Physically Consistent Humanoid Loco-Manipulation using Latent Diffusion Models

Anonymous Author(s)
Affiliation
Address
email

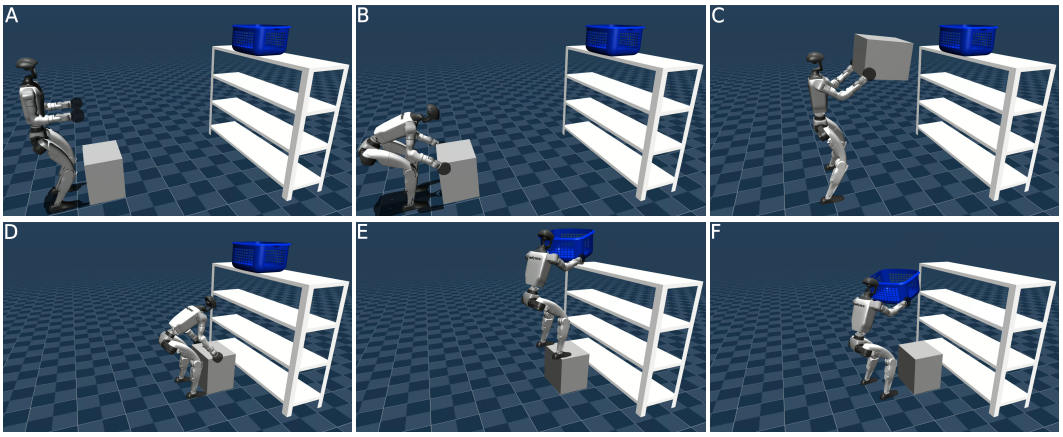


Figure 1: A long-horizon loco-manipulation task generated with our proposed method; the robot moves a box to enable reaching for a laundry basket.

Abstract: This paper uses the capabilities of latent diffusion models (LDMs) to generate realistic RGB human-object interaction scenes to guide humanoid loco-manipulation planning. To do so, we extract from the generated images both the contact locations and robot configurations that are then used inside a whole-body trajectory optimization (TO) formulation to generate physically consistent trajectories for humanoids. We validate our full pipeline in simulation for different long-horizon loco-manipulation scenarios and perform an extensive analysis of the proposed contact and robot configuration extraction pipeline. Our results show that using the information extracted from LDMs, we can generate physically consistent trajectories that require long-horizon reasoning.

Keywords: Contact Planning, Humanoids, Generative Models

1 Introduction

It has been long argued that humanoids are the best platform to replace humans in repetitive and dangerous tasks, because of the similarities in their morphologies. However, the complexity of these platforms poses significant challenges that have hindered the progress and we still do not see humanoid robots reliably doing real-world tasks. In particular, humanoids are high-dimensional systems with highly unstable dynamics (compared to wheeled and four-legged robots) which renders their planning problem highly challenging. Furthermore, performing any reasonable loco-manipulation task requires a long-horizon reasoning procedure and none of the existing methods can scale to such problems. The similarity between the human and humanoid morphologies can come to rescue in such a case, as the robot can imitate the behavior of humans doing the same task. Thanks to the recent advances in generative models, it is nowadays possible to generate a desired

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

human behavior from text prompts. While the outputs of these models do not respect the geometrical and physical constraints of the real world, they can guide the existing optimization frameworks to find physically consistent motions quickly.

In this paper, we develop a framework to rapidly synthesize plausible 3D human-object interaction scenes using latent diffusion models (LDMs) [1] for 2D image generation, without the need for ad hoc heuristics or 3D richly annotated data, and use the retargeted motion inside a whole-body trajectory optimization (TO) formulation to generate physically consistent motions to achieve complex long-horizon tasks.

The main contributions of this work are as follows:

- We introduce, to the best of our knowledge, the first pipeline that plans both contacts and robot configurations for humanoid loco-manipulation using LDMs.
- We integrate our proposed robot configuration and contact planner within a whole-body TO formulation to generate physically consistent trajectories.
- We validate our approach in simulation on two challenging long-horizon scenarios, and perform an extensive analysis with various baselines.

