2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10610283

EgoPAT3Dv2: Predicting 3D Action Target from 2D Egocentric Vision for Human-Robot Interaction

Irving Fang[1,*], Yuzhong Chen[1,*], Yifan Wang[1,*], Jianghan Zhang[1,†], Qiushi Zhang[1,†], Jiali Xu[1,†], Xibo He[1], Weibo Gao[2], Hao Su[2], Yiming Li[1], Chen Feng[1,✉]

https://ai4ce.github.io/EgoPAT3Dv2/

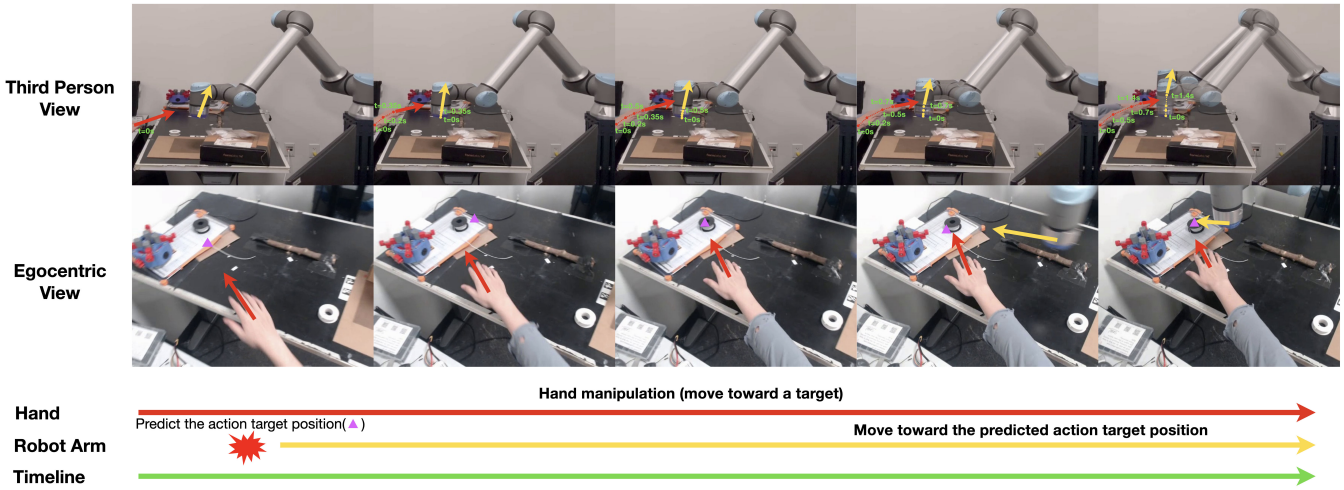

Fig. 1: **Real-World Demonstration of EgoPAT3Dv2**. A human wearing a helmet camera manipulates objects in a shared workspace with a UR10E cobot. The cobot tries to reach the anticipated 3D action target with the shortest Cartesian path.

***Abstract*— A robot's ability to anticipate the 3D action target location of a hand's movement from egocentric videos can greatly improve safety and efficiency in human-robot interaction (HRI). While previous research predominantly focused on semantic action classification or 2D target region prediction, we argue that predicting the action target's 3D coordinate could pave the way for more versatile downstream robotics tasks, especially given the increasing prevalence of headset devices. This study expands EgoPAT3D, the sole dataset dedicated to egocentric 3D action target prediction. We augment both its size and diversity, enhancing its potential for generalization. Moreover, we substantially enhance the baseline algorithm by introducing a large pre-trained model and human prior knowledge. Remarkably, our novel algorithm can now achieve superior prediction outcomes using solely RGB images, eliminating the previous need for 3D point clouds and IMU input. Furthermore, we deploy our enhanced baseline algorithm on a real-world robotic platform to illustrate its practical utility in straightforward HRI tasks. The demonstrations showcase the real-world applicability of our advancements and may inspire more HRI use cases involving egocentric vision. All code and data are open-sourced and can be found on the project website.**

I. Introduction

To make robots more viable in our daily lives, intelligent and safe human-robot interaction (HRI) is essential. In the past, much work has been done to make robots' motion more legible and expressive to humans [1, 2] and make robots provide more intuitive visual feedback [3]. Equally important in HRI is the ability of robots to anticipate human actions and adapt their own accordingly. While many robotics research on human action anticipation adopt either a third-person view camera [4] or a camera mounted on the robot [5], addressing it using egocentric vision, namely visual input from the human's perspective, enjoys great potential and unique benefits for HRI due to the increasing prevalence of low-cost egocentric cameras (e.g., in mixed reality headsets or lifelogging devices) and the rich information they capture on both the environment and the human egomotion [6].

As a common task setup in HRI, object manipulation in a workspace shared by humans and robots is our focus in this work. Previous egocentric action anticipation research often studies 2D target region prediction [7, 8], trajectory forecasting [9], or the prediction on video of fixed length [10]. To fill the gap between those works and real-world manipulation HRI, we need *online predictions of 3D target coordinates* on variable-length videos. This results in our previous work on *EgoPAT3D* [11] which provides the first dataset and baseline method capable of such 3D forecasting.

