

# 大規模言語モデルの活用による効率的なロボット制御の学習

高城 頌太<sup>1</sup> 谷口 尚平<sup>1</sup> 中野 聡大<sup>1</sup>  
 上田 亮<sup>1</sup> 松尾 豊<sup>1</sup> 谷中 瞳<sup>1</sup>  
<sup>1</sup> 東京大学

{takashiro,taniguchi,nakano.akihiro,matsuo}@weblab.t.u-tokyo.ac.jp  
 {ryoryoueda,hyanaka}@is.s.u-tokyo.ac.jp

## 概要

GPT-3 や ChatGPT に代表される大規模言語モデルは、様々なタスクで汎用的に高い性能を示し、自然言語処理のみならず様々な領域での応用が広がっている。本稿では、大規模言語モデルをロボット制御の模倣学習に活用し、その学習効率やサンプル効率の向上への寄与について検証を行う。実験では、RLBench というベンチマークデータセットを用いて、提案手法の有効性を検証する。具体的には、単腕ロボットがモノをゴミ箱に捨てるというタスクをデモンストレーションデータから模倣学習する際に、大規模言語モデルへのプロンプティングを活用して、タスクを解く手順を説明するテキストを生成し、それを補助情報として用いることで、学習効率の向上を図る。

## 1 はじめに

GPT-3 [1] に代表される大規模言語モデルの登場により、自然言語処理の様々なタスク（機械翻訳、質問応答、コードの自動生成・補完 [2] など）を高精度に解くことが可能になりつつある [3]。特に、昨年秋に公開された ChatGPT [4] はチャット形式のインターフェースと高精度な文章生成性能をもつことから、世界的に注目を集めたことは記憶に新しい。また、こうした大規模言語モデルの活用は、自然言語処理のみに留まらず、様々な領域まで広がり始めている。

近年は、ロボット制御の文脈でも、大規模言語モデルを活用する研究が行われ始めている [5, 6]。ロボット制御において、大規模言語モデルの活用が注目されている背景として、強化学習や模倣学習に代表される機械学習ベースのロボット制御のサンプル効率の悪さが挙げられる [7]。インターネット上から大量の学習データを収集することが容易な画像や

自然言語などの領域と異なり、ロボット制御はデータを収集するコストが高く、効率よく学習する方法論の確立が重要である。高い汎用性をもつ大規模言語モデルの活用は、このようなロボット学習の課題を解決する糸口になりうると考えられる。例えば、床に落ちているモノを掴んでゴミ箱に捨てるというタスクを単腕ロボットに学習させることを考えた場合、ロボットは (1) モノにロボットハンドを近づけて、(2) グリッパーを操作してモノを把持し、(3) ハンドをゴミ箱の上に移動させて (4) グリッパーを開いてモノをゴミ箱の中に落とす、という多段階の操作をデータから学習する必要がある。このような段階的な操作の必要性は、人間であれば常識的に持ち合わせている知識であるが、事前知識を持たないロボットはこれを一から学習することが要求され、大量のデータと学習時間が必要となってしまう。一方で、大規模言語モデルは、ウェブ上の大量のテキストデータから学習されており、人間が持ち合わせている常識的な知識に関する質問にも非常に精度よく回答できることが知られている [1, 8]。特に、大規模言語モデルに対して、タスクの説明や少数の正解例を与えることで、新しいタスクへ適応させるテクニックは**プロンプティング**と呼ばれ、その有効性が翻訳や文書要約などの様々なタスクで検証されている [3]。そこで、前述のようなロボットにとって複雑なタスクを解くための手順（以下、**処理指示文**と呼ぶ）を、大規模言語モデルへのプロンプティングを通して獲得し、それを補助情報としてロボットの制御モデルに与えてやることで、学習を促進することが期待できる。

本稿では、このような大規模言語モデルへのプロンプティングを活用したロボット学習の手法を提案し、その有効性を検証する。具体的には、前述のようなモノをゴミ箱に捨てるタスクをデモンストレーションデータからの模倣学習によって学習する際