2 Related work

Classical approaches for planning and control of loco-manipulation for humanoids consider the effect of manipulated objects on the locomotion system as a disturbance [2, 3, 4, 5]. However, for general loco-manipulation problems, concurrent consideration of both locomotion and manipulation is essential. To reduce the complexity of the holisitc loco-manipulation planning, more advanced approaches relied on splitting the system into simpler coupled dynamical systems [6], using heuristics to separate zones in which locomotion or loco-manipulation or manipulation occurs [7], splitting the object path planning and locomotion planning problems [8], or using a predefined contact sequence [9]. [10] used a hierarchy of optimal controllers to perform loco-manipulation automatically, augmenting the locomotion problem with logic predicates for manipulation [11]. However, they demonstrated only quadrupedal loco-manipulation with single arm, which is simpler than a humanoid with two arms.

There have been recent efforts on the use of Deep Reinforcement Learning (DRL) for loco-manipulation tasks in the real world [12, 13, 14, 15]. However, these approaches are limited to very simple manipulation tasks with a quadruped and cannot reason about the complex, long-horizon humanoid loco-manipulation tasks.

Recent advances in imitation learning have shown promise in generating loco-manipulation policies from teleoperation demonstrations [16, 17]. However, generating teleoperated demonstrations for humanoid robots is extremely difficult compared to other manipulation settings [18], as the system is highly unstable and can easily fall down. [19] used TO to generate demonstrations that are then imitated using DRL. However, TO is a local approach and would fail to generate long-horizon trajectories that require reasoning.

3 Method

In this section, we present our approach to plan contacts and robot configurations to guide a TO procedure for arbitrarily long-horizon humanoid loco-manipulation tasks. Our approach does not rely on task-specific heuristics or 3D interaction datasets. Instead, we propose a pipeline that introduces an optimization-based approach that leverages LDMs to generate realistic human-object interaction 2D scenes, given a high-level description of the desired interactions. These 2D RGB scenes are used to extract the contact locations and robot configurations that are later used by TO (Section 4). The pipeline overview is illustrated in Fig. 2.

3.1 Planning Contacts & Robot Configurations

The planner receives high-level instructions P (that can come from a language model) and RGB-D images of the objects in the scene as input. The high-level plan consists of ordered sequences of text prompts describing how to break down the long-horizon task. For tasks involving placement, we assume to receive the target 3D location and yaw of the object. The RGB-D images consist of the RGB frames R_s and respective depth images D_s of the objects to be manipulated. The output of the planner is the sequence of 3D contact locations L and associated robot configurations C. The planning process consists of three main steps, which are detailed in the sections below.

