M2T2: Multi-Task Masked Transformer for Object-centric Pick and Place

Anonymous Author(s)
Affiliation
Address
email

Abstract: With the advent of Large Language Models (LLMs) and large-scale robotic datasets [1, 2, 3, 4, 5, 6], there has been tremendous progress in high-level decision-making for object manipulation. Unfortunately, these generic models are restricted by the inability of low-level pick-and-place primitives in terms of object generalization. In contrast, existing specialized models [7, 8] excel in low-level manipulation of unknown objects but only work for a single task. To bridge this gap, we present M2T2, a solution for picking and placing arbitrary objects in cluttered scenes. M2T2 is a transformer model which predicts pick and place poses for all (unknown) objects given just a point cloud observation of the scene. M2T2 is trained with a large-scale synthetic dataset consisting over 160K scenes and achieves zero-shot *sim2real* transfer on the real robot. It also significantly outperforms the state-of-the-art task-specific baseline system [8, 7] by about 19% in overall performance and as much as 37.5% in challenging scenes were the object needs to be re-oriented for collision-free placement with precision.

Keywords: Object Manipulation, Pick-and-place, Multi-task Learning

1 Introduction

The successful completion of many complex manipulation tasks such as object rearrangement relies on robust action primitives that can handle a large variety of objects. Recently, there has been great progress made in object rearrangement with [1, 2, 3, 4, 5, 6] where the focus is mostly on the high-level planning. However, these efforts are typically restricted on the robustness of low-level skills such as having few objects in the scene, using soft objects to accommodate for manipulation skills, hard-coded low-level actions, and a limited or reduced dimensionality in action space. On the other hand, there are works that focus on manipulation skills in complex challenging environments [7, 9, 10, 11, 12] that are specialized models trained for a single task.

In this work, we propose **M**ulti-**T**ask **M**asked **T**ransformer (M2T2), a object-centric transformer model for picking and placing given the point cloud of the scene. M2T2 is based on Mask2Former [13] and it simultaneously generates predictions for both the picking and placing tasks. For the picking task, it generates a dense set of grasps for all the objects in the scene in a single forward pass. For placing, conditioned on the point cloud of scene and the segmented point cloud of object in hand, it predicts where in the scene the objects can be placed. In addition to the placement position, it also predicts a set of discretized rotation for each placement point where the object can be placed (see Fig. 1) This is important for scenarios where placing the objects without collisions is not feasible without re-orienting the object. M2T2 predicts target gripper poses for pick and place tasks and as a result offers more flexibility compared to end-to-end approaches [14]. It allows for integration with existing motion planning and control policies and can also facilitate multi-step planning due to its ability to predict goal poses for all the objects in the scene. In addition, M2T2 predicts multiple goal poses which is essential for overcoming the collision and kinematic restrictions of robot at test-time. Overall our contributions are as follows:

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

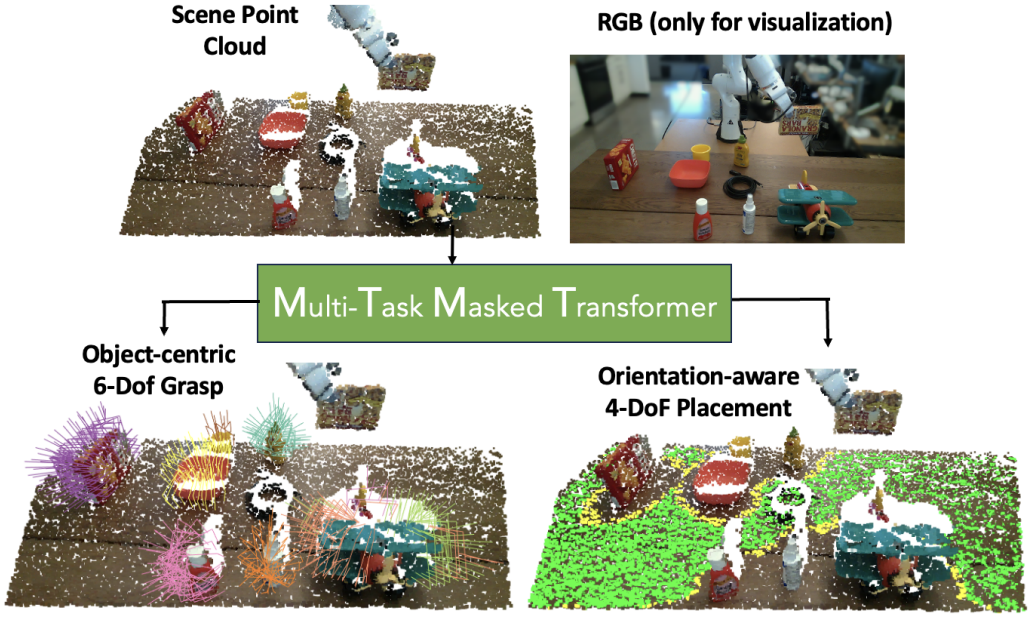


Figure 1: We propose M2T2, a multi-task model for 6-DoF picking and placing. M2T2 takes a raw 3D point cloud and predicts 6-DoF grasps per object (lower left) and orientation-dependent placements. The lower right plot shows placement locations (green) where the object can fit in any orientation and locations (yellow) where only a subset of orientations are possible.

1. We present a unified transformer model for object-centric, 6-DoF grasping and placing. Our experimental results show that M2T2 outperforms state-of-the-art methods [7, 8] in terms of success rate and output diversity.
2. A large-scale dataset for training M2T2, consisting of 164K cluttered scenes and 8.8K [15] different objects. This is the first large-scale dataset for object placement in stable poses, that we are aware of.
3. We show that M2T2 achieves zero-shot *sim2real* transfer for pick-and-place in unknown scenes and objects, outperforming baseline methods by about 19%.

2 Related Work

Multi-Task Learning Recent attention in multi-task learning has focused on using a single Transformer model for predicting across multiple AI tasks. For instance, LLMs [16] perform several tasks including question answering, text summarization and machine translation. Such consolidated models have also emerged in visual processing, such as the Segment Anything model for segmentation [17], ImageBind [18] for learning a common representation across different modalities, and Mask2Former [13] for panoptic, semantic, and instance segmentation.

Multi-Task Learning in Robotics Multi-task learning and foundation models have seen recent interest in the community for learning low-level manipulation skills. Multi-task learning has been shown to be improve sample-efficiency in robotics, such as in learning a common representation for multiple tasks [19, 20, 21, 22]. CLIPort [23] presented a language-conditioned imitation learning agent, based of Transporter Nets [24], that predicts 3-DoF actions prompted with CLIP features. VIMA proposed multimodal prompts, combining language and image instructions to prompt a transformer agent [6] to predict 3-DoF actions for object manipulation. PerAct [25] presented a 6-DOF language conditioned agent that can learn multiple tasks with just a handful of keyframe demonstrations in the real world. However, none of these works showed generalization to new objects or

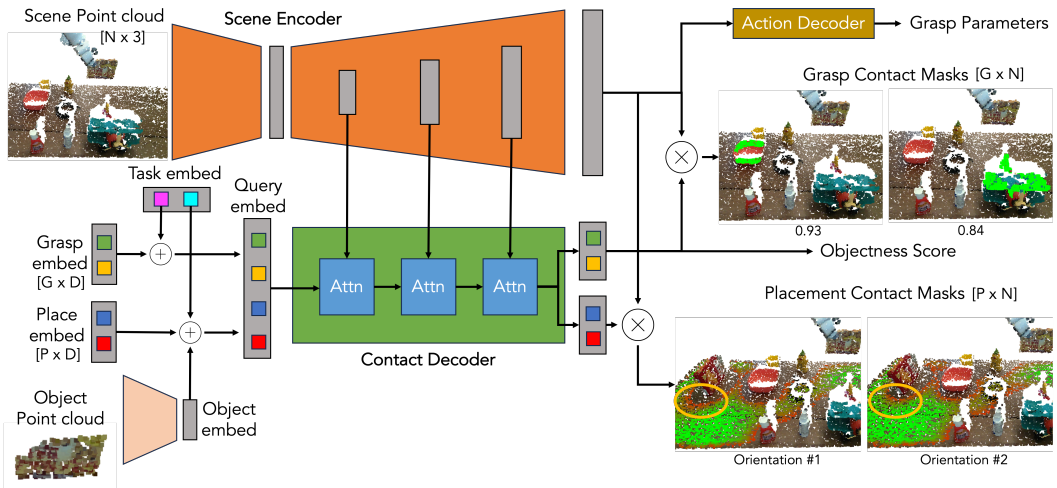


Figure 2: M2T2 is a transformer-based model architecture for generating pick-and-place poses.

