Differentiable Task Graph Learning: Procedural Activity Representation and Online Mistake Detection from Egocentric Videos

<u>Luigi Seminara</u>, Giovanni Maria Farinella, Antonino Furnari



University of Catania | Department of Mathematics and Computer Science

Procedure Understanding

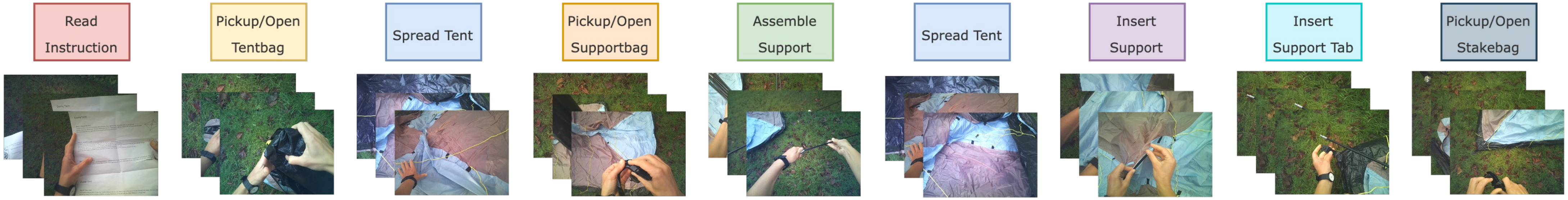
Bike Repair

Health

Procedure Understanding

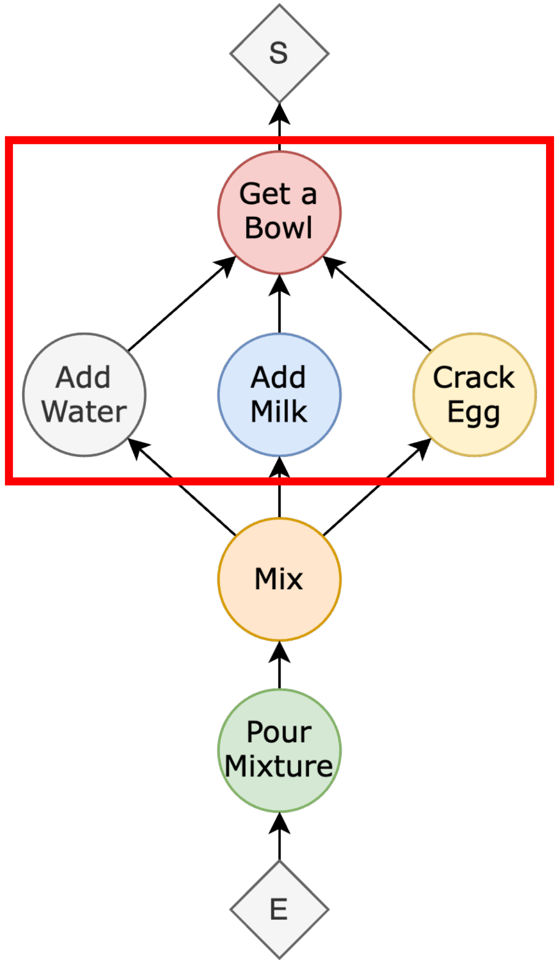
Procedure: Assemble a tent

Key-steps:



- Jang, Youngkyoon, et al. "Epic-tent: An egocentric video dataset for camping tent assembly." Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. 2019.

Task Graph



- Peddi, Rohith, et al. "CaptainCook4D: A dataset for understanding errors in procedural activities." *arXiv preprint arXiv:2312.14556* (2023).
- Grauman, Kristen, et al. "Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

Task Graph Maximum Likelihood (TGML)

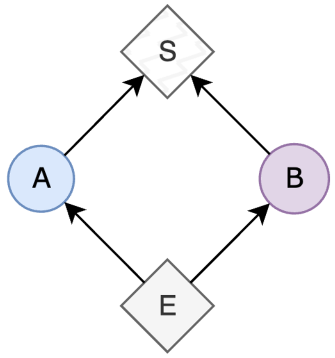
Observed sequence

S A B E

Goal: Estimate

$$P(<S, A, B, E> | \bar{Z}) = P(S|\bar{Z}) \cdot P(A|S, \bar{Z}) \cdot P(B|S, A, \bar{Z}) \cdot P(E|S, A, B, \bar{Z})$$

Graph G



Adjacency Matrix Z

	S	A	B	E
S	0	0	0	0
A	1	0	0	0
B	1	0	0	0
E	0	1	1	0

We can estimate the probability of observing key-step K_i given the set of observed key-steps K_J and the constraints imposed by $\bar{Z}$, following Laplace's classic definition of probability:

$$P(K_i | K_{\mathcal{J}}, \bar{Z}) = \frac{\text{number of favorable cases}}{\text{number of possible cases}} = \frac{\mathbb{1}(\sum_{j \in \bar{\mathcal{J}}} \bar{Z}_{ij} = 0)}{\sum_{h \in \bar{\mathcal{J}}} \mathbb{1}(\sum_{j \in \bar{\mathcal{J}}} \bar{Z}_{hj} = 0)}$$

