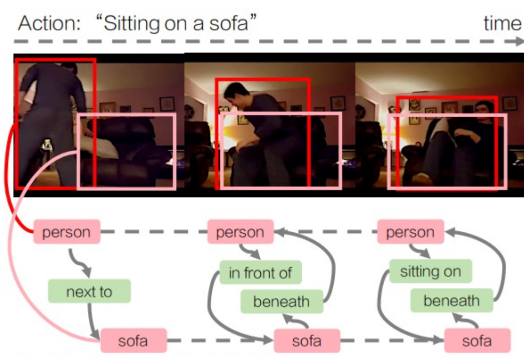

Scene Graphs + Graph Pooling
Geng et al., AAAI, 2021



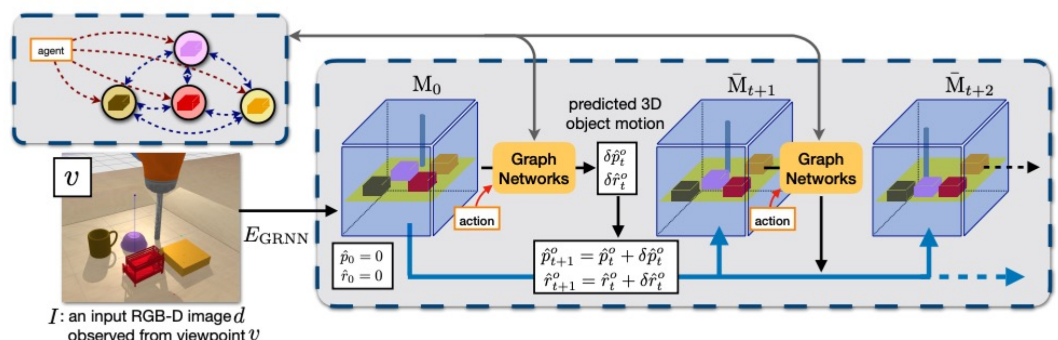

Scene Graphs + Knowledge Distillation
Pan et al., CVPR, 2020




Scene Graphs + Graph Alignment
Jiang and Han., AAAI, 2020

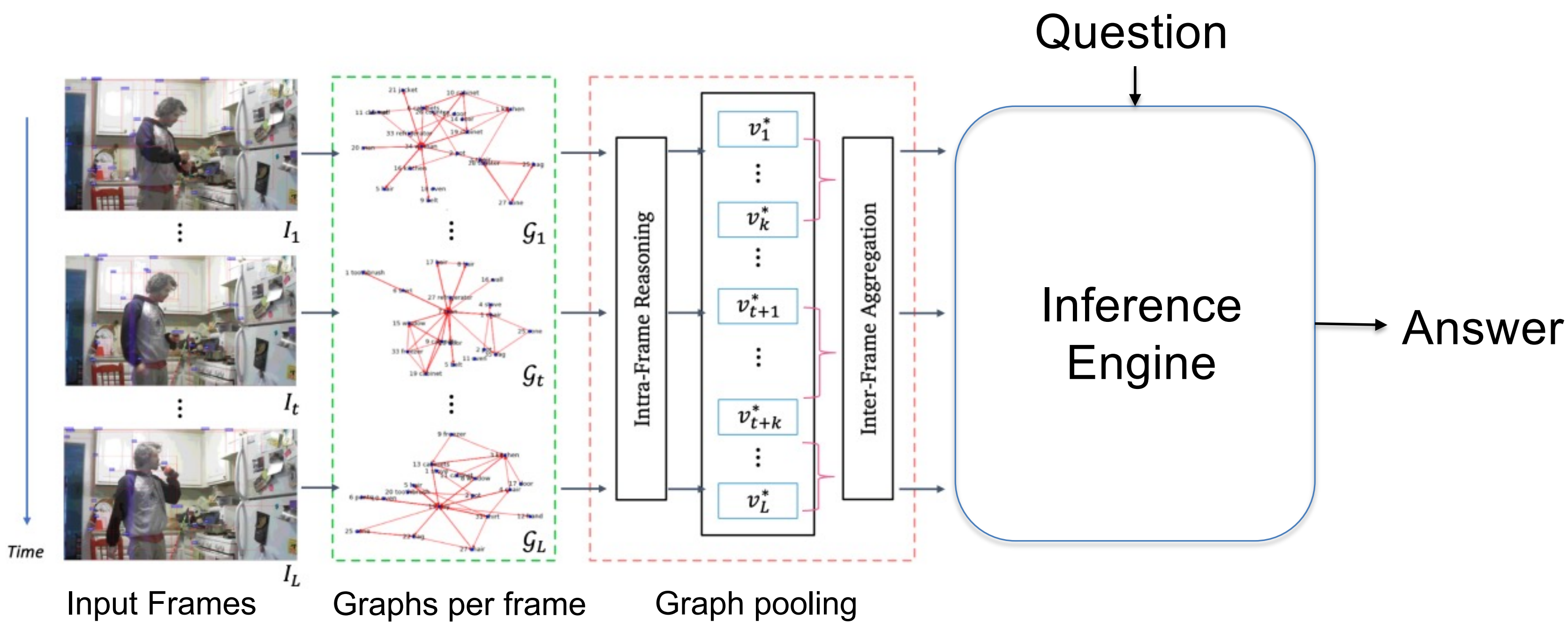



Scene Graphs for Action Recognition
Ji et al., CVPR, 2020




Scene Graphs for Video Prediction
Tung et al., CoRL, 2020

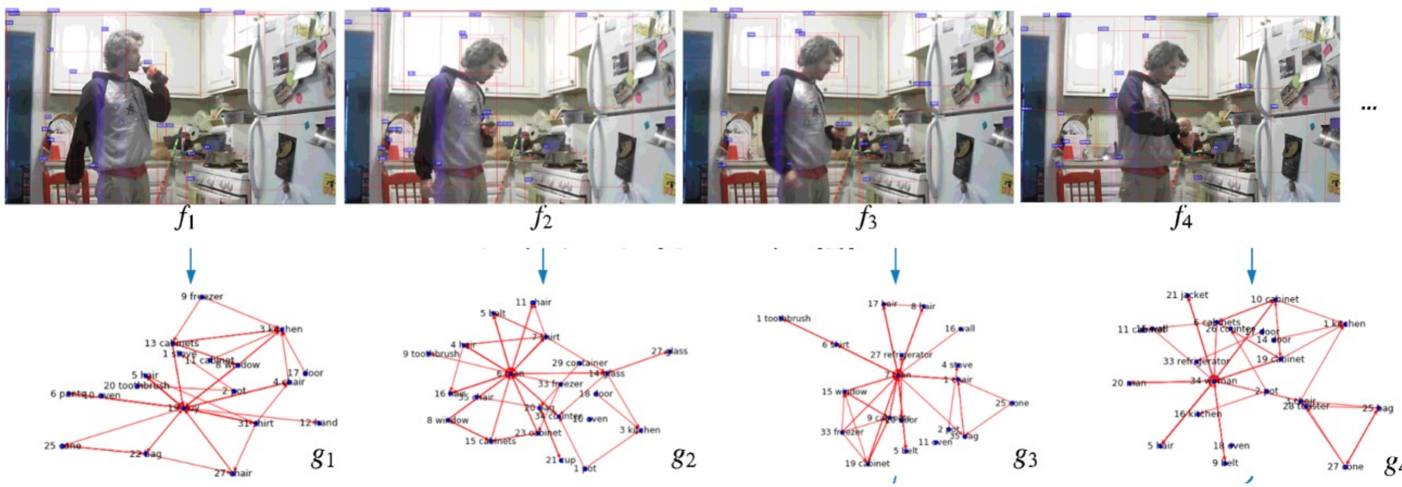
# Standard Video Question Answering Pipeline




Dynamic Graph Representation Learning for Video Dialog via Multi-Modal Shuffled Transformers, Geng et al., AAAI, 2021

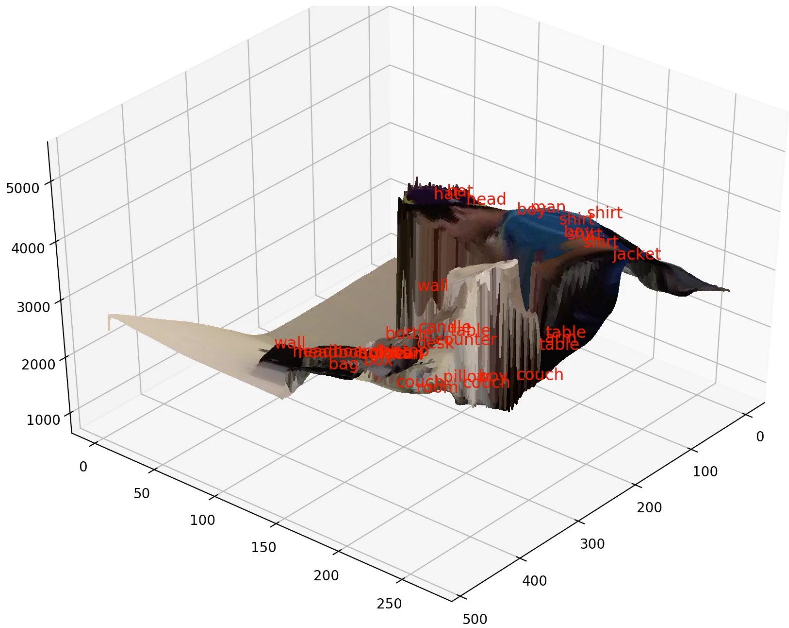
# Key Questions

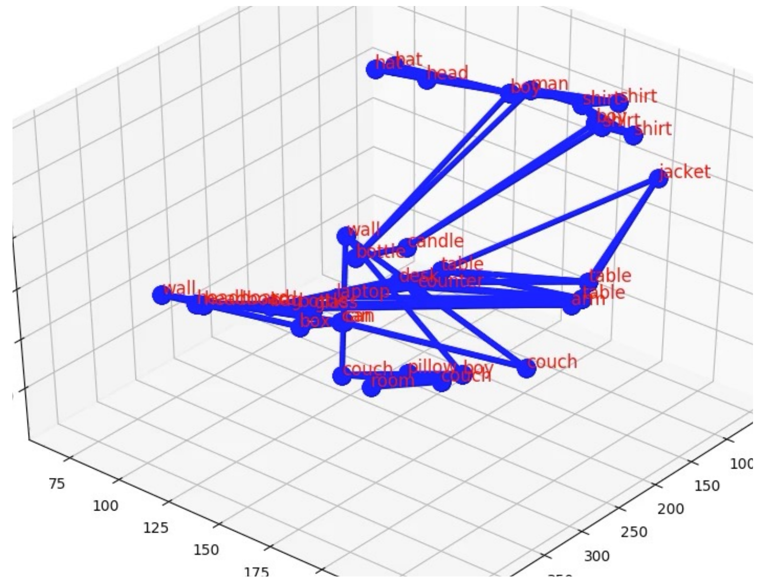
- Isn't constructing a **scene graph for every video frame redundant**? Usually several of the objects in the scene (and their relationships) will not change from frame to frame?
- Won't the learning and inference be **computationally challenging** for long video sequences if we create a scene graph for every frame?



# Key Insights

- Video frames are 2D views of a 3D space in which various events happen spatio-temporally. Can we use this **3D knowledge** to build a scene graph?