environments and there is no sim-to-real in these works. PaLM-E [20] trained a LLM with multi-modal prompts but still relied on a language conditioned policy [5, 14] to execute low-level control in the real world. M2T2 is trained with simulated data and it generalizes to unseen objects and scenes in the real world. It utilizes the full 6-DoF action space and provides more flexibility to be interfaced with other planning and controller modules.

Object Grasping 6-DoF grasping is a very well studied in robotics [26, 27, 12, 28, 29, 30, 7]. These methods typically take the point cloud [26, 27, 12, 28] or voxelization of the scene [29, 30] and predict 6-DoF poses that gripper needs that gripper needs to move to. However, all these works are geared toward just grasping task. M2T2 is closest to the formulation of [7] for representing grasps using contact mask and additional parameters specifying the orientation of grasps and the offset from the contact point. We apply the same formulation for placement task as well.

Object Placement With the increase in robustness of grasping methods, there is a rise in number of works that consider object placement as well. The object placement works can fall into two categories: 1) language guided [31, 32, 33] where the placement positions are predicted conditioned on the language commands 2) Unconditional placement methods [34, 8] where the goal is to find all the placement positions given the object point cloud and scene point cloud. CabiNet [8] utilizes a highly efficient model [34] to learn the collisions given the scene point cloud, object point cloud, and large batches of object poses. Placement positions are found by doing rejection sampling against the same model. M2T2 predicts placement poses directly without the need for rejection sampling.

3 Technical Approach

M2T2 takes as input the 3D point cloud of the scene and predicts target poses for the gripper to either pick or place objects. M2T2 predicts 6-DoF (3-DoF translation + 3-DoF orientation) grasp poses for the pick task and 4-DoF (3-DoF translation + 1-DoF planar orientation) placement poses for placing task. One key difference between our formulation of placement compared to prior works [8, 34] is that we consider not only the placement location, but also the planar rotation of the object to be placed. This ensures that the object can be re-oriented before placed to fit into cluttered spaces.

3.1 Architecture

M2T2 consists of four parts: scene encoder, object encoder, contact decoder and action decoder. The overall architecture is illustrated in Fig. 2. We describe the key elements of the architecture below and more details can be found in the supplementary.

Scene Encoder: The scene encoder encodes a 3D point cloud of the scene into multi-scale feature maps that serve as context for the contact decoder. Specifically, it produces four feature maps that are $1/64, 1/16, 1/4, 1$ times the input size respectively. Each feature vector in the feature maps is grounded to a point in the input point cloud. We adapt a PointNet++ [35] designed for semantic segmentation as the scene encoder, but in principle, any network that produces multi-resolution feature maps from 3D point clouds can serve as the scene encoder.

Object Encoder: The object encoder is only used for placement task. It is based on PointNet++ [35] and encodes 3D point cloud of the placement object to a single feature vector as the placement query.

Contact Decoder: The contact decoder is the core module of M2T2. It is a transformer, based on Mask2Former [13], that predicts where to make contact for both grasping and placing. We used the grasp representation of [7] for grasping where each grasp is anchored around the visible point on the object that makes contact with the gripper during grasping and the model predicts additional parameters specifying the relative transform of grasp with respect to the contact point. We extend this representation for placing by defining contact point as the location where the center of object point cloud projects on the table.

