物体中心オプティカルフローによる言語条件付き Open-Loop 軌道生成

○神原元就 †，妹尾幸樹 †, 鶏内朋也 ‡, 王亜楠 ‡, 杉浦孔明 †
† 慶應義塾大学　　‡KDDI 総合研究所

本研究では，RGB 画像と言語指示から対象物の 2D フロー及びエンドエフェクタの軌道を生成する手法を提案する．主要な貢献は言語との整合性を高める Semantic Consistency Loss 及び，視覚と言語特徴を統合する Prompt-Conditioned Cross-Modal Adapter の導入である．結果より，提案手法は既存手法を上回る品質のフローの生成及び物体操作成功率を達成した．

1. はじめに

ロボットによる物体操作は，サービスロボットにとって基盤となる重要な能力であり，安全性，効率性，および幅広い動作が可能であることが求められる．言語指示に基づき軌道を生成し，物体操作を行うことができる Vision-Language Action モデル (VLA) の性能は，視覚言語モデルの発展に伴い向上している．一方で，多くの VLA において，訓練データに含まれる動作については高い成功率を達成するものの，ゼロショット転移性能は依然として限られている．この課題に対処するために，我々は，物体中心のフローに基づき動作生成を行うアプローチに焦点を当てる．このアプローチでは，操作前の RGB 画像と自然言語指示が与えられたとき，ポリシーは (i) 操作対象の物体中心の 2D フローと，(ii) 対応するエンドエフェクタ軌道を順に生成する．このアプローチは，自然言語指示の示す動きと embodiement を分離したモデル化を可能とする．これにより，フロー生成モジュールはロボットデータ以外にも多様な人間による物体操作動画を用いて訓練が可能となるため，より未知の物体や自然言語指示に頑健なモデルが構築できる．さらに，ロボットの教示データはフローからエンドエフェクタの軌道への変換を行うためだけに用いればよいため，訓練において必要なロボットの教示データを削減することが可能である．

これらの利点がある一方で，既存のフローベースのアプローチには 2 つの大きな課題がある．一つ目に，多くの手法が大規模なデータセットで訓練されておらず，狭い範囲の動作への適用に限られている．次に，closed-loop 軌道生成が一般的である点が挙げられる．すなわち，各ステップで，画像及び言語指示に基づいて次時刻のフロー及びウェイポイントを予測する．この方法は物体操作中に外乱が生じる場合では有効である．一方で，推論回数の増大に伴い実行時間の増大につながる他，各ステップでのエラーが蓄積されてしまうという欠点が存在する．

これらの課題を解決するために，我々は LILAC を提案する．LILAC は，言語指示に基づき 2D フローを生成した後，その 2D フローをエンドエフェクタの軌道へと変換する視覚言語動作生成モデルである．また，物体操作前の画像及び言語指示から生成したフローに基づき，一連の物体操作のための軌道をオフライン生成することが可能である．本論文の主な貢献は以下の通りである．

- 我々は，RGB 画像と言語指示から物体中心の 2D フローを生成し，それをエンドエフェクタ軌道に変換する VLA である，LILAC を提案する．
- 我々は，視覚情報，言語指示，および視覚的プロンプトを統合しつつ動作生成を行うこと可能にする，Prompt-Conditioned Multimodal Adapter を導入する．
- 我々は，訓練時に Semantic Reconstruction Loss を導入する．これは，モデルに対して言語指示による条件付けを明示的に行いながら軌道を生成することを促す役割を持つ．

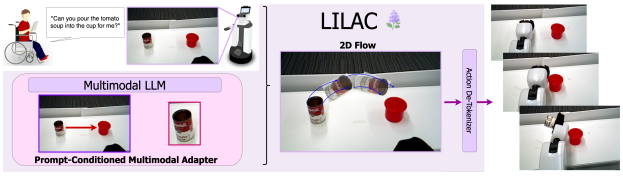

図 1: LILAC の概要図.

