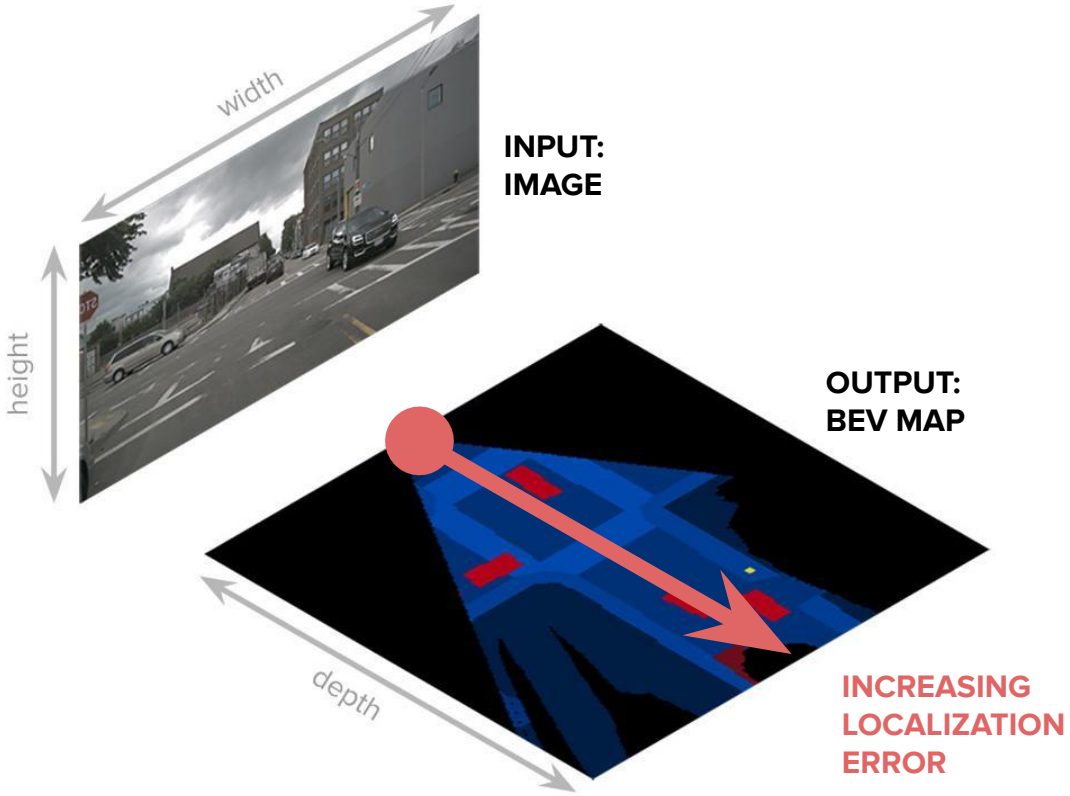
Addressing the shortcomings of BEV Mapping

SHORTCOMINGS

Localization error in BEV Mapping/Scene understanding methods



1. Localization **error increases for distant objects**.

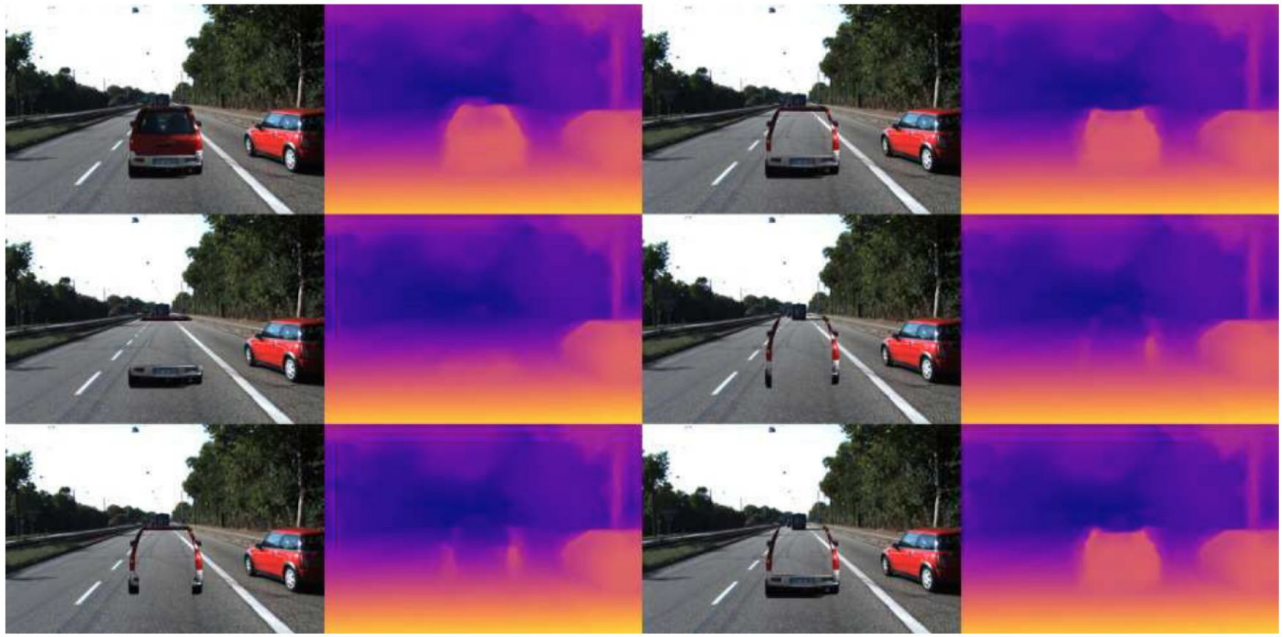
2. Depth estimation = primary challenge of object localization in BEV mapping.

SHORTCOMINGS

Visual cues used by scene understanding methods

1. Most methods rely on **object-ground intersections** as context for depth reasoning (Dijk and Croon 2019).

2. But these become **unreliable for distant objects**.



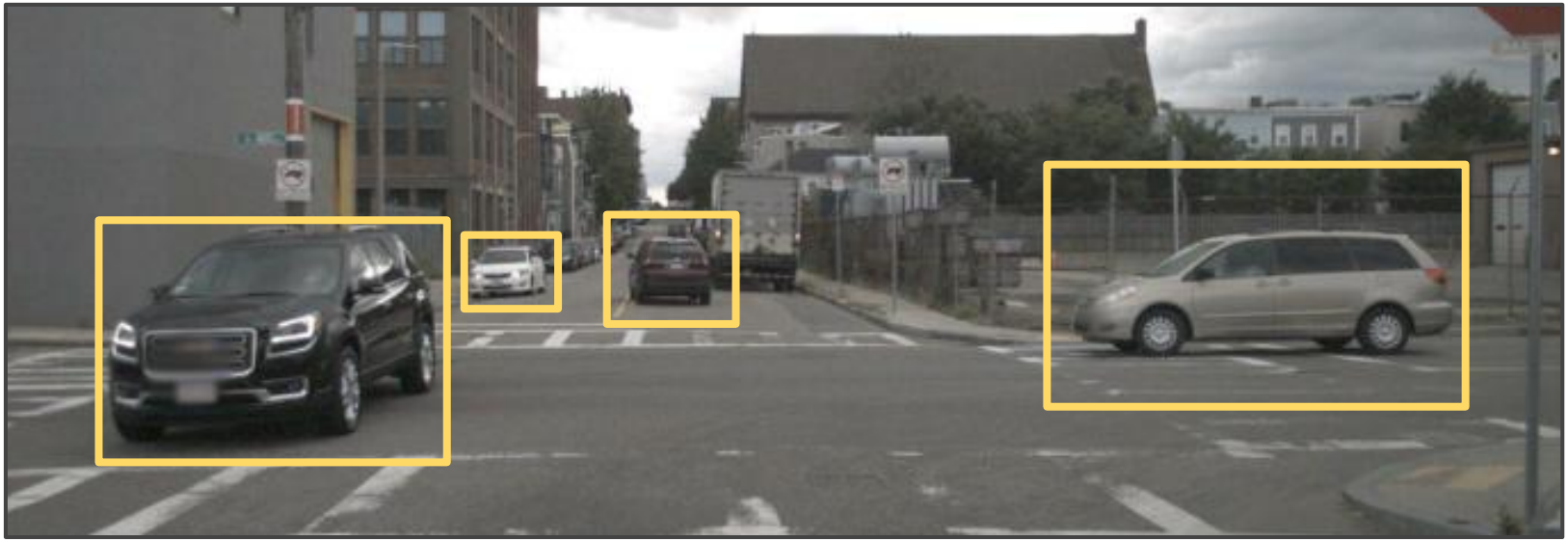
Dijk, T.V. and Croon, G.D., 2019. How do neural networks see depth in single images?

PROPOSAL

Localizing distant objects

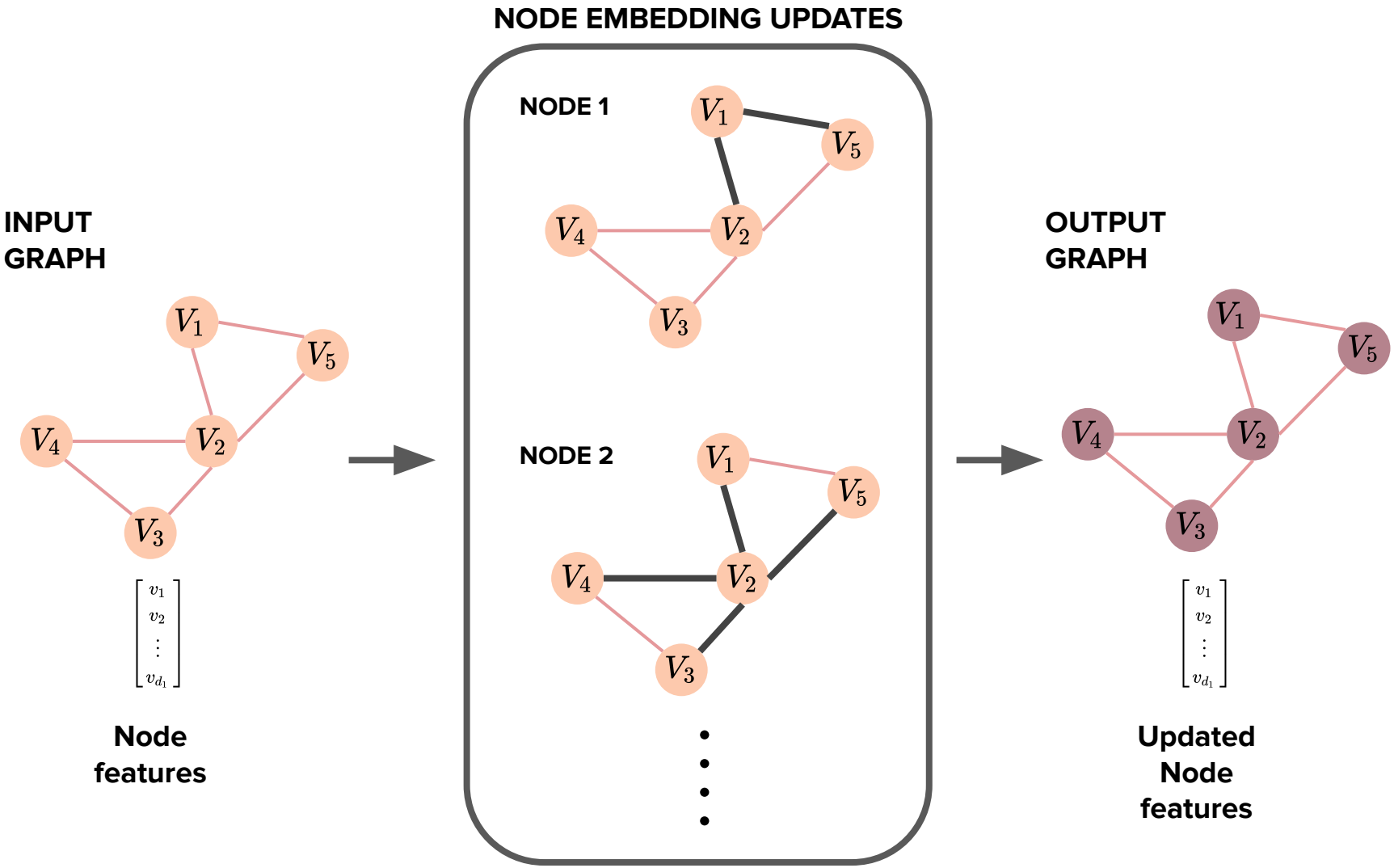
1. When shadows are unreliable, **localize object depth by comparing objects to each other**.

