DUQIM-Net: Probabilistic Object Hierarchy Representation for Multi-View Manipulation: Supplementary material

Vladimir Tchuiev, Yakov Miron, and Dotan Di Castro

February 23, 2022

This document provides supplementary material to [1]. Therefore, it should not be considered a self-contained document, but instead regarded as an appendix of [1]. Throughout this report, all notations and definitions are in compliance to the ones presented in [1].

1 Derivation of Eq. (3) in [1]

For object i, the probability of it not carrying other objects in the pile, i.e. being "on top" of the clutter is denoted $\mathbb{P}(\neg\epsilon_i|I)$. This probability can calculated via multiplying for all objects $i \neq j$ the probability that object j is *not* placed above object i, i.e. $1 - \mathbb{P}(\exists\epsilon_{ji}|I)$. As such, we can compute $\mathbb{P}(\neg\epsilon_i|I)$ via:

$$\mathbb{P}(\neg\epsilon_i|I) = \prod_{j\neq i}(1 - \mathbb{P}(\exists\epsilon_{ji}|I)) = \prod_{j\neq i}(1 - A_{ji}). \tag{1}$$

By applying log to both sides of the equation, we can write the above as a sum of logarithms:

$$\log(\mathbb{P}(\neg\epsilon_i|I)) = \log(1) + \sum_{j\neq i}\log(1 - A_{ji}). \tag{2}$$

Note that $A_{ii} = 0,\ \forall i$. Recall the definition of $\mathbb{P}(\neg\epsilon|I)$:

$$\mathbb{P}(\neg\epsilon|I) \triangleq \left[\mathbb{P}(\neg\epsilon_1|I), \cdots, \mathbb{P}(\neg\epsilon_{N_\mathcal{V}}|I)\right]^T, \tag{3}$$

then we can aggregate all $\mathbb{P}(\neg\epsilon_i|I)$ to a single vector, and write Eq. (2) as a vector equation:

$$\log\begin{bmatrix}\mathbb{P}(\neg\epsilon_1|I)\\ \vdots \\ \mathbb{P}(\neg\epsilon_{N_\mathcal{V}}|I)\end{bmatrix} = \begin{bmatrix}\log(1) + \sum_{j\neq 1}\log(1 - A_{j1})\\ \vdots \\ \log(1) + \sum_{j\neq N_\mathcal{V}}\log(1 - A_{jN_\mathcal{V}})\end{bmatrix}. \tag{4}$$

Note that all elements of A can be reordered such that they correspond to A^T; As such, we receive a batch formulation to compute the entire $\mathbb{P}(\neg\epsilon|I)$ vector:

$$\log(\mathbb{1}_{N_\mathcal{V}\times N_\mathcal{V}} - A^T)\cdot\mathbb{1}_{N_\mathcal{V}} = \log\mathbb{P}(\neg\epsilon|I). \tag{5}$$

2 Derivation of Eq. (4) in [1]

Consider a series of images $I_1, ...I_t, ..., I_k \in H$, then for element t the probability of relation ϵ_{ij} existing can be expressed as a function of $t-1$ as follow via the Bayes Rule:

$$\mathbb{P}(\exists\epsilon_{ij}|I_{1:t}) = \frac{\mathbb{P}(\exists\epsilon_{ij})\mathbb{P}(I_{1:t}|\exists\epsilon_{ij})}{\mathbb{P}(I_{1:t})}. \tag{6}$$

We assume that all images are independent from each other, thus the joint image likelihood term $\mathbb{P}(I_{1:t}|\exists\epsilon_{ij})$ can be written as a product of image likelihoods:

$$\mathbb{P}(I_{1:t}|\exists\epsilon_{ij}) = \prod_{\tau=1}^{t}\mathbb{P}(I_\tau|\exists\epsilon_{ij}). \tag{7}$$