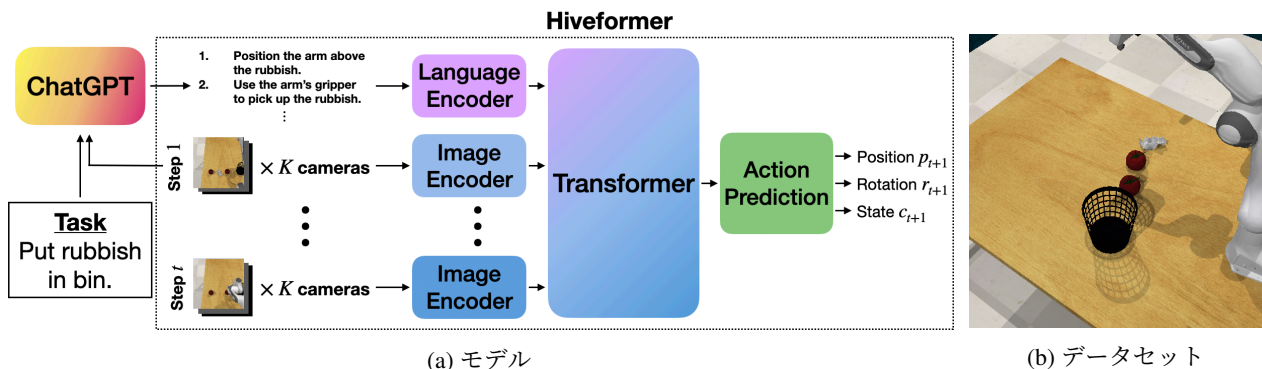


図 1: (a) 提案手法の概念図。提案手法では、HiveFormer におけるテキスト入力として、タスクの説明文の代わりに、ChatGPT へのプロンプティングの結果得られるタスクを解く手順の説明文を用いる。(b) RL Bench データセットの PutRubbishInBin タスクのエピソード例。いずれのエピソードも 1 つのゴミ箱、1 つのゴミ、2 つの無関係なオブジェクトから構成される。配置されるオブジェクトの形状はエピソード間で共通であり、初期の位置関係のみランダムで変化する問題設定である。

に、タスクを解く手順を ChatGPT へのプロンプティングによって獲得し、それをロボットの行動を決定する方策モデルへ補助情報として入力する手法を提案する。実験では、RLBench [9] というシミュレータ環境でのロボット制御のベンチマークを利用し、学習過程におけるタスクの成功率の推移や、データ効率を従来手法と比較する形で、提案手法の有効性を検証する。

## 2 関連研究

深層学習を用いたロボット制御モデルを学習させる上で、十分な学習データを収集するコストは高い。そのため、より効率の良い学習方法に関して様々な研究が行われている [7]。とりわけ、複数ステップを要する構成的なタスクにおいて効率的な学習手法の開発は重要である。構成的なタスクの例として、例えば、家具を組み立てるタスク [10] や積み木を積み上げるタスク [11, 12] がある。このようなタスクは、制御モデルを学習する上で特に難しいことが知られており、汎用的な方法論は未だに確立されていない。

一方で、深層学習において言語モデルは、GPT モデルの進展 [13, 14, 1, 4] から分かるように、データセットのサイズとモデルのパラメータをスケールさせることで、より汎用性のあるモデルへと発展してきた。近年では、GPT-3 [1] をはじめとした大規模言語モデルについて、プロンプティングを行うことで新しいタスクに少ないデータのみで適応できることが検証されている [3]。検証を通じて、大規模言語モデルは、人間が持ち合わせている一般常識を獲