2. 関連研究

自然言語指示及び画像を入力とし軌道を生成する VLA は広く研究されている [1–4]．これらの手法は，自然言語指示により条件づけられたタスクを実行するという点で，我々の手法と関係が深い．一方で，こうしたモデルは多くの場合で膨大な教示データを必要とする．例えば $\pi 0$ [1] は事前訓練のために 10K 時間もの教示データを収集し使用している．また，こうしたモデルは，多くの場合 zero-shot 性能が十分でない．そのため，未知環境・タスクでの安全・効率的な実行を行うため，タスクの成否を予測する研究も行われている [5–7]．これらのことより，VLA は，訓練のために実際のロボット・環境を用いて収集した大量の教示データが求められるため，訓練にかかるコストが極めて大きいという課題がある．他方，フローに基づく軌道生成モデルでは，物体操作指示文に基づいて物体の動きに関するフローを生成できれば，結果的にエンドエフェクタの軌道を生成することができる．このフロー生成については，物体中心のフローを生成する場合，学習データはマニピュレータによる物体操作に加え人間による物体操作のデータも使用可能である．従って，あらゆるマニピュレータ・人間による物体操作に関する動画データセットを用いることができるため，低コストで大量の学習データを収集可能であるという利点が存在する．

3. 提案手法

我々は，open-loop 設定において視覚的観察と自然言語指示からタスク関連の動作軌道を生成する手法，LILAC を提案する．図 2 に提案手法の概要を示

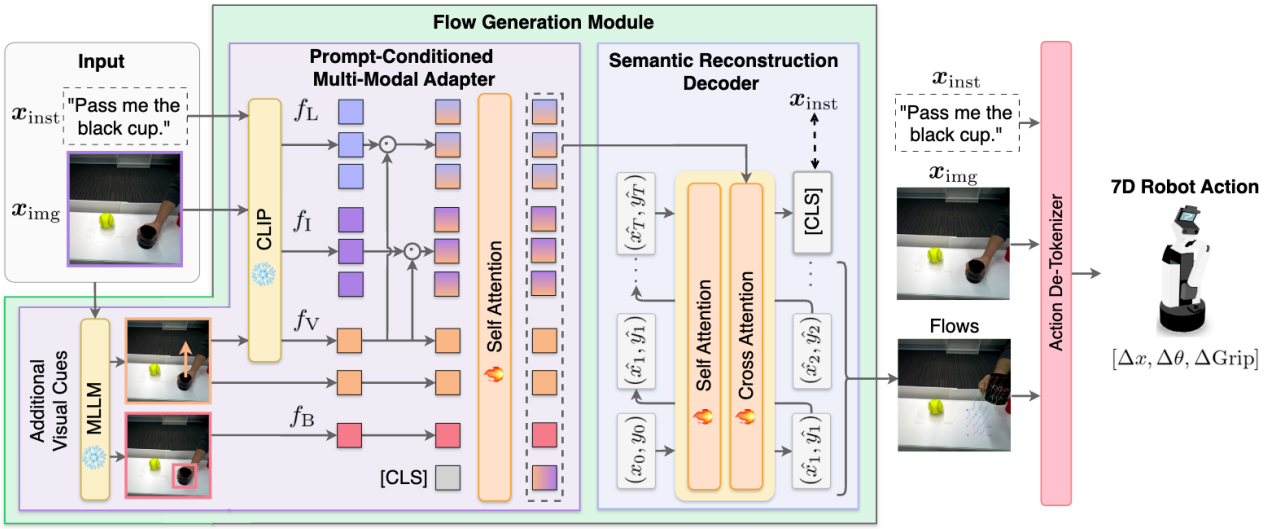

図 2: LILAC におけるフロー生成モジュールのネットワーク図.

す．LILAC は，フロー生成モジュールと Action De-Tokenizer の 2 つの主要なモジュールで構成されている．

3.1 問題設定

本研究では，RGBD 画像，自然言語指示，及びカメラパラメータに基づきエンドエフェクタの軌道を生成するタスクを扱う．また，エンドエフェクタの軌道を生成するにあたり，入力 RGB 画像 $\boldsymbol{x}_{\text{img}}$ と自然言語指示 $\boldsymbol{x}_{\text{inst}}$ に基づき P ステップから構成されるフロー $\mathcal{T} = \{T_1, T_2, \ldots, T_P\}$ を予測するタスクを扱う．各ステップにおける $T_p = \{(x_p^t, y_p^t)\}_{t=1}^{H}$ は，H タイムステップ分の点 p の 2D 座標を表す．フロー生成時は $\boldsymbol{x}_{\text{img}}$ 及び $\boldsymbol{x}_{\text{inst}}$ のみをモデル入力とする．すなわち，深度画像 $\boldsymbol{x}_{\text{depth}}$ とカメラパラメータはエンドエフェクタ軌道への変換時にのみ与えられるとする．ここにおいて，フロー生成時に入力情報に深度情報を含めないことには，以下の利点がある．公開されている大規模動画データセットのうち，深度画像を含むデータセットはほとんど存在しない．従って，入力における視覚情報を RGB 画像のみとすることで，フロー生成モデルの訓練に使用するためのデータセットを構築する上で，より多くの動画データセットを利用可能となる．