2. How?
   a. **Message-passing** across a graph of the objects



GRAPHS

Message-passing and Graph Convolutions



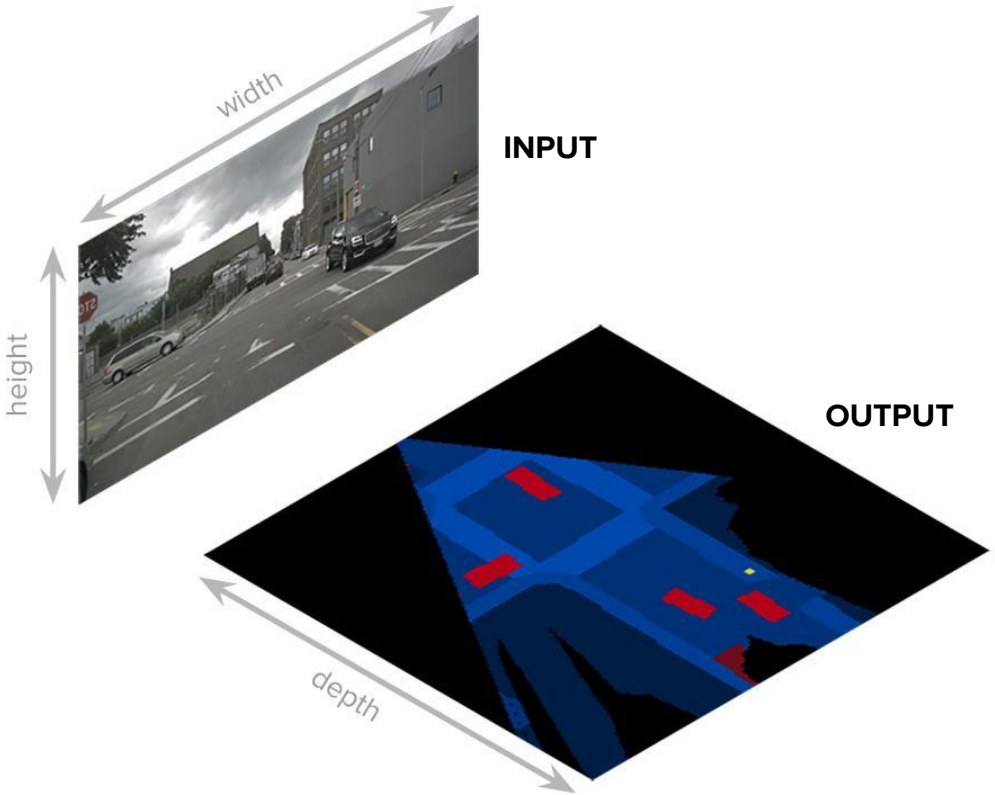
Methodology

Saha, A., Mendez, O., Russell, C. and Bowden, R., 2022. "The Pedestrian next to the Lamppost" Adaptive Object Graphs for Better Instantaneous Mapping.

OVERVIEW

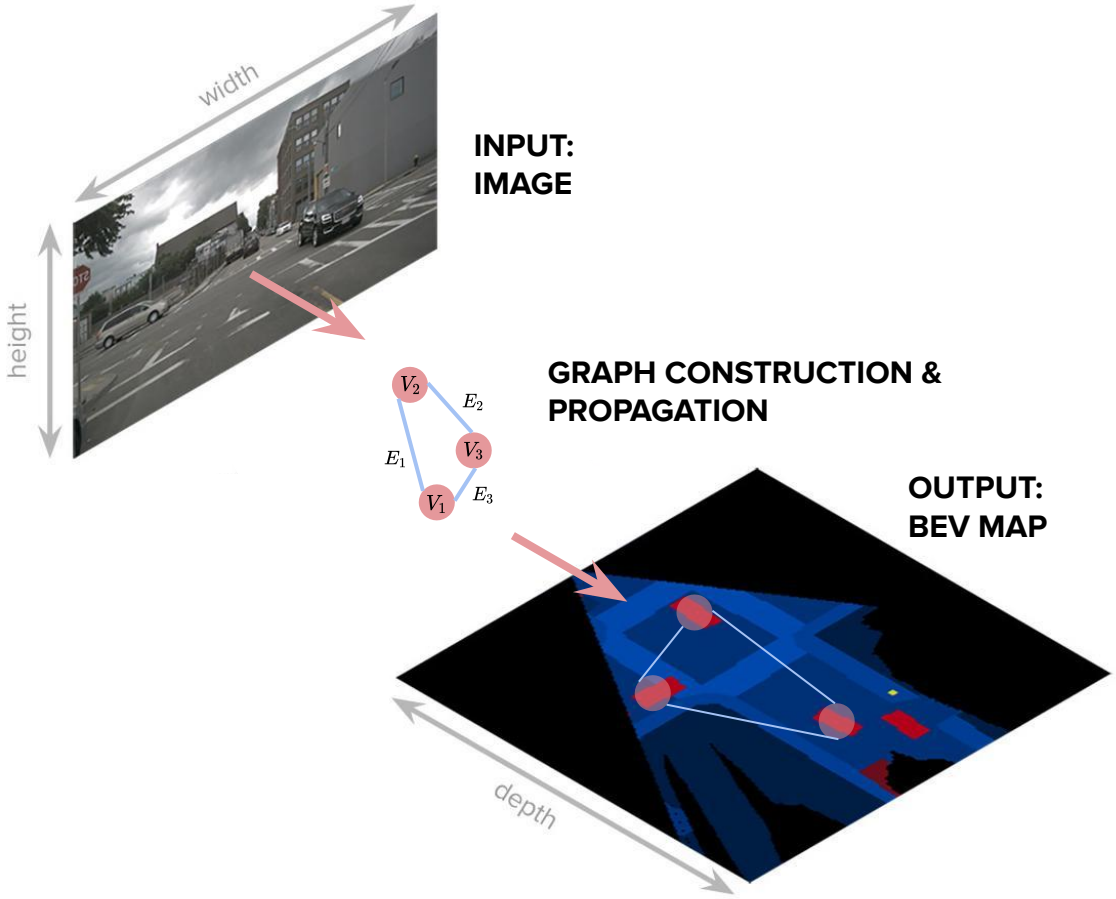
BEV Mapping

1. **Goal:** learn a model that takes a monocular input **image** and generates a semantically segmented **BEV** of the scene - with **improved localization of distant objects**.



APPROACH OVERVIEW

Graphs for object-object reasoning

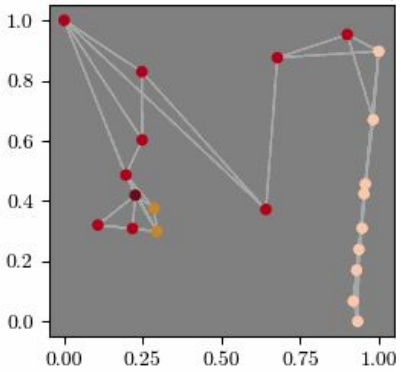


APPROACH OVERVIEW

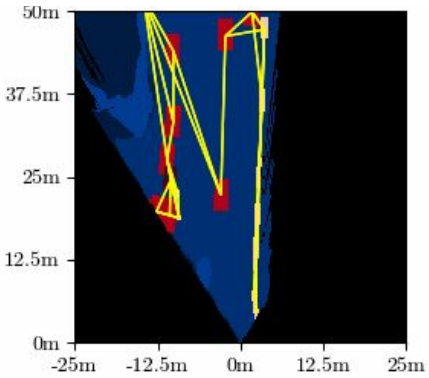
Graphs for object-object reasoning

INPUT IMAGE and INITIAL GRAPH

LATENT GRAPH



BEV PREDICTION



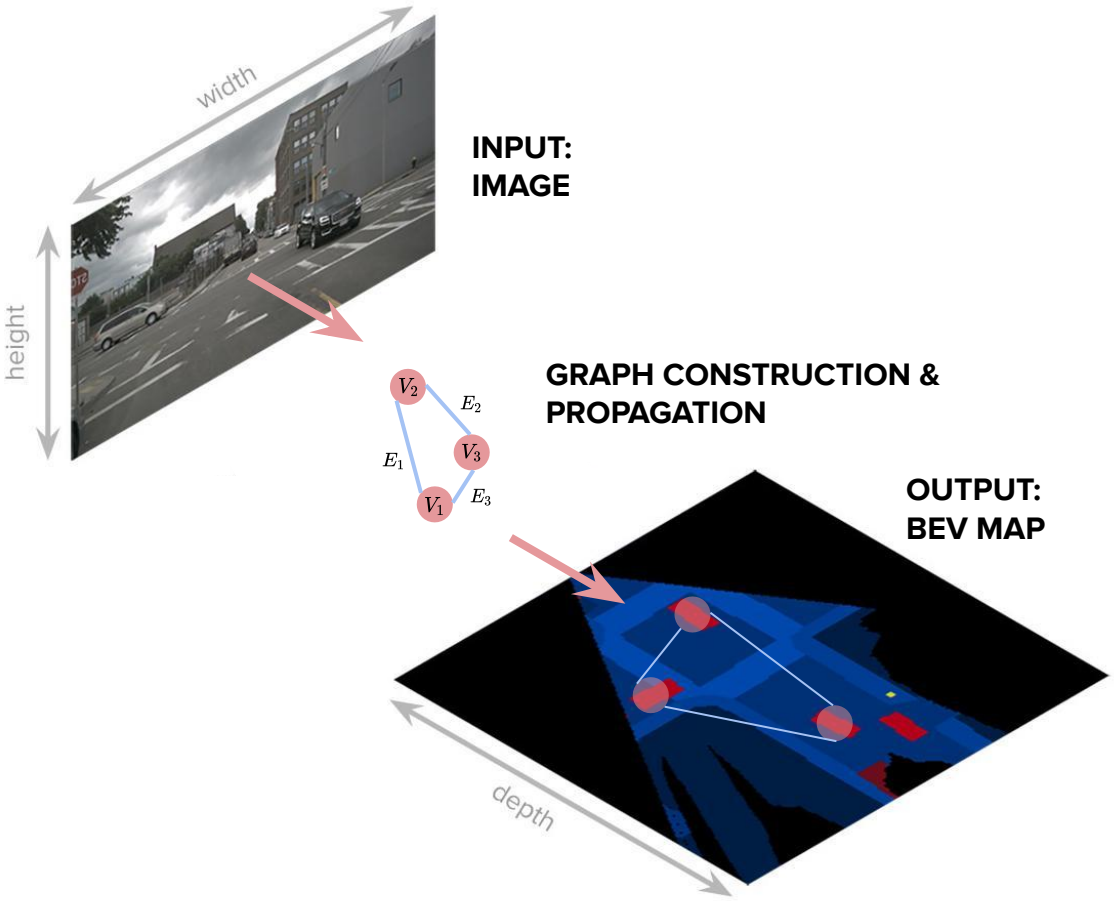
1. **Learn spatial relationship** between objects to reason about scene layout.