得し、順序立てた手続的な推論も行うことが可能であることが報告されている [6, 15]。

模倣学習や強化学習などを用いた学習ベースのロボット制御においても、画像情報に加えてこうした大規模言語モデルによる観測やタスクに関する言語情報を入力として取り入れることで、性能向上を図る手法が注目を集めている [5, 6, 16, 17]。GPT-3 のような大規模言語モデルがロボット学習に寄与できる要素としては、学習効率や汎化性能の向上が挙げられる。人間の一般常識に相当するような知識を持ち合わせる大規模言語モデルからその知識を引き出すことで、新しいタスクを素早く学習し、適応できることが期待できる。しかし、現状では、ロボット学習における大規模言語モデルの活用は、物体の説明文生成のような比較的単純な補助情報の付与のみに留まっており、制御の方策自体を直接ガイドするような枠組みは確立されていない。本稿では、そのような枠組みの確立の第一歩として、デモンストレーションデータをもとに制御方策を学習する枠組みである模倣学習 [18, 19] において、大規模言語モデルを活用して制御モデルの方策を補助する枠組みに関して検討する。

## 3 手法

本稿では、Transformer [20] をベースとした模倣学習手法である HiveFormer [21] をベースラインとして採用する。HiveFormer では、各時刻における複数視点からの画像とタスクの説明文を、それぞれ U-Net (セグメンテーションを行うのに一般的に使われる畳み込みニューラルネットワーク) [22] と

CLIP（ウェブ上の大量のデータを用いて画像とテキストが対応するように学習させたモデル）[23]の言語エンコーダに入力して、特徴抽出を行い、それらをTransformerに入力して、その出力を再び別のU-Netに入力することで、最終的な行動（グripperの位置・回転・状態の制御値）を出力する。長期的なステップ数にわたる行動の予測が必要となるが、keyframe discovery [24]を適用することで、取る行動が大きく変化する時刻を特定し、よりマクロなステップでの予測が可能となっている。

提案手法では、図 1a のように、HiveFormer に与えるタスクの説明文の代わりに、大規模言語モデルへのプロンプティングを通して得られた、処理指示文（タスクを解く手順についての説明文）をテキストの入力として与えることで、モデルの学習を促進することを目的とする。本稿では、大規模言語モデルとして、ChatGPT [4]を用いて検証を行う。図 2 は、ChatGPT へのプロンプティングとその結果の例である。このように、段階的な操作が必要なタスクにおいて、ChatGPT へ適切なプロンプトを与えることで、具体的な手順を説明したテキストを引き出し、制御モデルの学習効率を高めることが期待できる。

## 4 実験

本実験では、大規模言語モデルから生成された処理指示文（タスクの解き方）によって、どの程度ロボット制御の学習が効率化するかについて実験を行う。具体的には、ロボット制御タスクにおいて、ChatGPT からの処理指示文があり/なしの場合を比較してどれほど学習が効率化されたかを比較する。

### 4.1 シミュレーション環境とデータセット

ロボット制御のシミュレーション環境としては、図 1b に示す RLBench データセットの PutRubbishInBin タスクを用いた。RLBench データセットは7次元の自由度を持つ manipulator を用いた100種類のロボット制御タスクを含んだベンチマークである。7次元の自由度は、manipulator のグripperの位置を表す3次元と、その回転を表す4次元ベクトルを指す。PutRubbishInBin タスクはロボット制御タスクの一つであり、複数の物体からゴミを特定しゴミ箱に入れるタスクである。

本手法では模倣学習を用いるため、エキスパートによるデモンストレーションデータが必要となる。そのため、100 エピソード分のエキスパートデータ

What are the 4 steps for the robotic arm to place the rubbish in the bin? The instructions must be short, informative, and concise. Please give me the coordinates of where to move it. Coordinates are given by (x, y, z).

State: bin (0.10, 0.21, 0.85), rubbish (0.23, -0.30, 0.77), tomato1 (0.22, -0.15, 0.78), tomato2 (0.16, -0.02, 0.78)

Instruction:

1. Move the robotic arm to the position of the rubbish: (0.23, -0.30, 0.77)
2. Grip the rubbish with the gripper on the robotic arm.
3. Move the robotic arm to the position of the bin: (0.10, 0.21, 0.85)
4. Release the grip on the rubbish.