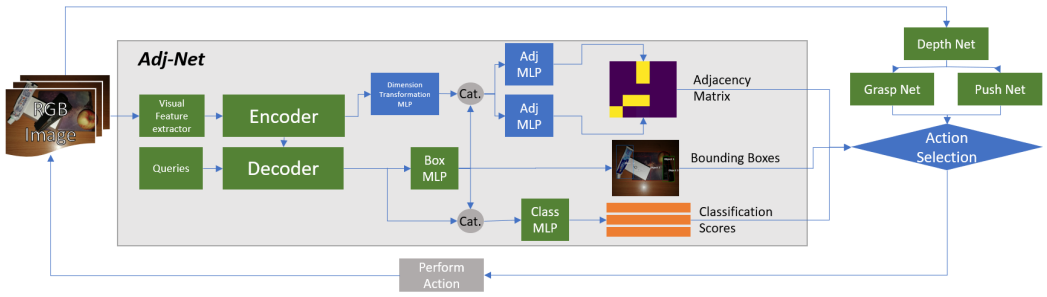


Figure 1: High-level diagram of DUQIM-Net with pushing. The addition compared to fig. [FIG] in the main paper is the pushing-net block near the grasp-net. As such, the pushing-net inputs a push affordance map to the action selection block.

Assuming uninformative priors on $\mathbb{P}(I_\tau)$ for every $\tau \in [1,t]$ and $\mathbb{P}(\exists\epsilon_{ij})$, using Bayes Rule we can deduce that $\mathbb{P}(I_\tau|\exists\epsilon_{ij})$ is proportional to $\mathbb{P}(\exists\epsilon_{ij}|I_\tau)$:

$$\mathbb{P}(I_\tau|\exists\epsilon_{ij}) = \frac{\mathbb{P}(\exists\epsilon_{ij}|I_\tau)\mathbb{P}(I_\tau)}{\mathbb{P}(\exists\epsilon_{ij})} \propto \mathbb{P}(\exists\epsilon_{ij}|I_\tau), \tag{8}$$

and by substituting the above equation in Eq. (7) and subsequently in Eq. (6), we can deduce that $\mathbb{P}(\exists\epsilon_{ij}|I_{1:t})$ is proportional to all the relation probabilities conditioned on the separate images:

$$\mathbb{P}(\exists\epsilon_{ij}|I_{1:t}) \propto \prod_{\tau=1}^{t} \mathbb{P}(\exists\epsilon_{ij}|I_\tau). \tag{9}$$

To complete the equation, the product in the above equation is normalized by dividing with the sum of the same product with the complementary product which represents the probability that the relation ϵ_{ij} does not exist:

$$\mathbb{P}(\exists\epsilon_{ij}|I_{1:t}) = \frac{\prod_{\tau=1}^{t} \mathbb{P}(\exists\epsilon_{ij}|I_\tau)}{\prod_{\tau=1}^{t} (1-\mathbb{P}(\exists\epsilon_{ij}|I_\tau)) + \prod_{\tau=1}^{t} \mathbb{P}(\exists\epsilon_{ij}|I_\tau)}, \tag{10}$$

and by referring to the definition of A_{ij} in the main paper, extending to k, and considering that all $I_t \in H \quad \forall t = [1,k]$, we reach the formulation in the main paper for element-wise posterior adjacency matrix inference:

$$A_{ij}^{post} = \frac{\prod_{I\in H} A_{ij}(I)}{\prod_{I\in H}(1-A_{ij}(I)) + \prod_{I\in H} A_{ij}(I)}. \tag{11}$$

3 DUQIM-Net With Pushing

Multiple works nowadays (e.g. [2]) combine pushing and grasping to increase grasp success rates and action efficiency. In applications where grasping success might be more important than the grasping order, we may consider applying a push to make the objects easier to grasp by scattering them. As such, we present a variation of DUQIM-Net that utilizes object pushing in addition to grasping and taking other viewpoints. The push is performed by closing the gripper claws and performing a 10 centimeter push towards the designated angle. As the grasping network, the pushing network receives the depth estimation parameters, creates a push affordance map, and inputs it into the action selection block. The modified DUQIM-Net diagram is presented in Fig. 1.