The contact decoder takes the multi-resolution feature maps from the scene encoder and outputs $\hat{G}$ grasp contact masks, each assigned with an objectness score indicating how likely this contact mask belongs to an object. It also outputs $\hat{P}$ placement contact masks for different orientations, indicating valid placement locations for each orientation of the placement object. Overall, the contact decoder has three sets of learnable embeddings: G grasp embeddings where G it the maximum number of objects, P placement embeddings where P is the number of discretized orientations, and two task embeddings. The embedding vectors are used to construct a set of query tokens. First, the grasp embeddings are added to the the grasp task embedding, resulting in G grasping queries. Then, the placement query from the object encoder is added to the placement embeddings and the placement task embedding to produce P placement queries. The final output tokens of the transformer is cross-attended with the per-point feature map produced by the scene encoder. The attention matrix is then passed through a sigmoid and thresholded to produce $\hat{G}$ grasp contact masks and $\hat{P}$ placement contact masks. The G grasp tokens are also used to predict the grasp objectness scores by an MLP.

Action Decoder: The action decoder is used for predicting additional continuous parameters for certain tasks. In our case, only grasping needs this module. Specifically, it is a 3-layer MLP that takes the per-point feature map from scene encoder and predicts a 3D approach direction, a 3D contact direction and a 1D grasp width for each point. See 3.2 for how these values are used together with the contact point to reconstruct the grasp pose.

3.2 Model Inference

Grasp Pose Prediction: During inference, we first select the contact masks whose objectness score is above 0.5. Then, for each point $\mathbf{p}$ within the contact mask, we take the corresponding grasp parameters from the action decoder to reconstruct a 6-DoF grasp pose $(R_{\text{grasp}}, \mathbf{t}_{\text{grasp}}) \in \text{SE}(3)$ as

$$\mathbf{t}_{\text{grasp}} = \mathbf{p} + \frac{w}{2}\mathbf{c} + d\mathbf{a} \tag{1}$$

$$R_{\text{grasp}} = \begin{bmatrix} | & | & | \\ \mathbf{c} & \mathbf{c} \times \mathbf{a} & \mathbf{a} \\ | & | & | \end{bmatrix} \tag{2}$$

where $\mathbf{c}$ is the unit 3D contact direction, $\mathbf{a}$ is the unit 3D approach direction, w is the 1D grasp width and d is the constant distance from the gripper base to the grasp baseline (the line between two fingers). We refer readers to Contact-Grasp-Net paper for more details on the formulation [7].

Placement Pose Prediction: The P placement contact masks represent valid placement locations when the object in the gripper is rotated by P discrete planar rotations R_{planar}. To recover the placement poses, we first compute the bottom center $\mathbf{b}$ of the object point cloud which is used as the reference point for contact. Then, we use forward kinematics to obtain the gripper's current pose

$(R_{\text{ee}}, \mathbf{t}_{\text{ee}})$. The 6-DoF placement pose $(R_{\text{place}}, \mathbf{t}_{\text{place}})$ can be computed as

$$\mathbf{t}_{\text{grasp}} = \mathbf{p} + R_{\text{planar}}(\mathbf{t}_{\text{ee}} - \mathbf{b}) \tag{3}$$

$$R_{\text{grasp}} = R_{\text{planar}} R_{\text{ee}} \tag{4}$$

3.3 Training Objective

Grasping: The grasping objective consists of four terms: objectness loss L_{obj}, mask loss L_{mask}, parameter loss L_{param} and ADD-S loss $L_{\text{ADD-S}}$.

Because the number of objects N in the scene is unknown, we set G, the number of grasp queries, to a large number (see Sec. 4.2 for ablations). For each output grasp token g_i, we compute an objectness score between 0 and 1, o_i, and a contact mask, M_i^{grasp}. o_i is used to determine whether M_i^{grasp} corresponds to a graspable object and should be added to the inference output. We first compute the following cost for each pair of prediction $(o_i, M_i^{\text{grasp}})$ and ground truth mask M_j^{gt}

$$C_{ij} = 1 - o_i + \text{BCE}(M_i^{\text{pred}}, M_j^{\text{gt}}) + \text{DICE}(M_i^{\text{pred}}, M_j^{\text{gt}}) \tag{5}$$

where BCE is binary cross entropy and DICE is the DICE loss from [36]. Then, we use Hungarian matching to obtain the set of predictions $\mathcal{M} = \{m_i\}$ that minimizes the total cost $\sum_{j=1}^{N} C_{m_j j}$. We compute the objectness loss by labeling all matched tokens as positive and others as negative

$$L_{\text{obj}} = \frac{1}{G} \sum_{i=1}^{G} -\left[\mathbb{1}(i \in \mathcal{M}) \log(o_i) + (1 - \mathbb{1}(i \in \mathcal{M})) \log(1 - o_i)\right] \tag{6}$$