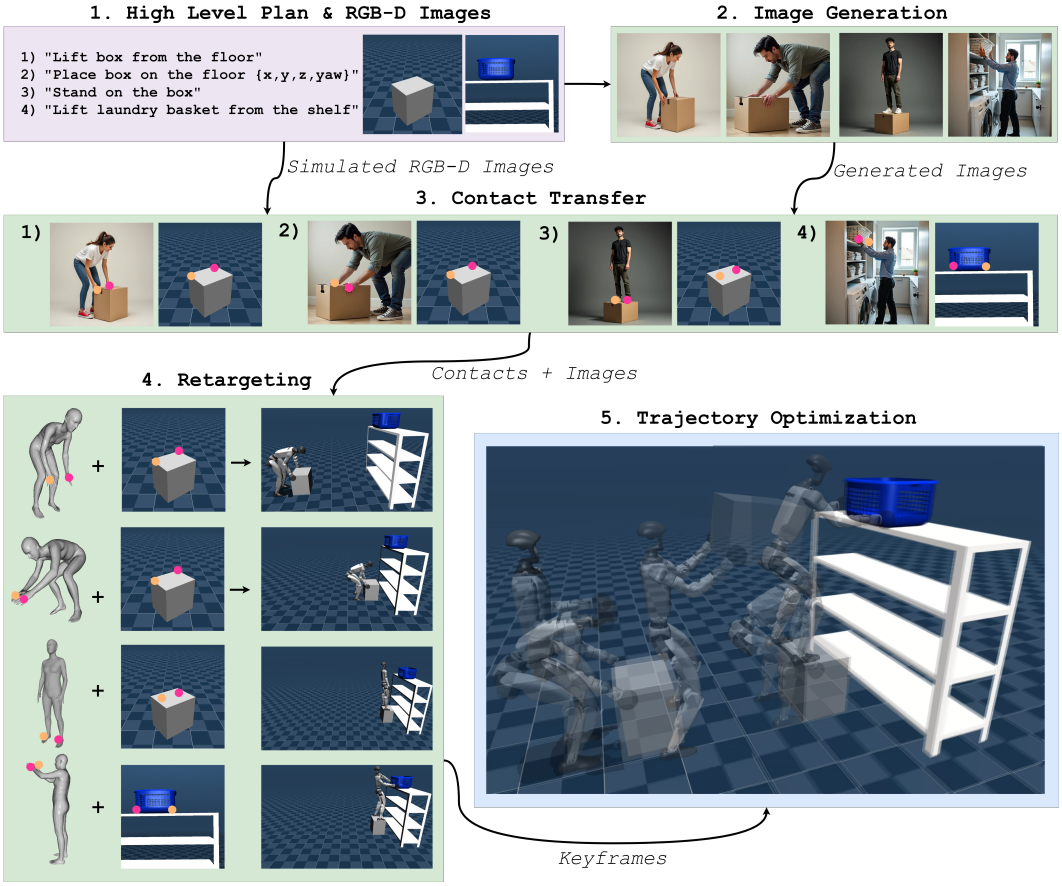


Figure 2: Pipeline overview.

3.1.1 Image Generation

Given P, we use a state-of-the-art latent diffusion model (LDM) [20] to generate a collection of images R_g demonstrating how to accomplish the long-horizon task. The instructions consist of an ordered sequence of short text prompts describing very minimally the expected interaction with the objects in the scene, as shown in Fig. 2. Using directly these prompts leads to images that do not depict a full-body person, which is essential for our pipeline. In fact, to extract the contact locations of the hand and feet, and the respective robot configuration the vast majority of the human body has to be visible in the generated image. Therefore, we automatically append a static set of words to the task prompts to generate a full-body person. The additional words are mainly a general description of the person's hair color and clothing style, which forces the LDM to generate the correct interaction but also a full-body. Given an instruction, we modify it in the following way:

"*A scene of a person* {***predicate***}*+ing* {***subtask prompt without predicate***}*. The person has dark hair and is wearing casual clothes such a shirt, jeans, and sneakers.*"

where {***predicate***} is the verb describing the task's action and is always the first word in the task prompt, while {**task prompt without predicate**} is what remains of the prompt after removing the predicate and it mainly describes the object to which the action applies and its position in the environment. The generated images are then fed to the contact transfer (Sec. 3.1.2) and retargeting (Sec. 3.1.3) modules.

3.1.2 Contact Transfer

The contact transfer stage extracts and maps the 2D contact information from the images in R_g to the corresponding 3D contact locations in the scene to manipulate the object of interest for the task. To achieve this we use a three-step approach as shown in Fig. 3.

First, we compute the 2D semantic masks of the objects in R_g and R_s. To do so, we use a Vision Language Model (VLM) [21] to perform open vocabulary object detection that returns the bounding box coordinates enclosing the objects. However, the obtained bounding box is not tight enough for the point cloud extraction later. Hence, we apply a visual segmentation foundation model [22] that further refines the VLM output and returns a per-pixel segmentation of the objects.