Task Graph Maximum Likelihood (TGML)

$$P(K_i|K_{\mathcal{J}},\bar{Z}) = \frac{\text{number of favorable cases}}{\text{number of possible cases}} = \frac{\mathbb{1}(\sum_{j\in\bar{\mathcal{J}}}\bar{Z}_{ij}=0)}{\sum_{h\in\bar{\mathcal{J}}}\mathbb{1}(\sum_{j\in\bar{\mathcal{J}}}\bar{Z}_{hj}=0)}$$

Observed sequence

S A B E

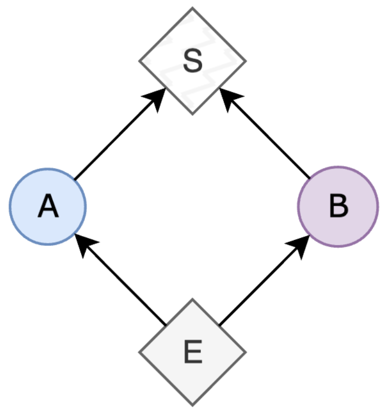
Goal: Estimate

$$P(<S,A,B,E>|\bar{Z}) = P(S|\bar{Z})\cdot P(A|S,\bar{Z})\cdot P(B|S,A,\bar{Z})\cdot P(E|S,A,B,\bar{Z})$$

Estimate:

$$\boxed{P(S|\bar{Z}) = 1}$$

Graph G

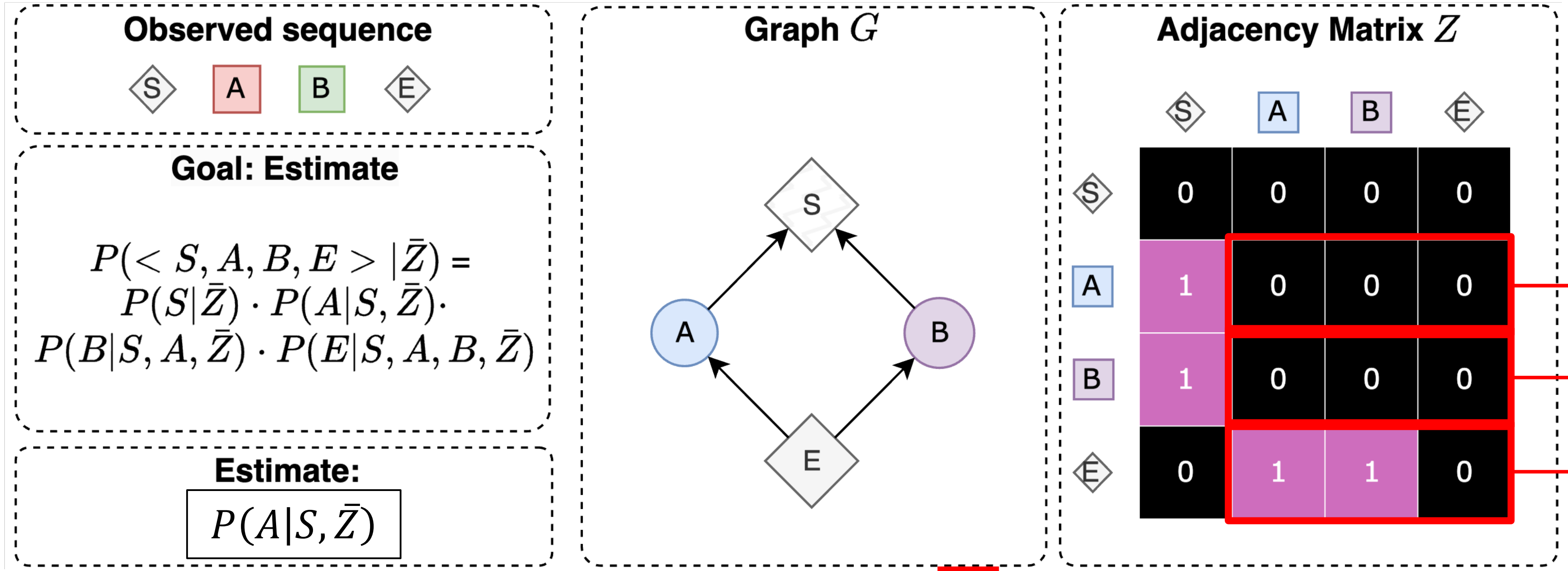


Adjacency Matrix Z

	S	A	B	E
S	0	0	0	0
A	1	0	0	0
B	1	0	0	0
E	0	1	1	0

Task Graph Maximum Likelihood (TGML)