We compute the mask loss between the matched masks and the ground truth as

$$L_{\text{mask}} = \frac{1}{N} \sum_{j=1}^{N} \text{BCE}(M_{m_j}^{\text{pred}}, M_j^{\text{gt}}) + \text{DICE}(M_{m_j}^{\text{pred}}, M_j^{\text{gt}}) \tag{7}$$

The parameter loss is defined between the output of the action decoder and the ground truth grasp parameters. We only apply this loss to $\mathcal{C}^{\text{gt}}$, the set of contact points of ground truth grasps. We discretize the grasp width w into 8 bins following [7] (the bin center is used in inference). The parameter loss is defined as

$$L_{\text{param}} = \frac{1}{|\mathcal{C}^{\text{gt}}|} \sum_{i \in \mathcal{C}^{\text{gt}}} -\mathbf{c}_i^{\text{pred}} \cdot \mathbf{c}_i^{\text{gt}} - \mathbf{a}_i^{\text{pred}} \cdot \mathbf{a}_i^{\text{gt}} + \text{CrossEntropy}(w_i^{\text{pred}}, w_i^{\text{gt}}) \tag{8}$$

where $\mathbf{c}$ are unit contact directions and $\mathbf{a}$ are unit approach directions.

The ADD-S loss is introduced by [7] and is critical for good grasp confidence estimation (see Sec. 4.2 for ablations). To compute it, we first need to define 5 key points $\{\mathbf{v}_k\}$ on the gripper. Then, for each pair of predicted grasp and ground truth grasp, we compute the total distance between transformed key points

$$d_{ij} = \sum_{k=1}^{5} \|(R_i^{\text{pred}} \mathbf{v}_k + \mathbf{t}_i^{\text{pred}}) - (R_j^{\text{gt}} \mathbf{v}_k + \mathbf{t}_j^{\text{gt}})\| \tag{9}$$

Next, we find the closest ground truth to each prediction $n_i = \text{argmin}_j d_{ij}$ and compute ADD-S as

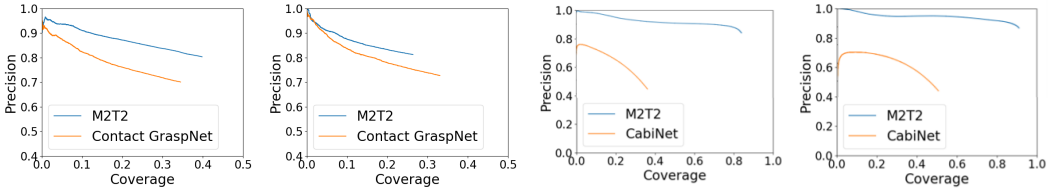
$$L_{\text{ADD-S}} = \frac{1}{|\mathcal{C}^{\text{pred}}|} \sum_{i \in \mathcal{C}^{\text{pred}}} s_i d_{in_i} \tag{10}$$

where $\mathcal{C}^{\text{pred}}$ is the set of contact points of predicted grasps. Note that the ADD-S loss is weighted by the predicted confidence s_i. As a result, predicted grasps that are far away from any ground truth grasp will get a larger penalty on confidence, which improves grasp confidence estimation.

Placing: The placing objective is defined as a combination of BCE and DICE [36] between the predicted and ground truth placement masks

$$L_{\text{placing}} = \frac{1}{P} \sum_{i=1}^{P} \text{BCE}(M_i^{\text{pred}}, M_i^{\text{gt}}) + \text{DICE}(M_i^{\text{pred}}, M_i^{\text{gt}}) \tag{11}$$

This is the only loss for placing since no other learnable quantities are needed to reconstruct the placement poses (see 3.2).



(a) Grasping Seen (b) Grasping Unseen (c) Placing Seen (d) Placing Unseen

Figure 3: M2T2 outperforms task-specific models – Contact-GraspNet [7] for grasping and CabiNet [8] for placing – on objects from seen categories (a,c) and unseen categories (b,d).