An additional hyperparameter q_{th} used to determine whether a push or grasp is performed if $\mathcal{H}_{max}(A) \leq \mathcal{H}_{th}$. Consider the masked grasp quality map $\mathcal{Q}_{bb}$, created by cropping the bounding boxes that correspond to objects that can be grasped according to Adj-Net, a grasped is performed if the maximum grasp quality in $\mathcal{Q}_{bb}$ exceeds q_{th}, otherwise a push is performed at the maximal value of the push affordance map. The modified DUQIM-Net algorithm is presented in Alg. 1.

Algorithm 1 Action selection algorithm with pushing

Require: $\mathcal{E}$, $\mathcal{H}_{th}$, q_{th} ▷ Gets view and switching parameters
$a \leftarrow \emptyset$
$A, \mathcal{C}_{bb} \leftarrow \text{AdjNet}(\mathcal{E})$ ▷ Run Adj-Net
while $a \notin \{\textbf{grasp}, \textbf{push}\}$ **do** ▷ View until grasp/push
  $\mathcal{Q}, \mathcal{P}, \mathcal{M} \leftarrow \text{Nets}(\mathcal{E})$ ▷ Run DuqimNet
  $\mathcal{Q}_{bb} \leftarrow \mathcal{Q}(\mathcal{C}_{bb})$ ▷ Mask grasp quality map
  if $\mathcal{H}_{max}(A) \leq \mathcal{H}_{th}$ & $\max(\mathcal{Q}_{bb}) \geq q_{th}$ **then**
    $a \leftarrow \{\textbf{grasp:} \ \arg\max(\mathcal{Q}_{bb})\}$ ▷ Perform best grasp.
  else if $\mathcal{H}_{max}(A) \leq \mathcal{H}_{th}$ & $\max(\mathcal{Q}_{bb}) < q_{th}$ **then**
    $a \leftarrow \{\textbf{push:} \ \arg\max(\mathcal{P})\}$ ▷ Perform best push
  else if $\mathcal{H}_{max}(A) > \mathcal{H}_{th}$ **then**
    $a \leftarrow \{\textbf{view:} \ \arg\max(\mathcal{M})\}$ ▷ Select informative view
    $A_{prev} \leftarrow A$
    $\mathcal{E} \leftarrow \text{PerformAction}(\mathcal{E}, a$ ▷ Update view
    $A, \mathcal{C}_{bb} \leftarrow \text{AdjNet}(\mathcal{E})$ ▷ Run Adj-Net on new view
    $A \leftarrow \text{FuseAdjacency}(A_{prev}, A)$ ▷ Posterior adj. mat.
  end if
end while
return a ▷ Returns the best grasping or pushing action.

Method	Grasp Att.	Grasp Suc.	Views Added	Pushes
GR-ConvNet	6.14	5.0	0	0
GR-ConvNet + VPG	6.32	5.14	0	0.54
Ours	5.76	5.10	2.08	0
Ours + VPG	5.8	5.16	2.4	0.74

Table 1: Average raw results of all the approaches we tested, in terms of grasp attempts, grasp successes, additional viewpoints added, and pushes performed.

4 Experimental Results for DuqimNet With Pushing

We conducted experiments for the scenarios presented in the main paper with the addition of a pushing network for both our GR-ConvNet baseline, and with Adj-Net activated. We utilize the pushing module of VPG [2] as our pushing network pre-trained on 30 real-life objects. To facilitate pushes in addition to grasps, we consider pushes on objects that support order as object order errors. In addition, we define the grasp order error (GOE) metric that only considers the object order errors caused by grasping an incorrect object.

The raw experiment results are presented in table 1; in general, using Adj-Net increases the number of successful grasps, subsequently reduces the number of grasp attempts, with no clear difference between when push is used and when it is not. DUQIM-Net performs more pushes when Adj-Net is activated because the partial grasping map $\mathcal{Q}_{bb}$ is a subset of $\mathcal{Q}$, therefore $\max \mathcal{Q}_{bb} \leq \max \mathcal{Q}$.