However, *EgoPAT3D* has its limitations. First, there is a lack of a real-world HRI demonstration (unlike ours in Figure 1), which is critical to justify our research efforts into this 3D target coordinate prediction problem, and to inspire the future transfer of such methods to wearable robots and robotic prostheses. Second, the requirement of 3D inputs (point clouds or depth images) leads to a *bulky wearable* (a helmet mounted with a Kinect Azure sensor)

[1]New York University, Brooklyn, NY 11201, USA
[2]North Carolina State University, Raleigh, NC 27695, USA
*,†Equal contributions.
✉ Corresponding author (cfeng@nyu.edu). This work is supported by NSF Grant 2026479, and by NYU IT High-Performance Computing resources, services, and staff expertise.

that is disadvantageous in practice. Furthermore, we find *image-only methods more desirable* than using point cloud and IMU readings together. Point clouds are generally less accessible than simple RGB images. IMU data also increases sensing costs. Finally, we believe the diversity of the original dataset can be increased to boost its potential for better *real-world generalization*. Addressing those limitations leads to the following contributions of **EgoPAT3Dv2** in this work:

1) We propose a better algorithm exploiting our priors about human hand movement to achieve significant 3D prediction accuracy improvement while using only RGB image input without 3D point clouds and IMU readings.
2) We double the size of the original dataset by introducing more diverse background scenes and people of different skin complexions, hoping to make algorithms trained on this dataset more generalizable in the real world.
3) We deploy our algorithm to a real cobot and enable it to perform some human-robot interaction tasks, such as reaching the predicted action target with the shortest Cartesian path and proactively avoiding human action in a shared workspace.

II. Related Work

A. *Human-Robot Interaction with Egocentric Human Action Anticipation.*

Traditionally, human-robot interaction (HRI) research focuses on optimizing the robot's physical movement [1, 12, 13, 14] so that the robot's actions and execution timings are more legible and expressive to humans around them. Alternatively, researchers design various visual feedback from robots to facilitate safer and more intuitive human-robot interaction [3, 12, 15]. However, due to algorithmic and computing constraints, especially when it comes to processing high-dimensional visual data, the ability of the robot to classify or anticipate human actions before its motion generation is a relatively under-served area of research.

With the recent reviving interest in machine learning, we started to see more works focusing on integrating human action anticipation with HRI research. [4, 16] show that early prediction of human action using a Gaussian mixture model or inverse optimal control helps generate safer motion in a shared workspace. In [17], a Bayesian network is designed to consider human wait time in a collaborative bin manipulation task. [5] uses an encoder-decoder recurrent neural network to predict action sequences to maximize reward in a human-robot collaborative assembly setting. While these works show promising results, the perception is often done on an overhead camera or a camera mounted on the robot.

Recently, more HRI research started to look into human egocentric vision. In [18], Kim et al. design a soft wearable robot that can understand human actions through visual input from smart glasses. Planamente et al. in [19] propose a multi-modal neural network to predict diverse human actions under changing environments. [20] investigates the usability of a wheelchair controlled by human intention anticipation in egocentric vision. Marina-Miranda and Traver [6] use egocentric vision as a surrogate for head and eye gestures and accurately anticipate the wearer's action. [21, 22] classify human hand gestures in egocentric vision to give corresponding commands to robot systems. [10] is very similar to our workflow. However, because their work aims to automatically turn unlabelled human collaboration videos into training data for imitation learning on simple human-robot collaboration, the network only predicts a 2D region where the hand will be after a fixed time interval (i.e., 1 second). Meanwhile, our network incorporates a recurrent network to deal with the uncertain length of the human action sequence to produce 3D target coordinates. So far, most of the existing literature in HRI produces semantic-level classification or prediction with egocentric vision, while our work predicts a 3D coordinate of human action target, potentially allowing more precise human-robot interactions.

While egocentric vision can be closely woven with Augmented Reality (AR) [23], our work does not involve AR technologies. For a more thorough review of AR in human-robot interaction, we refer our readers to this survey [24].

B. *Egocentric Vision and Datasets.*

Egocentric vision is not only adopted for human action anticipation in the context of HRI. It has also proven to be useful in navigation [25], trajectory planning [26, 27], hand pose estimation and segmentation [28, 29] and scene understanding [30, 31]. Due to the unlimited potential, we refer our readers to the following survey papers for a more thorough investigation of egocentric vision's application on video summarization [32], hand analysis [33], future prediction [34] and futuristic use cases [35].

While there are several existing works on human action anticipation in egocentric vision, they either produce semantic level prediction with action labels [36], hand or walk trajectory [9, 37] or region [10, 37], or focusing on predicting a future after a fixed interval [10, 37]. Our method focuses on predicting a 3D action target coordinate given an online streaming video with variable length.

There also exists a large amount of egocentric datasets. EPIC-KITCHENS [38] and EGTEA Gaze+ [39] facilitate research [40] on action anticipation, usually on a semantic level. Datasets like EPIC-TENT [8] or 100DOH[7], on the other hand, provide annotation for region prediction, which is a 2D bounding box that will indicate the hand's intended location in the future. Ego4D [41] allows action anticipation, region prediction, and walk trajectory prediction at the same time. Unlike previous work, which usually focuses on action labels or 2D prediction, EgoPAT3D [11], along with our enhancement on it, provides 3D target coordinates on videos of varying lengths. For a more thorough review of the different modalities and annotations that different Egocentric datasets provide, we refer our readers to [11, 35].

III. Method

A. *Overview*

This section will describe the mathematical notation we will use to rigorously define the egocentric 3D action target prediction problem and the specific details of our improved algorithm *EgoPAT3Dv2*. Please refer to Sec. IV-B.1 for more discussions and comparisons on some design choices.