3.4 Synthetic Data Generation

We build a synthetic dataset with 132K cluttered scenes for training and evaluating 6-DoF picking and placing methods. There are 64K training scenes and 2K test scenes for picking and placing each. There are 1 to 15 objects per scene scattered on a randomly sized table mounted with a Franka Emika robot arm. The objects are sampled from the ACRONYM [15] dataset, which contains 8.8K object models, each labeled with 2K grasps. The objects are from 252 different categories, 12 of which are excluded from training. Half of the test scenes contain only objects from the 12 unseen categories. For each scene, we render a 512×512 depth image from a random viewpoint above the table to generate the scene point cloud. We include example images of the dataset in the supplementary.

4 Experimental Evaluation

4.1 Evaluation in Simulation

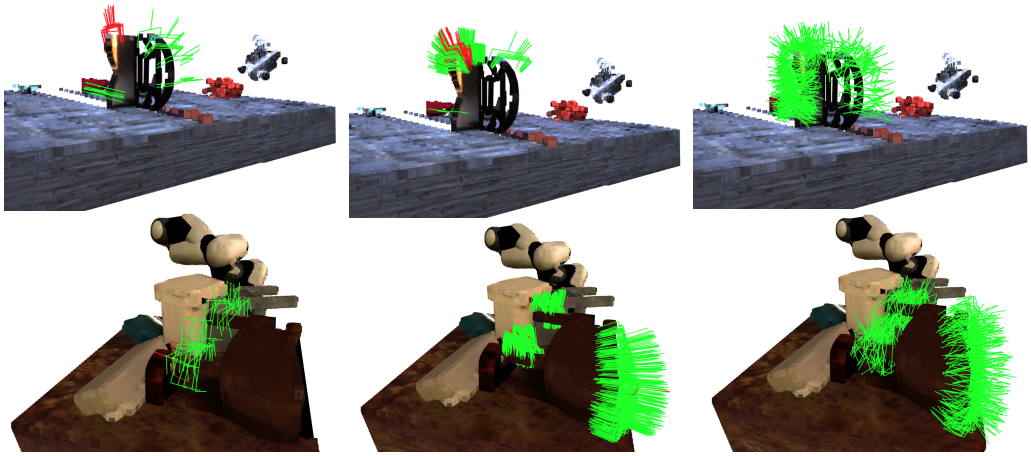
Evaluation Metric: We use the precision-coverage curve as the metric for our evaluation in simulation. To plot this curve, we start with a confidence threshold of 1 and add grasps/placements to the set of predictions by incrementally lowering the confidence threshold until 0.5. In the mean time, we keep track of two numbers: precision, the percentage of successful grasps/placements, and coverage, the percentage of ground truth grasps/placements that are within 5 cm of any predicted pose. Finally, we plot the coverage on the x-axis and precision on the y-axis. In practice, we found that this curve is a good indicator of a model's performance in the real world.

A grasp is considered successful if the gripper does not collide with the scene (including occluded parts) and is stable. We evaluate the stableness of a grasp by shaking the grasped object for 5 seconds in the Isaac gym simulator [37] with a *physx* physics engine, identical to the evaluation in ACRONYM [15]. A placement is considered successful if both the gripper and the object are collision-free and the bottom of the object is less than 5 cm from the correct placement surface.

M2T2 vs. Specialized Baseline Models: We compare M2T2 against two state-of-the-art specialized models: Contact-GraspNet [7] for grasping and CabiNet [8] for placing. The results are summarized in Fig. 3. M2T2 outperforms both models by a significant margin, especially on the placement. This shows the advantage of orientation reasoning for placement. In many cases, it is not possible to find a good placement pose without rotating the object in hand. For grasping, we found that Contact-GraspNet predicts fewer grasps which can be attributed to the fact that Contact-Graspnet only takes the cropped point cloud around the object as the input and also it has less model capacity compared to M2T2.