Second, using the objects' masks and the depth information we proceed to compute the objects' point clouds by performing a 2D to 3D lifting procedure. For the simulated images R_s we have the ground truth depth D_s from the simulated RGB-D camera in the MuJoCo [23] simulator and its correct camera intrinsic parameters. However, for R_g we are missing both its depth estimate and correct camera intrinsics. This is because LDMs only output RGB images and do not adhere to a specific camera model during the image generation. Therefore, to estimate each generated image depth we leverage a zero-shot metric depth geometric foundation model [24] to obtain the estimated metric depth D_g. The missing camera intrinsics are computed using an empirical trial and error approach where we found that using the LDMs' image resolution as the focal lengths and half the focal lengths for the principal point offsets leads to a reasonable point cloud geometry, without too much distortion. Finally, we also apply a noise removal process to the point clouds to remove outliers.

Third, we use a sampling-based optimization approach that combines the semantic scores from a semantic-aware foundation model and the object geometries to transfer the 2D contact locations from R_g to the 3D world. The task of finding correct semantic correspondences across images is a challenging one, and especially so in our case. This is mainly due to the fact that during the image generation process, we have limited control over the generated object properties, such as viewpoint, shape, and texture, leading to significant intra-class variation between the generated and the simulated objects. To solve the semantic correspondence problem, we use a semantic-aware foundation model [25] to obtain semantic matches between the images. However, depending exclusively on the model is not reliable, as the intra-class variation can be large leading to incorrect mappings that deteriorate the output trajectory (Sec. 5.2). Therefore, we propose a sampling-based algorithm that refines the correspondences from the semantic-aware model using the objects' point cloud geometries. The underlying idea is that correct semantic matches should result in a good geometrical overlap between the objects' point clouds. Hence, we generate a semantic score pool for some sampled 2D points on the objects' masks found in R_g, such that for each sampled point we obtain the top N most plausible correspondences in the respective objects' masks found in R_s from the model. To find the set of semantic matches that best aligns the objects' geometries, we formulate a sampling-based algorithm that searches randomly within the semantic pool. More precisely, given the sampled semantic correspondences we compute the respective 3D correspondences and solve for the rigid-body transform with the Singular Value Decomposition (SVD) algorithm. The transform is then further refined with the Iterative Closest Point (ICP) algorithm. We repeat this process for 10 iterations and pick the rigid-body transform that obtained the highest overlapping score between the objects' point clouds. In practice, we found that within 3 iterations the best transform is already found. Finally, we compute the 3D contact locations L by applying the transform to the 3D lifted hand and feet 2D locations obtained by running a human pose estimator [26] on the images in R_g.

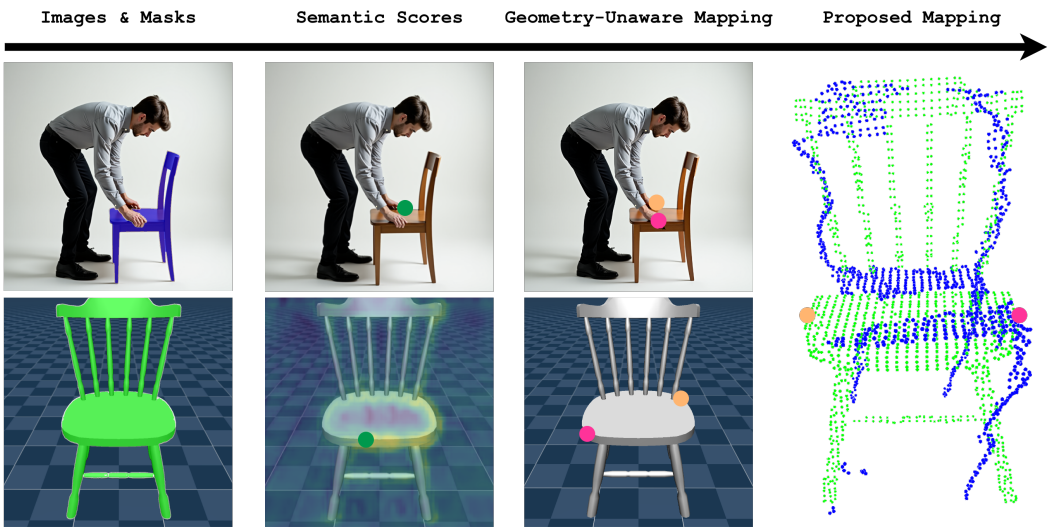


Figure 3: Contact extraction procedure.