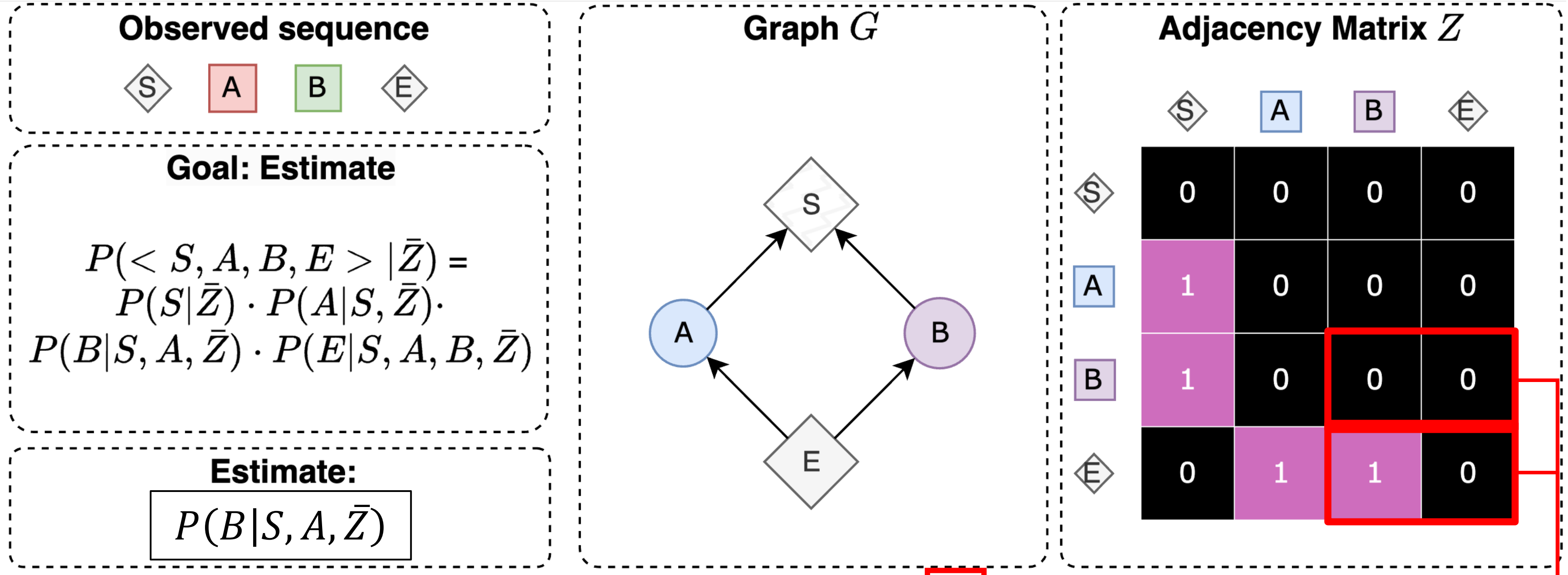
$$P(K_i|K_{\mathcal{J}},\bar{Z}) = \frac{\text{number of favorable cases}}{\text{number of possible cases}} = \frac{\mathbb{1}(\sum_{j\in\bar{\mathcal{J}}}\bar{Z}_{ij}=0)}{\sum_{h\in\bar{\mathcal{J}}}\mathbb{1}(\sum_{j\in\bar{\mathcal{J}}}\bar{Z}_{hj}=0)}$$



$$P(A|S,\bar{Z}) = \frac{\boxed{1}}{\boxed{1}+\boxed{1}+\boxed{0}} = 0.5$$

Task Graph Maximum Likelihood (TGML)