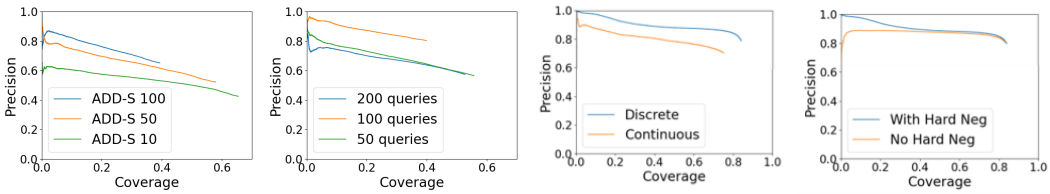
4.2 Ablations

Choice of ADD-S: We find that setting a larger weight for the ADD-S loss has a critical impact on the grasping performance. As shown in Fig. 5a, setting lower ADD-S increases grasp coverage at the expense of precision which is not desirable.



(a) Contact-GraspNet (b) M2T2 (Ours) (c) Ground Truth

Figure 4: M2T2 generates a more diverse distribution of grasps compared to Contact-Graspnet [7].



(a) ADD-S weight (b) Number of tokens (c) Discrete rotation (d) Hard negative mining

Figure 5: Ablation studies

Number of Grasp Queries: We experimented with different number of grasp query tokens and find 100 to be an appropriate number, as shown by the results in Fig. 5b.

Discrete vs. Continuous Rotation: We compared our model with a variant where there is only a single placement mask and the placement rotations are regressed just like the grasp parameters. As shown by Fig. 5c, it is better to have a set of placement masks corresponding to discrete rotations. Since multiple orientations of the object can be valid for a given placement location, regressing to a single rotation does not model the multi-modality of placement orientations.

Importance of Hard Negative Mining: As mentioned in Sec. 3.3, we use hard negative mining (similar to [34, 8, 38]) by applying the mask loss for 1024 points with the largest loss. Without this trick, the quality of most confident placements become significantly worse (see Fig. 5d).

4.3 Real Robot Experiments

Hardware Setup: We evaluated M2T2 on a tabletop manipulation setting with a 7-DOF Franka Panda Robot and a parallel jaw gripper. For perception, we used a single extrinsically calibrated Intel Realsense L515 RGB-D camera overlooking the scene. We use the model from [39] to run instance segmentation, the neural collision checker from CabiNet [8] for reactive path planning in a model predictive control framework. We want to emphasize that while we did not use CabiNet [8]

Table 1: Real Robot Experiments

Model	Pick	Place	Place re-orient	Overall
M2T2 (ours)	**85.7%**	**72.2%**	**62.5%**	**61.9%**
Contact-GraspNet [7] + CabiNet [8]	76.2%	56.2%	25.0%	42.9%

Figure 6: Our robot experimental setup. *Left:* Scenes where the target object (highlighted in red) needs to be reoriented to be placed in the placement region (shown in green). *Right:* A example of a scene where objects are sequentially moved from the right to the initially empty region on the left.

for collision-free object placement, we still use it for motion generation. The user selects the target object by clicking on the external camera image and specifies the placement region by drawing a segmentation mask on the same image. All inference is run on a NVIDIA Titan RTX gpu.

Results: Table 1 shows the success rate of 21 pick-and-place sequences in 7 different scenes. Each scene contains more than 5 objects and all objects are unseen during training. We do not provide 3D models for any of the object. The placing success rate is conditioned on picking success, i.e. overall success = picking success $\times$ placing success. We designed four scenes where the object has to be re-oriented before placing to fit in the target region, as shown on the left in Fig 6. We can see that M2T2 significantly outperforms the baseline system, which is a combination of state-of-the-art task-specific models. Notably, M2T2 is 9.5% higher for grasping than [7] and 37.5% higher than [8] for the more challenging re-orientation placement. 2/3 pick failures for M2T2 were for cup objects, which were out-of-distribution during training. 4/5 pick failures for the baseline [7] was because the model did not generate grasps at all or the generated ones were not reachable based on Inverse Kinematics (IK). On the other hand, M2T2 generates grasps with higher coverage and hence have a greater chance of being within the robot's kinematic workspace. We only use 8 bins for the orientation discretization, and further increasing the discretization granularity could potentially reduce our placement error. The real robot executions are included in the supplementary video and the website.