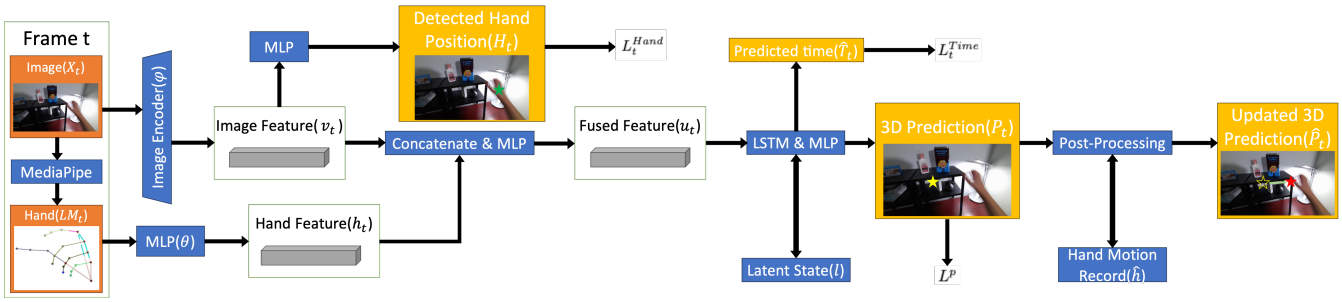

Fig. 2: **Algorithm Workflow.** Visual and hand features are extracted from RGB images and fused with an MLP. The fused feature is fed into an LSTM to produce the initial prediction, which is then adjusted by post-processing that considers our prior knowledge about manipulation. Note that LSTM and Post-Processing both rely on previous frames' information.

B. *Problem Formulation and Notation*

We briefly review the problem formulation from the original EgoPAT3D [11]. A video clip $\mathbf{C}$ of length T is said to be made of T consecutive frames $\{X_1, X_2, \cdots, X_T\}$. We want to predict the 3D target location $P_t^{gt} \in \mathbb{R}^3$ at each frame t. Namely, we want to predict the hand's target location (x, y, z) in the manual manipulation task.

From the machine learning perspective, we define our algorithm as a function f that takes in all the frames up until the current frame X_t and produces a regression output $\hat{P}_t$. We aim to find a f so that $\hat{P}_t = f(X_{1:t})$ will be as close to P_t^{gt} as possible, which is in the coordinate system of X_t.

Note that the target location is acquired at the very last frame of the video clip, but its value is transformed to the coordinate system of each video frame by using the transformation matrices computed from incremental ICP as explained in the original EgoPAT3D [11].

It is important to note that, in order to make the algorithm more useful in real-world applications, we restrict our algorithm to operating in online mode. When predicting y at frame X_t, the algorithm only has access to the frames from the start up until the current frame $\{X_1, X_2, \cdots, X_t\}$, but not all the frames in the future. Namely, the algorithm does not have access to future frames $\{X_{t+1}, X_{t+2}, \cdots, X_T\}$.

Finally, in this formulation, the frames $\{X_1, X_2, \cdots, X_T\}$ can be in various modalities. It can be point cloud, IMU readings, RGB pictures, etc or a combinations of the aforementioned modalities. We will be specific about the modality when we compare different algorithms.

C. *Improved Baseline Algorithm*

We improve the baseline method in EgoPAT3D [11] so it can perform online 3D target coordinate prediction with only RGB inputs. At each frame, the algorithm uses two backbone networks to extract **visual** and **hand** features. A feature fusion MLP network then fuses the features and passes them into a Long-Short Term Memory (LSTM) network [42] to generate 3D coordinate predictions for a continuous time sequence. The generated 3D target prediction for every frame is then post-processed before we acquire our final predictions. The workflow can be seen in Fig. 2.

*1) **RGB and Hand Feature Encoding**:* We employ ConvNeXt_Tiny [43] (denoted by (ψ)) pre-trained on ImageNet-1K to extract visual features $v_t = \psi(X_t)$ from each RGB frame. The weights of it are not frozen. The choice of ConvNeXt_Tiny is discussed in more detail in Sec. IV-B.1. Hand landmarks $LM_t^1, LM_t^2 \cdots LM_t^{21}$ are firstly extracted by the Hand API from Google's MediaPipe [44]. The underlying model is Google's proprietary technology. If no hand is detected, then all landmarks are set to 0. A multi-layer perceptron (MLP) denoted by (ϕ) is then used to encode hand landmarks to features $h_t = \phi(LM_t^{21stack})$. After the feature encodings, the two features were concatenated and fed into another MLP to obtain the fused feature $u_t = MLP(cat(v_t, h_t))$ for a single frame.

*2) **Online 3D Target Prediction**:* We use a 2-layer LSTM to process the fused feature. The steps to handle LSTM outputs are similar to the original EgoPAT3D baseline [11]. We divide a 3D space into grids of dimension $1024 \times 1024 \times 1024$ and aim to generate a confidence score for each grid. The choice of granularity at 1024 is empirically supported in [11]. We used three separate MLPs to process the output of the LSTM and obtain the confidence scores in three dimensions. For example, without loss of generality, for dimension x at frame t, let $g \in \mathcal{R}^{1024}$ denote all the grids in the x-dimension, where we normalize the coordinates of each grid to be in [-1, 1]. The score vector $s_t^x \in \mathcal{R}^{1024}$ is computed by $s_t^x = MLP_X(LSTM(u_t, l_{t-1}))$, where l_{t-1} is the learned hidden representation and l_0 is set to be 0. A binary mask $m_t^x \in \mathcal{R}^{\infty} \prime \in \triangle$ is used to remove the value for all the grids where the confidence is less than a threshold γ. Let $s_t^x[i]$, $m_t^x[i]$ denote the score and mask for the i-th grid, we have that:

$$m_t^x[i] = \begin{cases} 1, & i \in j | s_t^x[j] > \gamma \\ 0, & i \in j | s_t^x[j] \leq \gamma \end{cases}$$