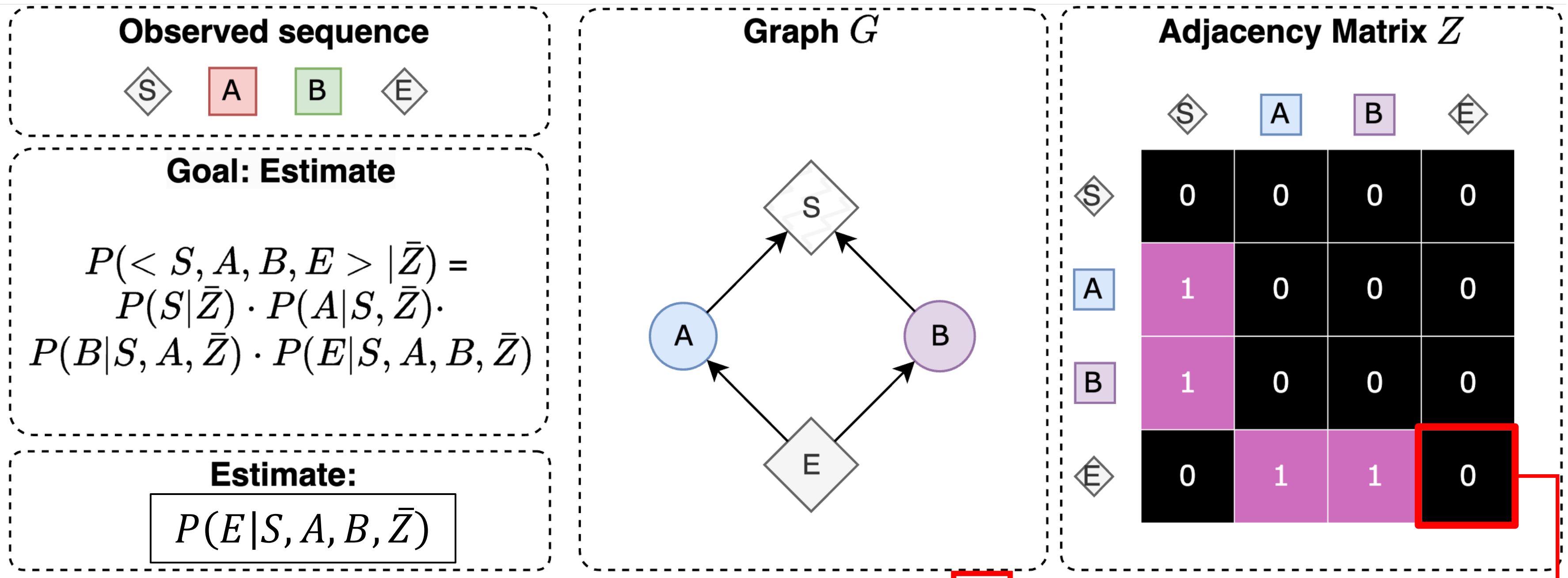
$$P(K_i|K_{\mathcal{J}}, \bar{Z}) = \frac{\text{number of favorable cases}}{\text{number of possible cases}} = \frac{\mathbb{1}(\sum_{j \in \bar{\mathcal{J}}} \bar{Z}_{ij} = 0)}{\sum_{h \in \bar{\mathcal{J}}} \mathbb{1}(\sum_{j \in \bar{\mathcal{J}}} \bar{Z}_{hj} = 0)}$$



$$P(B|S, A, \bar{Z}) = \frac{\boxed{1}}{\boxed{1} + \boxed{0}} = 1$$

Task Graph Maximum Likelihood (TGML)

$$P(K_i|K_{\mathcal{J}},\bar{Z}) = \frac{\text{number of favorable cases}}{\text{number of possible cases}} = \frac{\mathbb{1}(\sum_{j\in\bar{\mathcal{J}}}\bar{Z}_{ij}=0)}{\sum_{h\in\bar{\mathcal{J}}}\mathbb{1}(\sum_{j\in\bar{\mathcal{J}}}\bar{Z}_{hj}=0)}$$



$$P(< S,A,B,E >|\bar{Z}) =$$
$$= 1 \cdot 0.5 \cdot 1 \cdot 1 = 0.5$$

$$P(E|S,A,B,\bar{Z}) = \frac{1}{1} = 1$$

Task Graph Maximum Likelihood (TGML)

Modeling Sequence Likelihood for a Weighted Graph To enable gradient-based learning, we consider the general case of a continuous adjacency matrix $Z \in [0,1]^{(n+2)\times(n+2)}$. We generalize the concept of "possible cases" discussed in the previous section with the concept of "feasibility of sampling a given key-step K_i, having observed a set of key-steps $K_{\mathcal{J}}$, given graph Z", which we define as the sum of all weights of edges between observed key-steps $K_{\mathcal{J}}$ and K_i: $f(K_i|K_{\mathcal{J}}, Z) = \sum_{j\in\mathcal{J}} Z_{ij}$. Intuitively, if key-step k_i has many satisfied pre-conditions, we are more likely to sample it as the next key-step. We hence define $P(K_i|K_{\mathcal{J}}, Z)$ as "the ratio of the feasibility of sampling K_i to the sum of the feasibilities of sampling any unobserved key-step":

$$P(K_i|K_{\mathcal{J}}, Z) = \frac{f(K_i|K_{\mathcal{J}}, Z)}{\sum_{h\in\bar{\mathcal{J}}} f(K_h|K_{\mathcal{J}}, Z)} = \frac{\sum_{j\in\mathcal{J}} Z_{ij}}{\sum_{h\in\bar{\mathcal{J}}}\sum_{j\in\mathcal{J}} Z_{hj}} \tag{3}$$

Figure 2 illustrates the computation of the likelihood in Eq. (3). Plugging Eq. (3) into Eq. (1), we can estimate the likelihood of a sequence y given graph Z as:

$$P(y|Z) = P(S|Z)\prod_{t=1}^{|y|} P(K_{y_t}|K_{\mathcal{O}(y,t)}, Z) = \prod_{t=1}^{|y|} \frac{\sum_{j\in\mathcal{O}(y,t)} Z_{y_t j}}{\sum_{h\in\overline{\mathcal{O}(y,t)}}\sum_{j\in\mathcal{O}(y,t)} Z_{hj}}. \tag{4}$$

Where we set $P(K_{y_0}|Z) = P(S|Z) = 1$ as sequences always start with the start node S.