図 2: ChatGPT へ与えるプロンプト（黒字）と生成された処理指示文（青字）の例。

を作成し、模倣学習に用いた。

### 4.2 大規模言語モデルに与えるプロンプト

ChatGPT は、これまでの GPT-3 などのモデルとは異なり、対話型のインタフェースを介して大規模言語モデルと「会話」することが可能である。会話のスタイルとして、質問応答やロールプレイなどの方法も可能だが、今回は、文章補完の形でプロンプティングを行った。

プロンプト文としては様々な文章が考えられるが、Hiveformer の入力として用いるにあたって、(1) 処理指示に関する情報のみ生成されること、(2) 生成されるステップ数が固定であること、(3) 各処理指示の文章が冗長でないこと、(4) 実際の現在の状況に対応する処理指示が生成されること、の4つが求められると考えた。例えば、付録の図 4 に示すように、プロンプト文の情報が不十分だと生成される処理指示文は曖昧かつ冗長になることがある。したがって、これらの要件を満たすようにプロンプト文を指定することで、Hiveformer の入力に適した処理指示文を生成した。特に、ステップ数はHiveformerのマクロステップ数と一致させることで画像情報と言語情報がより対応づけられる形で学習が進むように図った。また、エピソードの初期状態をテキストとして表すことで、エピソードに対応した手順の生成を促した。実際に使用したプロンプトと ChatGPT を通して得られた結果の例を図 2 に示す。

### 4.3 実験設定

本実験では、ベースラインとして HiveFormer [21]を用いた。HiveFormer はカメラからの RGB 画像と、距離センサからの Depth 情報（RGBD）、そしてタス



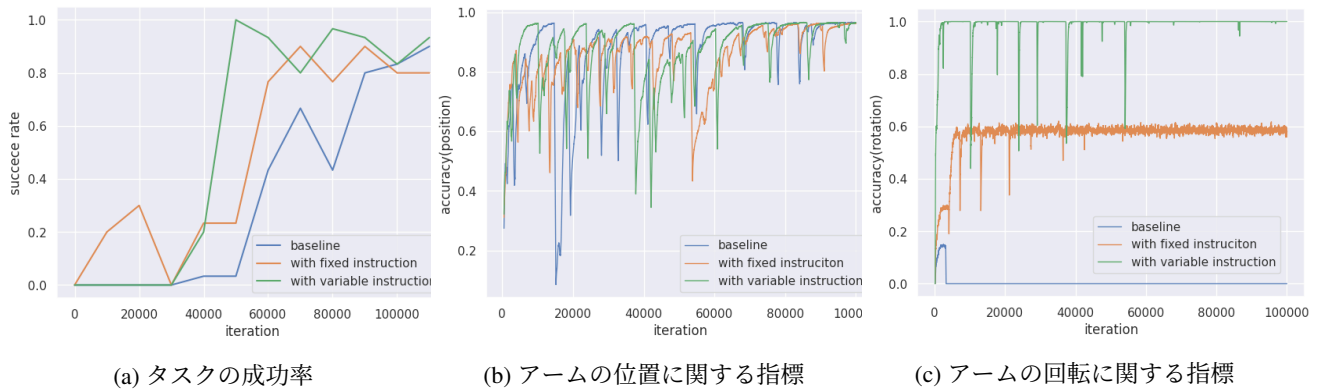


図 3: 学習中の評価指標とタスクの成功率に関する図. 成功率は 30 エピソード中の成功割合. 評価指標は予測値と真の値の誤差がある一定の閾値以下の時に 1, それ以外は 0 として各イテレーションで平均を取っている.

ク自体の説明文を入力として模倣学習を行うマルチタスク学習のフレームワークであるが, 本実験では 1 つのタスクのみを解くシングルタスク学習として用いる. また, Hiveformer ではタスク自体の説明文 (“Put rubbish in bin.”) を与えているが, 本実験では, ChatGPT を用いて生成された処理指示文を与える.