3.2 フロー生成モジュール

Prompt-Conditioned Multi-Modal Adapter. 言語指示を視覚情報と効果的にアラインさせ，さまざまな動作に適用するために，Prompt-Conditioned Multi-Modal Adapter を導入する．既存のアプローチは，自然言語指示を物体に関するフローと直接アラインさせるように訓練を行う．言語トークンは動作の表現としては粗く，一方でフローは密な表現となるため，これら 2 つのモダリティを直接整合させることは困難である．我々のモジュールは，最初に言語指示から粗い動きのスケッチを抽出し，それを視覚的プロンプトの形でエンコーダに入力することで，この粒度のギャップを狭める．これよりプロンプトは，ネットワークが高レベルの言語表現とよりよくアラインするフローに関する潜在表現を獲得することに繋がる．このモジュールの入力は，$\boldsymbol{x}_{\text{img}}$ と $\boldsymbol{x}_{\text{inst}}$ である．

本モジュールではまず，事前訓練済み視覚言語モデルを用いて，$\boldsymbol{x}_{\text{img}}$ 及び $\boldsymbol{x}_{\text{inst}}$ から，(i) 望ましい物体の動きを示す変位ベクトル $\boldsymbol{d} \in \mathbb{R}^2$ と，(ii) 操作対象を特定するバウンディングボックス $\boldsymbol{b} \in \mathbb{R}^4$ を予測する．$\boldsymbol{d}$ は視覚言語モデルから獲得された後矢印としてレンダリングされ RGB フレームに重畳することで，視覚的プロンプト $\boldsymbol{x}_{\text{vp}}$ が生成される．この矢印で表現されるプロンプトによって，自然言語の指示がどのように物体の動きに変換されるかについての粗い手がかりをネットワークに与えることが可能となる．この手法は，既存のプロンプトベースの物体操作に関連する手法 [8,9] においても有効であることが示されている．

続いて，このモジュールでは 3 つのモダリティを次式により統合する．

$$\boldsymbol{h}_{\text{mm}} = f_\theta(\boldsymbol{x}_{\text{img}}, \boldsymbol{x}_{\text{inst}}, \boldsymbol{x}_{\text{vp}}), \tag{1}$$

ここで，f_θ はトランスフォーマデコーダである．得られたマルチモーダル特徴 $\boldsymbol{h}_{\text{mm}}$ は，フローを生成する後続モジュールにおいて条件付けとして利用する．

Semantic Reconstruction Decoder. 本モジュールでは，$\boldsymbol{h}_{\text{mm}}$ を条件付けとして，トランスフォーマデコーダによって自己回帰的に軌道系列を生成する．具体的には，causal self-attention と $\boldsymbol{h}_{\text{mm}}$ への cross-attention を用いるトランスフォーマ層によって，以下のように入力系列を処理する．

$$\boldsymbol{h}_t = \text{CausalSelfAttention}([\boldsymbol{h_i}]_{i=0}^{t-1}), \tag{2}$$

$$\boldsymbol{h}'_t = \text{CrossAttention}(\boldsymbol{h}_t, \boldsymbol{h}_{\text{mm}}), \tag{3}$$

ここで，CausalSelfAttention$(\cdot)$ と CrossAttention$(\cdot)$ は，それぞれ causal self-attention 及び cross-attention を表す．また，t $(t = 1, \ldots, T-1)$ はフローの系列長を示す．訓練時は，トランスフォーマデコーダにおいて，出力 $\boldsymbol{h}_T$ を CLS トークンとしてみなし，言語指示埋め込みとのアラインメントを行う．

フロー生成モジュールで用いられる損失関数は，座標に関する誤差損失と，我々が提案する Semantic Reconstruction Loss $\mathcal{L}_{\text{semantic}}$ を組み合わせたものである．この $\mathcal{L}_{\text{semantic}}$ は，言語指示と生成フローとのより良いアラインメントのために導入する．具体的には，$\boldsymbol{h}_T$ 及び言語指示埋め込みを近づけるように設計された損失であり，損失関数としては L1 損失関数を用いる．