2. Use object graphs to propagate context between objects.

3. Sets new SOTA in BEV mapping across Argoverse, Lyft and nuScenes.

MOTIVATION

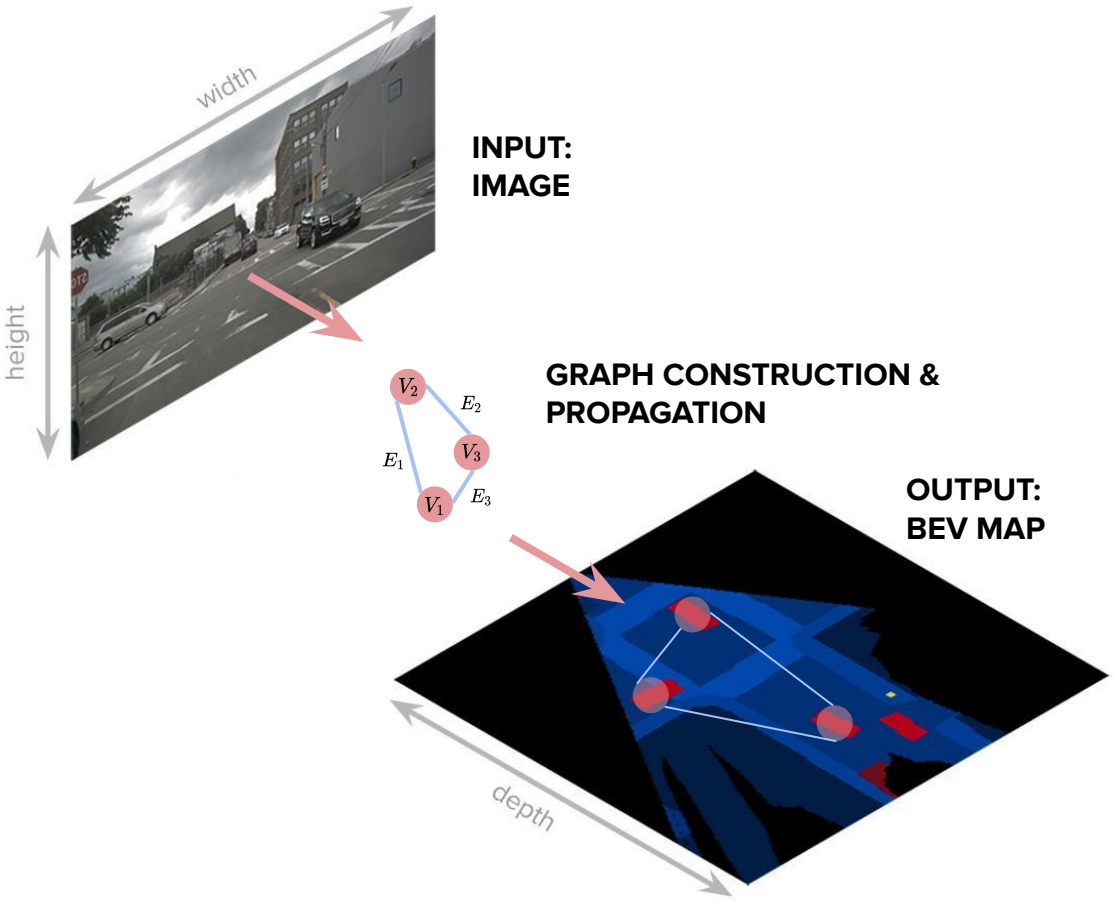
Why use graph representations?



1. Graphs encode explicit **geometric relationships** between objects
    a. Current BEV networks model this implicitly.
2. Graphs allow **nonlocal communication** between entities.
    a. CNN typically requires many downsampling operations to do this.

MOTIVATION

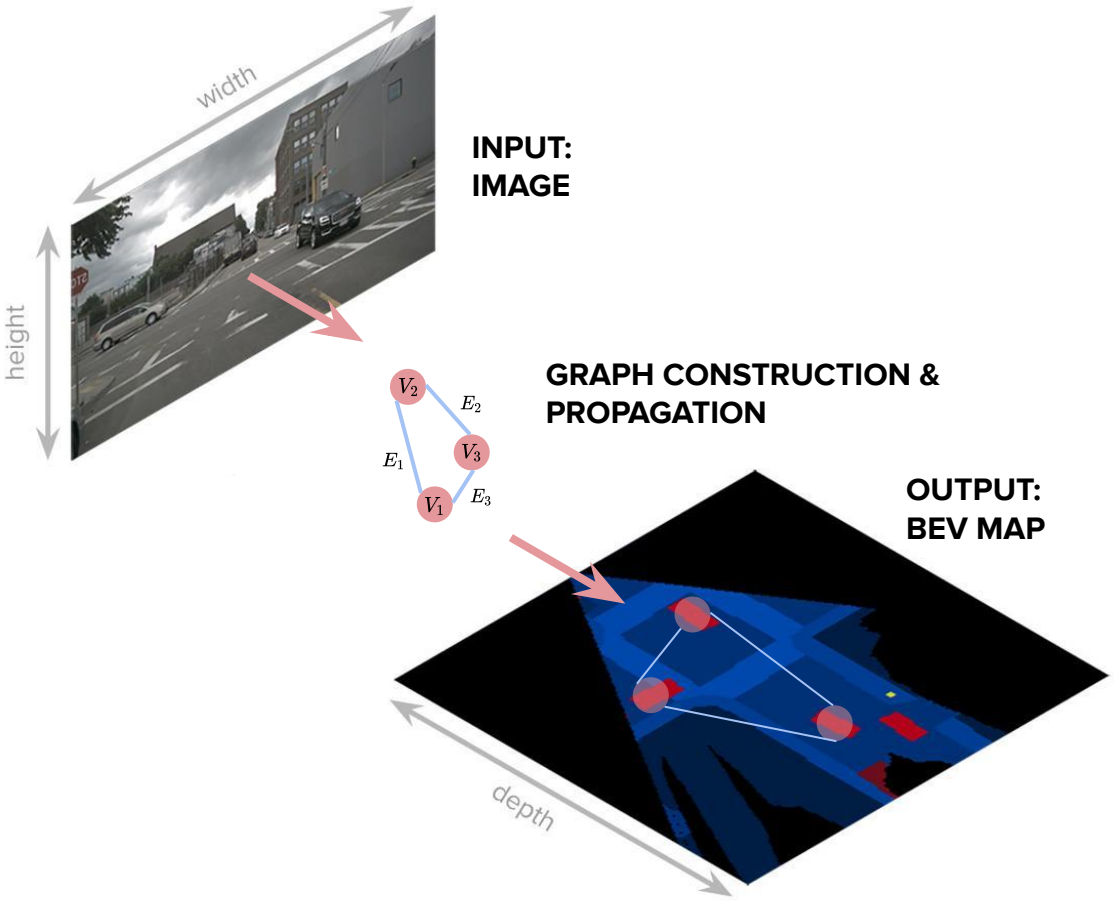
Why learn node embeddings?



1. Message passing between nodes allows depth reasoning by **comparing an object's appearance** and other features.
    a. Current BEV networks do not specifically use such cues/such object-object reasoning.

MOTIVATION

Why learn edge embeddings?

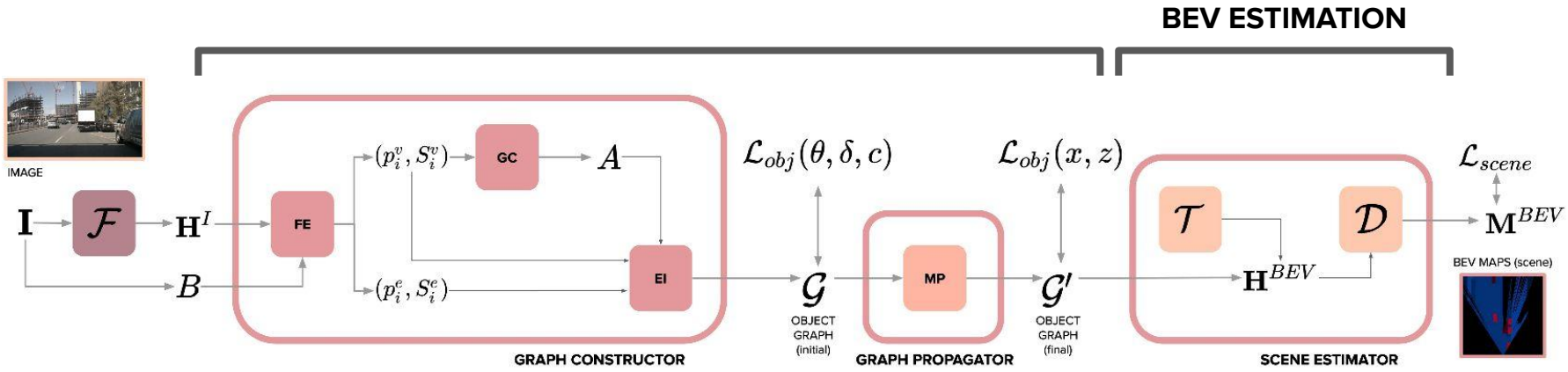


1. Learning edge embeddings allows us to place further constraints on object localization.
    a. **Forces congruency** between predicted object locations.

METHOD

Overview

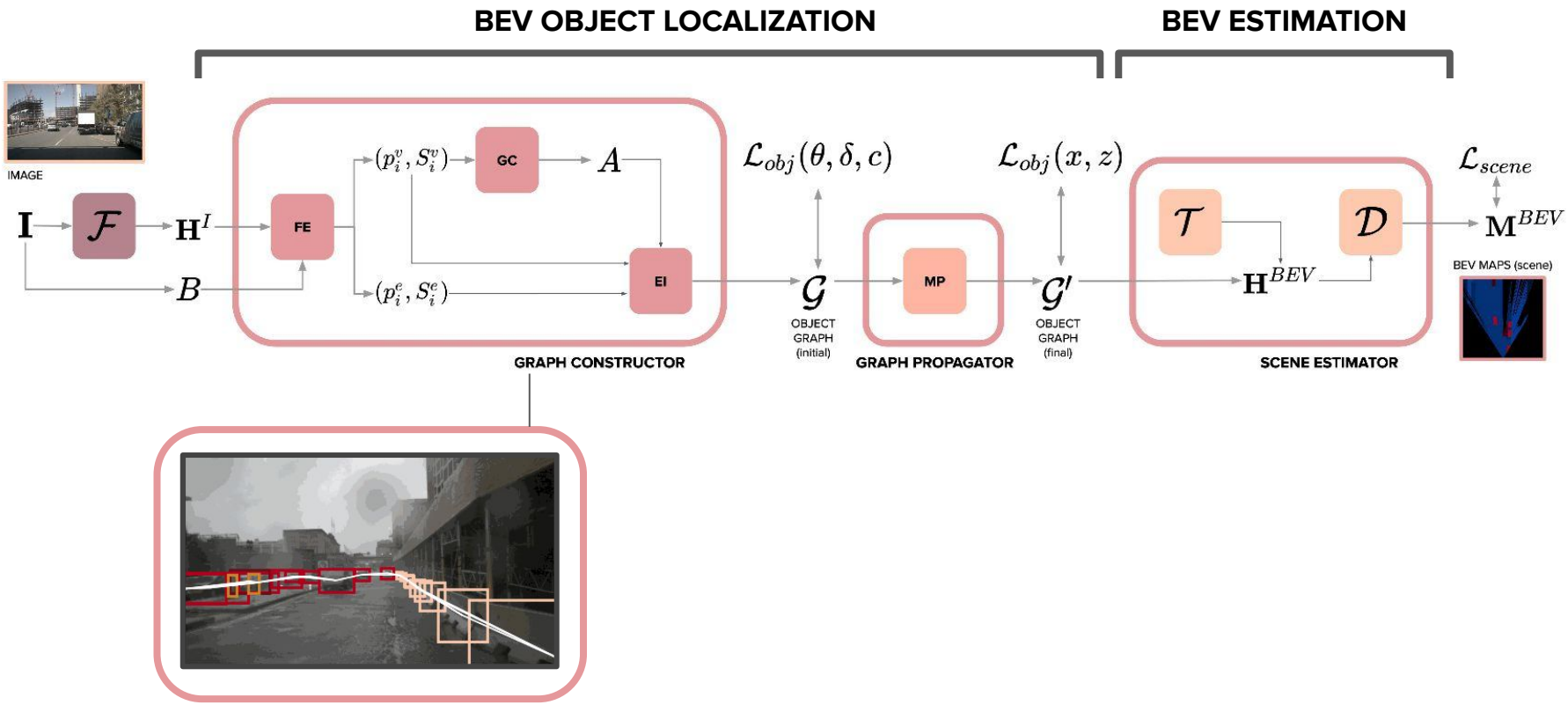
1. Two stage process: **object localization** followed by complete **BEV estimation**.