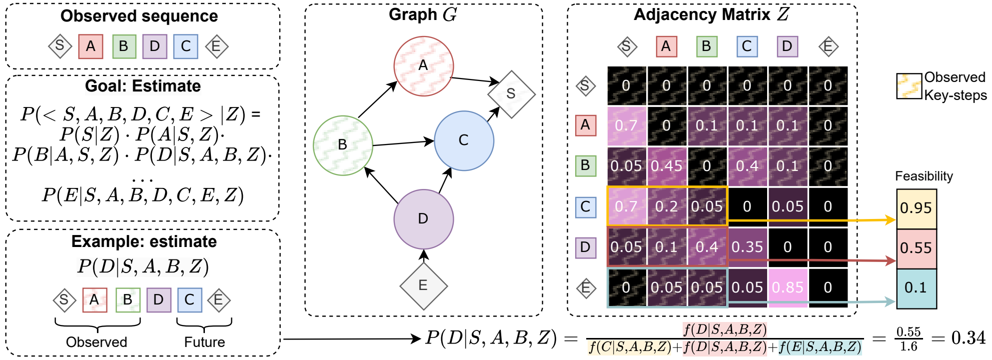


Figure 2: Given a sequence $< S, A, B, D, C, E >$, and a graph G with adjacency matrix Z, our goal is to estimate the likelihood $P(< S, A, B, D, C, E > |Z)$, which can be done by factorizing the expression into simpler terms. The figure shows an example of computation of probability $P(D|S, A, B, Z)$ as the ratio of the "feasibility of sampling key-step D, having observed key-steps S, A, and B" to the sum of all feasibility scores for unobserved symbols. Feasibility values are computed by summing weights of edges $D \to X$ for all observed key-steps X.

Task Graph Maximum Likelihood Loss Function Assuming that sequences $y^{(i)} \in \mathcal{Y}$ are independent and identically distributed, we define the likelihood of $\mathcal{Y}$ given graph Z as follows:

$$P(\mathcal{Y}|Z) = \prod_{k=1}^{|\mathcal{Y}|} P(y^{(k)}|Z) = \prod_{k=1}^{|\mathcal{Y}|}\prod_{t=1}^{|y^{(k)}|} \frac{\sum_{j\in\mathcal{O}(y^{(k)},t)} Z_{y_t j}}{\sum_{h\in\overline{\mathcal{O}(y^{(k)},t)}}\sum_{j\in\mathcal{O}(y^{(k)},t)} Z_{hj}}. \tag{5}$$

We can find the optimal graph Z by maximizing the likelihood in Eq. (5), which is equivalent to minimizing the negative log-likelihood $-\log P(\mathcal{Y}, Z)$, leading to formulating the following loss:

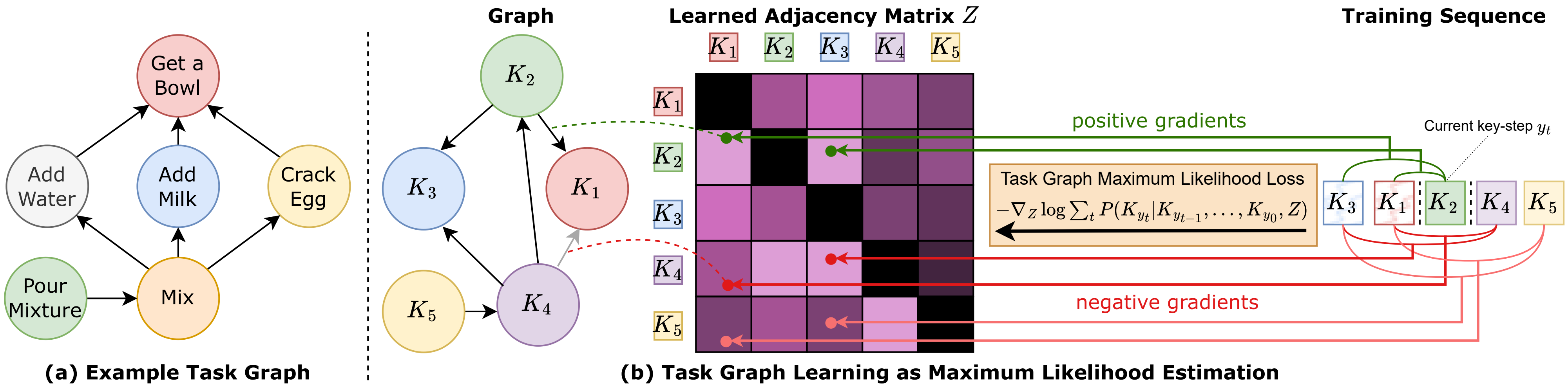
$$\boxed{\mathcal{L}(\mathcal{Y}, Z) = -\sum_{k=1}^{|Y|}\sum_{t=1}^{|y^{(k)}|}\left(\log \sum_{j\in\mathcal{O}(y^{(k)},t)} Z_{y_t j} - \beta \cdot \log \sum_{\substack{h\in\overline{\mathcal{O}(y^{(k)},t)}\\ j\in\mathcal{O}(y^{(k)},t)}} Z_{hj}\right) \tag{6}}$$