5 Conclusion

In this paper we present Multi-Task Masked Transformer (M2T2), an object-centric transformer model for pick-and-place of unknown objects in clutter. We train M2T2 with a large-scale dataset of over 160K scenes of pick-and-place in clutter. M2T2 outperforms state-of-the-art specialized models in 6-DoF grasping [7] and placing [8], by about 19% in overall performance in the real world. M2T2 is especially proficient in re-orienting objects for precise collision-free placement. In future, we are going to add more tasks to the model to learn a stronger representation suitable for variety of tasks.

Limitations: While it is convenient to perform pick-and-place with just a single depth camera observation, M2T2's performance is bounded by the extent of (self) occlusion and the visibility of contact points. M2T2 currently only generalizes to tabletop scenes but this could be improved by training with more diverse procedurally generated synthetic data such as in [8, 40]. Additionally, the grasp predictions for each token can spread across multiple nearby objects even though the training, we want each pick token to predict grasps corresponding to only one object.

References

[1] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023.

[2] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*, 2022.

[3] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

[4] N. Yokoyama, A. Clegg, E. Undersander, S. Ha, D. Batra, and A. Rai. Adaptive skill coordination for robotic mobile manipulation. In *arXiv preprint arXiv:2210.06407*, 2023.

[5] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. In *arXiv preprint arXiv:2210.06407*, 2022.

[6] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv: Arxiv-2210.03094*, 2022.

[7] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[8] A. Murali, A. Mousavian, C. Eppner, A. Fishman, and D. Fox. CabiNet: Scaling neural collision detection for object rearrangement with procedural scene generation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2023. URL https://arxiv.org/abs/2304.09302.

[9] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

[10] S. Song, A. Zeng, J. Lee, and T. Funkhouser. Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations. *Robotics and Automation Letters*, 2020.

[11] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg. Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*, 2017.

[12] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox. 6-dof grasping for target-driven object manipulation in clutter. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6232–6238. IEEE, 2020.

[13] B. Cheng, A. Choudhuri, I. Misra, A. Kirillov, R. Girdhar, and A. G. Schwing. Mask2former for video instance segmentation. *arXiv preprint arXiv:2112.10764*, 2021.

[14] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[15] C. Eppner, A. Mousavian, and D. Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.

[16] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[18] R. Girdhar, A. El-Nouby, Z. Liu, M. Singh, K. V. Alwala, A. Joulin, and I. Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

[19] L. Pinto and A. Gupta. Learning to push by grasping: Using multiple tasks for effective learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2161–2168. IEEE, 2017.

[20] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.

[21] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *Conference on Robot Learning*, 2020.

[22] W. Yuan, C. Paxton, K. Desingh, and D. Fox. Sornet: Spatial object-centric representations for sequential manipulation. In *Conference on Robot Learning*, pages 148–157. PMLR, 2022.

[23] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. In *Proceedings of the 5th Conference on Robot Learning (CoRL)*, 2021.

[24] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. *Conference on Robot Learning*, 2020.

[25] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[26] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.

[27] A. Mousavian, C. Eppner, and D. Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2901–2910, 2019.

[28] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020.

[29] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan. Volumetric grasping network: Real-time 6 dof grasp detection in clutter. In *Conference on Robot Learning*, 2020.

[30] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu. Synergies between affordance and geometry: 6-dof grasp detection via implicit representations. *Robotics: science and systems*, 2021.

[31] W. Liu, T. Hermans, S. Chernova, and C. Paxton. Structdiffusion: Object-centric diffusion for semantic rearrangement of novel objects. *arXiv preprint arXiv:2211.04604*, 2022.

[32] W. Liu, C. Paxton, T. Hermans, and D. Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[33] I. Kapelyukh, V. Vosylius, and E. Johns. Dall-e-bot: Introducing web-scale diffusion models to robotics. *IEEE Robotics and Automation Letters*, 2023.

[34] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox. Object rearrangement using learned implicit collision functions. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6010–6017, 2021.

[35] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[36] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016.

[37] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[38] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[39] A. Mousavian, L. Manuelli, B. Okorn, Y. Xiang, C. Eppner, A. Murali, and D. Fox. Object-seeker: A unified framework for one-shot object detection, tracking, and instance segmentation of everyday objects. In *arXiv*, 2023.

[40] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox. Motion policy networks. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.