METHOD

Overview

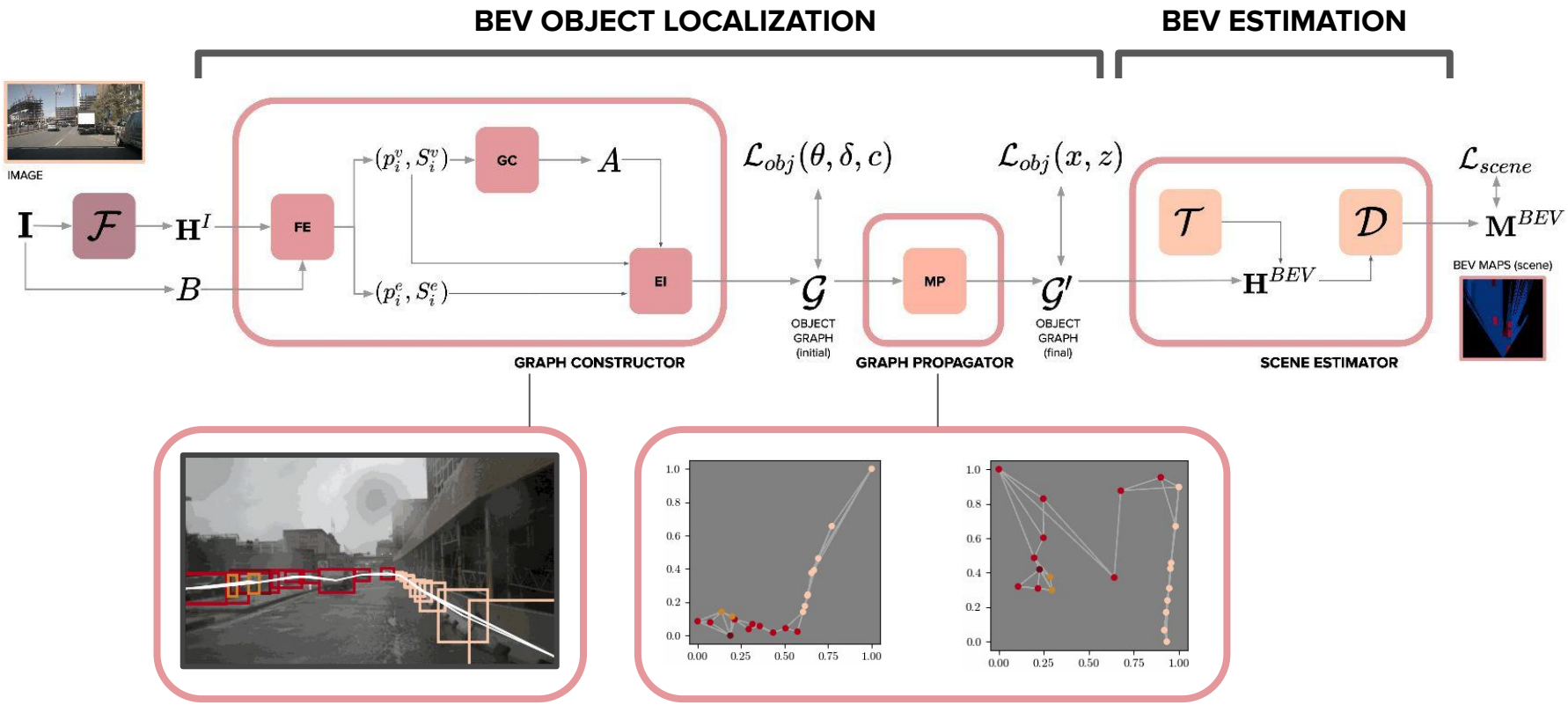
1. Two stage process: object localization followed by complete BEV estimation.
2. **Graph Constructor**: construct graph of objects in image



METHOD

Overview

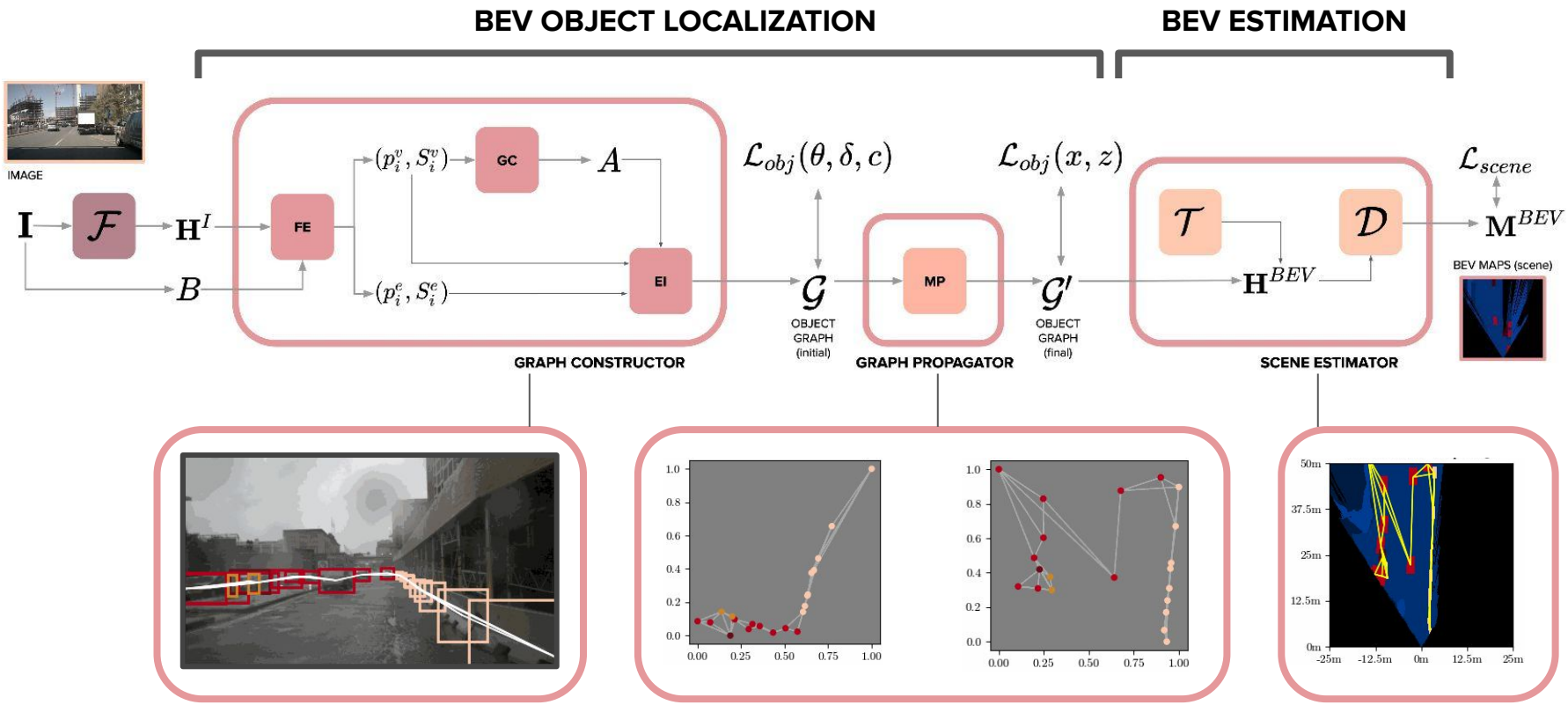
1. Two stage process: object localization followed by complete BEV estimation.
2. Graph Constructor: construct graph of objects in image
3. **Graph Propagator**: message pass to localize objects in BEV



METHOD

Overview

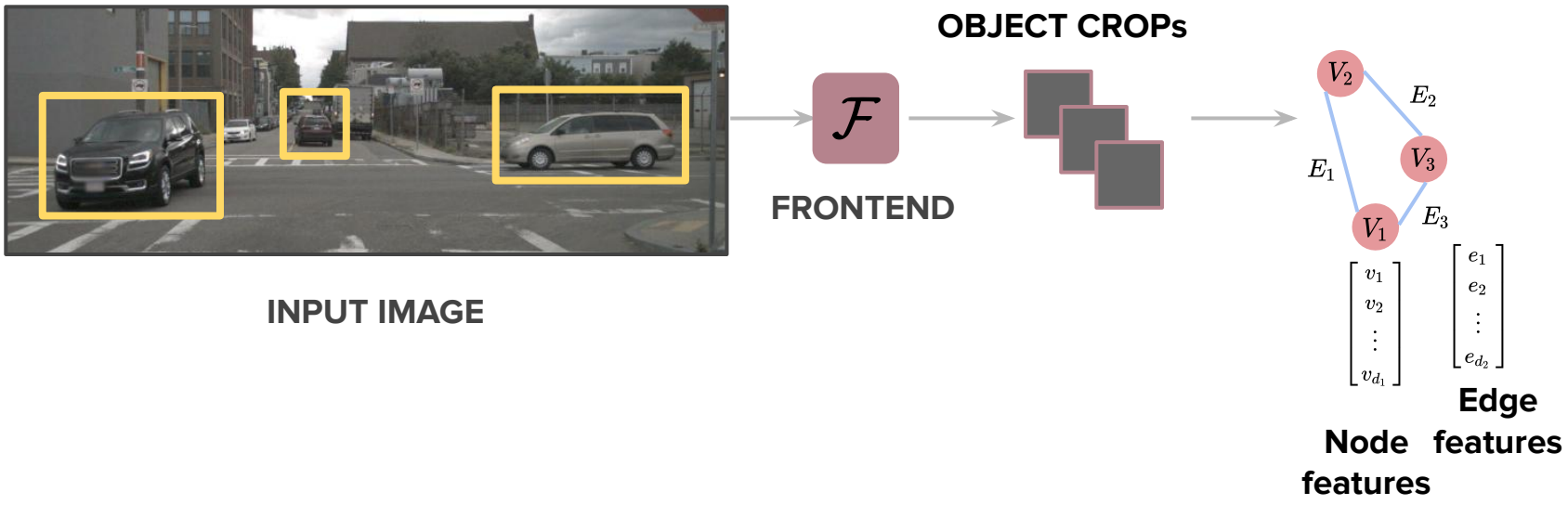
1. Two stage process: object localization followed by complete BEV estimation.
2. Graph Constructor: construct graph of objects in image
3. Graph Propagator: message pass to localize objects in BEV
4. **Scene Estimator**: generate BEV maps