To avoid penetrations between the real object and L, we apply a simple heuristic to L to make sure these are projected to the closest object's surface.

3.1.3 Retargeting

In the retargeting stage, we use the depicted humans in R_g and L to obtain the robot configuration R. R consists of a 35D vector describing the robot's 6D base state and joint angles for each actuated joint of the system. Extracting this configuration is an important step in our proposed pipeline, as this complementary information to the contacts guides the TO to a better local minima. However, we can't map directly the human configuration to the robot, due to differences in the number of degrees of freedom, limb length, and height. Therefore, we formulate an Inverse Kinematics (IK) based retargeting process that remaps the extracted human configurations from the generated images to a kinematically feasible robot configuration. Figure 4 shows an outline of the retargeting process.

First, we extract the human configuration we wish to remap, that consists of the human's joint angles, foot positions, and base orientation. To do so, we use WHAM [26] a 3D human pose estimation module that returns the 3D joint positions and the joint orientations given a 2D image depicting a human body. For the joint orientations, as our humanoid has a subset of the degrees of freedom obtained from WHAM, we follow a similar approach to [27], where we only consider the human's joints that have a corresponding match on the robot. The foot positions are extracted from the 3D body model by computing the relative distance between the human's pelvis and the left and right ankles, however we only consider the planar coordinates as the foot height is set based on the task. The base orientation is obtained by first applying the rigid body transformation, computed during the contact transfer stage (Sec. 3.1.2), to the 3D body model and then computing the relative orientation between the human's pelvis and the simulated object of which we know the full state. Finally, we apply the IK retargeting which uses as constraints the hand and feet contact locations, and feet pitch angle. The joint angles only act as a regularization term to guide the IK output towards a human-like configuration.

4 Trajectory Optimization

In this section, we outline our TO formulation and how the extracted contact locations and robot configurations are used within the formulation. We use the centroidal dynamics coupled with whole-body kinematics formulation similar to [28].

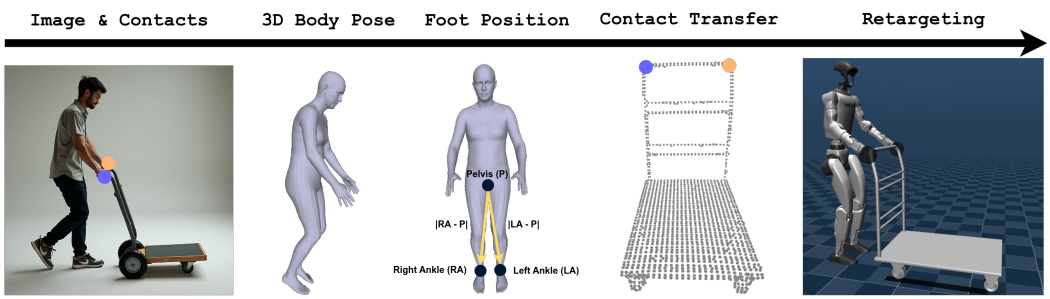


Figure 4: Keyframe extraction procedure.

4.1 Task-generic Cost

To avoid task-specific tuning, the stagewise cost L_{stage} is designed to regularize the robot locomotion with minimum heuristics:

$$L_{\text{stage}} := b_{\text{st}} L_{\text{st}} + (1 - b_{\text{st}}) L_{\text{wk}} + L_{\text{reg}} + L_{\text{slack}}, \tag{1}$$

where b_{st} is the Boolean flag indicating whether the robot is in a stance phase, $L_{\text{st}}, L_{\text{wk}}$ are respectively the cost of stance and walking phases, L_{reg} is the common regularization term, and L_{slack} is the penalty of the slack variables used for constraints. All costs are in quadratic form $w||(\cdot)||_2^2$, where w is the weight.