よって, HiveFormer の入力として,

1. RGBD とタスクの説明文を与えた場合 (baseline)
2. RGBD とエピソード毎で固定の指示文を与えた場合 (with fixed instruction)
3. RGBD とエピソード毎で物体の位置によって変化する指示文を与えた場合 (with variable instruction)

の 3 種類を用いて ChatGPT の出力する処理指示文の有効性を検証した. 各実験設定における入力文章は, 付録の図 5 に示すように, baseline, fixed instruction, variable instruction の順により具体的な指示になっている.

また, 手法の評価方法としては (1) タスクの成功率の推移, (2) 学習の収束の速さの 2 点を評価した.

## 4.4 結果

図 3a に学習中のタスクの成功率を示す. 横軸が学習のイテレーション回数, 縦軸が成功率を示している. 図より, タスクの指示文が具体的になるにつれて, 少ない学習回数で高い成功率を達成していることがわかる.

また, 図 3b, 3c に, 各手法における評価指標の値を示している. 図より, ロボットアームの位置に関する指標は各手法で差は見られなかったが, ロボットアームの回転に関する指標では, タスクの指示文

が具体的になるにつれて高い値を示しており, より学習効率が向上しているといえる.

## 4.5 考察

本実験から, 提案手法のアプローチは, 段階的な手順が必要なタスクにおいて, 学習効率の向上に寄与することが確認できた. 一方で, エピソードによらず固定のタスク説明文を加えた場合との比較においては, 改善の幅は比較的小さい結果となった. この結果から, 制御モデルの学習においては, 物体の位置などのような各エピソードに固有の情報よりも, タスク内で共通する一般的な操作に関する情報が, より学習効率を向上させる上での補助情報として有効に機能しているのではないかと考えられる. また, 提案手法が, 特にグリッパーの回転角の操作に関して, 大きく性能が改善した点については, ChatGPT によるタスクの処理指示文によって, グリッパーの位置の操作に関する補助情報が得られたために, 処理指示文にない回転角の制御についての学習が促進されたのではないかと考察できる.

## 5 おわりに

本稿では, 深層学習を用いたロボット制御において, ChatGPT へのプロンプティングによって得た言語情報を補助情報として与えた場合の学習効率への影響について実験を行った. RLBench データセットの PutRubbishInBin タスクを用いた実験結果から, 言語モデルによるタスクの手順説明は, 制御モデルの学習効率の向上に寄与することが確認できた. 今後は, 別のタスクにおける実験を加えることで, 提案手法の汎用性に関する検証をさらに進めていく.

## 謝辞

本研究は JST さきがけ JPMJPR21C8 の助成を受けたものです。

## 参考文献

- [1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Nee-lakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. **Advances in neural information processing systems**, Vol. 33, pp. 1877–1901, 2020.
- [2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. **arXiv preprint arXiv:2107.03374**, 2021.
- [3] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **arXiv preprint arXiv:2107.13586**, 2021.
- [4] OpenAI. Chatgpt: Optimizing language models for dialogue, 2022. <https://openai.com/blog/chatgpt>.
- [5] Dhruv Shah, Blazej Osinski, Brian Ichter, and Sergey Levine. Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. 2022.
- [6] Allen Z Ren, Bharat Govil, Tsung-Yen Yang, Karthik R Narasimhan, and Anirudha Majumdar. Leveraging language for accelerated learning of tool manipulation. In **Workshop on Language and Robotics at CoRL 2022**.
- [7] Oliver Kroemer, Scott Niekum, and George Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. **The Journal of Machine Learning Research**, Vol. 22, No. 1, pp. 1395–1476, 2021.
- [8] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. **arXiv preprint arXiv:2206.07682**, 2022.
- [9] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment. **IEEE Robotics and Automation Letters**, 2020.
- [10] Youngwoon Lee, Edward S Hu, and Joseph J Lim. Ikea furniture assembly environment for long-horizon complex manipulation tasks. In **2021 IEEE International Conference on Robotics and Automation (ICRA)**, pp. 6343–6349. IEEE, 2021.
- [11] Jungtaek Kim, Hyunsoo Chung, Jinhwi Lee, Minsu Cho, and Jaesik Park. Combinatorial 3d shape generation via sequential assembly. **arXiv preprint arXiv:2004.07414**, 2020.
- [12] Seyed Kamyar Seyed Ghasemipour, Satoshi Kataoka, Byron David, Daniel Freeman, Shixiang Shane Gu, and Igor Mordatch. Blocks assemble! learning to assemble with large-scale structured reinforcement learning. In **International Conference on Machine Learning**, pp. 7435–7469. PMLR, 2022.
- [13] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.
- [15] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. **arXiv preprint arXiv:2205.11916**, 2022.
- [16] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. **arXiv preprint arXiv:2204.01691**, 2022.
- [17] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. **arXiv preprint arXiv:2210.01911**, 2022.
- [18] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In **IJCAI**, 2018.
- [19] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In **Conference on robot learning**, pp. 1113–1132. PMLR, 2020.
- [20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. **Advances in neural information processing systems**, Vol. 30, , 2017.
- [21] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In **CoRL (oral)**, 2022.
- [22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In **International Conference on Medical image computing and computer-assisted intervention**, pp. 234–241. Springer, 2015.
- [23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In **International Conference on Machine Learning**, pp. 8748–8763. PMLR, 2021.
- [24] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. **IEEE Robotics and Automation Letters**, Vol. 7, No. 2, pp. 1612–1619, 2022.