The metrics results are presented in table 2; Adding a pushing net does not appear to have a significant impact on grasp success, and the action-efficiency is reduced. While as expected OOE is increased via the application of pushes, the GOE is smaller when applying pushing, implying that DUQIM-Net with pushing is suited for specific scenarios where the cost of pushing object is small, while grasping them incorrectly and potentially dropping them is costly. However, in scenarios where the manipulated objects are too fragile, it is preferable to utilize DUQIM-Net without pushing.

Method	Grasp Suc.%	Action Eff.%	OOE%	GOE%
GR-ConvNet	85.2	85.2	24.6	24.6
GR-ConvNet + VPG	85.5	81.3	22.22	17.8
Ours	91.1	71.5	10.0	10.0
Ours + VPG	91.2	66.1	14.65	7.61

Table 2: Average metrics results of all the approaches we tested, for grasp success percentage, action efficiency, object order error (OOE), and grasp order error (GOE).

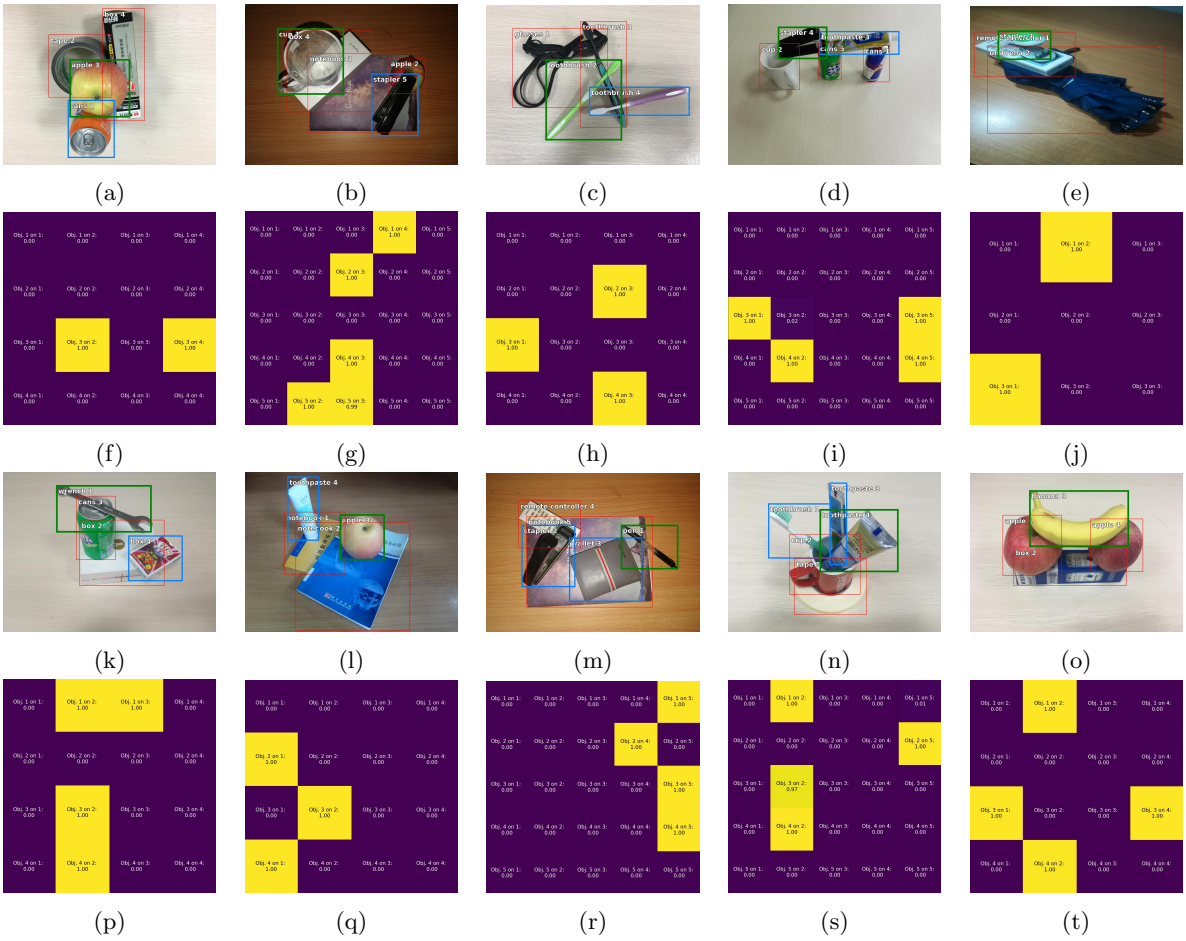


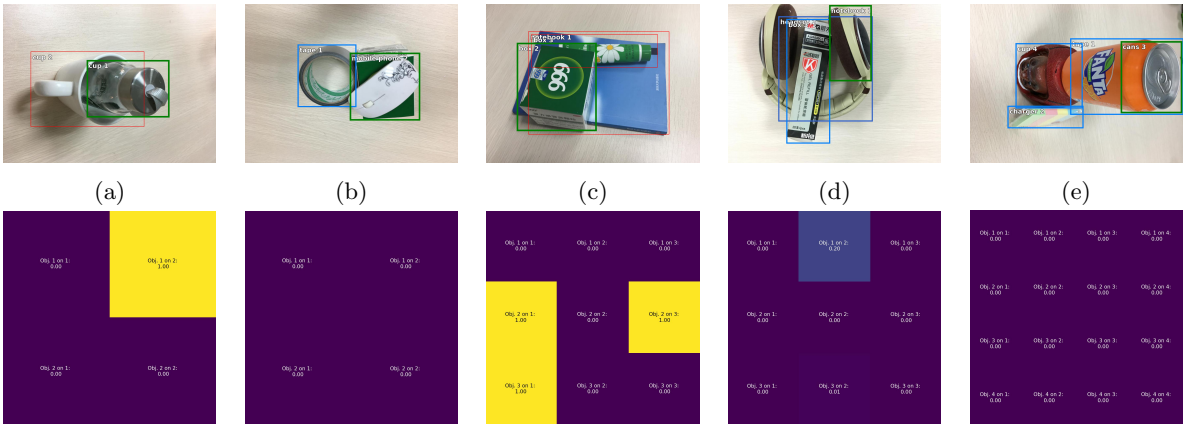
Figure 2: Adj-Net outputs for part of the VMRD test set. For each row of images with bounding boxes and the most probable class labels, the corresponding inferred adjacency matrix is displayed below, with yellow and blue colors correspond to high and low relationship probability respectively.

5 Adj-Net Example Results

In this section we present selected Adj-Net output results for images in the VMRD [3] test set for the RN101 with 31 classes version. In Fig. 2 we present images with correct bounding box, classification, and object relationship inference. Adj-Net can infer the correct relationship with stretched images, e.g. Fig. 2e, and even objects placed in a container, e.g. a cup like in Fig. 2n.

Fig. 3 presents failure cases from the VMRD test set; In Fig. 3a Adj-Net infers the correct relationship, but the bottle is incorrectly classified as a cup due to its transparency. In Fig. 3b Adj-Net failed to detect the both under the mouse, and the latter's flat shape induces an incorrect classification as a mobile phone. In Fig. 3c Adj-Net miss-classified the toothpaste as a box. Fig. 3d and 3e present completely incorrect inference; In Fig. 3d Adj-Net has failed to infer the relation between the box and the headset, in addition to "detecting" an object which does not exist. In Fig. 3e Adj-Net fails to infer the relationship between the can and the cup, miss-classifies the cup and the notebook as they are non-standard in terms of shape, and detects an additional tape object that does not exist.