Models

2. We propose two approaches to task graph learning…

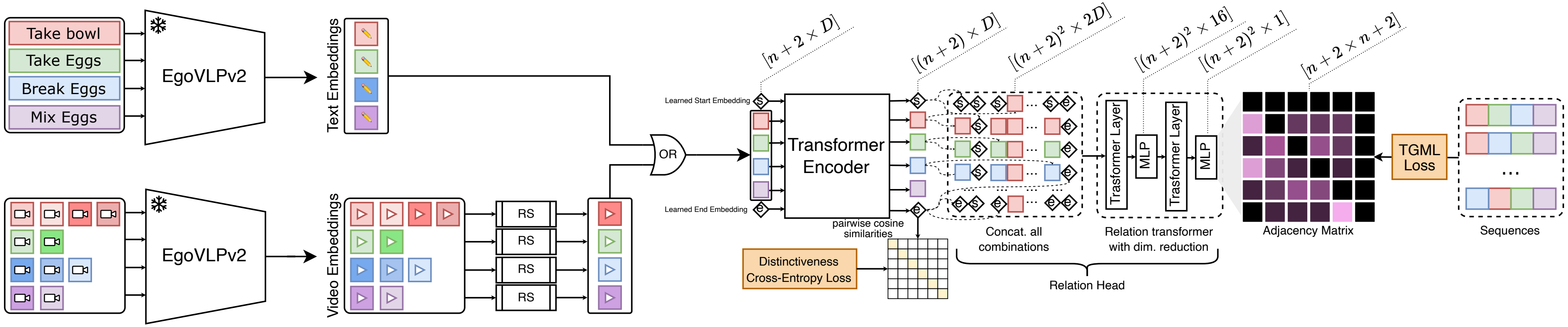
Models – Direct Optimization (DO)