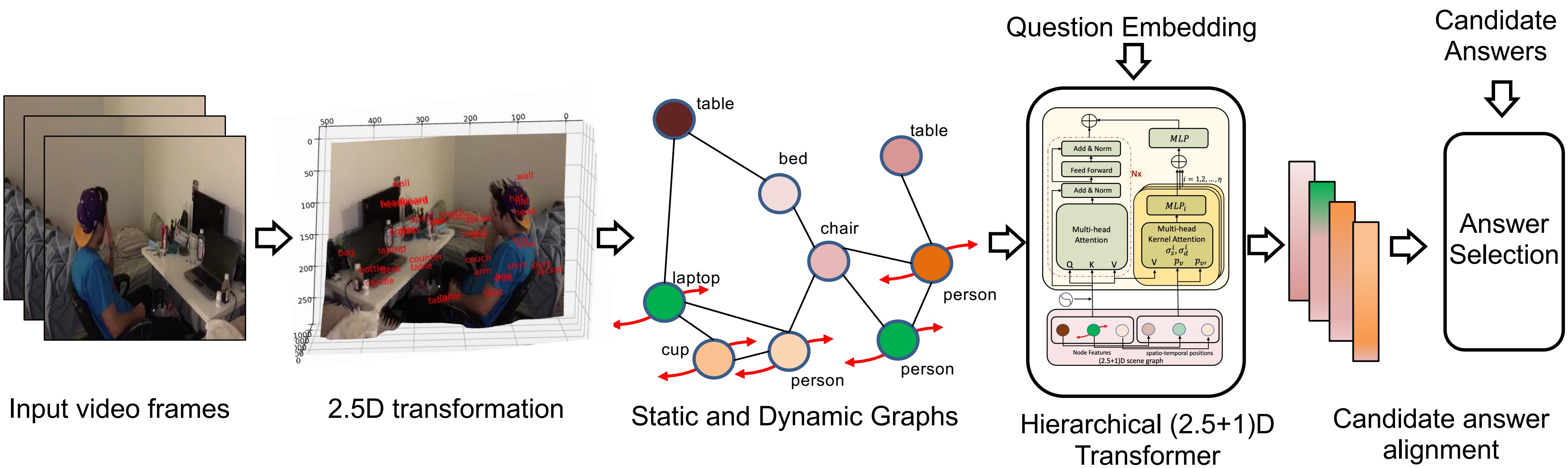



# Key Insights

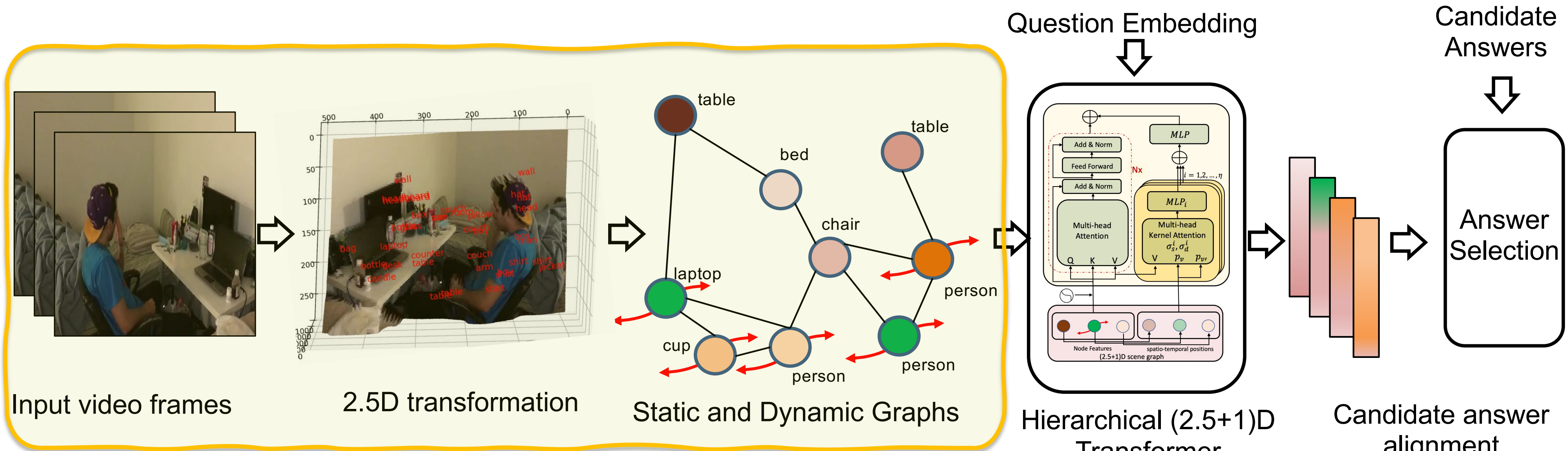
- Video frames are 2D views of a 3D space in which various events happen spatio-temporally. Can we use this **3D knowledge** to build a scene graph?

- Advantages:
  - A 3D scene graph could remove redundant object nodes
  - Objects are disentangled from their views and thus could help with occlusion reasoning (e.g., objects are visible in some views but not in all)
  - A smaller graph implies less memory footprint and faster training/inference