The masked score is then calculated by $\hat{s}_t^x = m_t^x \odot s_t^x$ where $\odot$ denotes the element-wise dot product. Then, we can get the estimated target position value for dimension x at frame t as:

$$x_t \in \mathcal{R} = (\hat{s}_t^x)^T g$$

*3) **Post-Processing**:* We conduct post-processing for each result produced by the LSTM to incorporate human prior knowledge. The specific reasoning behind this design choice is further explained in Sec. IV-B.1. For each frame t, we choose the coordinate of the landmark that marks the end of the index finger to be the 2D hand position $\acute{h}_t$. The

predicted 3D target position P_t (in meter) was transformed into 2D position $\acute{P}_t$ in pixel values with the help of camera intrinsic parameter K and image resolution (4K in our case). We ignore the depth information in this transformation. We calculate the hand position offset between each frame by $\hat{h}_t = ||\acute{h}_t - \acute{h}_{t-1}||_2$ and keep track of the max historical hand position offset $\bar{h}_t = max(\hat{h}_t)$ for $i < t$. The final 2D position is calculated as $\bar{P}_t = \acute{P}_t * \frac{\hat{h}_t}{\bar{h}_t} + \acute{h}_t * (1 - \frac{\hat{h}_t}{\bar{h}_t})$. The 2D result is then transformed back to a 3D position $\hat{P}_t$ with the pre-transformed depth, again with camera intrinsic parameter K and image resolution, to serve as the final prediction.

D. *Improved Loss Function*

Our loss is a modification from the truncated weighted regression loss (TWRLoss) proposed in EgoPAT3D [11]. The TWRLoss L^p directly calculates the loss between ground truths and predictions. Additionally, We incorporate two new losses **Hand Position Loss** L^{Hand} and **Time Loss** L^{Time}. They do not directly supervise the difference between predictions and ground truths but instead aim to incorporate human prior knowledge about manipulation. The overall loss of our training paradigm can be written as $L = \sum_{t=1}^{T} w_t(L_t^p + \delta(L_t^{Hand} + L_t^{Time}))$ where w_t is a linear weight from 2 to 1 with respect to time t, and δ is a hyperparameter that acts as a weight for the two new losses. More details about the design choices are discussed in Sec. IV-B.1.

*1) **Hand Position Loss**:* We want the visual feature extractor to focus more on the hand, so we introduce a task for the feature extractor and an additional MLP $H_t = MLP(v_t)$ to predict the hand position for each frame. Only frames with a hand detected will be included in the hand position loss:

$$L_t^{Hand} = \begin{cases} (H_t - \acute{h}_t)^2, & \text{hand in frame } t \\ 0, & \text{hand not in frame } t \end{cases}$$

*2) **Time Loss**:* We want the LSTM to be able to differentiate between early stages and late stages without introducing hardcoded positional encoding that relies on knowing the length of the clip beforehand, so we introduce another task for LSTM and an additional MLP $\hat{T}_t = MLP(LSTM(u_t, l_{t-1}))$ to predict where the current frame is relative to the whole clip $\bar{T}_t = \frac{t}{T}$. The time loss L_t^{Time} is calculated by $L_t^{Time} = (\hat{T} - \bar{T}_t)^2$.

IV. Experiment

We conducted two separate experiments.

- **Experiment 1:** Algorithms are trained and tested on the EgoPAT3D dataset with the same protocol as in the EgoPAT3D paper [11].
- **Experiment 2:** Algorithms are trained and tested on our enhanced dataset which is to be explained in Sec. V.

A. *Experiment Setup*

*1) **Dataset**:* **For Experiment 1**, we follow the exact same dataset preparation process in the EgoPAT3D paper[11]. The models are trained and validated with data from five scenes. The test set is divided into **seen** and **unseen**. The seen part of the test set is made of unused data from the five training scenes. The unseen part is from 6 unseen scenes. While we cap the training clips at 25 frames, the validation and test sets have no frame number limit. Note that we made a few minor corrections to the ground truth label in the original dataset. After the corrections, the baseline from [11] performs better than in the original paper.

For Experiment 2, we added nine new scenes to the training, validation, and seen test set and two new scenes to the unseen test set. Again, the clips from the validation and test sets have no length limit, while the clips in the training set are capped at 25 frames.

*2) **Implementation Details**:* The baselines are trained and tested with the same hyperparameters open-sourced by the authors of EgoPAT3D [11]. However, we modified the distributed training used in the scripts from PyTorch's DataParallel to Distributed DataParallel. The baseline model performs better than the original one because of different behaviors [45] in Distributed DataParallel, such as gradient gathering. All our *EgoPAT3Dv2* models are trained with the Adam optimizer [46]. The learning rate is 10^{-4}, and the weight decay rate is 10^{-5}, with no hyperparameter search conducted. We deployed our training to four RTX 8000 GPUs, with a batch size of 8 on each, 32 in total. MediaPipe's hand detection and tracking confidences are all set to 0.5. The weight δ for the two new losses is set to 0.1.

All models are trained with the same set of three different random seeds and with PyTorch's `deterministic=True` and `benchmarking=False` to ensure maximal reproducibility. The results are the average of the three trials.

B. *Quantitative Results and Discussions*

Every test clip is evenly divided by frames into ten stages during evaluation. For clips that cannot be evenly divided by ten, more frames are allocated toward the early stages. As a result, errors in the early stages will have a slightly bigger impact on the average error.