4.2 Keyframe Cost

The full configuration of the robot and the contact locations generated from the planning module constitute a *keyframe*, which provides waypoints for a long-horizon loco-manipulation task. We add the following cost term for the keyframe robot pose:

$$L_{\text{kf}}^{\text{b}} := W_{\text{kf}}^{\text{b}} [(r_{\text{base},z} - r_{\text{base},z}^{\text{kf}})^2, ||\boldsymbol{\Theta} - \boldsymbol{\Theta}^{\text{kf}}||_2^2]^\top, \tag{2}$$

where $W_{\text{kf}}^{\text{b}} \in \mathbb{R}^{1\times 2}$ is the cost weight for which we used $[100, 10]$ in this paper. $(\cdot)^{\text{kf}}$ denotes the corresponding value at a certain keyframe, $r_{\text{base},z}$ is the z component of the robot base, and $\boldsymbol{\Theta}$ is the aforementioned base orientation. The keyframes also indicate the desired relative foot position w.r.t. the object, which can then be used to compute a global position reference $\mathbf{r}_f^{\text{des}}$. It is added as an additional term to the corresponding stance phase cost L_{st}:

$$L_{\text{kf}}^{\text{f}} := W_{\text{kf}}^{\text{f}} ||(r_{lf,xy} + r_{rf,xy})/2 - \mathbf{r}_f^{\text{des}}||_2^2, \tag{3}$$

where W_{kf}^{f} is the keyframe foot position weight for which we used 5e2 in this paper.

The overall TO problem can be formulated as

$$\begin{aligned} \min . \quad & \frac{1}{N} \sum_{i=0}^{N} [L_{\text{stage}}^i + L_{\text{col}}^i] + \sum_{j \in \mathcal{K}} L_{\text{kf}}^{\text{b},j} \\ \text{s.t.} \quad & \text{Dynamics, Contacts, Collision.} \end{aligned} \tag{4}$$

5 Experiments

In this section, we present the results of applying our proposed pipeline for two different scenarios, each involving a long-horizon task. The first scenario (S1) consists of fetching a laundry basket placed on top of a shelf. As the basket is not easily reachable, the robot needs to move a box close to the shelf and step on top of it to be able to reach the basket. The second scenario (S2) consists of moving a box placed on top of a table using a trolley and then pushing the trolley. We used the MuJoCo simulation environment [23] and the Unitree G1 humanoid robot for all the

visualization and comparisons. For both scenarios the pipeline took several minutes, with the main bottlenecks being the image generation, the sematic-aware foundation model inference, and the trajectory optimization. This currently hinders real-time capabilities.

First, we demonstrate the promise of our framework in generating physically plausible trajectories for both scenarios. We also compare our results to a TO baseline that is guided only through contacts from a semantic-aware foundation model. Note that without providing such information, it was impossible for TO to solve the tasks, as both tasks require long-horizon reasoning that is unfeasible to do with local TO.

Second, we perform an ablation study on our proposed contact extraction process (Sec. 3.1.2), where we compare a geometry-unaware contact transfer and our method to see how much our proposed contact refinement improves the resulting trajectories.

Third, we carry out an ablation study on the effect of each component of the keyframe information (Sec. 3.1.3) on the TO output. We refer the reader to the supplementary video for the additional qualitative results.

5.1 Physically Plausible Trajectories

We compare the output of the TO results (Sec. 4) when using the output of our proposed planning pipeline (Sec. 3.1) and a naive approach. Our pipeline feeds both the refined contact information and the respective robot configuration extracted from LDMs to TO. The naive approach only gives the contact locations obtained directly from the semantic-aware foundation model to the TO. In both cases, we use a minimal set of collision penalties constraints.

Figure 5 presents the total amount of negative collision penetrations at each timestep of the trajectory from MuJoCo, while enabling collision penalties for both scenarios. Our proposed pipeline maintains a collision-free behavior throughout the whole trajectory, while a naive approach experiences significant negative penetrations during the whole trajectory. One might argue that all possible collision constraints could be enabled in the TO to obtain a collision-free motion. However, in such a case TO fails to solve the task and gets stuck in a local minima.