# Proposed Architecture: (2.5+1)D Scene Graph Reasoning

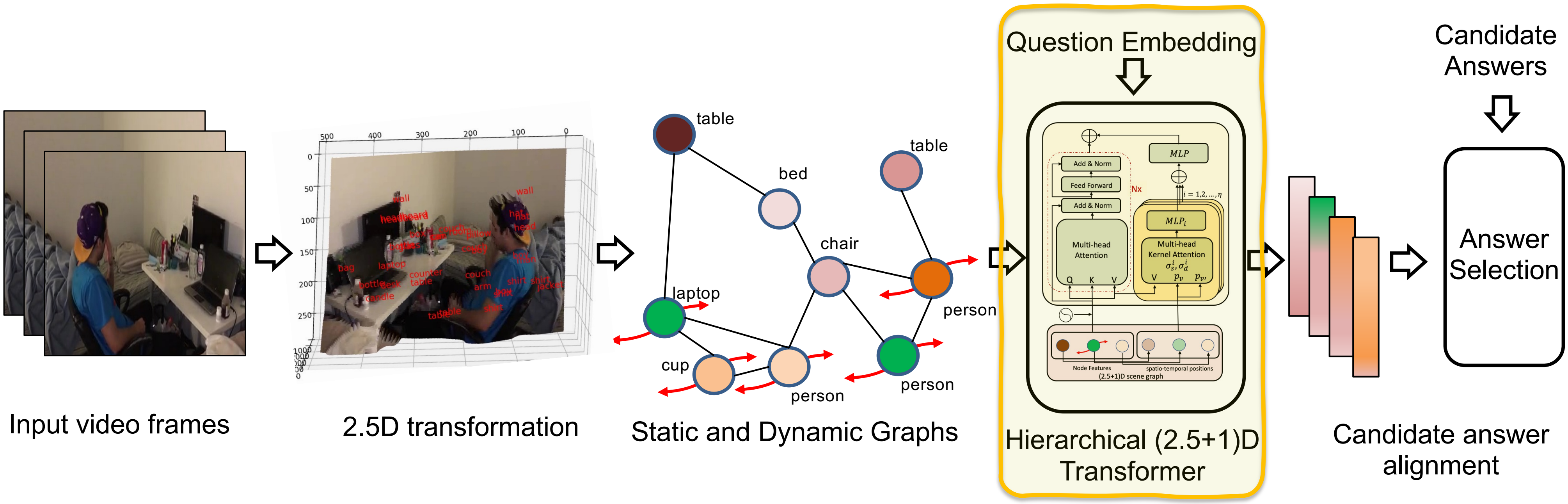


# Proposed Architecture: (2.5+1)D Scene Graph Reasoning



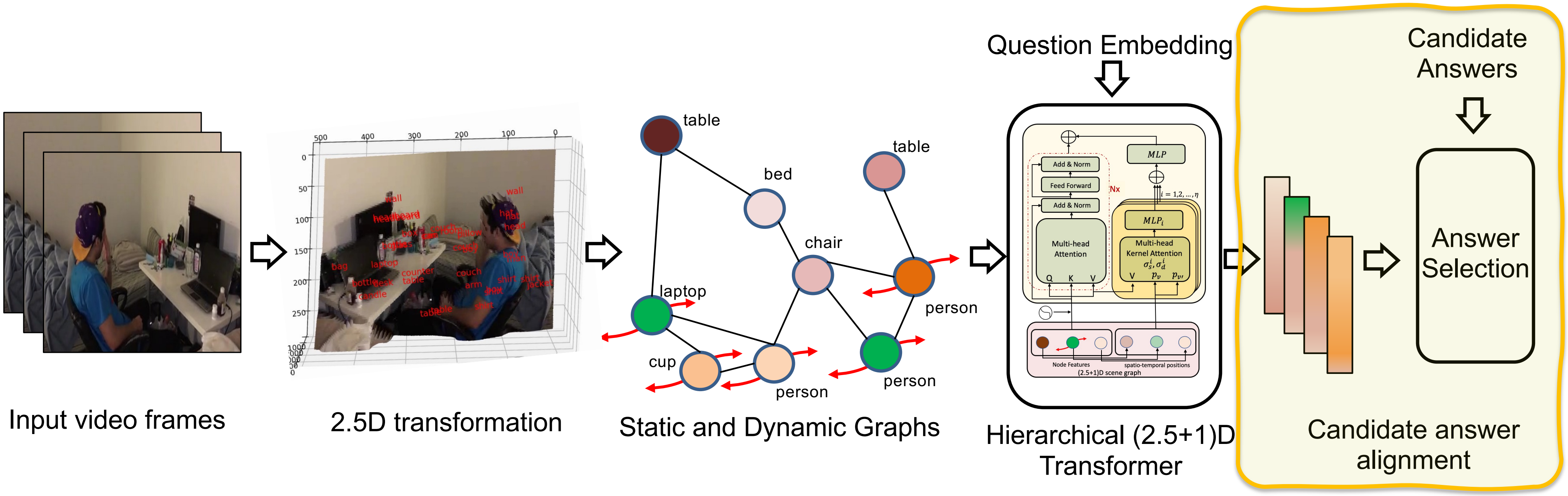

(2+1)D → (2.5+1)D Scene Graphs

# Proposed Architecture: (2.5+1)D Scene Graph Reasoning



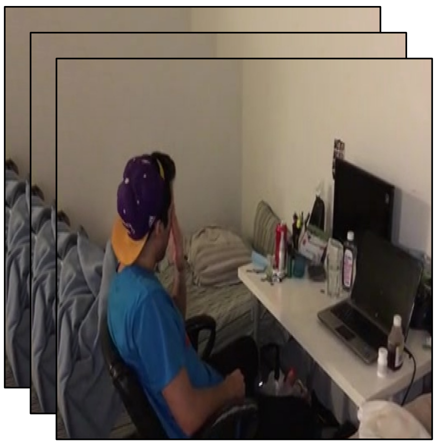

Fusion Engine

# Proposed Architecture: (2.5+1)D Scene Graph Reasoning

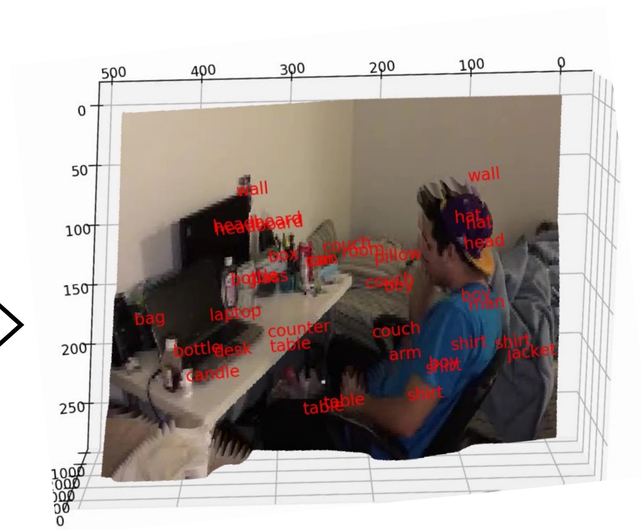



Inference Engine /Losses

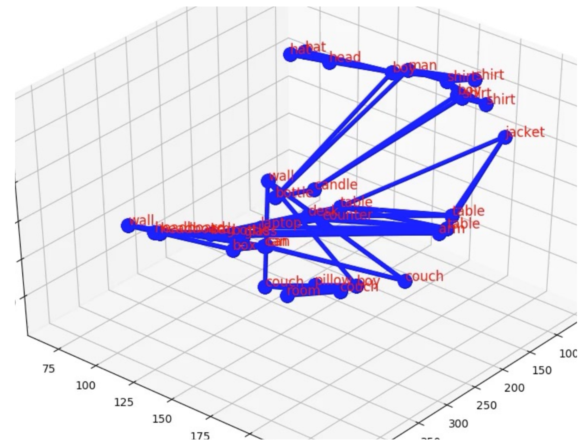
# 2D → 2.5D Scene Graphs Construction



Input video frames



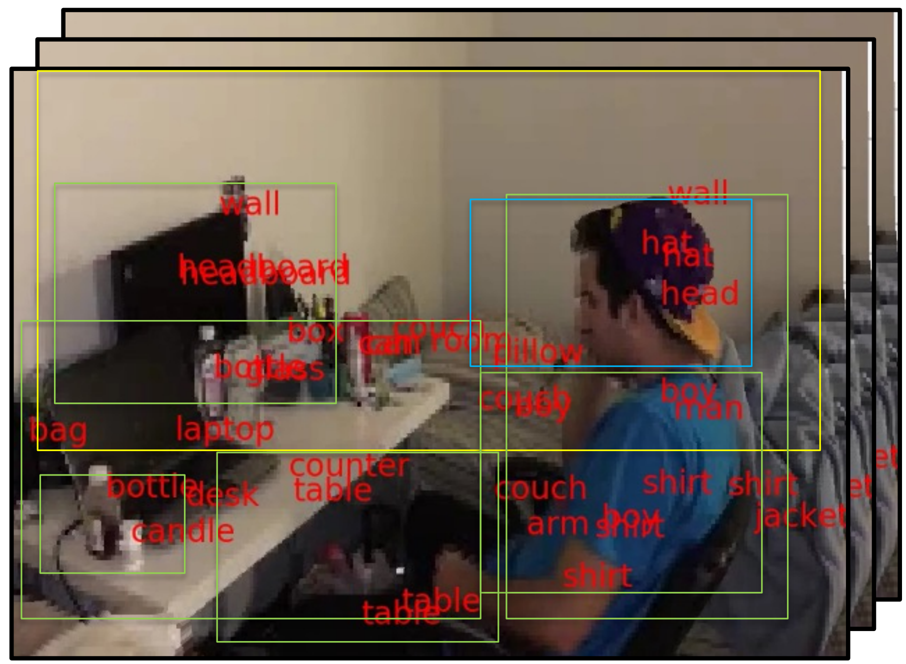
2.5D transformation



2.5D scene graphs

# 2D Scene Graphs

- For every video frame,
  - We use a Fast-RCNN object detector to find
    - the bounding boxes of objects
    - their object classes
    - and their feature vectors

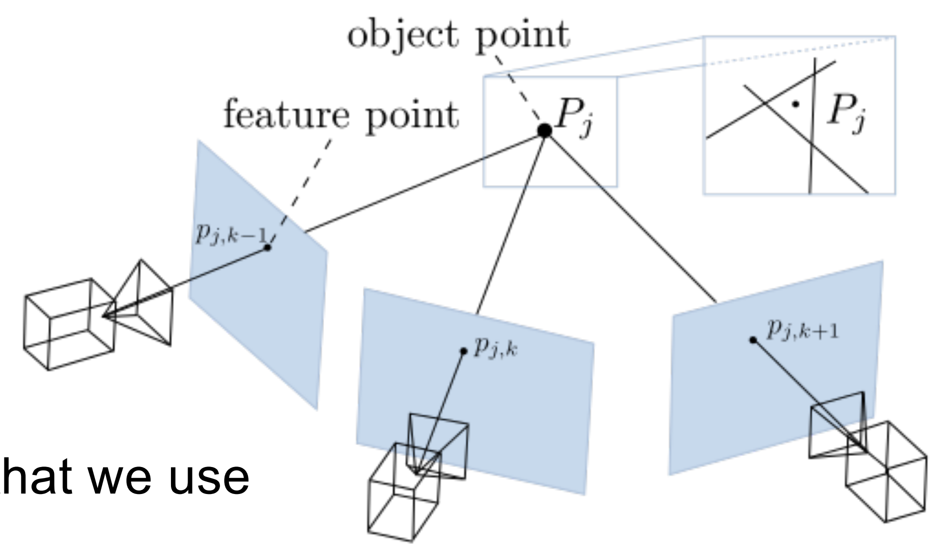



Each box (and its attributes) forms a node in the graph

But, what are the edges for the *graph*?
They will come later.

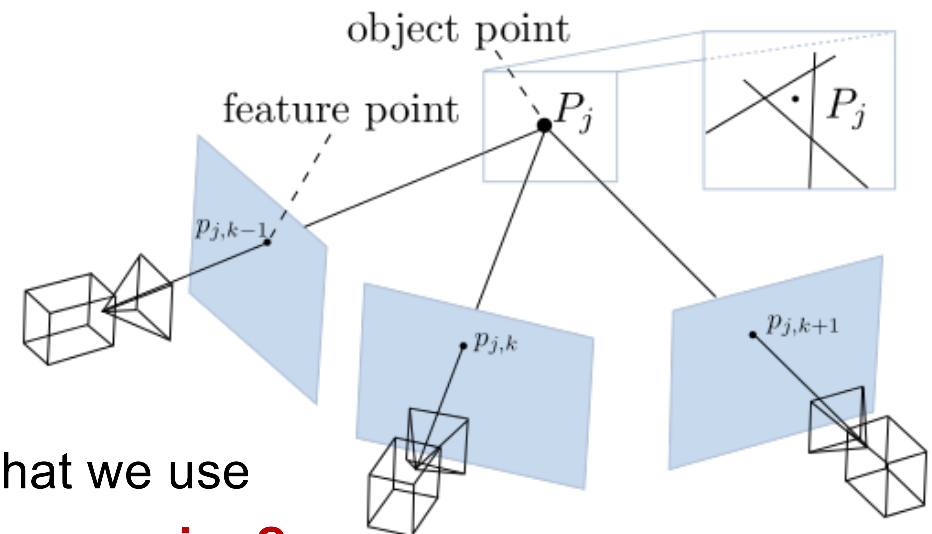
# (2.5+1)D Spatio-Temporal Scene Graphs

- How to remove the redundancy in the graph nodes from all video frames?
  - The video is a view of happenings in a 3D space
  - Ground each repeated 2D graph node to a single 3D graph node in a 3D space
- **Challenges:**
  - How to construct the 3D scene graph?
  - Usually needs
    - a static scene,
    - the camera parameters,
    - multiple overlapping views, etc.
  - Unavailable for arbitrary internet videos that we use



# (2.5+1)D Spatio-Temporal Scene Graphs

- How to remove the redundancy in the graph nodes from all video frames?
  - The video is a view of happenings in a 3D space
  - Ground each repeated 2D graph node to a single 3D graph node in a 3D space
- **Challenges:**
  - How to construct the 3D scene graph?
  - Usually needs
    - a static scene
    - the camera parameters
    - multiple overlapping views, etc.
  - Unavailable for arbitrary internet videos that we use




**Do we need an accurate reconstruction for reasoning?**

# 2D → 2.5D Scene Graphs

- For every video frame,
  - We use MiDAS Monocular → 3D pseudo-depth mapping algorithm
  - to produce a 2.5D approximate depth map and ground FRCNN bounding boxes in it
  - For each box, we use its 2.5D centroid as its 3D location attribute


Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer, Ranftl et al., PAMI, 2020

# (2.5+1)D Spatio-Temporal Scene Graphs

- To construct the 3D scene, we need to register the objects into a common 3D coordinate frame.
  - We split the FRCNN bounding boxes into two object classes
    - **Static**: that usually do not move in the scene (e.g., table, bed, window, etc.)
    - **Dynamic**: that may move in the scene (e.g., people, cup, car, etc.)
  - We use only the static object classes for registration to create the 3D scene.
  - We use the first video frame as the key frame and propose to progressively map all other frames to the coordinate frame defined by the first frame
  - For frames that do not have overlaps with the first frame (such as shot changes), we use the respective first frame of the shot as its coordinate frame

# Static Sub-Graphs

- We merge two static nodes into a single 2.5D static scene graph node, if they:
  - are temporally close, and
  - have the same object classes, and
  - bounding boxes overlap by more than $\gamma$, and
  - 3D object centroids are closest

$$\mathrm{C}(v_t, v_{t'}) := (c_{v_t} = c_{v_{t'}}) \wedge \mathrm{IoU}(\mathrm{bbox}_{v_t}, \mathrm{bbox}_{v_{t'}}) > \gamma$$

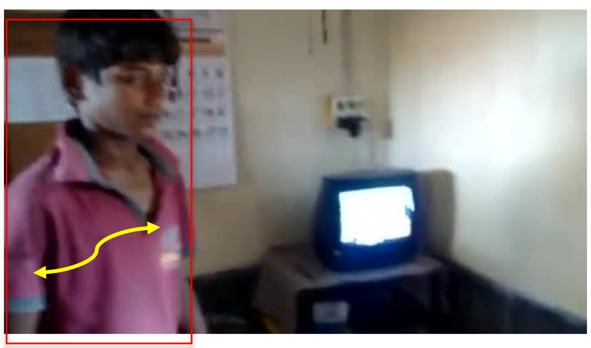
$$\mathrm{match}(v_t) = \underset{\substack{v_{t'} \in V^s_{t-\delta} \cup \cdots \cup V^s_{t-1} \\ \text{such that } \mathrm{C}(v_t, v_{t'}) = 1}}{\arg\min} \left\| p_{v_t} - p_{v_{t'}} \right\|$$

# Dynamic Sub-Graphs

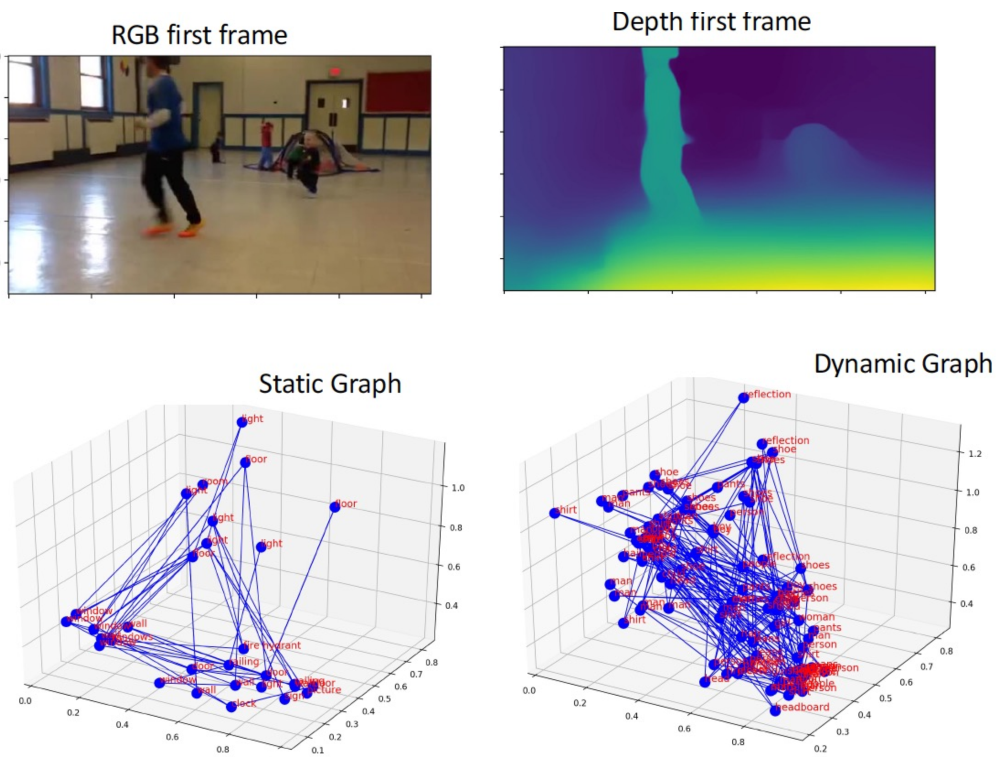
- For dynamic objects in the scene, we do not merge their scene graph nodes:
  - Since their informative cues may change from frame-to-frame
  - and their spatio-temporal dynamics are important for reasoning.