METHOD

Graph constructor

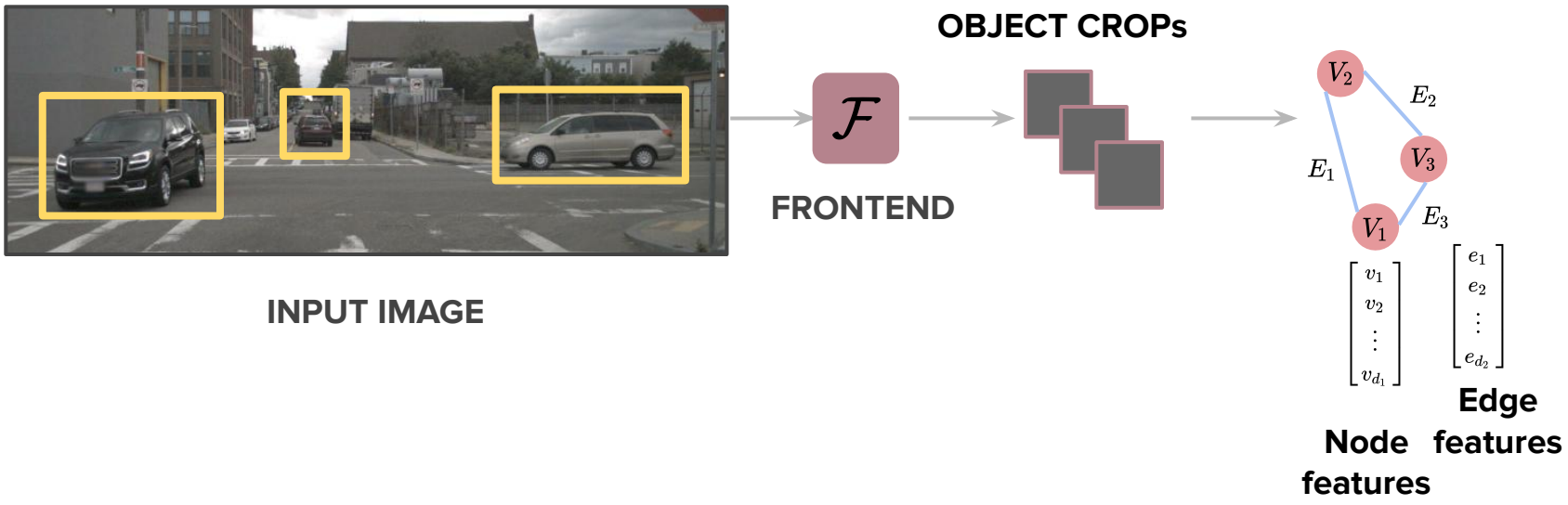
1. Role: construct **graph from objects in input image**



METHOD

Graph constructor

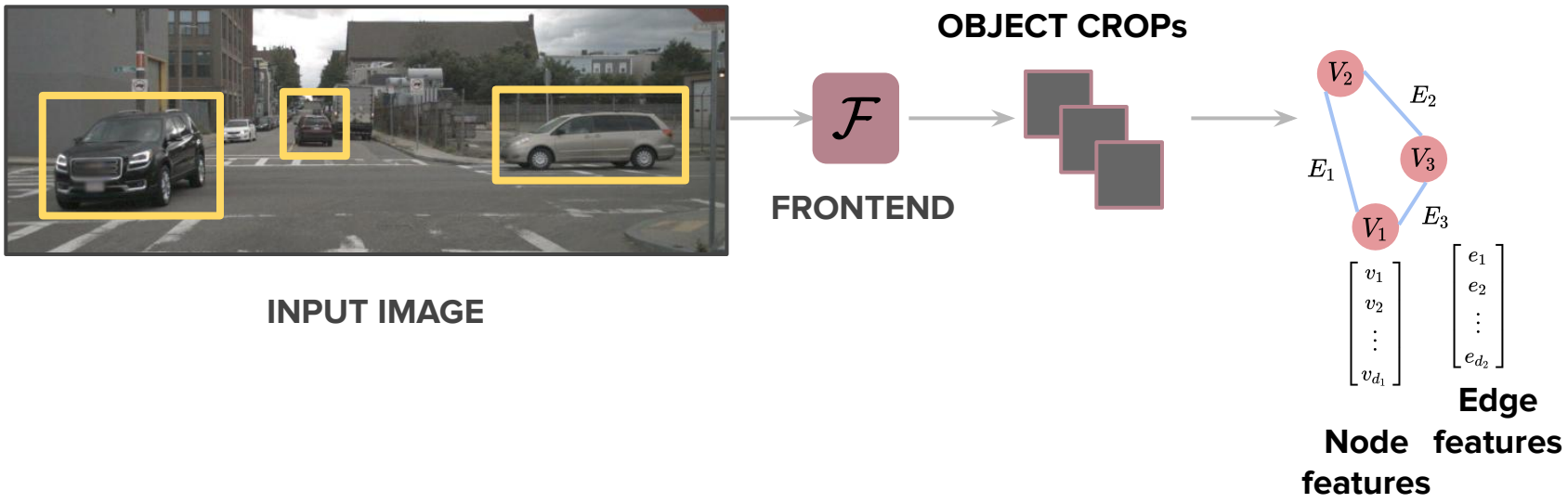
1. Role: construct graph from objects in input image.
2. Each **object corresponds to a node**.
3. Nodes assigned image object features.



METHOD

Graph constructor

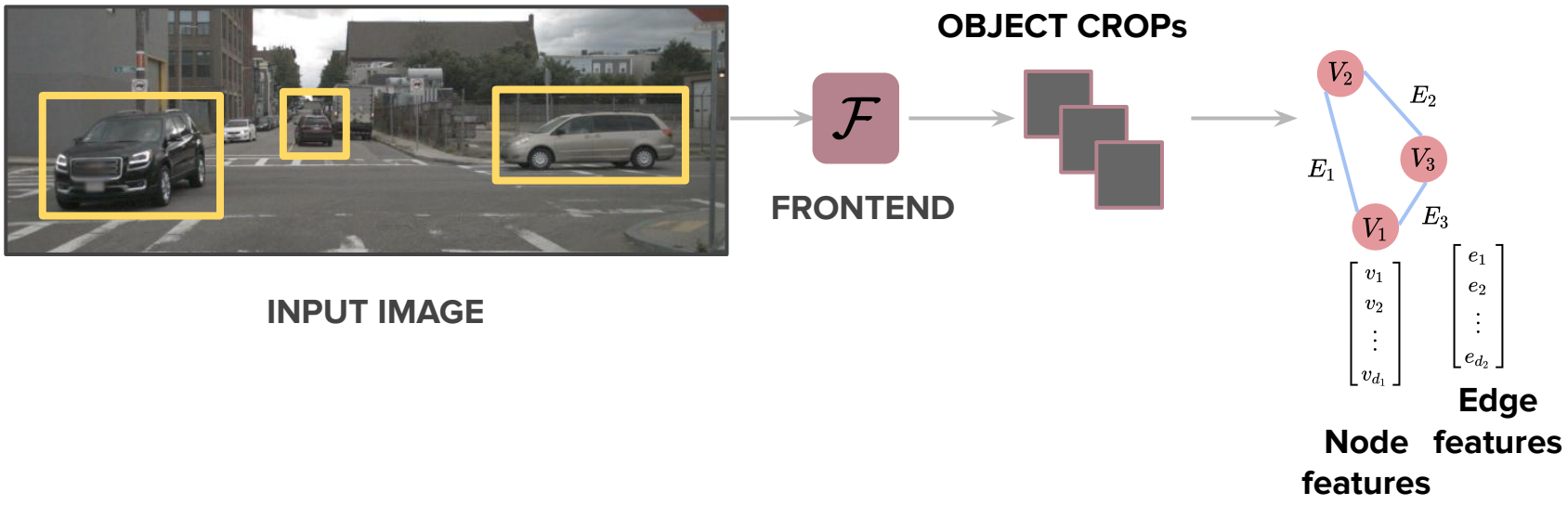
1. Role: construct graph from objects in input image
2. Each object corresponds to node.
3. Nodes assigned image object features.
4. **Edge structure** based on kNN in latent object feature space.
5. Edges assigned image features between objects.



METHOD

Graph constructor

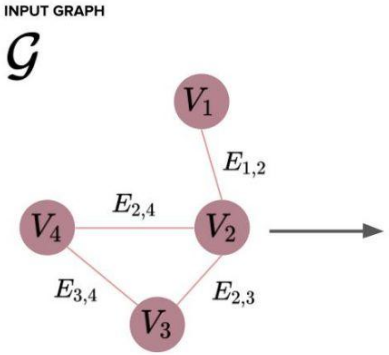
1. Role: construct graph from objects in input image
2. Each object corresponds to node.
3. Nodes assigned image object features.
4. Edge structure based on kNN in latent object feature space.
5. Edges assigned image features between objects.
6. Creates graph with **initial node and edge embeddings**.



METHOD

Graph propagation

1. Role: message pass across graph to **learn embeddings for localisation**.
2. **Two update mechanisms** per round:
    a. Node-level update
    b. Edge level update



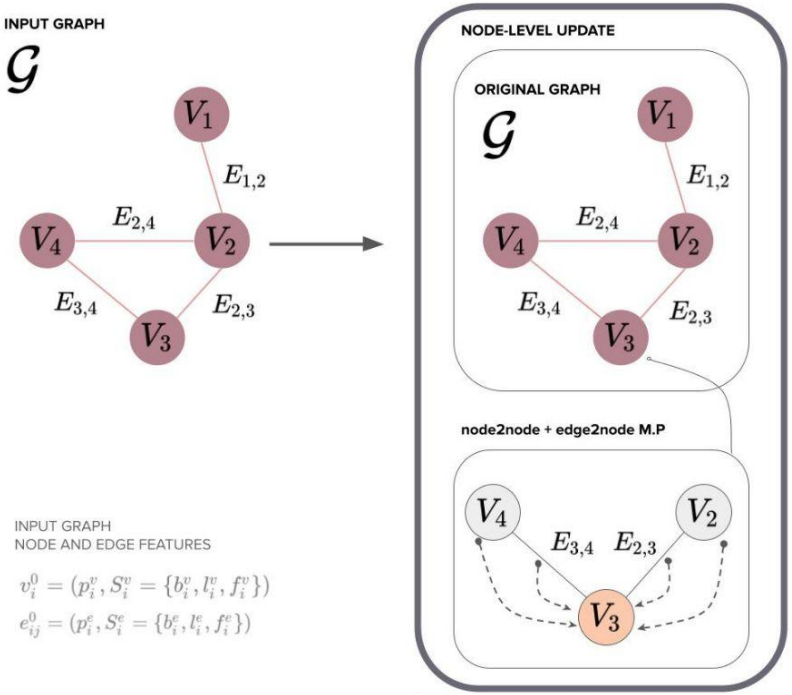
INPUT GRAPH
NODE AND EDGE FEATURES

$$v_i^0 = (p_i^v, S_i^v = \{b_i^v, l_i^v, f_i^v\})$$
$$e_{ij}^0 = (p_i^e, S_i^e = \{b_i^e, l_i^e, f_i^e\})$$

METHOD

Graph propagation

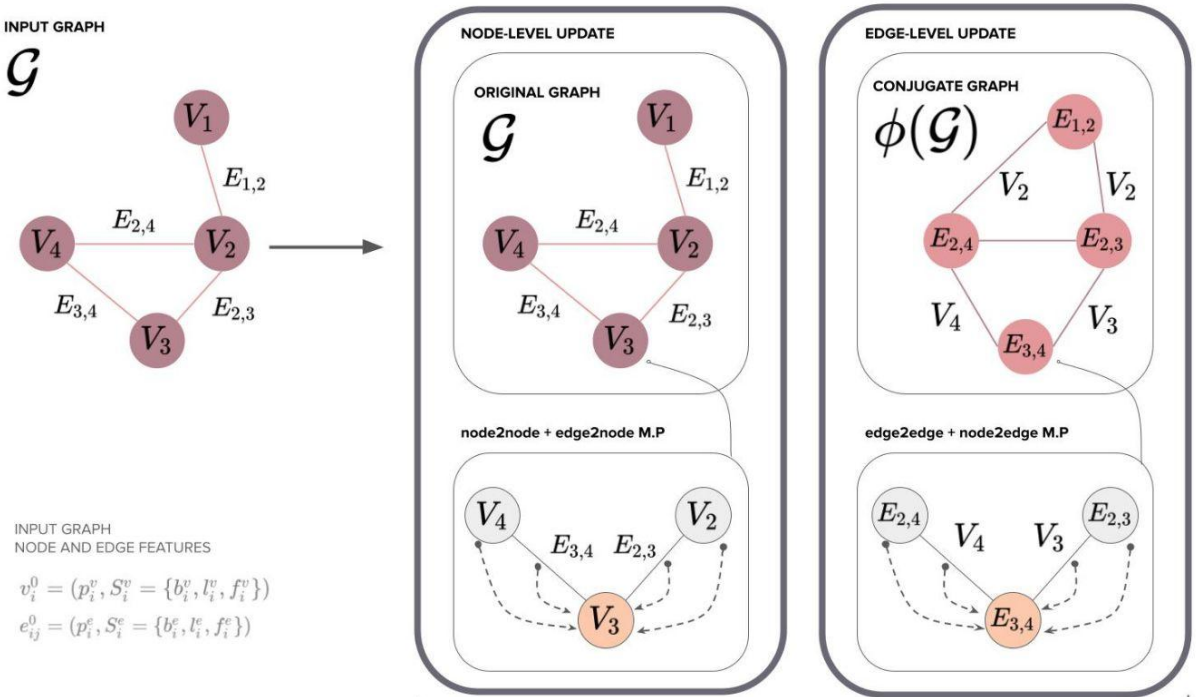
1. Role: message pass across graph to learn embeddings for localisation.
2. Two update mechanisms per round:
   a. Node-level update
   b. Edge level update
3. **Node update**: weighted average of node and edge states of neighborhood.