*1) **Experiment 1**:* Table I contains the results and associated ablation studies for **Experiment 1**. The details of the ablation studies are explained below:

1) **Random** has a 3D point randomly generated within the range of [max, min] of the training set's labels
2) **EgoPAT3D_VF** and **EgoPAT3D** are two baseline algorithms from the original EgoPAT3D paper [11]. **EgoPAT3D_VF** uses only point cloud as input, while **EgoPAT3D** additionally takes in the IMU data. Note that, as mentioned in Sec. IV-A.1 and Sec. IV-A.2, our baseline implementations actually perform better than those in the original paper.
3) **ConvNeXt_Only** and **ResNet50_Only** simply replace the PointConv [47] in the **EgoPAT3D_VF** with a pre-trained ConvNeXt_Tiny in [43] and pre-trained ResNet50 in [48], respectively. It does not have any of the modifications mentioned in Sec. III
4) **Post+Hand** does not have the new losses discussed in Sec. III-D but has post-processing and hand features.

TABLE I: **Prediction error (cm) of different models on the EgoPAT3D seen scenes.** The lower, the better. Post denotes post-processing. Hand denotes hand position input into the model. L^{Hand} denotes the Hand Position Loss and L^{Time} denotes the Time Loss. Red, green, and blue fonts denote the top three performance.

Description	Method	Modality	Overall	Early prediction					Late prediction				
				10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Non-Learning	Random	N/A	49.44	50.83	47.43	47.74	49.01	47.99	50.06	47.30	48.67	50.01	48.45
Baseline	EgoPAT3D_VF	Point Cloud	18.75	23.45	21.73	20.11	18.71	17.52	16.65	16.15	16.02	15.97	16.15
	EgoPAT3D	Point Cloud, IMU	16.70	20.76	19.15	17.85	16.84	15.72	15.01	14.54	14.29	14.23	14.35
Variants	ConvNeXt_Only	RGB	15.90	20.63	19.65	18.84	17.69	16.02	14.34	12.67	11.13	10.11	9.79
	ResNet50_Only		16.39	20.69	19.82	18.92	17.73	16.09	14.71	13.33	12.30	11.78	11.67
	Post+Hand		15.59	20.81	19.78	19.02	17.96	16.28	14.34	12.23	10.43	8.20	7.06
	Post+L^{Hand}+L^{Time}		15.66	21.72	20.58	19.68	18.33	16.22	13.79	11.46	9.59	7.50	6.59
	Hand+L^{Hand}+L^{Time}		15.41	20.38	19.09	18.36	17.38	15.71	13.96	12.11	10.42	9.32	8.82
	Post+Hand+L^{Hand}		15.63	20.56	19.81	19.13	18.27	16.49	14.43	12.24	11.38	8.10	6.94
	Post+Hand+L^{Time}		15.60	20.90	19.65	18.86	17.89	16.35	14.39	12.30	10.51	8.22	7.08
	ResNet101_Replace		15.54	20.98	19.95	19.33	18.22	16.28	13.96	11.70	9.90	7.67	7.05
	EgoPAT3Dv2_Transformer		16.09	20.73	19.81	19.12	18.37	16.91	15.12	13.11	11.49	9.16	8.14
	EgoPAT3Dv2_Full	Point Cloud, IMU, RGB	16.19	21.22	19.50	18.43	17.53	16.25	14.81	13.34	12.26	10.80	10.09
EgoPAT3Dv2	EgoPAT3Dv2	RGB	14.97	20.36	19.07	18.34	17.43	15.66	13.68	11.57	9.67	7.50	6.60

5) **Post+L^{Hand}+L^{Time}** does not have a hand feature extractor, so there is no MLP to fuse the hand features with the vision feature. However, it has the post-processing and the new losses.
6) **Hand+L^{Hand}+L^{Time}** does not have the post-processing mentioned in Sec. III-C.3 but has hand features and the two losses.
7) **Post+Hand+L^{Hand}** and **Post+Hand+L^{Time}** either Hand Position Loss or Time Loss mentioned in Sec. III-D while keeping other components the same as the *EgoPAT3Dv2* algorithm
8) **ResNet101_Replace** replaces the ConvNeXt_Tiny in *EgoPAT3Dv2* with a ResNet101.
9) **EgoPAT3Dv2_Transformer** relaces the LSTM in *EgoPAT3Dv2* with a transformer. In our low-data regime, the transformer does not perform as well, which has been observed in similar works [49, 50].
10) **EgoPAT3Dv2_Full** adds point cloud data and IMU data. They are processed and fused as in **EgoPAT3D**. This experiment shows that simply providing more information by increasing data modality does not translate to better accuracy.

Discussion on RGB Backbones. As we can see with **ConvNeXt_Only** and **ResNet50_Only**, simply replacing the PointConv with a CNN network feature extractor can substantially improve the performance, especially late prediction. We hypothesize that 2D RGB input is adequate for simple monocular depth estimation due to the depth cues that come with it and because our manipulation task has limited depth variety. At the same time, feature extractors pre-trained on the massive ImageNet [51] can be more capable than three PointConv layers trained from scratch. Along with **ResNet101_Replace**, we show that ConvNeXt_Tiny (28,589,128 #params) performs significantly better than ResNet50 (25,557,032 #params) and larger ResNet101 (44,549,160 #params) in our algorithm. The small number of parameters guarantees strong inference speed for real-time tasks. More discussion on performance can be found in Sec.VI

Discussion on Hand Feature. One issue not discussed in the original EgoPAT3D baseline [11] is that the visual feature extraction is only at the global hierarchy, as PointConv produces a global feature for each point cloud input. Nothing is explicitly done about the hand manipulating the object, even though prior knowledge tells us that the action target will be highly correlated to the hand's location and movement through time. In our algorithm, hand features are fused into the pipeline, and experiment **Post+Loss** shows that early-stage prediction performance suffers substantially without the hand features.