3.3 Action De-Tokenizer

Action De-Tokenizer は生成された 2D フローをエンドエフェクタの軌道に変換する．このモジュールへの

表 1: 提案手法およびベースライン手法の定量的比較結果

手法	Fractal					Bridge v2				
	ADE↓	AUC↑	P@5↑	P@10↑	P@20↑	ADE↓	AUC↑	P@5↑	P@10↑	P@20↑
Im2Flow2Act [10]	56.46	0.190	0.134	0.162	0.274	63.19	0.177	0.133	0.154	0.245
FLIP [11]	44.41	0.165	0.069	0.141	0.284	41.95	0.200	0.098	0.180	0.322
LILAC w/o srl	28.94	0.357	0.148	0.320	0.602	31.14	0.325	0.163	0.300	0.512
LILAC w/o vp	33.65	0.253	0.090	0.209	0.458	37.09	0.211	0.097	0.179	0.356
LILAC (full)	**26.98**	**0.434**	**0.214**	**0.421**	**0.668**	**29.44**	**0.396**	**0.229**	**0.382**	**0.576**

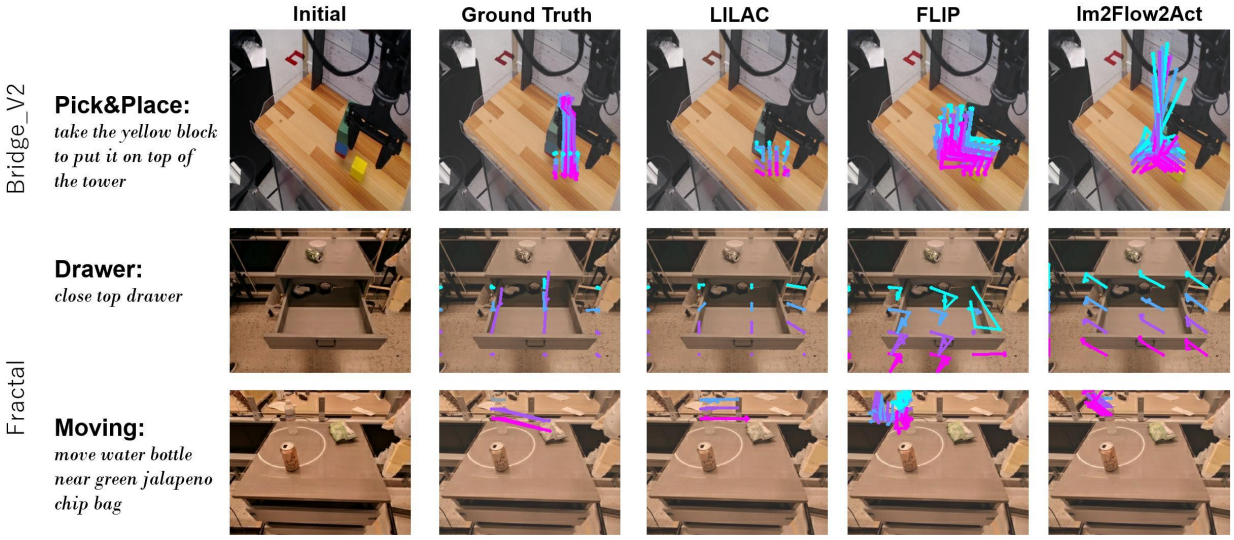

図 3: 提案手法およびベースライン手法の定性的結果

入力は，2D フロー，深度画像 $\boldsymbol{x}_{\text{depth}}$，およびカメラパラメータである．出力は，エンドエフェクタの軌道である．まず，このモジュールは [12] と同様のアルゴリズムを用いて，粗いエンドエフェクタ軌道を生成する．最終的に，トランスフォーマデコーダにおいて，2D フロー，粗いエンドエフェクタ軌道，$\boldsymbol{x}_{\text{img}}$，および $\boldsymbol{x}_{\text{inst}}$ を用いて，精緻化されたエンドエフェクタ軌道を生成する．

4. 実験

実験設定. 我々は新しいベンチマークである RobotFlow ベンチマークを構築した．物体中心のフロー条件付きアクション生成に使用されるデータセットは，通常，動画の各フレーム全体に対してトラッキング点を均一に配置することによって構築される．一方で，この方法ではトラッキング点が画像全体に存在するため，エンドエフェクタ等操作対象の物体以外についてもトラッキングが行われてしまうという課題が存在する．そのため，フィルタリングを行わずにそうしたデータを用いてモデルを訓練すると，不要なフローやノイズまで学習し，非効率的なフロー生成が行われてしまう．この課題に対処するために，我々はマニピュレータによる物体操作タスクのための物体中心のフローを含む RobotFlow ベンチマークを新しく構築した．我々のベンチマークは，Fractal [2] と Bridge v2 [13] サブセットの 2 つのサブセットに基づいて構築されている．ベンチマークは以下の手順により構築した．まず，Qwen-2.5-VL [14] を使用して，自然言語指示に基づいて対象物体のバウンディングボックスを生成した．次に，生成されたバウンディングボックス内で n 個のトラッキング点を均一に配置した．これらの点について，CoTracker-3 [15] を用いて動画クリップ全体でのフローを生成した．結果として得られたフローを正解フローとして扱った．