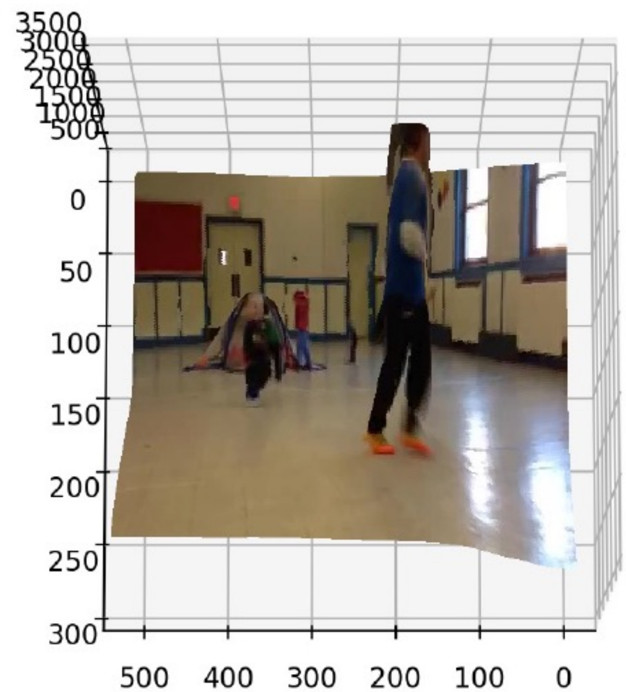
# Dynamic Sub-Graphs

- For dynamic objects in the scene, we do not merge their scene graph nodes:
  - Since their informative cues may change from frame-to-frame
  - and their spatio-temporal dynamics are important for reasoning.
  - We augment the frame-level FRCNN features of the dynamic objects with **motion features (I3D) capturing spatio-temporal dynamics** within these boxes.



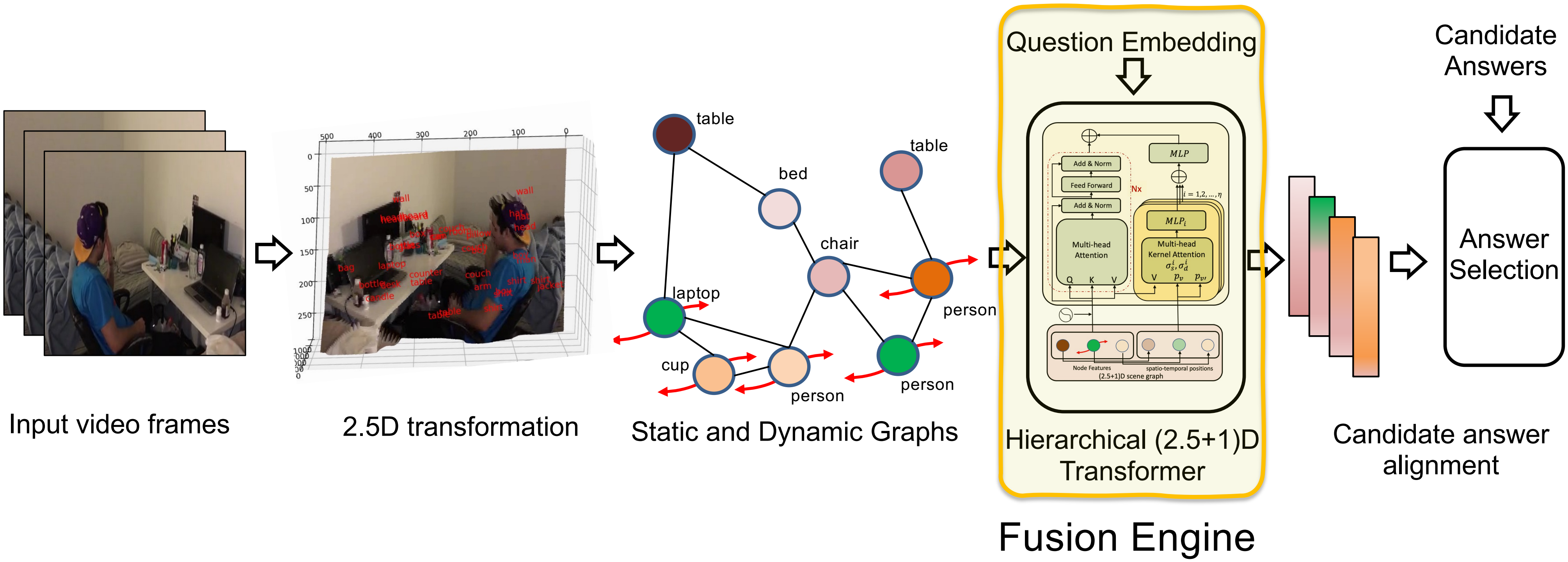
# (2.5+1)D Spatio-Temporal Scene Graphs




2.5D rendered scene using the depth map

# Hierarchical (2.5+1)D Transformer

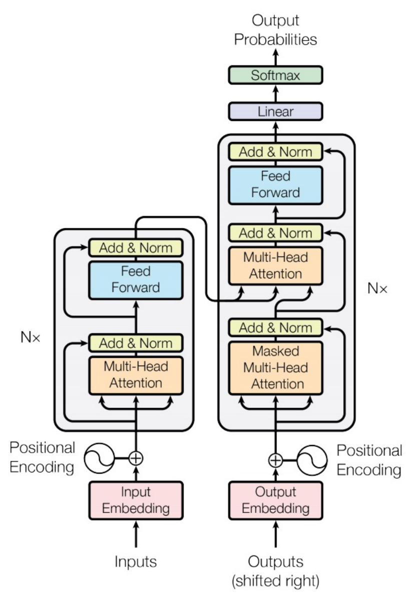


# Hierarchical (2.5+1)D Transformer

- Key idea:

  To augment a standard Transformer architecture with an attention model that captures the spatio-temporal proximity of the (2.5+1)D scene graph nodes.

Vaswani et al., NeurIPS, 2017




$$\Big\Vert_{i=1}^{k} \text{softmax}\left(\frac{\mathbf{Q}_F^i {\mathbf{K}_F^i}^\top}{\sqrt{r_k}}\right) \mathbf{V}_F^i$$

Multi-head Transformer Attention

**Multi-head (2.5+1)D Kernel Attention**

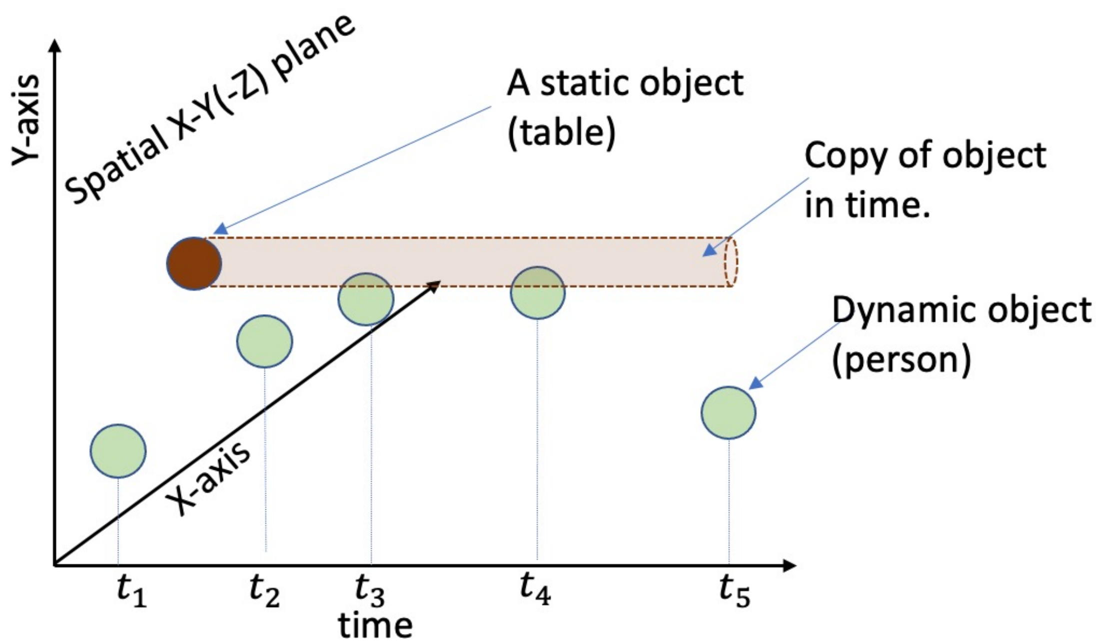
$$\Big\Vert_{i=1}^{k} \text{softmax}\, \mathbf{K}(V', V' | \sigma_s, \sigma_d) \mathbf{V}_F^i$$

$$\kappa(v_1, v_2 | \sigma_s, \sigma_t) = \exp\left(-\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2} - \frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t}\right),$$