Fig. 4 presents a visualization of the attention maps for Adj-Net that uses DETR as a backbone instead of Deformable DETR as in the main paper for visualization purposes. The attention of the model is centered around the object's edges for the corresponding row, attempting to determine whether said object is placed on the object in the corresponding column, with the strongest attention is at the corners. For relationships that are not present in the image, the attention map is more spread apart, as if Adj-Net "searches" for possible indicators for the relationship to exist.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 3: Adj-Net outputs failure cases for part of the VMRD test set. For each row of images with bounding boxes and the most probable class labels, the corresponding inferred adjacency matrix is displayed below, with yellow and blue colors correspond to high and low relationship probability respectively. The number of each object corresponds to a row and column in the adjacency matrix.

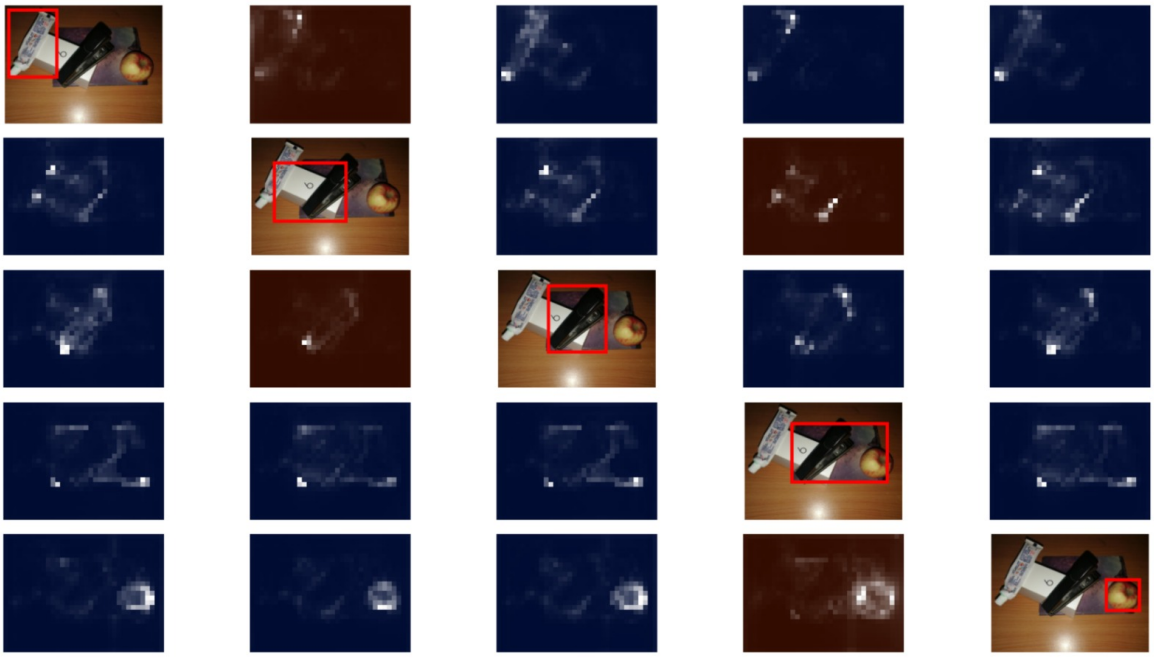


Figure 4: Attention maps of the last encoder layer for Adj-Net that uses DETR as a backbone for visualization. The 5 by 5 grid represents the adjacency matrix. Blue and red visualizations represent adjacency matrix elements in which $\mathbb{P}(\epsilon_{ij}|I)$ around 0 and 1 respectively. The diagonals show the object bounding box of the corresponding row and column, and by definition $\mathbb{P}(\epsilon_{ii}|I) = 0$ for all i so no information is lost in this figure.

References

[1] V. Tchuiev, Y. Miron, and D. Di-Castro, "Duqim-net: probabilistic object hierarchy representation for multi-view manipulation," *IEEE Robotics and Automation Letters, Submitted*, 2022.

[2] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[3] H. Zhang, X. Lan, X. Zhou, Z. Tian, Y. Zhang, and N. Zheng, "Visual manipulation relationship network for autonomous robotics," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 118–125.