2. …based on **Direct Optimization (DO)** of the adjacency matrix…

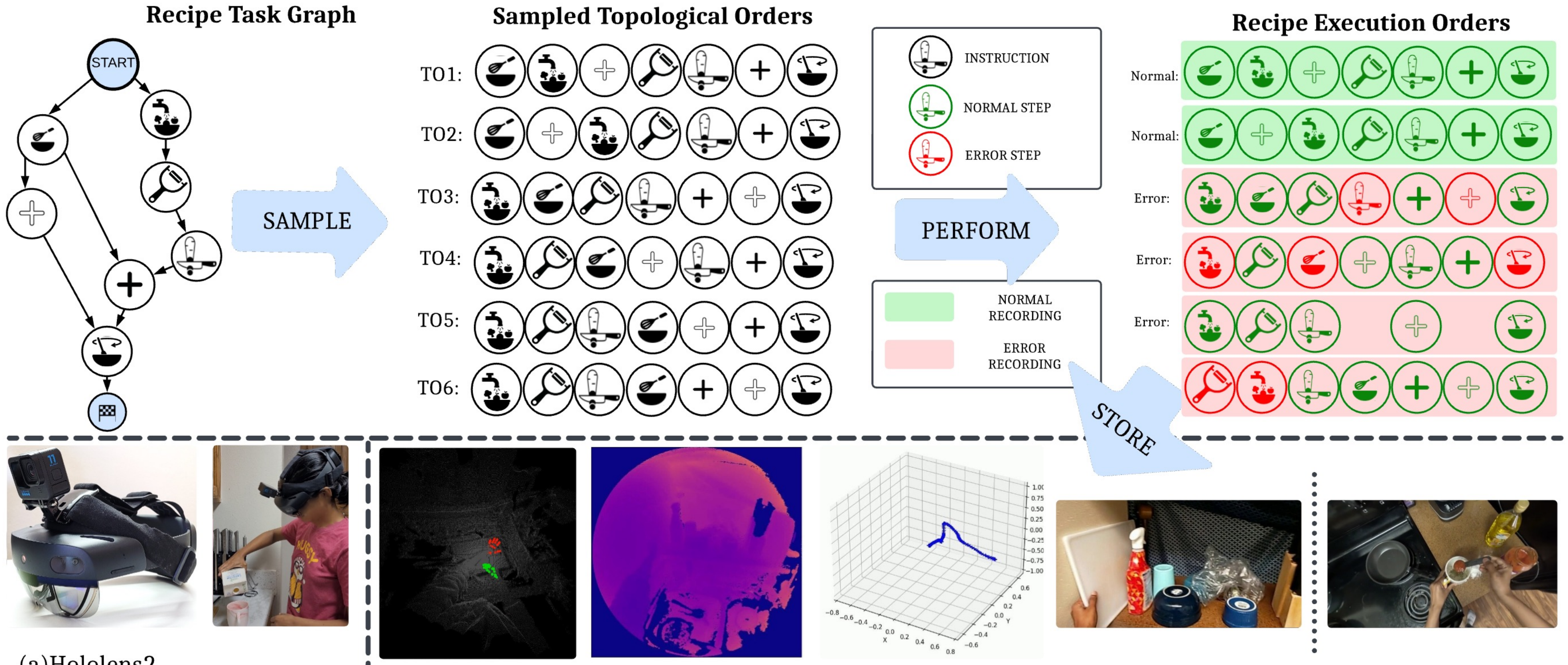


Models – Task Graph Transformer (TGT)

2. …and a transformer based on the processing of textual descriptions of key-steps or video embeddings **Task Graph Transformer (TGT).**



Experiments on CaptainCook4D



(a)Hololens2 and GoPro (b)Participant (c)Hololens2 3D Hand Joints (d)Hololens2 Depth (e)Hololens2 Camera Trajectory (f)Hololens2 RGB (g)GoPro RGB

- Peddi, Rohith, et al. "CaptainCook4D: A dataset for understanding errors in procedural activities." *arXiv preprint arXiv:2312.14556* (2023).

Experiments on CaptainCook4D

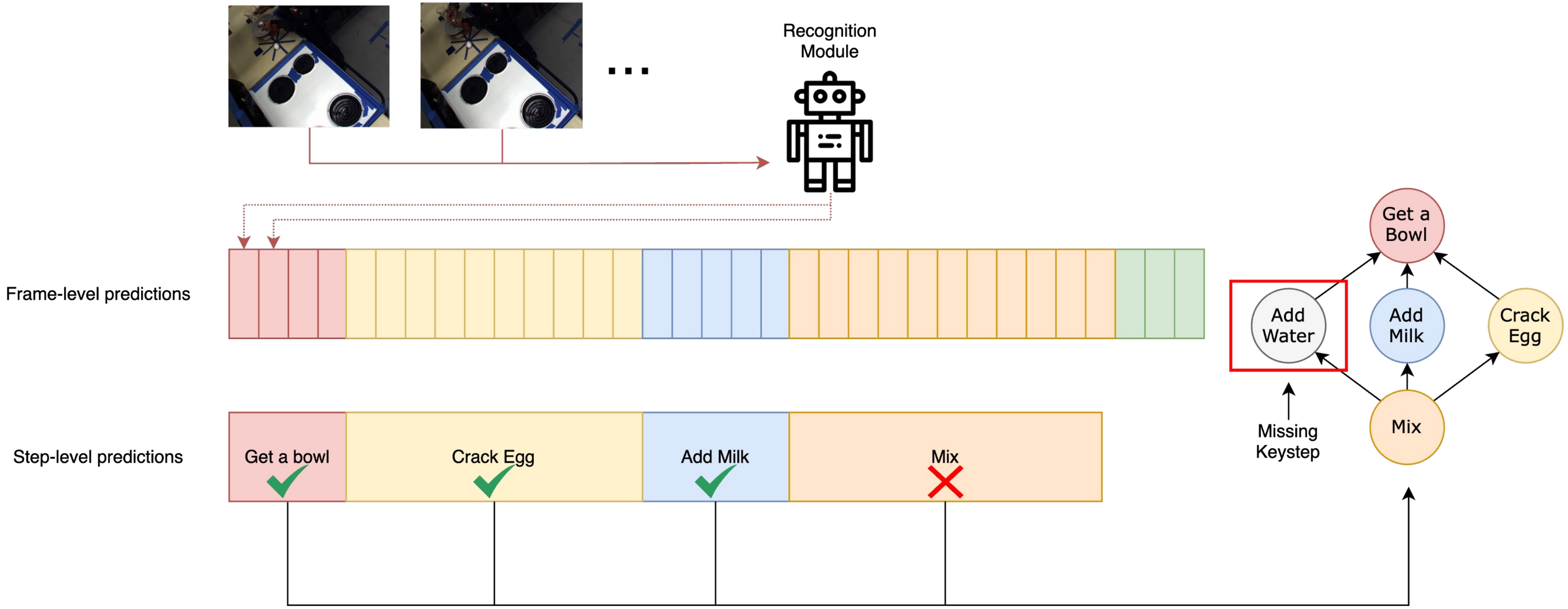
Method	Precision	Recall	F_1
MSGI [39]	11.9	14.0	12.8
LLM	52.9	57.4	55.0
Count-Based [3]	66.7	55.6	60.6
MSG^2 [20]	70.9	71.6	71.1
TGT-text (Ours)	79.9 ±8.8	81.9 ±6.9	80.8 ±8.0
DO (Ours)	**86.4** ±1.5	**89.7** ±1.5	**87.8** ±1.5
Improvement	+15.5	+18.1	+16.7