Discussion on Post-Processing. One salient issue we observe in the baseline of EgoPAT3D [11] is that the network's accuracy stagnates and degrades when the hand moves towards the action target during the later stages. We experimented using PointConv as a feature extractor to perform hand detection on the very last frame and achieved an error of less than 0.2 cm. This experiment shows that the LSTM network does not understand when the manipulation task is approaching the end. To tackle this issue, we adopt a post-processing procedure that considers hand movement. We observe that hand movement tends to stabilize toward the end of a manipulation task so one can place an item stably. Therefore, in our post-processing, we put more weight on the hand detector when the stabilization of hand locations is detected, and we trust the LSTM's "raw" output more when it is in the early stages and the hand is still moving fast. We can see in **Hand+L^{Hand}+L^{Time}** that late-stage performance suffers significantly when there is no post-processing.

Discussion on New Losses. The **Hand Location Loss** tries to tackle the same problem as the Hand Feature input. Visual features should not be simple global features but focus on the hand. The **Time Loss** is trying to solve a similar problem as post-processing. We want the LSTM network to pay more attention to the progression of a manipulation task without introducing a hardcoded positional encoding that will be at odds with our online prediction setting. By observing **Post+Hand**, **Post+Hand+L^{Hand}** and **Post+Hand+L^{Time}**, we see that the combination of the two losses can create a substantial performance boost compared to using only one.

*2) **Experiment 2**:* Table II contains the results and associated ablation studies for **Experiment 2**. The details of the ablation studies are explained below.

- **EgoPAT3D** is the original baseline of EgoPAT3D evaluated on the unseen part of the EgoPAT3D test set.

TABLE II: **Prediction error (cm) of *EgoPAT3Dv2*.** The models are trained separately on the original EgoPAT3D and the enhanced EgoPAT3Dv2 dataset. They are then tested on the seen and unseen test sets of the original EgoPAT3D and enhanced EgoPAT3Dv2 dataset. Red, green, and blue fonts denote the top three performance.

Description	Ovr.	Early prediction					Late prediction				
		10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
EgoPAT3D	18.8	22.8	20.4	19.3	18.6	18.0	17.6	17.3	17.00	16.9	17.0
T1_D1_Seen	15.0	20.4	19.0	18.3	17.4	15.7	13.7	11.6	9.7	7.5	6.6
T1_D1_Unseen	16.3	21.2	19.6	18.6	17.6	16.3	15.1	13.7	12.0	10.9	10.4
T1_D2_Seen	21.6	28.0	26.6	25.4	23.8	21.5	19.2	17.0	15.5	14.3	14.00
T1_D2_Unseen	21.5	24.0	22.5	21.5	20.3	18.9	17.6	15.8	14.2	13.2	12.8
T2_D1_Seen	15.2	20.0	19.2	18.8	18.0	16.2	14.1	11.8	9.9	7.8	6.9
T2_D1_Unseen	16.6	21.6	20.6	19.8	18.7	17.1	15.4	13.2	11.0	10.00	9.8
T2_D2_Seen	15.2	22.0	20.7	19.5	17.9	15.3	12.7	10.2	8.5	7.4	7.1
T2_D2_Unseen	17.6	22.7	21.8	20.9	19.7	18.00	16.0	13.8	12.00	11.2	11.0

- **T1** means the model is trained with the original training set of EgoPAT3D.
- **T2** means the model is trained on our enhanced dataset. The enhanced dataset contains the original EgoPAT3D training set and our additions.
- **D1** means the model is evaluated on the test set of the original EgoPAT3D.
- **D2** means the model is evaluated on the test set of our enhanced dataset, which contains the original EgoPAT3D's test set and our additions.
- **Seen** means that the model is evaluated on the seen part of the test set, as described in Sec. IV-A.1
- **Unseen** means that the model is evaluated on the unseen part of the test set, as described in Sec. IV-A.1

As we can see in **T1_D1_Unseen**, *EgoPAT3Dv2* also performs significantly better on the unseen test set of the original EgoPAT3D dataset compared to the original baseline.

Through all the models with **T2** labels, we can see when trained with additional data, our best model maintains a competitive performance when evaluated on the original EgoPAT3D dataset, on both seen and unseen scenes. At the same time, our model with additional training data can achieve better generalization when facing a more extensive and diverse test set.

V. Dataset Enhancement

A. *Size*

We doubled the size of the EgoPAT3D dataset. The total number of available clips increases from 4129 to 9579. The seen test set of the new data has clip lengths ranging from 10 to 115 frames, with a mean of 24. The unseen test set runs from 8 frames to 42 frames, with an average of 20. The distribution of clip length can be seen in Fig. 3

B. *Diversity*

We have nine additional individuals as human subjects, compared to 2 in the first version. The volunteers are from different countries with different skin complexions.

We also added 12 new scenes that are substantially different from those in the original EgoPAT3D dataset, with a diverse array of objects for humans to manipulate.

C. *Data Collection and Annotation*

The data collection process follows the same procedure in the original EgoPAT3D paper [11] with a Microsoft Azure Kinect camera mounted on a helmet.

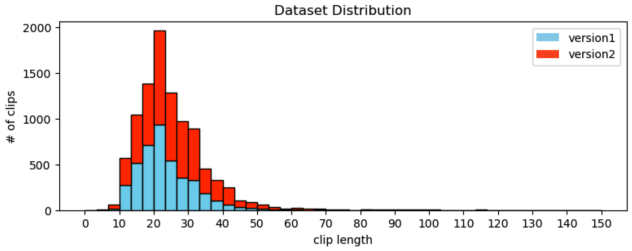

Fig. 3: Distribution of the clip length in EgoPAT3D and EgoPAT3Dv2 dataset

Instead of the semi-automatic data annotation process described in [11], we opted for a more reliable manual annotation method. While we still use MediaPipe [44] for hand detection, we have humans to verify the results and manually annotate if MediaPipe fails to perform the detection.

