Reshaping Robot Trajectories Using Natural Language Commands:
A Study of Multi-Modal Data Alignment Using Transformers

Arthur Bucker¹, Luis Figueredo¹, Sami Haddadin¹, Ashish Kapoor², Shuang Ma², Rogerio Bonatti²
¹Technische Universität München, ²Microsoft

Abstract—Natural language is the most intuitive medium for us to interact with other people when expressing commands and instructions. However, using language is seldom an easy task when humans need to express their intent towards robots, since most of the current language interfaces require rigid templates with a static set of action targets and commands. In this work, we provide a flexible language-based interface for human-robot collaboration, which allows a user to reshape existing trajectories for an autonomous agent. We take advantage of recent advancements in the field of large language models (BERT and CLIP) to encode the user command, and then combine these features with trajectory information using multi-modal attention transformers. We train the model using imitation learning over a dataset containing robot trajectories modified by language commands, and treat the trajectory generation process as a sequence prediction problem, analogously to how language generation architectures operate. We evaluate the system in multiple simulated trajectory scenarios, and show a significant performance increase of our model over baseline approaches.

Fig. 1: Our system allows a user to send natural language commands to reshape robot trajectories relative to objects in the environment.

I. INTRODUCTION

Large language models such as BERT [1], GPT3 [2] and Megatron-Turing [3] have radically improved the quality of machine-generated text, along with our ability to solve natural language processing tasks. Beyond just language, we see a shift in machine learning architectures in multiple domains, as the dominant design paradigm changes from designing task-specific models towards the use of large foundational pre-trained models [4]. Several of these large models already combine multiple data modalities such as text, images, video, depth, and even the temporal dimension [5]–[8]. The use of foundational models is appealing because they are trained on broad datasets over a wide variety of downstream tasks, and therefore provide general skills which can be used directly or with minimal fine-tuning to new applications [4].

The field of robotics traditionally uses extremely task and hardware-specific models, which have to be re-trained and even re-designed if there are minor changes in robot dynamics, environment, and operational objectives. This inflexible machine learning approach is ripe for innovation with the use of foundational models [4], in particular when it comes to task specification in ambiguous scenarios (What should I do?) and task learning that can generalize across multiple environments (How should I do it?). Recent works have just started to explore the use of pre-existing foundational models from language and vision towards robotics [9]–[13], and also the development of robotics-specific foundational models [8, 14].

Our work aims to leverage information contained in existing vision-language foundational models to fill the gap in existing tools for human-robot interaction. Even though natural language is the richest form of communication between humans, modeling human-robot interactions using language is challenging because we often require vast amounts of data [11, 15]–[17], or classically, force the user to operate within a rigid set of instructions [18, 19]. To tackle these challenges, our framework makes use of two key ideas: first, we employ large pre-trained language models to provide rich user intent representations, and second, we align geometrical trajectory data with natural language jointly with the use of a multi-modal attention mechanism.

As seen in Fig 1, we focus our study on robotics applications where a user needs to reshape an existing robot trajectory according to specific operational constraints. This class of use cases often arises in human-robot interaction when autonomous agents that employ traditional motion planners (e.g. A∗, RRT® [20] or MPC [21] concerned solely about obstacle avoidance and dynamics) need to be corrected by a user according to additional semantic or safety objectives. For instance, our goal is to enable a factory worker to quickly reconfigure a robot arm trajectory further away from fragile objects, or to allow a user to intuitively tell a robot barista to get a little closer to the cup in order to pour a wine bottle.

Then main contribution of this paper is to propose a novel system with a multimodal attention mechanism for semantic
trajectory generation. It can effectively align natural language features with geometrical cues jointly, and perform the goal of trajectory reshaping with a predictive trajectory decoder. The use of large pre-trained language models to obtain word embeddings allows us to offer a flexible and intuitive user interface, while lowering the requirements on the number of training examples. We validate the proposed models in a series of experiments in simulation and in real-world tests with a robotic arm. Finally, we show that the proposed trajectory reshaping method is highly preferred by users in comparison with baseline methods, both in terms of ease of use and performance.

II. RELATED WORK

Robots and language: As robots become more prevalent in environments outside of laboratories and dedicated manufacturing spaces, it is important to offer non-expert users simple ways of communication with machines. Natural language is an ideal candidate, given that interfaces such as mouse-and-keyboard, touchscreens and programming languages are powerful, but require extensive training for proper usage [22]. Multiple facets of language-based human-robot interaction have been studied in literature, such as instruction understanding [23, 24], motion plan generation [9, 12, 16, 25], human–robot cooperation [26], semantic belief propagation [18, 19], and visual language navigation [11, 27]. Most of the recent works in the field have shifted from representing language in terms of classical grammatical structure towards data-driven techniques, due higher flexibility in knowledge representations [22].