Method	Ordering	Fut. Pred.
Random	50.0	50.0
TGT-video	**77.3**	**74.3**
Improvement	+27.3	+24.3

- Peddi, Rohith, et al. "CaptainCook4D: A dataset for understanding errors in procedural activities." *arXiv preprint arXiv:2312.14556* (2023).

Online Mistake Detection

3. We assess the accuracy of the proposed task graph generation approach and showcase the usefulness of the learned graphs on the downstream task of online mistake detection.



- Flaborea, Alessandro, et al. "PREGO: online mistake detection in PRocedural EGOcentric videos." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

Online Mistake Detection

3. We assess the accuracy of the proposed task graph generation approach and showcase the usefulness of the learned graphs on the downstream task of online mistake detection.

	Assembly101-O							EPIC-Tent-O						
	Avg	Correct			Mistake			Avg	Correct			Mistake		
Method	F_1	F_1	Prec	Rec	F_1	Prec	Rec	F_1	F_1	Prec	Rec	F_1	Prec	Rec
Count-Based* [3]	26.0	9.2	4.8	85.7	42.8	97.8	27.4	56.6	92.5	92.8	92.2	20.7	20.0	21.4
LLM*	29.3	15.1	8.3	87.2	43.4	96.7	27.9	47.7	86.3	82.4	90.6	9.1	13.3	6.9
MSGI* [39]	33.1	22.7	13.1	84.4	43.5	93.4	28.3	44.5	66.9	51.6	95.2	22.0	73.3	12.9
PREGO* [13]	39.4	32.6	89.7	19.9	46.3	30.7	94.0	32.1	45.0	95.7	29.4	19.1	10.7	86.7
MSG^{2*} [20]	56.1	63.9	51.5	84.2	48.2	73.6	35.8	54.1	92.9	94.1	91.7	15.4	13.3	18.2
TGT-text (Ours)*	62.8	69.8	56.8	90.6	55.7	84.1	41.7	**64.1**	**93.8**	94.1	93.5	**34.5**	33.3	35.7
DO (Ours)*	**75.9**	**90.2**	98.2	83.4	**61.6**	46.7	90.4	58.3	93.5	94.8	92.4	23.1	20.0	27.3
Improvement*	+19.8	+26.3			+13.4			+7.5	+0.9			+12.5		
Count-Based$^+$ [3]	23.2	2.6	1.3	66.7	**43.9**	98.4	28.2	40.4	59.2	42.9	95.5	21.6	80.0	12.5
LLM$^+$	28.1	15.1	7.8	65.5	42.3	89.5	27.7	35.9	61.6	46.7	90.4	10.2	40.0	5.8
MSGI$^+$ [39]	28.4	14.0	7.8	67.9	42.7	90.7	28.0	40.4	59.2	42.9	95.5	21.6	80.0	12.5
PREGO$^+$ [13]	32.5	23.1	68.8	13.9	41.8	27.8	84.1	29.4	41.6	97.9	26.4	17.2	9.5	93.3
MSG^{2+} [20]	46.2	59.1	51.2	70.0	33.2	44.5	26.5	45.2	67.5	52.4	95.1	22.9	73.3	13.6
TGT-text (Ours)$^+$	53.0	67.8	62.3	74.5	38.2	46.2	32.6	43.8	**69.5**	55.8	92.1	18.2	53.3	11.0
DO (Ours)$^+$	**53.5**	**78.9**	85.0	73.5	28.1	22.5	37.3	**46.5**	69.3	54.4	95.2	**23.7**	73.3	14.1
Improvement$^+$	+7.3	+19.8			-5.7			+1.3	+1.2			+1.2		

- Flaborea, Alessandro, et al. "PREGO: online mistake detection in PRocedural EGOcentric videos." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024.

University of Catania | Department of Mathematics and Computer Science

IMAGE PROCESSING LABORATORY

NEURAL INFORMATION PROCESSING SYSTEMS

Thanks for your attention!

Luigi Seminara (luigi.seminara@phd.unict.it)

Antonino Furnari (antonino.furnari@unict.it)

Giovanni Maria Farinella (giovanni.farinella@unict.it)