VI. Real-World Demonstration

We deployed our algorithm on a UR10e robot for simple demonstrations that reflect a real-world scene where a human and a robot share a common workspace. In setting #1, if the predicted 3D action target enters a ball of radius of r around the robot's end-effector, the robot reactively moves away to avoid collision with the human. In setting #2, the robot tries to reach the target position with the shortest Cartesian path.

A hand-eye calibration using MoveIt [52] is performed as an eye-to-hand setup. The motion planning is done through MoveIt and OMPL [53].

In a PC with i7-13700F and RTX 4070 Ti, the algorithm can achieve a stable 17 FPS thanks to the small parameters count of ConvNeXt_Tiny.

The demo videos can be found on our project webpage. A frame-by-frame excerpt can be found in Fig. 1

VII. Conclusion and Limitation

In this work, we greatly improve the 3D human action target prediction performance on the EgoPAT3D [11] dataset. Our algorithm incorporates prior knowledge about manipulation into the learning process and also utilizes state-of-the-art vision pre-training to reduce the required number of sensing modalities. We also enhance the original EgoPAT3D dataset and improve its diversity. Finally, we deploy the algorithm on a real robot to demonstrate the potential of 3D human action prediction for human-robot interaction (HRI).

However, we realize the *limitations in our real-world demonstration, algorithm, and dataset.* The complexity and variety of our real-world demonstration is limited. Complex tasks with both hands visible should be attempted to investigate the robustness of the algorithm. The variety of robotic platforms in our HRI demonstration can also be expanded (e.g., wearable robots, robotic prostheses) since we want a generic egocentric vision algorithm.

The dataset and algorithm can also be further improved. Our current dataset and algorithm assume that there will only be one single hand throughout most of the videos, limiting its real-world use cases. The improvement of early-stage performance is also limited compared to late stages.

References

[1] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2013, pp. 301–308. 1, 2

[2] M. Avdic, N. Marquardt, Y. Rogers, and J. Vermeulen, "Machine body language: Expressing a smart speaker's activity with intelligible physical motion," in *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, ser. DIS '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1403–1418. 1

[3] E. Cha and M. Matarić, "Using nonverbal signals to request help during human-robot collaboration," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5070–5076. 1, 2

[4] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 11 2013. 1, 2

[5] P. Schydlo, M. Rakovic, L. Jamone, and J. Santos-Victor, "Anticipation in human-robot cooperation: A recurrent neural network approach for multiple action sequences prediction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5909–5914. 1, 2

[6] J. Marina-Miranda and V. J. Traver, "Head and eye egocentric gesture recognition for human-robot interaction using eyewear cameras," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7067–7074, 2022. 1, 2

[7] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, "Understanding human hands in contact at internet scale," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2020, pp. 9866–9875. 1, 2

[8] Y. Jang, B. Sullivan, C. Ludwig, I. D. Gilchrist, D. Damen, and W. Mayol-Cuevas, "Epic-tent: An egocentric video dataset for camping tent assembly," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 4461–4469. 1, 2

[9] W. Bao, L. Chen, L. Zeng, Z. Li, Y. Xu, J. Yuan, and Y. Kong, "Uncertainty-aware state space transformer for egocentric 3d hand trajectory forecasting," in *International Conference on Computer Vision (ICCV)*, October 2023. 1, 2

[10] J. Lee and M. S. Ryoo, "Learning robot activities from first-person human videos using convolutional future regression," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. 1, 2

[11] Y. Li, Z. Cao, A. Liang, B. Liang, L. Chen, H. Zhao, and C. Feng, "Egocentric prediction of action target in 3d," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022, pp. 20 971–20 980. 1, 2, 3, 4, 5, 6

[12] D. Szafir, B. Mutlu, and T. Fong, "Communication of intent in assistive free flyers," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, ser. HRI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 358–365. 2

[13] E. Mainprice, Jim. Akin Sisbot, T. Simeon, and R. Alami, "Planning safe and legible hand-over motions for human-robot interaction," in *IARP/IEEE-RAS/EURON Workshop on Technical Challenges for Dependable Robots in Human Environments (IARP)*, 2010. 2

[14] A. Zhou, D. Hadfield-Menell, A. Nagabandi, and A. D. Dragan, "Expressive robot motion timing," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, ser. HRI '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 22–31. 2

[15] S. Song and S. Yamada, "Bioluminescence-inspired human-robot interaction: Designing expressive lights that affect human's willingness to interact with a robot," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, T. Kanda, S. Sabanovic, G. Hoffman, and A. Tapus, Eds. ACM, 2018, pp. 224–232. 2

[16] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative re-planning for predicting human reaching motions in shared workspaces," *IEEE Transactions on Robotics*, vol. 32, 06 2016. 2

[17] K. P. Hawkins, N. Vo, S. Bansal, and A. F. Bobick, "Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 499–506. 2

[18] D. Kim, B. B. Kang, K. B. Kim, H. Choi, J. Ha, K.-J. Cho, and S. Jo, "Eyes are faster than hands: A soft wearable robot learns user intention from the egocentric view," *Science Robotics*, vol. 4, no. 26, p. eaav2949, 2019. 2

[19] M. Planamente, G. Goletto, G. Trivigno, G. Averta, and B. Caputo, "Toward human-robot cooperation: Unsupervised domain adaptation for egocentric action recognition," in *Human-Friendly Robotics 2022*, P. Borja, C. Della Santina, L. Peternel, and E. Torta, Eds. Cham: Springer International Publishing, 2023, pp. 218–232. 2