Fractal サブセットは合計 26,516 エピソードで構成されており，23,866 エピソードを訓練集合，1,325 エピソードを検証集合，1,325 エピソードをテスト集合とした．Bridge v2 サブセットには 10,672 エピソードが含まれ，9,605 エピソードを訓練集合，533 エピソードを検証集合，そして 534 エピソードをテスト集合として使用した．訓練中，我々は両方のサブセットの訓練セットを合わせて使用した．

実験において，Im2Flow2Act [10] 及び FLIP [11] の 2 つのベースラインを使用した．これらは，代表的なフローに基づく VLA であるため使用した．訓練は 2 つの NVIDIA H200 GPU を使用して行われた．LILAC におけるフロー生成モジュールと軌道生成モジュールの訓練時間は，それぞれ約 4 時間と 10 分であった．

定量的結果. 表 1 に，LILAC とベースライン手法との定量的比較結果を示す．各評価尺度において最良の値は太字で示した．評価には，フロー予測タスクにおいて標準的な Average Distance Error (ADE)，Precision@K(P@K)，および AUC を採用した．ここで，P@K は予測された点のうち，正解フローにおける点からユークリッド距離が K 以内である点の割合を表す．既存手法 [11,12] の設定に従い，前のタイムステップからの変位が閾値 δ_t を超える点のみを評価対象とした．

表 1 に示されているように，LILAC は全ての評価尺度においてベースラインを上回った．最良のベースラ

表 2: 実機実験における定量的結果.

手法 [%]	Pick up	Move	Pour	計
FLIP [11]	20	15	**15**	17
LILAC	**40**	**35**	**15**	**30**

(a) (b) (c)

図 4: 実機実験における実験設定.

イン手法と比較して，LILAC は Fractal サブセットで ADE 及び AUC においてそれぞれ 17.43 ポイント及び 0.244 向上した．同様に，Bridge v2 サブセットにおいては，ADE において 12.51 ポイント向上した．これらの結果は，提案手法がベースライン手法と比較してより高品質なフローを生成できたことを示している．

定性的結果． 図 3 に，LILAC およびベースライン手法の定性的結果を示す．1 つ目の例では，ロボットは黄色のブロックをタワーの上に置くように指示されている．ベースライン手法では，ブロックを塔に向かって直線的に移動させるフローを生成できていない一方で，提案手法によって生成されたフローは，ブロックを移動させるべき方向を正確に示している．

2 番目の例では，引き出しを閉じる動作を示すフローを生成することが期待される．しかし，ベースライン手法は引き出しを閉じる正しい方向を把握できず，歪んだフローを生成してしまう．一方，提案手法は正確に閉じる方向を捉えたフローの生成に成功している．これらの例は，提案手法が自然言語による指示と視覚的入力の双方と適切に整合したフローを生成できることを示している．

Ablation Study. LILAC の各モジュールの有効性を検証するために Ablation Study を実施した．表 1 にその結果を示す．Semantic Reconstruction Loss（w/o SRL）及び視覚的プロンプト（w/o VP）をそれぞれ除去し，その有効性を検証した．表 1 に示されているように，いずれの場合も各ベンチマーク及び評価尺度においてスコアが低下した．この結果より，これらの新規性がフロー生成モデルの性能向上に寄与していることが示された．

実機実験． LILAC の実機での性能を評価するため，実機実験を行った．図 4 に実験設定を示す．ロボットプラットフォームとして，トヨタ自動車製の Human Support Robot（HSR，図 4(a)）を用いた．本実験では ‘pick-up’，‘move’，および ‘pour’ の 3 つのタスクを扱った．各タスクについて，まず，フロー生成モジュールの訓練のため，100 エピソードずつ人間の物体操作データを収集した．続いて，後段の軌道変換モジュールを訓練するため，20 エピソードずつロボットの教示データを収集した．これらのデータ収集には，図 4(b) に示される物体群を用いた．各モデルは，人間とロボットのデータを統合したデータセットで学習され，未知の物体及び言語指示を用いて評価を行った．ロボットは各タスクにつき 20 回試行し，合計で 60 試行が行われた．評価の際には，図 4(c) に示される物体群を用いた．