**Multi-head (2.5+1)D Spatio-Temporal Kernel**

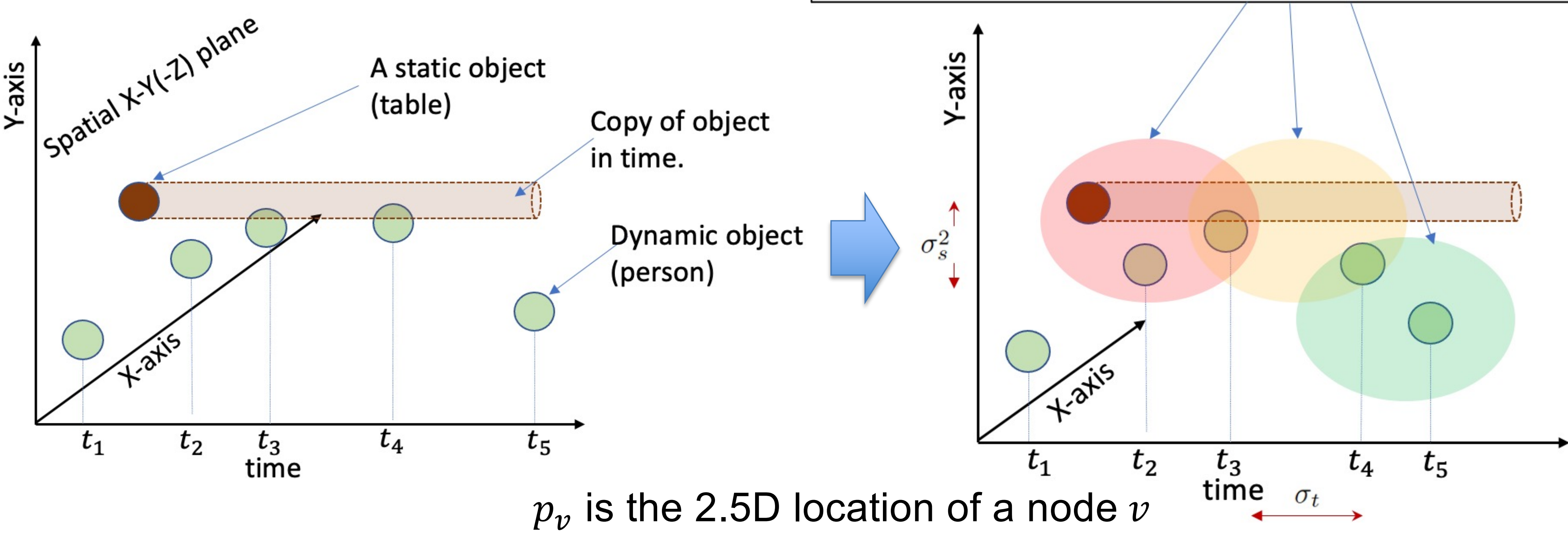
We use (2.5+1)D locations of queries and keys

# Spatio-Temporal Kernel Attention



# Spatio-Temporal Kernel Attention

$$\kappa(v_1, v_2|\sigma_s, \sigma_t) = \exp\left(-\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2} - \frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t}\right)$$

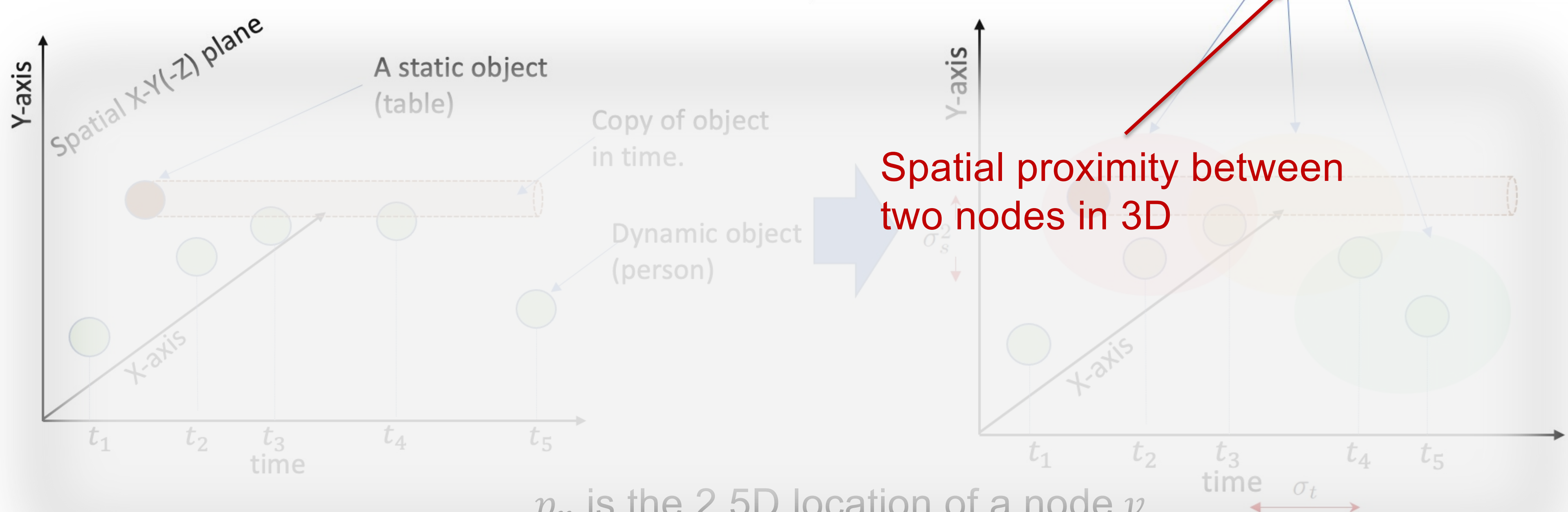



$p_v$ is the 2.5D location of a node $v$

# Spatio-Temporal Kernel Attention

$$\kappa(v_1, v_2|\sigma_s, \sigma_t) = \exp\left(-\boxed{\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2}} - \frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t}\right)$$

Spatial proximity between two nodes in 3D

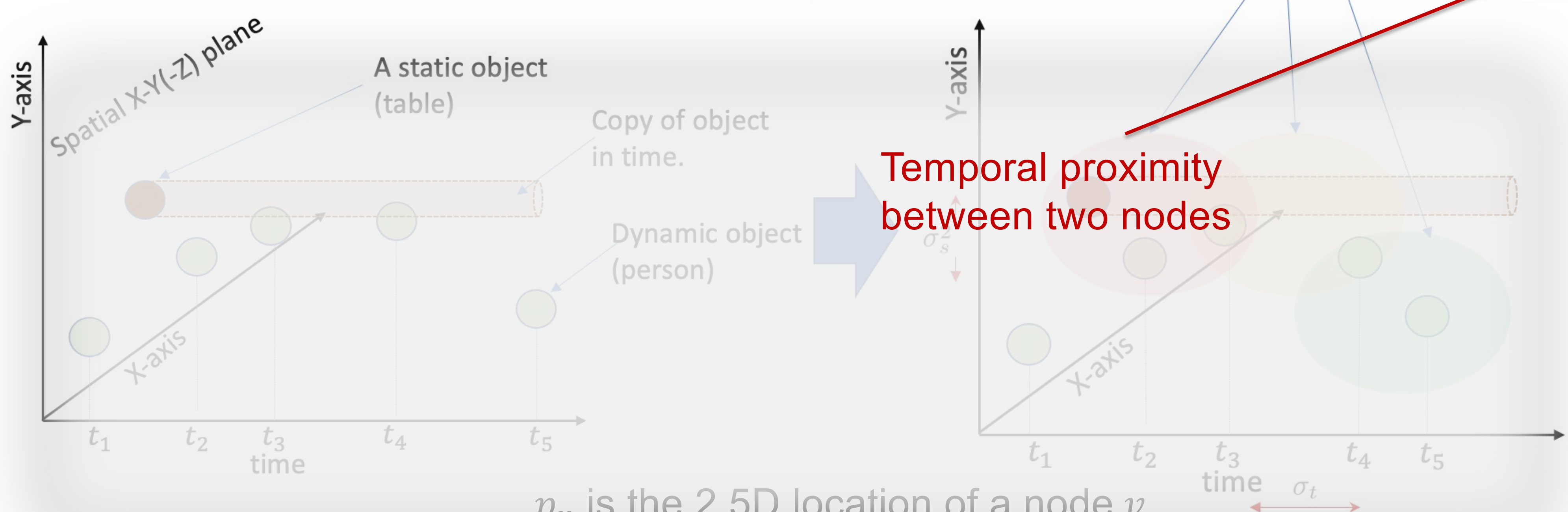



$p_v$ is the 2.5D location of a node $v$

# Spatio-Temporal Kernel Attention

$$\kappa(v_1, v_2|\sigma_s, \sigma_t) = \exp\left(-\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2} - \boxed{\frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t}}\right)$$

Temporal proximity between two nodes

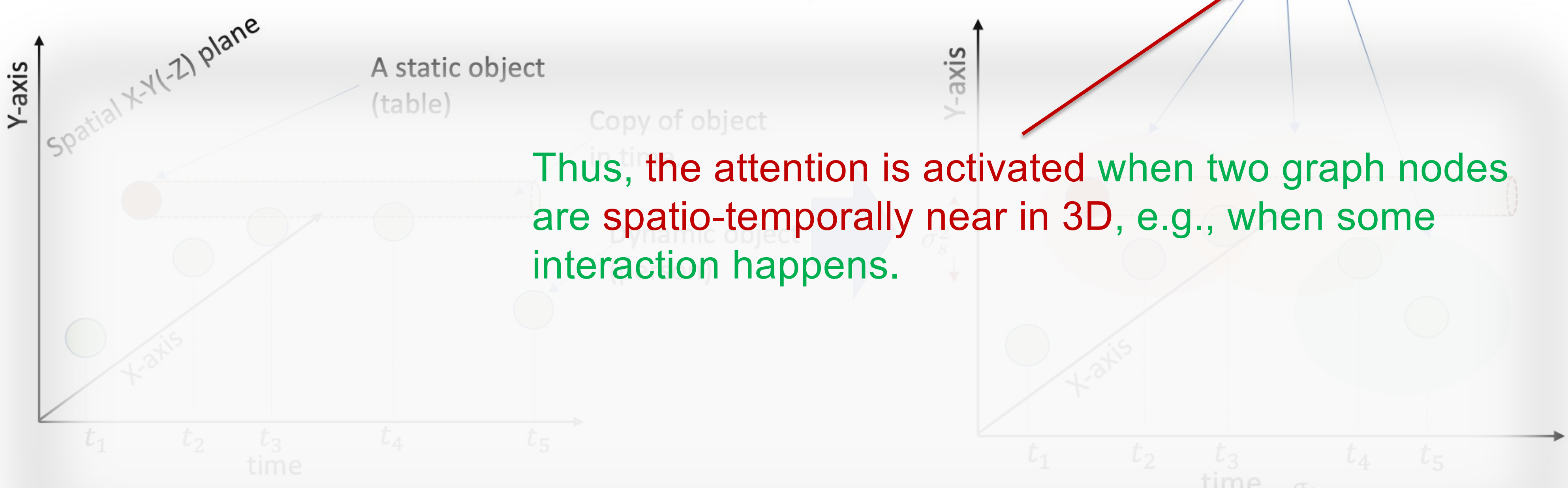



$p_v$ is the 2.5D location of a node $v$

# Spatio-Temporal Kernel Attention

$$F'_{3.5D} := \Big\Vert_{i=1}^{k} \operatorname{softmax} \mathbf{K}(V', V' | \sigma_s, \sigma_d) \mathbf{V}_F^i$$

$$\boxed{\kappa(v_1, v_2 | \sigma_s, \sigma_t) = \exp\left( - \frac{\left\| p_{v_1} - p_{v_2} \right\|^2}{\sigma_s^2} - \frac{\left\| t_{v_1} - t_{v_2} \right\|_1}{\sigma_t} \right)}$$