METHOD

Graph propagation

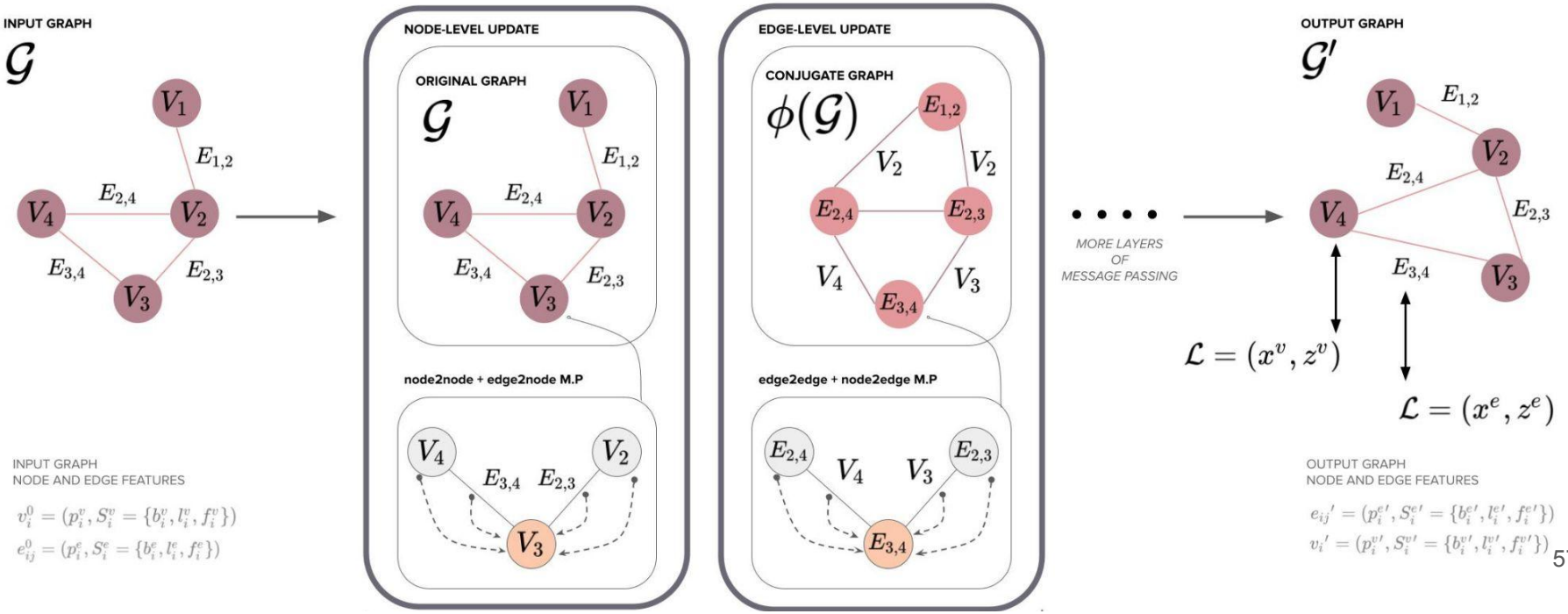
1. Role: message pass across graph to learn embeddings for localisation.
2. Two update mechanisms per round:
   a. Node-level update
   b. Edge level update
3. Node update: weighted average of node and edge states of neighborhood.
4. **Edge update**: same as node update but on conjugate of graph



METHOD

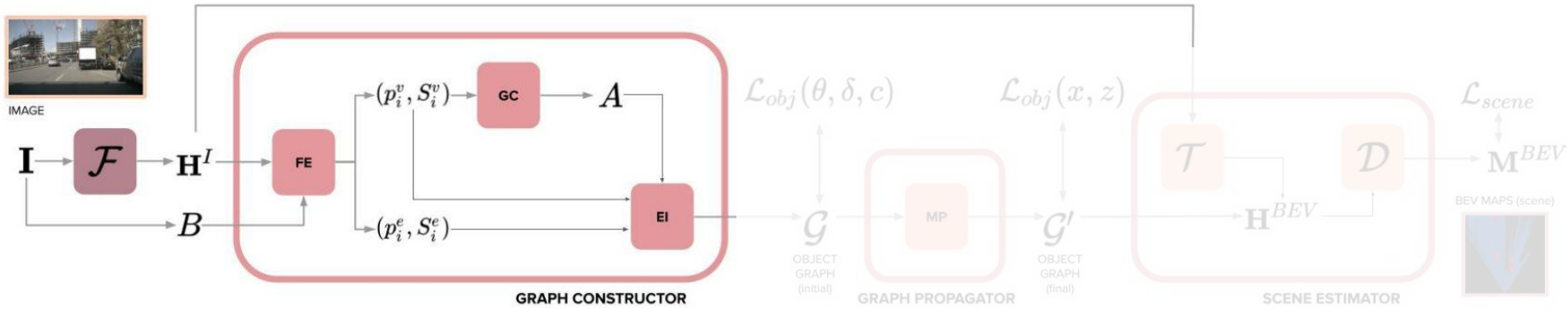
Graph Propagator

1. Role: message pass across graph to learn embeddings for localisation.
2. Two update mechanisms per round:
   a. Node-level update
   b. Edge level update
3. Node update: weighted average of node and edge states of neighborhood.
4. Edge update: same as node update but on conjugate of graph
5. Outputs graph with **updated node and edge embeddings**.



METHOD

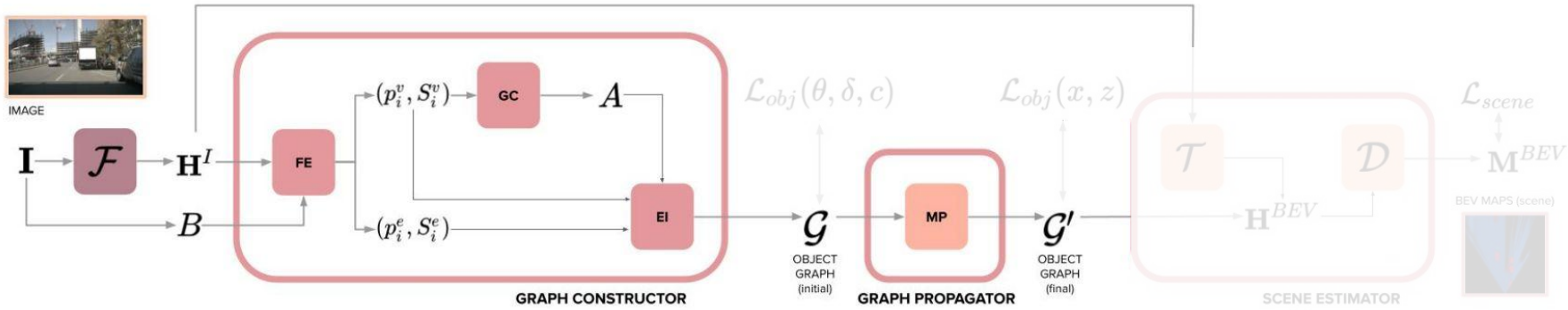
Architecture



1. **Graph Constructor** generates a graph of the scenes objects.

METHOD

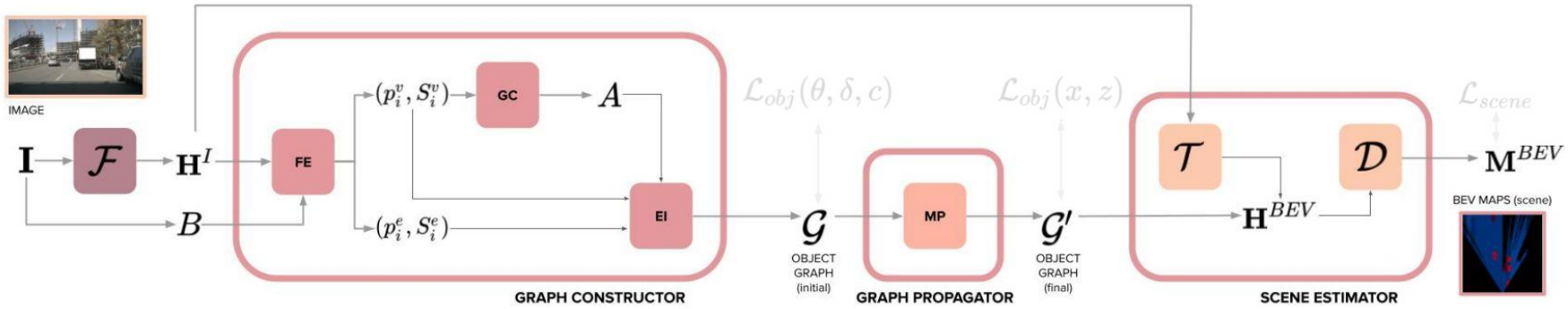
Architecture



1. Graph Constructor generates a graph of the scenes objects.
2. **Graph Propagator** updates node and edge embeddings through message-passing.

METHOD

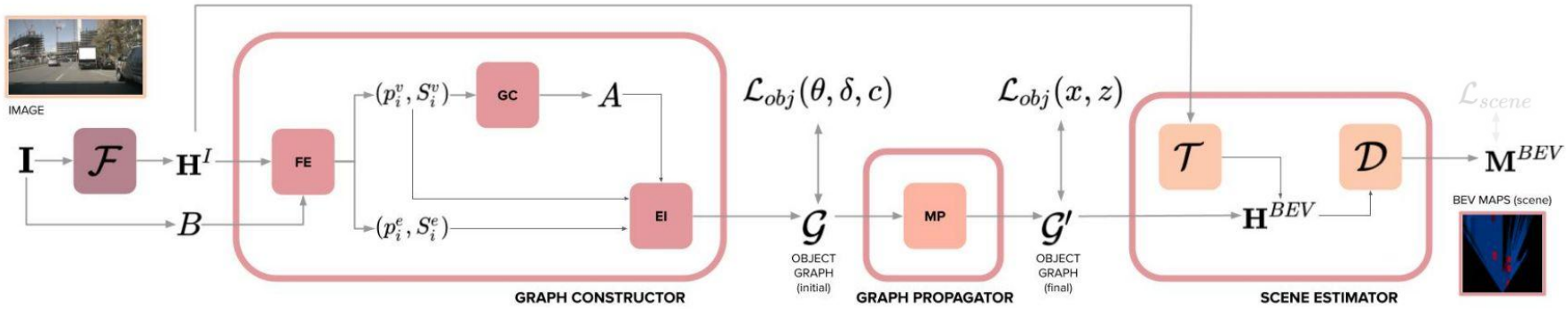
Architecture



1. Graph Constructor generates a graph of the scenes objects.
2. Graph Propagator updates node and edge embeddings through message-passing.
3. **Scene Estimator** generates BEV maps from node embeddings and image features.