表 2 に定量的比較結果を示す．LILAC は，各物体操作における成功率および全タスクにおける平均成功率の両方で，すべてのベースライン手法を上回った．これらの結果は，LILAC が予測するフローが動作における物体の動きをより忠実に表現しており，その結果としてエンドエフェクタの軌道の品質自体も向上したことを示している．

5. 結論

本研究では，Open-loop 軌道生成のためのフローに基づく VLA である, LILAC を提案した．LILAC は自然言語指示および物体操作前の画像をもとに 2D フローを生成し，さらに，少量の教示データによってエンドエフェクタ軌道を生成する．本研究の主要な寄与は以下のとおりである．

- RGB 画像と言語指示から物体中心の 2D フローを生成し最終的に軌道に変換する VLA である，LILAC を提案した．
- 視覚情報，言語指示，および視覚的プロンプトを統合しつつ動作生成を行うこと可能にする，Prompt-Conditioned Multimodal Adapter を導入した．
- モデルに対して言語指示による条件付けを明示的に行いながら軌道を生成することを促す役割を持つ Semantic Reconstruction Loss を導入した．
- 実験結果より，LILAC は既存手法を上回る品質のフロー生成に成功したほか，実機実験においても既存手法を上回るタスク成功率を達成した．

謝辞

本研究の一部は，JSPS 科研費 23K28168，JST ムーンショット，JSPS 特別研究員奨励費 JP23KJ1917 の助成を受けて実施されたものである．

参 考 文 献

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, et al., “π_0: A Vision-Language-Action Flow Model for General Robot Control,” RSS, 2025.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, et al., “RT-1: Robotics Transformer for Real-World Control at Scale,” arXiv preprint arXiv:2212.06817, 2022.

[3] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, et al., “OpenVLA: An Open-Source Vision-Language-Action Model,” CoRL, 2024.

[4] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, et al., “Octo: An Open-Source Generalist Robot Policy,” RSS, 2024.

[5] M. Goko, M. Kambara, D. Saito, S. Otsuki, and K. Sugiura, “Task Success Prediction for Open-Vocabulary Manipulation Based on Multi-Level Aligned Representations,” CoRL, 2024.

[6] Z. Liu, A. Bahety, and S. Song, “REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction,” CoRL, pp.3468–3484, 2023.

[7] M. Kambara and K. Sugiura, “Pre-Manipulation Alignment Prediction for Open-Vocabulary Object Manipulation Based on End-Effector Trajectories,” MVA, 2025.

[8] M. Hwang, J. Hejna, D. Sadigh, and Y. Bisk, “Motif: Motion instruction fine-tuning,” IEEE Robotics and Automation Letters, 2025.

[9] Y. Kuang, J. Ye, H. Geng, J. Mao, C. Deng, L. Guibas, et al., “RAM: Retrieval-based Affordance Transfer for Generalizable Zero-Shot Robotic Manipulation,” CoRL, 2024.

[10] M. Xu, Z. Xu, Y. Xu, C. Chi, G. Wetzstein, M. Veloso, and S. Song, “Flow as the Cross-domain Manipulation Interface,” CoRL, 2024.

[11] C. Gao, H. Zhang, Z. Xu, Z. Cai, and L. Shao, “FLIP: Flow-Centric Generative Planning for General-Purpose Manipulation Tasks,” ICLR, 2025.

[12] H. Bharadhwaj, R. Mottaghi, et al., “Track2act: Predicting Point Tracks from Internet Videos Enables Generalizable Robot Manipulation,” ECCV, pp.306–324, 2024.

[13] H. Walke, K. Black, T. Zhao, Q. Vuong, C. Zheng, P. Hansen, A. He, V. Myers, M. Kim, et al., “BridgeData V2: A Dataset for Robot Learning at Scale,” CoRL, pp.1723–1736, 2023.

[14] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, et al., “Qwen2.5-VL Technical Report,” arXiv preprint arXiv:2502.13923, 2025.

[15] N. Karaev, I. Makarov, J. Wang, et al., “CoTracker3: Simpler and Better Point Tracking by Pseudo-Labelling Real Videos,” arXiv preprint arXiv:2410.11831, 2024.