Multi-modal robotics representations: Representation learning is a rapidly growing field. The existing visual-language representation approaches primarily rely on BERT-style [1] training objectives to model the cross-modal alignments. Common downstream tasks consist of vision-answering, grounding, retrieval and captioning etc. [28]–[31]. Learning representations for robotics tasks poses additional challenges, as perception data is conditioned on the motion policy and model dynamics [4]. Visual-language navigation of embodied agents is a well-established field with clear benchmarks and simulators [32, 33], and multiple works explore the alignment of vision and language data by combining pre-trained models with fine-tuning [34]–[36]. To better model the visual-language alignment, [37] also proposed a co-grounding attention mechanism. In the manipulation domain we also find the work of [10], which uses CLIP [5] embeddings to combine semantic and spatial information. In this paper we also need to align the semantic information with geometry understanding in order to reshape trajectories according to the desired task specifications.

Transformers in robotics: Transformers were originally introduced in the language processing domain [38], but quickly proved to be useful in modeling long-range data dependencies other domains. Within robotics we see the first transformers architectures being used for trajectory forecasting [39] and reinforcement learning [40, 41]. Our work is the first to present a multi-modal transformer model to align visual-language understanding with robot actions for trajectory reshaping.

III. APPROACH

A. Problem Definition

Our overall goal is to provide a flexible language-based interface for human-robot interaction within the context of trajectory reshaping. One typical application for our systems is that of a user re-configuring a robotic arm trajectory that, although already avoids collisions, gets uncomfortably close to a particular fragile obstacles in the environment. We design the trajectory generation system with a sequential waypoint prediction decoder, which takes into account multiple data modalities from geometry and language into a transformer network. The modified trajectory should be as close as possible to the original one throughout its length and respect the original start and goal constraints, while obeying the user’s semantic intent. Fig.2 depicts the expected model behavior in a typical use-case scenario.

Let \( \xi_s : [0, 1] \rightarrow \mathbb{R}^2 \) be the original robot trajectory, which is composed by a collection of \( N \) waypoints \( \xi_o = \{(x_1, y_1), \ldots, (x_N, y_N)\} \). We assume that the original trajectory is a reasonable path from the start to the goal positions (i.e. avoids collisions) and can be pre-calculated using any desired motion planning algorithm, but falls short of the full task specifications. Let \( L_{in} \) be the user’s natural language input sent to correct the original trajectory, such as \( L_{in} = “\text{Stay away from the wine glass”}“ \). Let \( O = \{O_1, \ldots, O_M\} \) be a collection of \( M \) objects in the environment, each with a corresponding position \( P(O_i) \in \mathbb{R}^2 \) and semantic label, such as \( L(O_i) = “\text{glass”}“ \). Our goal is to learn a function \( f \) that maps the original trajectory, user command and obstacles towards a modified trajectory \( \xi_{mod} \), which obeys the user’s semantic objectives:

\[
\xi_{mod} = f(\xi_o, L_{in}, O)
\]

B. Proposed Network Architecture

We approximate the function \( f \) from Eq. 1 by a parametrized model \( f_\theta \), learned directly from data. This mapping is non-trivial since it combines data from multiple distinct modalities, and also ambiguous since there exist multiple solutions that satisfy the user’s objective. Fig. 3 displays our model architecture, which consists of distinct
Fig. 3: We proposed a trajectory reshaping system with multimodal attention mechanism. The whole pipeline consists of pretrained CLIP and BERT and trainable transformer encoder and decoders. Given the input of initial trajectory (green), objects with their position and semantic labels (blue), and natural language commands (yellow), the model is trained to generate new trajectories in a predictive manner.

feature encoders \( (q_{\text{BERT}}, q_{\text{CLIP}}, T_{\text{enc}}) \), whose outputs are fed into a multi-modal decoder transformer \( T_{\text{dec}} \) for the sequential prediction of the output trajectory \( \xi_{\text{mod}} \). In more detail:

**Language encoding:** We use a pre-trained language model encoder, BERT [1], to produce semantic features \( q_{\text{BERT}}(z^{in}|L_{in}) \) from the user’s input. The use of a large language model creates more flexibility in the natural language input, allowing the use of synonyms (shown in Section IV-A) and less training data, given that the encoder has already been trained with a massive text corpus. In addition, we use the pre-trained text encoder from CLIP [5] to extract latent embeddings from both the user’s text and the \( M \) object semantic labels \( q_{\text{CLIP}}(z|L) \), which enable us to compute a cosine similarity vector between the embeddings, and use this information to identify user’s target object.