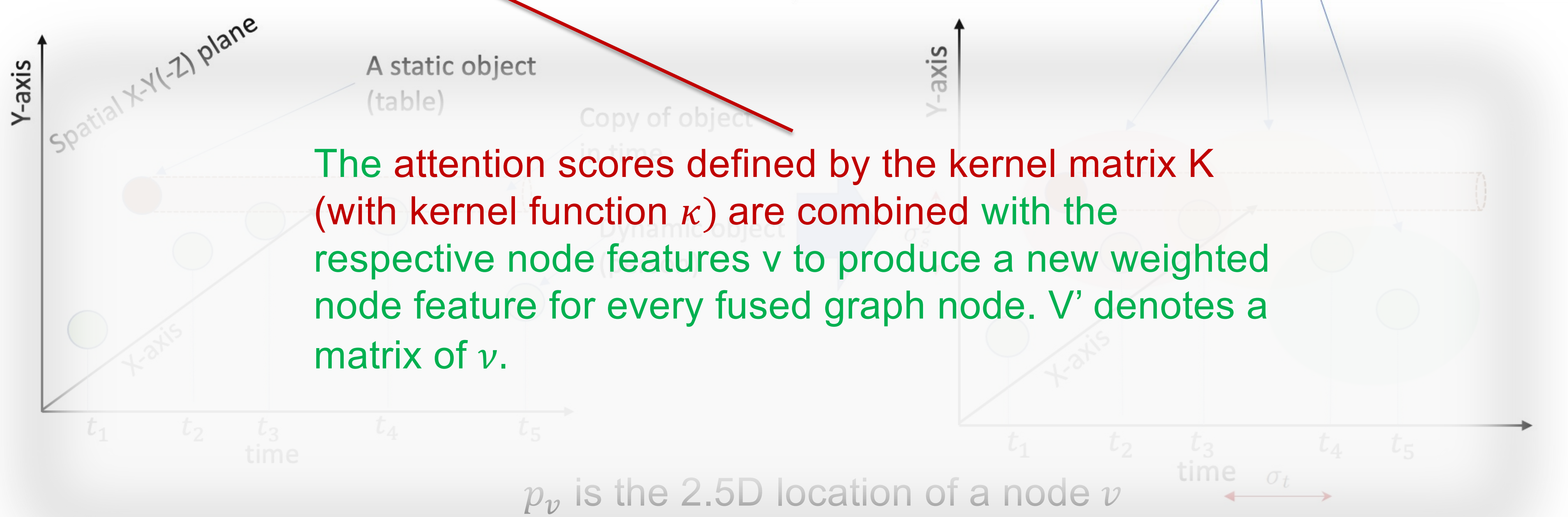

Thus, the attention is activated when two graph nodes are spatio-temporally near in 3D, e.g., when some interaction happens.

$p_v$ is the 2.5D location of a node $v$

# Spatio-Temporal Kernel Attention

$$F'_{3.5D} := \Big\Vert_{i=1}^{k} \mathrm{softmax}\, \mathbf{K}(V', V'|\sigma_s, \sigma_d)\mathbf{V}_F^i$$

$$\kappa(v_1, v_2|\sigma_s, \sigma_t) = \exp\left(-\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2} - \frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t}\right)$$



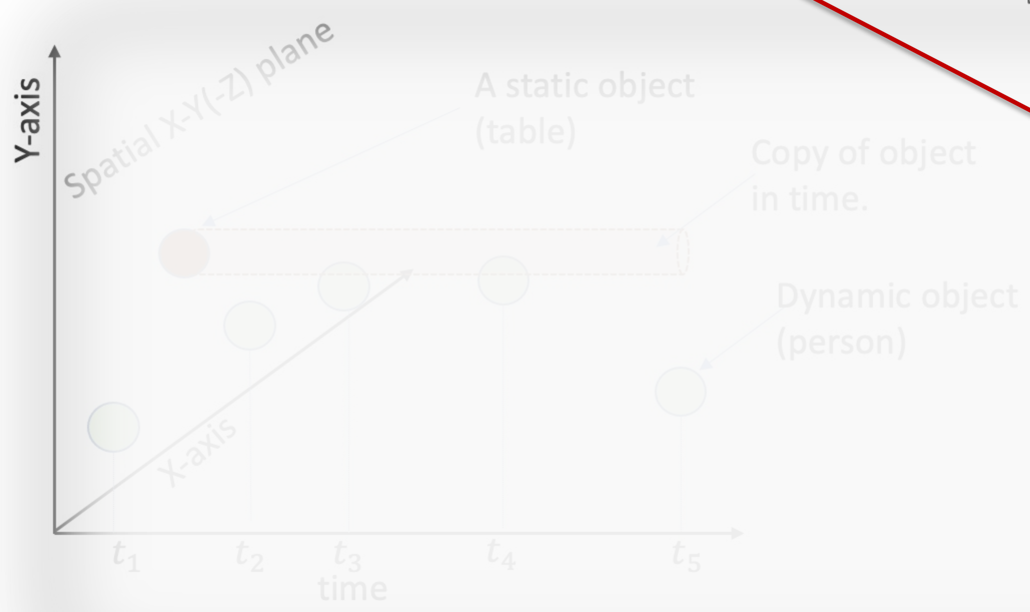

The attention scores defined by the kernel matrix K (with kernel function $\kappa$) are combined with the respective node features v to produce a new weighted node feature for every fused graph node. V' denotes a matrix of $v$.

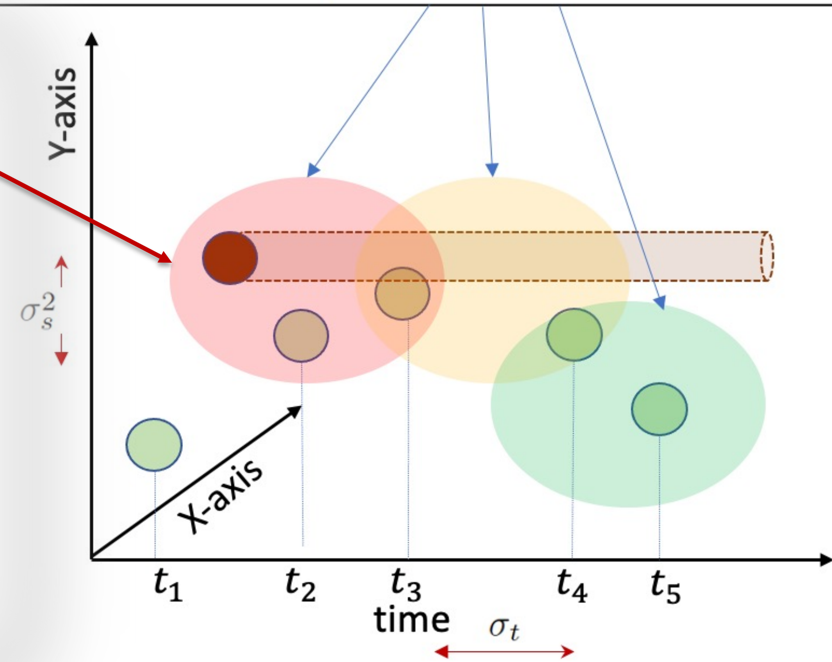
$p_v$ is the 2.5D location of a node $v$

# Spatio-Temporal Kernel Attention

$$F'_{3.5D} := \Big\|_{i=1}^{k} \operatorname{softmax} \mathbf{K}(V', V' | \sigma_s, \sigma_d) \mathbf{V}_F^i$$

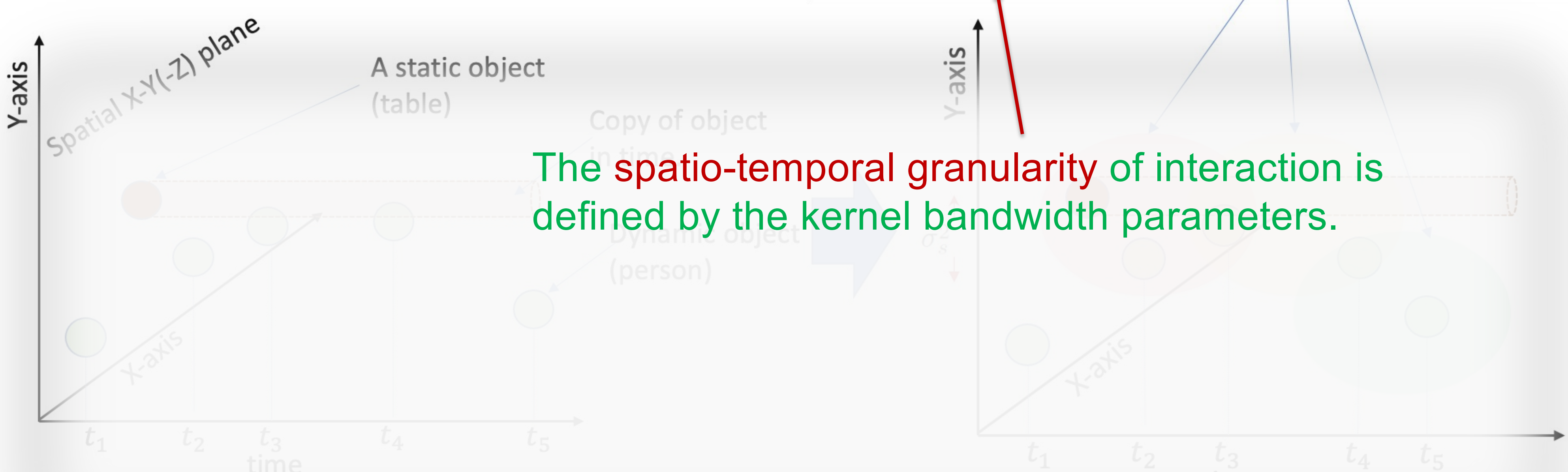
$$\kappa(v_1, v_2 | \sigma_s, \sigma_t) = \exp\left(-\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2} - \frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t}\right)$$



# Spatio-Temporal Kernel Attention

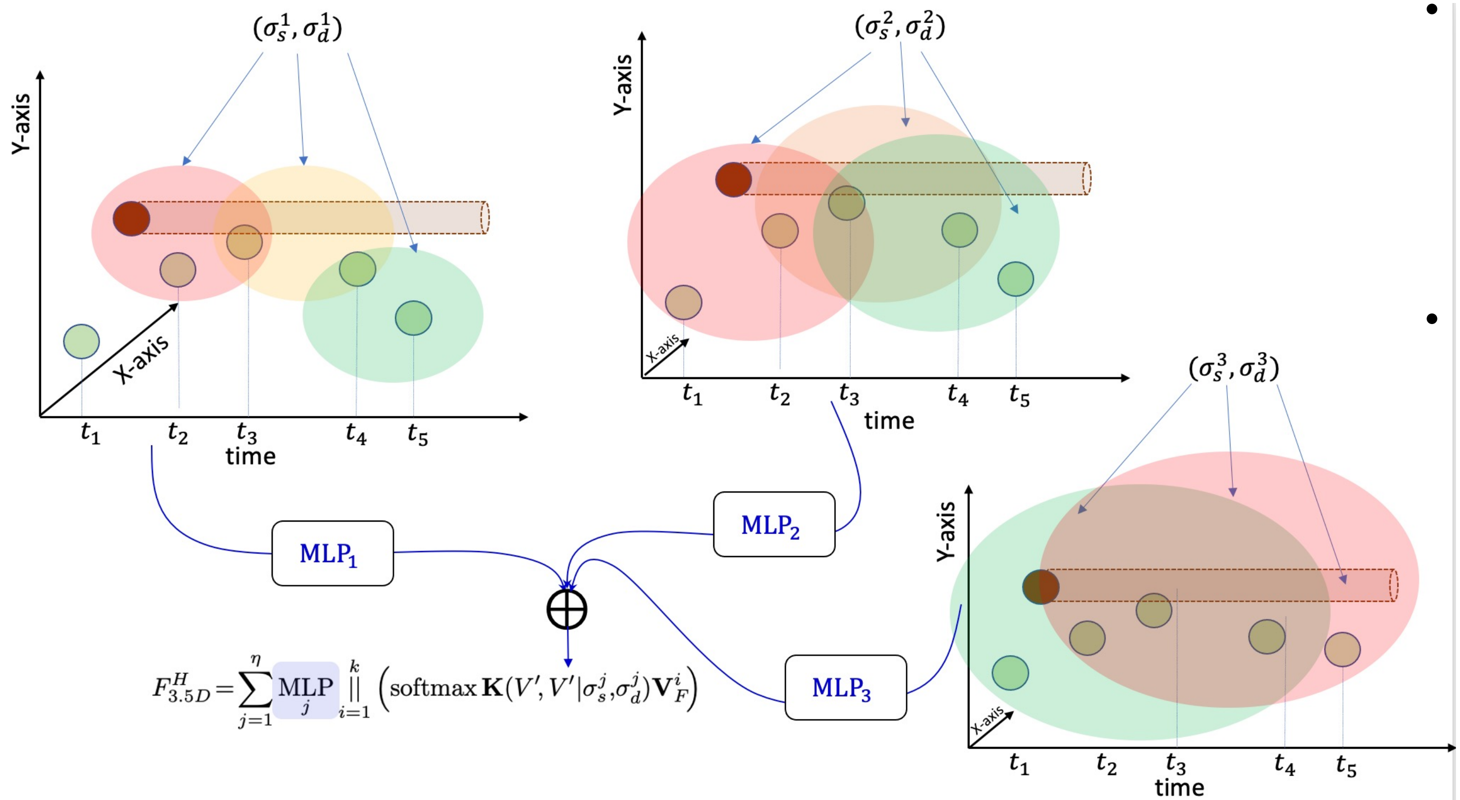
$$\kappa(v_1, v_2 | \sigma_s, \sigma_t) = \exp\left( -\frac{\|p_{v_1} - p_{v_2}\|^2}{\sigma_s^2} - \frac{\|t_{v_1} - t_{v_2}\|_1}{\sigma_t} \right)$$




The spatio-temporal granularity of interaction is defined by the kernel bandwidth parameters.