METHOD

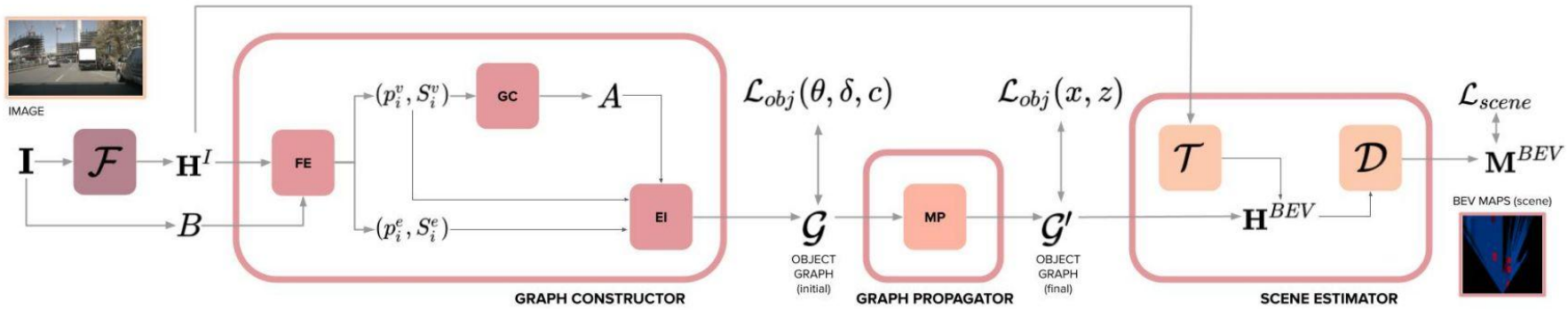
Architecture



1. Graph Constructor generates a graph of the scenes objects.
2. Graph Propagator updates node and edge embeddings through message-passing.
3. Scene Estimator generates BEV maps from node embeddings and image features.
4. **Node embeddings** supervised for object BEV positions.
5. **Edge embeddings** supervised for object midpoint BEV position.

METHOD

Architecture

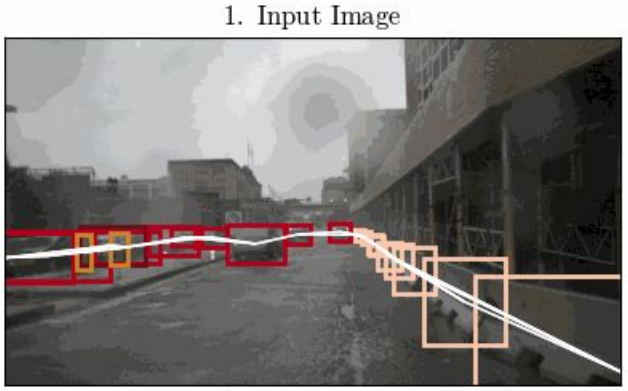


1. Graph Constructor generates a graph of the scenes objects.
2. Graph Propagator updates node and edge embeddings through message-passing.
3. Scene Estimator generates BEV maps from node embeddings and image features.
4. Node embeddings supervised for object BEV positions.
5. Edge embeddings supervised for midpoint BEV position.
6. **BEV maps** supervised at scene level with dice loss.

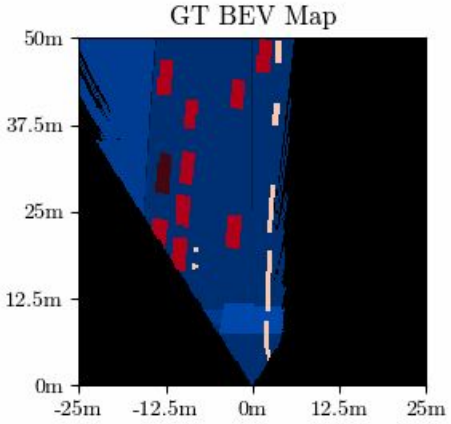
RESULTS

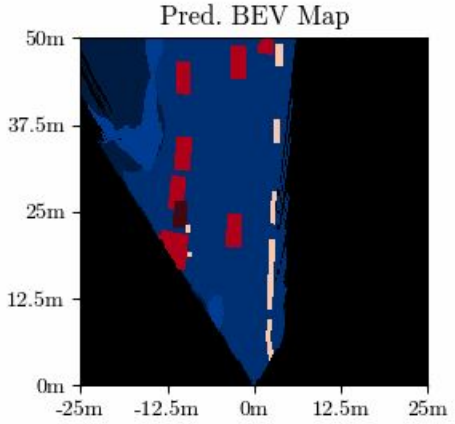
Visualising intermediate layers

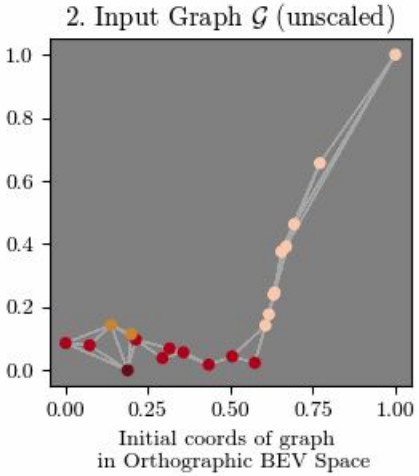
1. The Input Image I with candidate object regions B,
2. The Input Graph $\mathcal{G}$ constructed in BEV Orthographic space from B,
3. The Output Graph $\mathcal{G}\prime$ in BEV Orthographic space with refined node positions after message-passing across $\mathcal{G}$,
4. The predicted BEV Map with the Output Graph $\mathcal{G}\prime$ overlaid

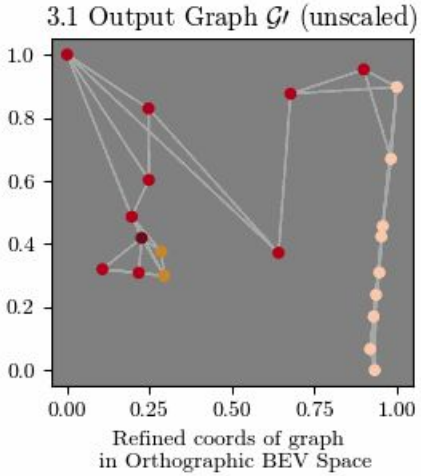


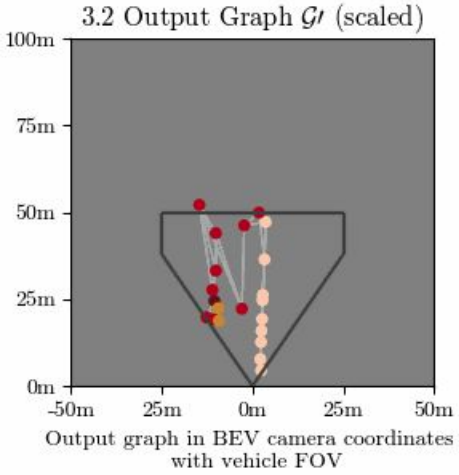
Input image with candidate object regions connected based on input graph $\mathcal{G}$

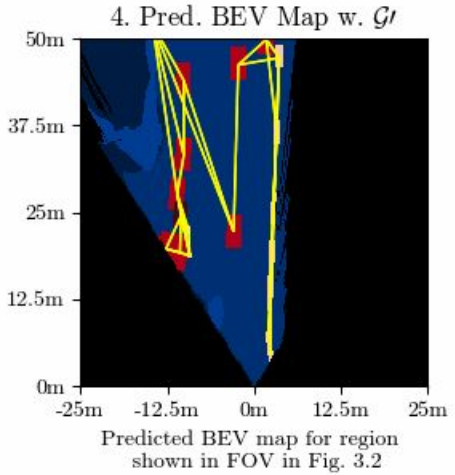












ABLATIONS
Key result

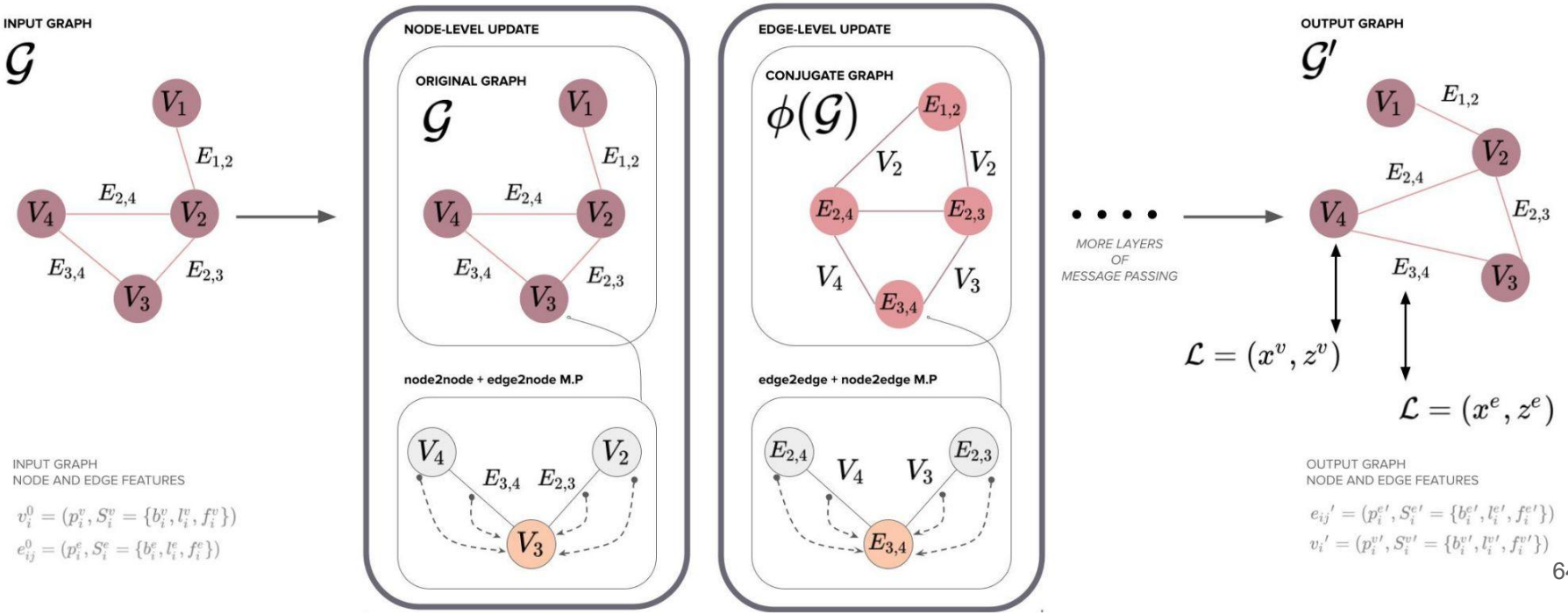
Best localisation accuracy achieved by the following message-passing:

1. node-to-node
2. edge-to-node
3. edge-to-edge
4. node-to-edge

Along with the following supervision:

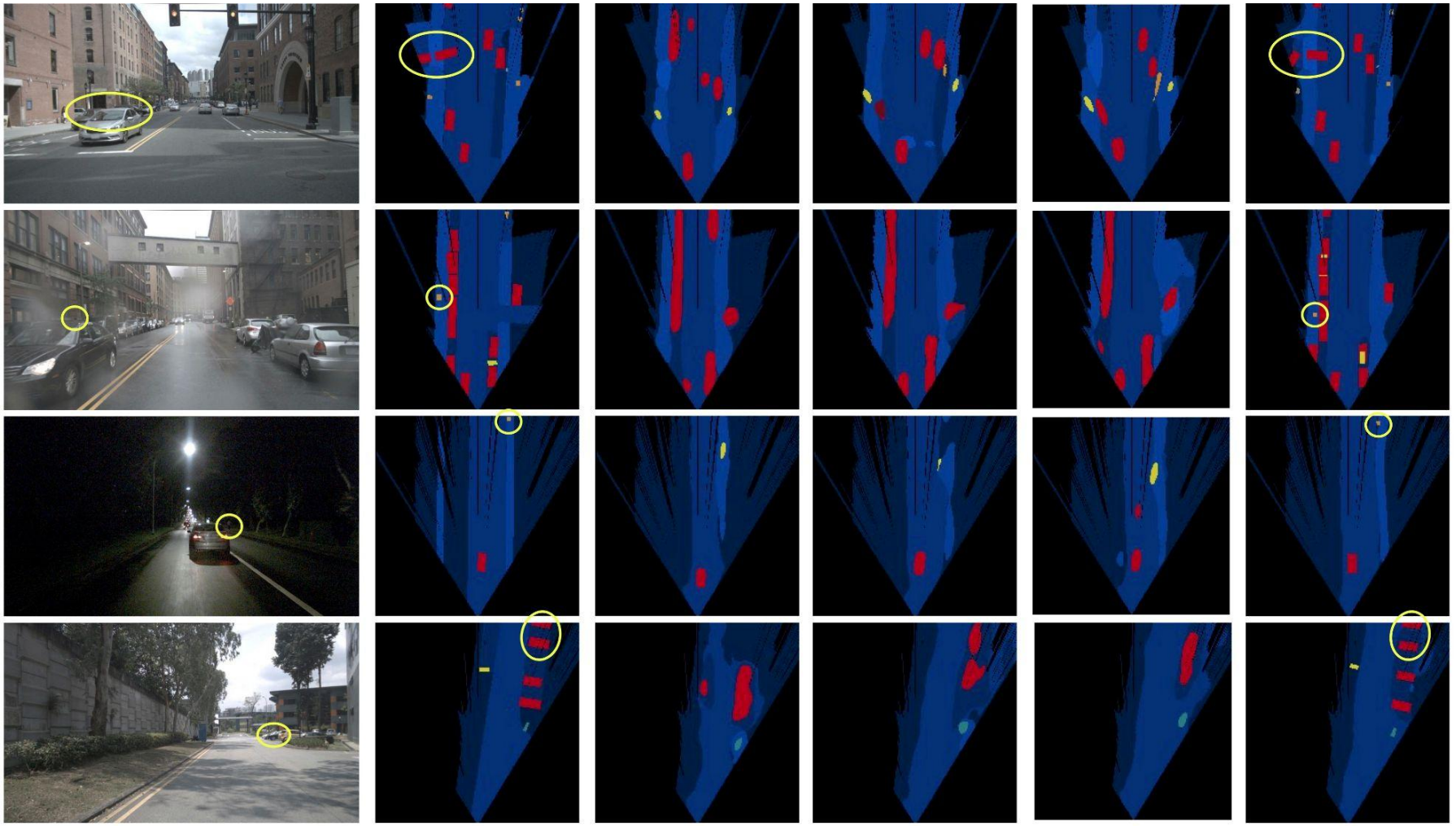
1. node
2. edge

Graph Propagation	Supervision	Objects Mean
n2n	nodes	20.0
n2n + e2n	nodes	21.1
n2n + e2n + e2e	nodes and edges	25.9
n2n + e2n + e2e + n2e	nodes and edges	**27.1**



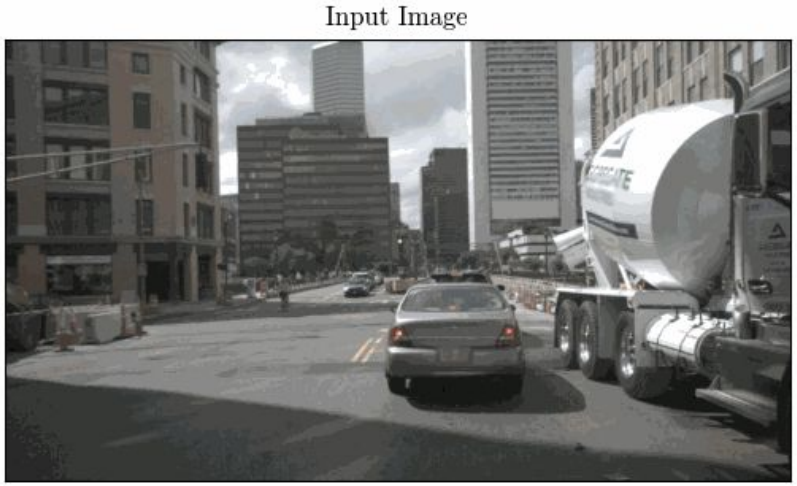
RESULTS

Localizing distant and partially occluded objects

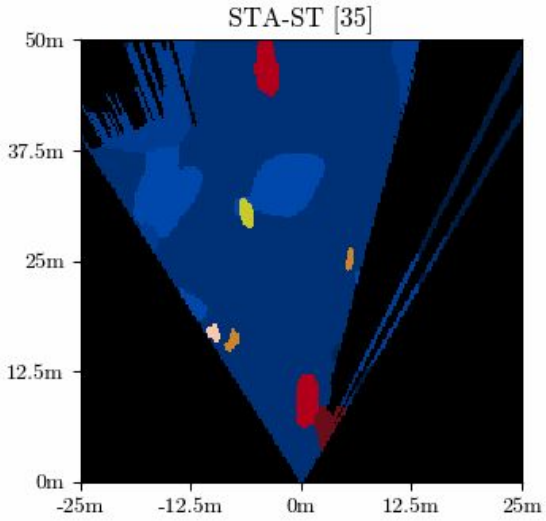


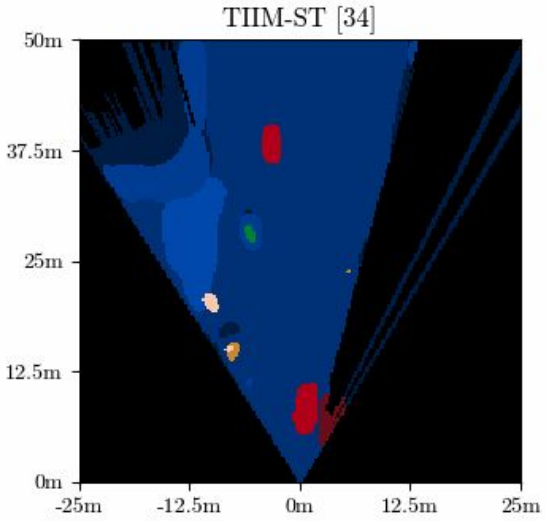
RESULTS

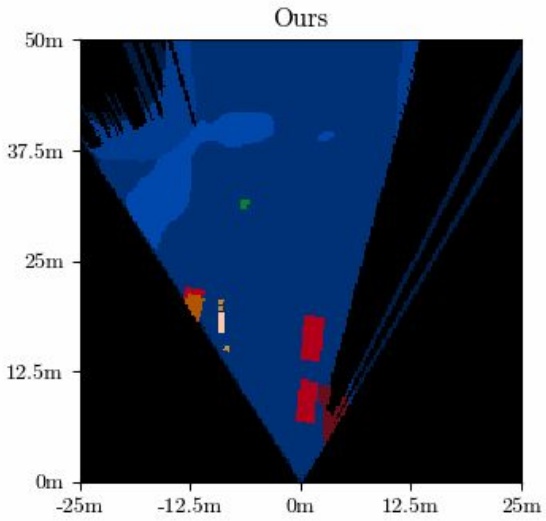
SOTA Comparison



Model	Mean IoU	Objects Mean IoU
PON [34]	19.1	12.9
STA-ST [37]	23.7	16.4
TIIM-ST [36]	25.7	18.1
Ours	**33.0**	**27.1**
Rel. improv (%)	32.4	50.0

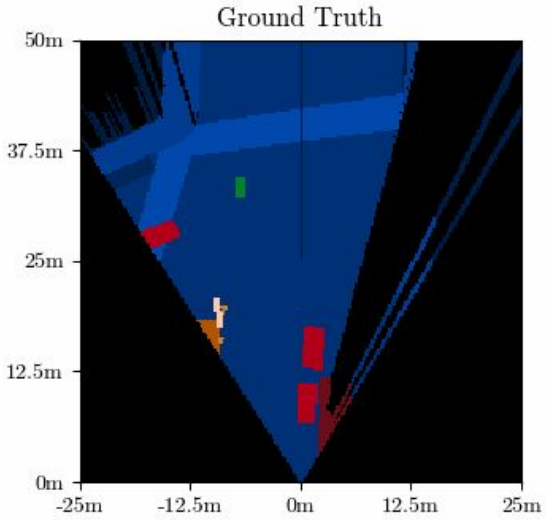


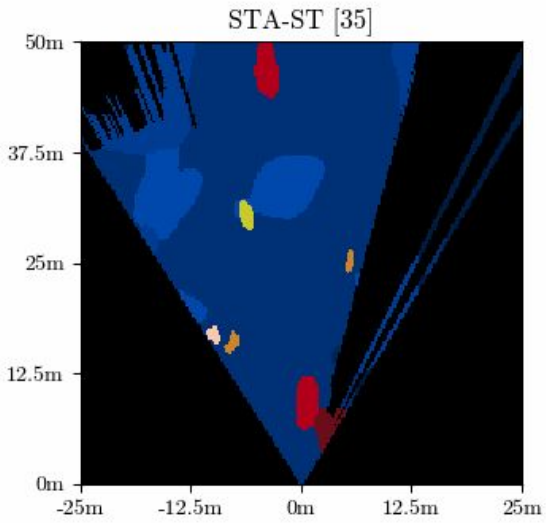


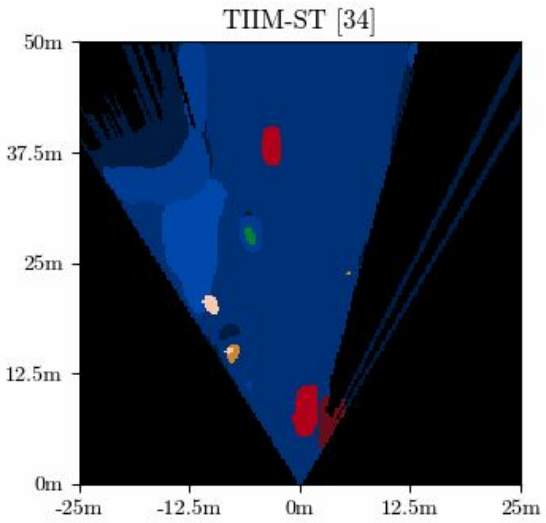


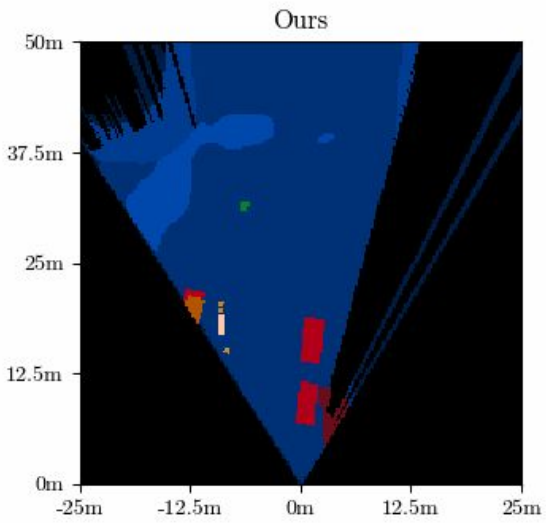
RESULTS

SOTA Comparison









Thanks for watching!
Questions?