The main idea behind using the 2 different language models simultaneously is that we aim to benefit from the specific types of corpus that each model was trained on. While CLIP was specifically trained on a corpus that correlates language and visual context, BERT on the other hand, learned from large generic language inputs, and because of that should be able to better capture the semantic intent of the NL input. Furthermore, in Section V we discuss how the CLIP model can potentially be used directly as input to the transformer-based model. We use imitation learning to train the model, using the Huber loss [42] between the predicted and ground-truth waypoint locations.

**Geometry encoding:** The original trajectory \( \xi_{o} \) is composed of low-dimensional tokens \( (x_{i}, y_{i}) \in \mathbb{R}^2 \). In order to extract more meaningful information from each waypoint, we apply a linear transform with learnable weights \( W_{\text{geo}} \) that projects each waypoint into a higher dimensional features space, following the example of [39]. The poses \( P(O_{i}) \) of each object are also processed with the same linear transform. We then concatenate both feature vectors and use a transformer-based feature encoder \( T_{\text{enc}} \). The use of a transformer is preferred for sequences because its architecture can attend to multiple time steps simultaneously, as opposed to recurrent networks, which suffer from vanishing gradient issues [39].

**Multi-modal transformer decoder:** Feature embeddings from both language and geometry are combined as input to a multi-modal transformer decoder block \( T_{\text{dec}} \). We generate the reshaped trajectory \( \xi_{\text{mod}} \) sequentially, analogously to common transformer-based approaches in natural language [2, 38]. Section IV-A compares sequential generation with other approaches such regressing to the entire trajectory at once. We also verify that a fully-connected architecture cannot achieve the same performance as the transformer-based model. We use imitation learning to train the model, using the Huber loss [42] between the predicted and ground-truth waypoint locations.

C. Synthetic Data Generation

Data collection in the robotics domain is challenging, specially when we require alignment between multiple modalities such as language and trajectories. Different strategies range from large-scale online user studies for language labeling [43] all the way to procedural trajectory-language pairs generation using heuristics [16]. Our work relies on a key hypothesis: the use of large-scale language models for feature encoding \( (q_{\text{BERT}}, q_{\text{CLIP}}) \) relieves some of the pressure in obtaining a diverse set of vocabulary labels, given that the text encoders are able to find semantic synonyms for different sentence structures. Therefore, we generated a small but meaningful set of examples with semantically-driven trajectory modifications. We employed an \( A^* \) planner to generate reasonable initial trajectories \( \xi_{o} \) in randomized environments with different object configurations, and based on a set of pre-determined semantic combinations, we used the CHOMP motion planner [44] to compute \( \xi_{\text{mod}} \) by modifying weights of different cost functions. Our vocabulary involved
different directions relative an object (closer or further away from -, to the left/right/front/back of -), intensity changes (a bit/little, much, very), and a thousand object labels sampled from the ImageNet vocabulary. We generated a total of 10,000 trajectory labels. Fig. 4 displays examples of original and reshaped trajectories.

D. Implementation Details
Our language encoders consist of pretrained BERT and CLIP models, with frozen weights. The output sentence embedding \( z^m \in \mathbb{R}^{1 \times 768} \) is concatenated with the similarity vector of size \( 1 \times M \) generated from the CLIP embeddings, and scaled to a feature vector of size \( 1 \times 256 \). For the geometrical inputs we concatenate an array with \( M \) object poses with a sequence of 100 waypoints from \( \xi_o \), and upscale the features to a vector of size \( 1 \times 256 \). \( T_{enc} \) is a 2-block transformer encoder, and \( T_{dec} \) is a 4-block transformer. Each transformer has 3 hidden layers with 512 fully-connected neurons in each. We empirically found it helpful to remove Layer and Batch Normalization [45] steps from both transformers in order to better retain the geometry information. The model is trained in a two-step fashion: first we augment the training data by randomly rotating and re-scaling the planning problem; second we fix the start and goal locations to always lay in the lower left and upper right corner of the map respectively, and fine-tune the network. We use the AdamW [46] optimizer with an initial learning rate \( \gamma = 1 \times 10^{-4} \) and a linear warm-up period of 15 epochs. We use a Nvidia Tesla V100 GPU with batch size of 64, and train the model for 500 epochs.