$p_v$ is the 2.5D location of a node $v$

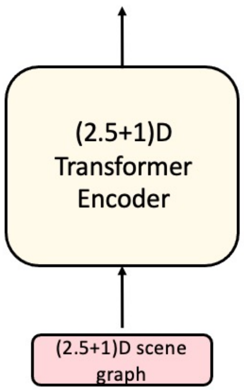
# Hierarchical Kernel Attention and Fusion

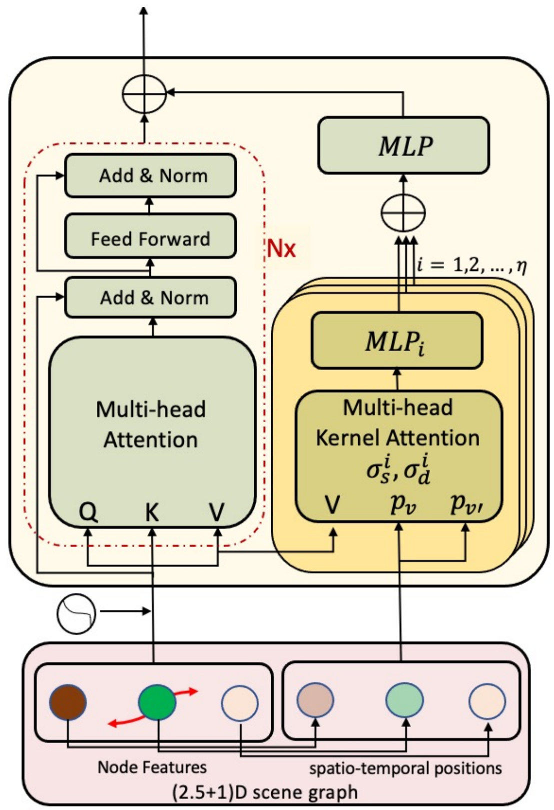



$$F^{H}_{3.5D} = \sum_{j=1}^{\eta} \underset{j}{\mathrm{MLP}} \Big\|_{i=1}^{k} \left(\mathrm{softmax}\,\mathbf{K}(V', V' | \sigma_s^j, \sigma_d^j)\mathbf{V}_F^i\right)$$

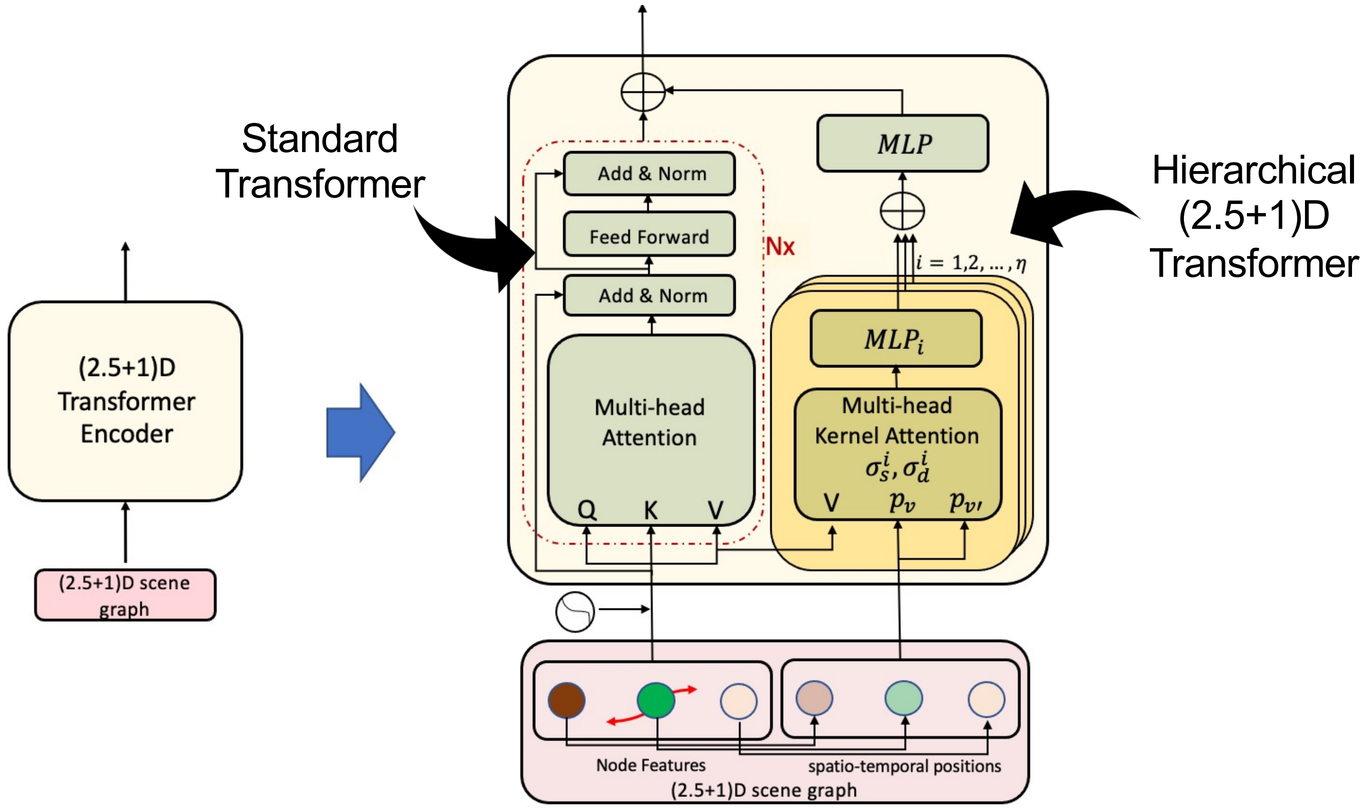
- Apply kernel attention at different granularity, each capturing interaction at different scales.
- Then, fuse the interaction features via an MLP.

# Hierarchical (2.5+1)D Transformer

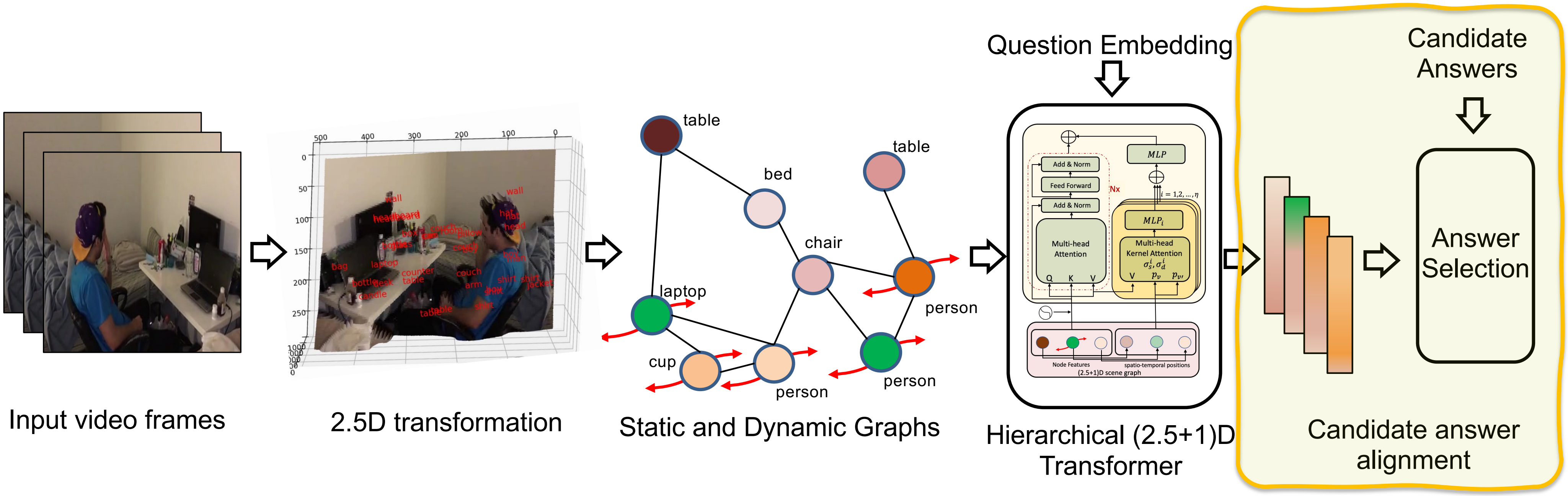


# Hierarchical (2.5+1)D Transformer



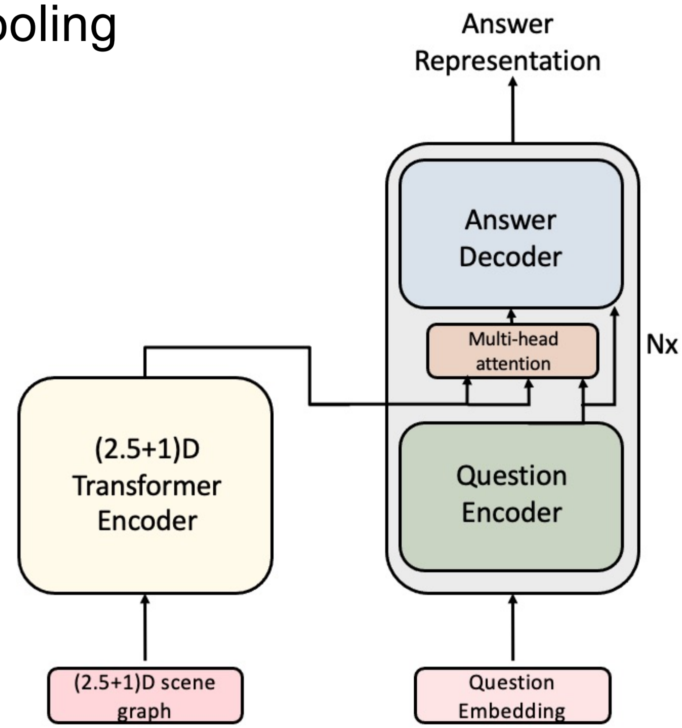
# Inference Engine




Inference Engine /Losses

# Inference Engine / Losses

- The *provided question* and the *candidate answers* are encoded using a multi-head standard Transformer followed by average pooling
- The model is trained using:
  – Softmax cross-entropy loss, and
  – Contrastive loss between the *embeddings* of correct answer and all candidates in a *batch*




The question embeddings are used to condition the (2.5+1)D fused graph features to generate an answer representation, that is cosine-aligned with candidate answers, selecting the best match.