## A ChatGPT を用いたプロンプティング例

ChatGPT を用いたプロンプティングを行うにあたって、プロンプト文に十分な情報が含まれていない場合、生成される手順が冗長になる傾向があることが確認された。例えば、図 4 では、「ロボットアームを用いて 4 ステップでゴミをゴミ箱に入れる方法は何か」と聞いており、実際に生成された手順は図 2 と類似している。しかし、各ステップにおける文に不要な情報（“This can be done by...”）が含まれてしまっている。

What are the 4 steps to put the rubbish in the bin using an robotic arm?

Instructions:

1. Position the robotic arm in front of the bin: The first step is to move the robotic arm so that it is in front of the bin and in the correct position to pick up the rubbish. This can be done by programming the arm to move to a specific location or by manually controlling it with a joystick or other input device.
2. Grab the rubbish: The next step is to use the robotic arm's gripper or other end effector to pick up the rubbish. This can be done by closing the gripper around the rubbish or by using a vacuum system to suck it up.
3. Move the arm to the bin: Once the rubbish has been picked up, the robotic arm needs to be moved over to the bin. This can be done by programming the arm to move to a specific location or by manually controlling it with a joystick or other input device.
4. Deposit the rubbish in the bin: The final step is to deposit the rubbish in the bin. This can be done by releasing the gripper or vacuum, allowing the rubbish to fall into the bin. It can also be done by using the robotic arm to place the rubbish carefully in the bin. Once the rubbish has been placed in the bin, the robotic arm can return to its starting position and wait for the next command.

図 4: 不十分なプロンプトを用いたときの、ChatGPT へ与えるプロンプト（黒字）と生成された手順（青字）の例。

## B 各手法における実際の入力文章例の比較

本実験では 3 つの実験設定を用いて有効性を検証した。各手法において、実際に Hiveformer に入力した文章例を図 5 に示す。図より、baseline, fixed instruction, variable instruction の順に具体的な指示になっていることが読み取れる。

[baseline]

pick up the rubbish and leave it in the trash can.

[with fixed instruction]

1. Position the arm above the rubbish.
2. Use the arm's gripper to pick up the rubbish.
3. Move the arm to the bin.
4. Release the rubbish into the bin by opening the gripper.

[with variable instruction]

1. Move the robotic arm to the position of the rubbish: (0.23, -0.30, 0.77)
2. Grip the rubbish with the gripper on the robotic arm.
3. Move the robotic arm to the position of the bin: (0.10, 0.21, 0.85)
4. Release the grip on the rubbish.

図 5: 各実験設定 (baseline, with fixed instruction, with variable instruction) における入力文章例。