IV. EXPERIMENTS
We execute experiments in both simulated and real environments to evaluate our trajectory reshaping model. Our goals are the following: 1) Investigate if the combination of pre-trained large-language models together with multimodal transformers can create efficient and generalizable human-robot interfaces; 2) Quantitatively and qualitatively compare our approach with other classes of human-robot interfaces from the user’s perspective; 3) Demonstrate our natural language method is applicable to real robots.

A. Simulation Experiments
We investigate multiple facets of the model’s capabilities and performance in a series of simulated experiments:

1. **How language influences trajectory behavior**: Given a fixed environment configuration, we evaluate the model’s ability to follow distinct natural language commands. Fig. 5 displays how a gradient in direction and intensity of language commands correctly modifies the resulting decoded path.

2. **How the model behaves in different planning problems**: To understand the effect of different object configurations and language commands onto the reshaped trajectory, we display in Fig. 4 a set of randomly sampled planning problems from our validation set. We can see from the image that in most cases the reshaped trajectory correctly models the desired user intent, and falls close to the ground-truth reshaped trajectory.

3. **Vocabulary and object diversity**: One key hypothesis assumed true when designing our model architecture was...
that the use of pre-trained large language models as feature encoders would make our pipeline amenable to a diverse set of natural language inputs, despite the relatively small amount of training examples. To test this hypothesis we compute results using with novel user commands, with vocabulary not present in our training language labels. Fig. 6 shows that our model still executes the expected behavior, being able to find the correct semantic meaning despite the new words.

**Baseline architectures:** We perform ablation studies comparing different variations of our proposed multi-modal transformer against baseline architectures. We employ a fully-connected network (FCN) for regression (5 hidden layers, with 512 neurons), that takes as input a single 1D vector composed of the concatenated trajectory, object positions and embedded language features (BERT and similarity vector from CLIP) and outputs a 1D vector with coordinates of all waypoints. The best fully-connected architecture and training procedure was found through a grid-search over the number of hidden layers, neurons, batch size and initial learning rate (64 models in total). Table I summarizes the results, and shows that the best architecture is composed by the multi-modal transformer without layer and batch norm, and using a sequential predictive decoder. The naive predictor referenced in the table shows the loss in the case where we simply copy the original trajectory as the model output, and serves as a baseline loss value. Similarly to previous studies in trajectory forecasting [39], we find that transformers drastically improve model performance, likely due to their unique ability to extract features combining features from distant waypoints.

![Fig. 6: Trajectory reshaping results using novel vocabulary (not seen in the training data) as the user input. Our model is able to correctly execute the desired semantic commands due to the large capacity of the BERT and CLIP text encoders.](image)

| Model         | Features | Test. Loss |
|---------------|----------|------------|
| Naive predictor | 2.4M     | 0.0051     |
| FCN w/ regression | 10.6M   | 0.0025     |
| **Ours**      |          | **0.002**  |

**TABLE I: Baseline architecture comparisons**

**B. User Evaluation in Real Robot Experiments**

We also evaluate our system with real-world experiments, and compare our method with the use of multiple human-robot interfaces. We use a 7-DOF PANDA Arm robot equipped with a claw gripper, and execute tasks on a 1×1m tabletop workspace. A standard desktop computer with an off-the-shelf GPU connected to the robot computes the original trajectories, executes our model, and runs low-level controls for the arm. We operate the model using 2D planar projections of the original robot trajectories, and respect the original waypoint heights when executing the reshaped motion plans. We use object positions given by markers, but we discuss the use of vision-based localization in Section V.

The goal of the study is to have the user control a robotic bartender. A traditional motion planning algorithm calculates an initial trajectory to transport a bottle of wine towards a cocktail shaker and pour the liquid inside (we leave the problem of learning how to make fancy drinks for future iterations of this work). This original trajectory comes dangerously close to toppling over a tower of crystal glasses, and the user needs to interact with the robot to make the end-effector trajectory safer. As seen in Fig. 7 we test 4 different human-robot interfaces: natural language (NL-oours), kinesthetic teaching (KT), trajectory drawing (Draw), and programming obstacle avoidance weights via a keyboard and mouse (Prog). A top-down view of the experimental platform is seen in Fig. 8. All user interactions followed a study protocol approved by the Technical University of Munich’s ethics committee, and we conducted a total of 10 interviews.