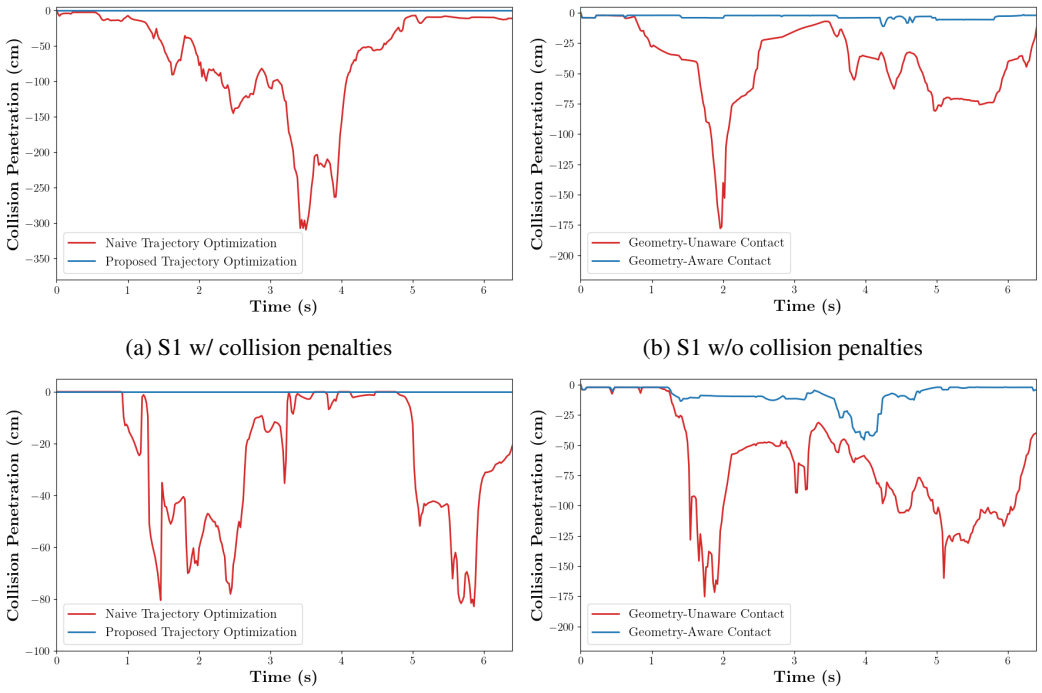
5.2 Geometry Improves Contact Transfer

We present a comparison between the trajectory outputs when using our proposed contact transfer (Sec. 3.1.2) and a geometry-unaware contact extraction process. Our contact extraction approach uses semantic and geometry cues from a semantic-aware foundation model and the objects' point clouds to refine incorrect semantic matches. On the other hand, the geometry-unaware contacts directly use the output of the model without any correction. Our metric for the comparison is the amount of collision penetration, both in terms of self-collisions and robot-object collisions. To study the effect of the contact extraction in isolation, we disable all collision penalties in the TO problem and use the same extracted robot configuration (Sec. 3.1.3) in both approaches.

Figure 5 shows the result of the ablation study for scenarios S1 and S2. For both S1 and S2, we clearly see that a geometry-unaware contact transfer leads to a higher number of negative collisions, and therefore a higher amount of negative penetration during the trajectory. While with our approach, there still exists some negative penetrations but these are substantially less for both scenarios and can be prevented with a minimal set of collisions (Sec. 5.1). However, to obtain a collision-free motion, a geometric-unaware approach would require significantly more collision constraints which makes the problem extremely non-convex with many local minima. Local TO in such cases gets stuck in local minima and is unable to solve the task.