# Experiments and Results

# Experiments: Datasets

- We used two datasets:
  - NExT-QA: Xiao et al., CVPR, 2020
    - A recent video QA dataset that goes beyond traditional VQA tasks
    - incorporates a significant number of why and how questions
    - consists of 3,870 training, 570 validation, and 1,000 test videos.
    - the task is to select one of the five candidate answers.
  - AVSD-QA: Alamri et al., CVPR, 2019
    - A variant of the Audio-Visual Scene Aware Dialog for the QA task.
    - consists of 7,985, 1,863, and 1,968 for training, validation, and test.
    - We use only the video features for this dataset (not dialog, text, or audio)

We follow standard training practices and report on standard evaluation metrics

# Quantitative Results

| Method | Accuracy (%)↑ |
|---|---|
| Spatio-Temporal VQA (Jang et al. 2019) | 47.94 |
| Co-Memory-QA (Gao et al. 2018) | 48.04 |
| Hier. Relation n/w (Le et al. 2020) | 48.20 |
| Multi-modal Attn VQA (Fan et al. 2019) | 48.72 |
| graph-alignment VQA (Jiang and Han 2020) | 49.74 |
| (2.5+1)D-Transformer (ours) | **53.40** |

NExT-QA

| Method | Mean Rank ↓ |
|---|---|
| Question Only (Alamri et al. 2019a) | 7.63 |
| Multimodal Transformers (Hori et al. 2019) | 7.23 |
| Question + Video (Alamri et al. 2019a) | 6.86 |
| MTN (Le et al. 2019) | 6.85 |
| ST Scene Graphs (Geng et al. 2021) | 5.91 |
| (2.5+1)D-Transformer (ours) | **5.84** |

AVSD-QA

| Method | Why (W) | How (H) | Avg. (W+H) | Prev&Next (P&N) | Present (P) | Avg. (P&N+P) | Count (C) | Location (L) | Other (O) | Avg. (C+L+O) | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|
| STVQA, IJCV'19 | 45.37 | 43.05 | 44.76 | 47.52 | 51.73 | 49.26 | 43.50 | 65.42 | 53.77 | 55.86 | 47.94 |
| CoMem, CVPR'18 | 46.15 | 42.61 | 45.22 | 48.16 | 50.38 | 49.07 | 41.81 | 67.12 | 51.80 | 55.34 | 48.04 |
| HCRN, CVPR'20 | 46.99 | 42.90 | 45.91 | 48.16 | 50.83 | 49.26 | 40.68 | 65.42 | 49.84 | 53.67 | 48.20 |
| HME, CVPR'19 | 46.52 | 45.24 | 46.18 | 47.52 | 49.17 | 48.20 | 45.20 | 73.56 | 51.15 | 58.30 | 48.72 |
| HGA, AAAI'20 | 46.99 | 44.22 | 46.26 | 49.53 | 52.49 | 50.74 | 44.07 | 72.54 | 55.41 | 59.33 | 49.74 |
| Ours | **52.39** | **48.36** | **51.33** | **50.91** | **54.28** | 52.30 | **46.02** | **77.08** | **58.31** | **62.58** | **53.4** |
| % improvement | +5.4 | +3.12 | +5.07 | +1.38 | +1.79 | +1.56 | +0.82 | +3.52 | +2.91 | +3.25 | +3.66 |

Performances on individual question classes in the NExT-QA dataset

# Ablative Results

| Method | NExT-QA Acc (%)↑ | AVSD-QA mean rank↓ |
|---|---|---|
| No dynamic graph | 52.49 | 5.97 |
| No static graph | 53.00 | 6.03 |
| No I3D | 52.65 | 6.09 |
| No hier. kernel | 52.90 | 5.97 |
| No ans. augment | 49.98 | 5.92 |
| No question condition | 50.39 | 5.96 |
| Full Model | **53.40** | **5.84** |

# Ablative Results

| Method | NExT-QA Acc (%)↑ | AVSD-QA mean rank↓ |
|---|---|---|
| No dynamic graph | 52.49 | 5.97 |
| No static graph | 53.00 | 6.03 |
| No I3D | 52.65 | 6.09 |
| No hier. kernel | 52.90 | 5.97 |
| No ans. augment | 49.98 | 5.92 |
| No question condition | 50.39 | 5.96 |
| Full Model | **53.40** | **5.84** |

| # | Ablation | Accuracy (%)↑ |
|---|---|---|
| 1 | Txr + I3D + FRCNN + QC | 47.90 |
| 2 | (1) + Ans. Aug. | 49.80 |
| 3 | Txr + V(2+1)D Txr + Ans. Aug. + QC | 52.40 |
| 4 | Txr + V(2.5+1)D Txr + Ans. Aug. + QC | 53.40 |
| 5 | (4) using all nodes (no pruning) | **53.50** |

# Ablative Results

| Method | NExT-QA Acc (%)↑ | AVSD-QA mean rank↓ |
|---|---|---|
| No dynamic graph | 52.49 | 5.97 |
| No static graph | 53.00 | 6.03 |
| No I3D | 52.65 | 6.09 |
| No hier. kernel | 52.90 | 5.97 |
| No ans. augment | 49.98 | 5.92 |
| No question condition | 50.39 | 5.96 |
| Full Model | **53.40** | **5.84** |

| # | Ablation | Accuracy (%)↑ |
|---|---|---|
| 1 | Txr + I3D + FRCNN + QC | 47.90 |
| 2 | (1) + Ans. Aug. | 49.80 |
| 3 | Txr + V(2+1)D Txr + Ans. Aug. + QC | 52.40 |
| 4 | Txr + V(2.5+1)D Txr + Ans. Aug. + QC | 53.40 |
| 5 | (4) using all nodes (no pruning) | **53.50** |

| Hier. levels | bandwidths $\sigma$ | Accuracy |
|---|---|---|
| 1-level | $0.01$ | 52.13 |
| 2-levels | $\{0.01, 0.1\}$ | 52.58 |
| 3-levels | $\{0.01, 0.1, 1.0\}$ | 52.97 |
| 4-levels | $\{0.01, 0.1, 1.0, 10.0\}$ | 53.20 |
| 5-levels | $\{0.01, 0.1, 1.0, 10, 20.0\}$ | 53.00 |

# Ablative Results

| | NExT-QA | AVSD-QA |
|---|---|---|
| Method | Acc (%)↑ | mean rank↓ |
| No dynamic graph | 52.49 | 5.97 |
| No static graph | 53.00 | 6.03 |
| No I3D | 52.65 | 6.09 |
| No hier. kernel | 52.90 | 5.97 |
| No ans. augment | 49.98 | 5.92 |
| No question condition | 50.39 | 5.96 |
| Full Model | **53.40** | **5.84** |

| # | Ablation | Accuracy (%)↑ |
|---|---|---|
| 1 | Txr + I3D + FRCNN + QC | 47.90 |
| 2 | (1) + Ans. Aug. | 49.80 |
| 3 | Txr + V(2+1)D Txr + Ans. Aug. + QC | 52.40 |
| 4 | Txr + V(2.5+1)D Txr + Ans. Aug. + QC | 53.40 |
| 5 | (4) using all nodes (no pruning) | **53.50** |

| Hier. levels | bandwidths $\sigma$ | Accuracy |
|---|---|---|
| 1-level | 0.01 | 52.13 |
| 2-levels | $\{0.01, 0.1\}$ | 52.58 |
| 3-levels | $\{0.01, 0.1, 1.0\}$ | 52.97 |
| 4-levels | $\{0.01, 0.1, 1.0, 10.0\}$ | 53.20 |
| 5-levels | $\{0.01, 0.1, 1.0, 10, 20.0\}$ | 53.00 |

| | AVSD-QA | NExT-QA |
|---|---|---|
| Full graph | 502.43 | 656.30 |
| Static graph | 97.26 | 68.68 |
| Dynamic graph | 136.10 | 430.83 |
| % node reduction | 53.6 | 23.9 |

# Qualitative Results

**What does** the man in grey do after sitting down int the middle?

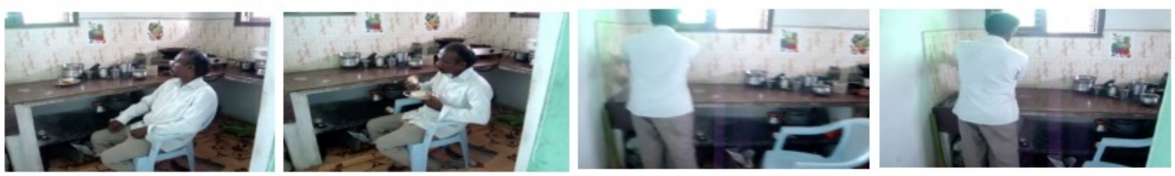
A1: talking on phone
A2: take the pipe
A3: smiling
A4: smell burger
A5: cross his legs

GT: smell burger
Ours: smell burger
HGA: take the pipe

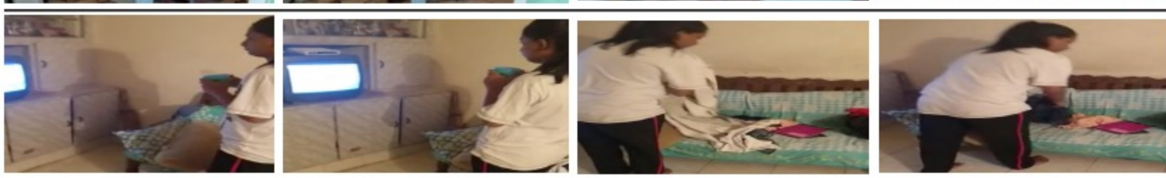
**Where** is the baby while him was fed milk?

A1: mobile
A2: in lady's arm
A3: pillow
A4: baby trolley
A5: living room

GT: in lady's arm
Ours: in lady's arm
HGA: living room



**why did** he get up ?
GT answer: the man got up to start cleaning the plate
Our answer: he stands up so he can go over to the stove
**Our rank = 4** **STSGR rank = 20**



**Does she** pick anything up from off the couch ?
GT answer: yes , she folds clothes that are on the couch
Our answer: yes , she folds clothes that are on the couch
**Our rank = 1** **STSGR rank = 11**

GT = Ground truth
HGA = Hierarchical Graph Alignment, Jiang and Han, AAAI, 2020
STSGR = Spatio-Temporal Scene Graphs, Geng et al., AAAI, 2021

# Summary and Future Work

- In this talk,
  - We looked at the problem of video question answering using scene graphs via reducing the **redundancy in the graph nodes**
  - Our key insight being to treat a video as a ”view” of a 3D space, and reconstruct an **approximate 2.5D scene graph for the 3D space**, removing redundant nodes.
  - We built a hierarchical (2.5+1)D Transformer using our proposed scene graph where we use the **spatio-temporal locations of the query and key pairs** for attention.
  - Our results on two recent Video QA datasets demonstrates **significant gain**
- **Going forward**
  - A more accurate 3D graph could improve results; e.g., 3D point clouds

Thank you!

For questions, write to
cherian@merl.com