![Fig. 7: Human-robot interfaces tested in the user study: a) natural language (NL), b) kinesthetic teaching (KT), c) programming via keyboard, and d) trajectory drawing.](image)

**Quantitative user evaluation:** We measured statistics on the number of iterations, success rate and total time taken for users to modify trajectories using the different interfaces. From Table II we see that the programming interface takes by far the longest for users to master, and requires numerous iterations. In the meanwhile, NL is the fastest option. We see a large number of failures for kinesthetic teaching and drawing because user inputs are often times kinematically
The natural language method proved to be the most robust, and we found no failure cases during in the study.

### TABLE II: Statistics collected over the user study experiment

| Interface | Avg. iterations | Success rate (%) | Avg. Time (s) |
|-----------|-----------------|------------------|--------------|
| NL        | 1.33            | 100              | 81           |
| KT        | 1.78            | 56.24            | 139          |
| Draw      | 1.89            | 64.7             | 120          |
| Prog      | 4.00            | 91.66            | 284          |

#### Qualitative user evaluation

After the experiments we asked users to rate the trajectories produced by different interfaces according to different criteria in a psychometric questionnaire:

1) How satisfied were you with the final robot motion?
2) How easy was refining the robot motion?
3) How safe was the final robot trajectory?
4) How natural was the human-machine interaction?
5) How predictable was the trajectory for you?

Table III summarizes the responses. We can see that most methods present a similar user satisfaction level except for programming, which was rated lower likely due to the difficulty of interaction. NL was rated as the easiest and most natural method, but at the same time was deemed less predictable than KT and drawing because with these two methods users have direct control over the final trajectory.

### TABLE III: User ratings collected in the user study

| Interface | Satisfied | Ease of use | Safety | Natural | Predictable |
|-----------|-----------|-------------|--------|---------|-------------|
| NL        | 90        | 92          | 92     | 98      | 72          |
| KT        | 90        | 88          | 88     | 78      | 96          |
| Draw      | 88        | 74          | 100    | 80      | 88          |
| Prog      | 62        | 58          | 82     | 62      | 48          |

#### Final experimental ratings

Overall, we find from the experiments that our proposed natural language model stands as a strong alternative to traditional human-robot interfaces. Kinesthetic teaching is often not a viable solution for real-world trajectory reshaping, depending on the robot’s size, form factor and actuator types. Trajectory drawing is also not a robust solution, as human-defined trajectories often extrapolate acceleration and kinematic constraints. By combining semantic and geometrical information, our method provides a natural and effective interface for trajectory reshaping.

### V. Conclusion and Discussion

In this paper, we present a novel system with multimodal attention mechanism for semantic trajectory reshaping. Given flexible natural language commands and an initial trajectory, it can effectively reshape the trajectory consistent with the language commands. The multimodal attention architecture provides a way for us to jointly align natural language features and the geometrical cues.

We verify through our experiments that by leveraging large pretrained language models like BERT and CLIP, our proposed system creates a flexible and intuitive user interface. Given that these foundational models train on massive corpus of data, we are able to train our robotics system with a smaller dataset, and let the language model find similarities between sentences if novel vocabulary is used.

By evaluating our methods on both simulation and real-world application scenarios, we show that our model outperforms baseline trajectory reshaping approaches in terms of loss values and quality of results. From the user’s perspective, we also perform a user study and show that users significantly prefer our natural language interface in opposition to other methods such as kinesthetic teaching or programming interfaces. Our method is faster to use, and results in a higher success rate.

Even though our study does not address the visual modality, we are confident that our current architecture would also be able to align this additional data through the CLIP encoder. For future iterations of this work we’re interested in using images of the objects as opposed to directly inputting the semantic labels to the model. In addition, in the future we are interested in designing methods that also consider causal relations among objects from the natural language commands in order for the robot to execute more complex task-driven behaviors.

### Acknowledgments

SH has a potential conflict of interest as shareholder of Franka Emika GmbH. This work was funded by the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant 5140951).

### References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[2] 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, vol. 33, pp. 1877–1901, 2020.

[3] S. Smith, M. Patwary, B. Norick, P. LeGresley, S. Rajbhandari, J. Casper, Z. Liu, S. Prabhumoye, G. Zerveas, V. Korthikanti et al., “Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model,” arXiv preprint arXiv:2201.11990, 2022.