[20] M. Kutbi, X. Du, Y. Chang, B. Sun, N. Agadakos, H. Li, G. Hua, and P. Mordohai, "Usability studies of an egocentric vision-based robotic wheelchair," *J. Hum.-Robot Interact.*, vol. 10, no. 1, jul 2020. 2

[21] H. Song, W. Feng, N. Guan, X. Huang, and Z. Luo, "Towards robust ego-centric hand gesture analysis for robot control," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*, 2016, pp. 661–666. 2

[22] P. Ji, A. Song, P. Xiong, P. Yi, X. Xu, and H. Li, "Egocentric-vision based hand posture control system for reconnaissance robots," *Journal of Intelligent & Robotic Systems*, vol. 87, no. 3, pp. 583–599, Sep 2017. 2

[23] H. Liang, J. Yuan, D. Thalmann, and N. Magnenat-Thalmann, "Ar in hand: Egocentric palm pose tracking and gesture recognition for augmented reality applications," *Proceedings of the 23rd ACM International Conference on Multimedia (ACMMM)*, 2015. 2

[24] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, "Augmented reality and robotics: A survey and taxonomy for ar-enhanced human-robot interaction and robotic interfaces," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI '22. New York, NY, USA: Association for Computing Machinery, 2022. 2

[25] E. Ohn-Bar, K. Kitani, and C. Asakawa, "Personalized dynamics models for adaptive assistive navigation systems," in *Conference on Robot Learning*, 2018. 2

[26] H. S. Park, J.-J. Hwang, Y. Niu, and J. Shi, "Egocentric future localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4697–4705. 2

[27] G. Bertasius, A. Chan, and J. Shi, "Egocentric basketball motion planning from a single first-person image," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5889–5898, 2018. 2

[28] T. Ohkawa, T. Yagi, A. Hashimoto, Y. Ushiku, and Y. Sato, "Foreground-aware stylization and consensus pseudo-labeling for domain adaptation of first-person hand segmentation," *IEEE Access*, vol. 9, pp. 94 644–94 655, 2021. 2

[29] T. Ohkawa, K. He, F. Sener, T. Hodan, L. Tran, and C. Keskin, "AssemblyHands: towards egocentric activity understanding via 3d hand pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 12 999–13 008. 2

[30] T. Nagarajan, S. K. Ramakrishnan, R. Desai, J. Hillis, and K. Grauman, "Egoenv: Human-centric environment representations from egocentric video," 2022. 2

[31] S. Tan, T. Nagarajan, and K. Grauman, "Egodistill: Egocentric head motion distillation for efficient video understanding," *Advances in Neural Information Processing Systems*, vol. 36, 2024. 2

[32] A. G. del Molino, C. Tan, J.-H. Lim, and A.-H. Tan, "Summarization of egocentric videos: A comprehensive survey," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 1, pp. 65–76, 2017. 2

[33] A. Bandini and J. Zariffa, "Analysis of the hands in egocentric vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, p. 6846–6866, apr 2020. 2

[34] I. Rodin, A. Furnari, D. Mavroeidis, and G. M. Farinella, "Predicting the future from first person (egocentric) vision: A survey," *Computer Vision and Image Understanding*, vol. 211, p. 103252, 2021. 2

[35] C. Plizzari, G. Goletto, A. Furnari, S. Bansal, F. Ragusa, G. M. Farinella, D. Damen, and T. Tommasi, "An outlook into the future of egocentric vision," 2023. 2

[36] M. Liu, S. Tang, Y. Li, and J. M. Rehg, "Forecasting human-object interaction: Joint prediction of motor attention and actions in first person video," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 704–721. 2

[37] S. Liu, S. Tripathi, S. Majumdar, and X. Wang, "Joint hand motion and interaction hotspots prediction from egocentric videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 3282–3292. 2

[38] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray, "The epic-kitchens dataset: Collection, challenges and baselines," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 43, no. 11, pp. 4125–4141, 2021. 2

[39] Y. Li, M. Liu, and J. M. Rehg, "In the eye of the beholder: Gaze and actions in first person video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 06, pp. 6731–6747, jun 2023. 2

[40] A. Furnari and G. Farinella, "Rolling-unrolling lstms for action anticipation from first-person video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4021–4036, nov 2021. 2

[41] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. Farinella, C. Fuegen, B. Ghanem, V. Ithapu, C. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. Park, J. Rehg, Y. Sato, J. Shi, M. Shou, A. Torralba, L. Torresani, M. Yan, and J. Malik, "Ego4d: Around the world in 3,000 hours of egocentric video," in *Proceedings - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. United States: IEEE Computer Society, 2022, pp. 18 973–18 990. 2

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997. 3

[43] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 966–11 976. 3, 4

[44] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for perceiving and processing reality," in *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 6

[45] S. Li and J. Zhu. Getting started with distributed data parallel. Accessed: 10 September 2023. [Online]. Available: https://pytorch.org/tutorials/intermediate/ddp_tutorial.html 4

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2015. 4

[47] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9613–9622. 4

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 4

[49] G. Melis, T. Kočiský, and P. Blunsom, "Mogrifier lstm," in *International Conference on Learning Representations (ICLR)*, 2020. 5

[50] P. Izsak, S. Guskin, and M. Wasserblat, "Training compact models for low resource entity tagging using pre-trained language models," in *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, 2019, pp. 44–47. 5

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255. 5

[52] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! task constructor for task-level motion planning," in *International Conference on Robotics and Automation (ICRA)*, 05 2019, pp. 190–196. 6

[53] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, https://ompl.kavrakilab.org. 6