5.3 Keyframes Reduce Penetration

In this section, we conduct ablation studies between the keyframe-guided TO and the TO without keyframes (we call this baseline NoKeyframe). Note that in this case, we use the refined contact



(a) S1 w/ collision penalties

(b) S1 w/o collision penalties

(c) S2 w/ collision penalties

(d) S2 w/ collision penalties

Figure 5: Collision penetrations comparison between the TO output using our proposed pipeline (blue) and a naive approach (red) for both the laundry scenario (S1) and the trolley scenario (S2) with and without collision penalties enabled.

information (Sec. 3.1.2) in all problems and the focus is to quantify the effect of the different components of the keyframe. To do that, we formulate various settings with different combinations of keyframe utilization. This includes the keyframe base cost in (2), the keyframe foot position cost as in (3), and the subtask-like warm start in Sec. 4.2. All formulations are provided with the same contact sequences and the same set of collision constraints as in Sec. 5.1.

5.3.1 S1 (Basket retrieval)

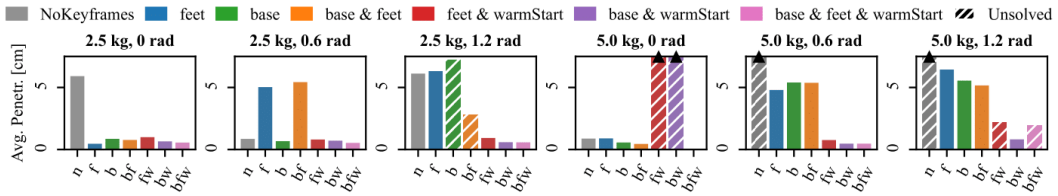
Three keyframes are used in this experiment: picking up the box, placing the box, and grabbing the basket while standing on the box. We also varied in the experiment the box mass in the range $\{2.5, 5.0\}$kg, and the initial robot yaw angle ϕ in the range $\{0, 0.6, 1.2\}$rad. We have observed in our experiments that these two parameters have a large impact on the performance of TO. We added only a minimal set of collision constraints as defined in Sec. 5.1 for S1. The results are shown in Fig. 6a.

5.3.2 S2 (Trolley pushing)

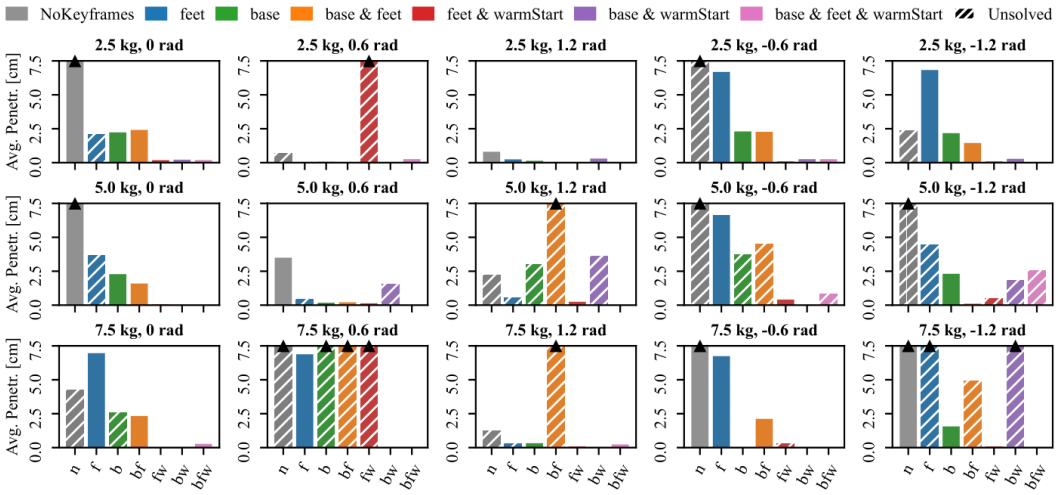
In this scenario, to describe the goal of pushing the trolley forward, an additional cost on the trolley's position is added. Three keyframes are used: picking up the box, placing the box, and the beginning of pushing the trolley. The set of tested box mass is $\{2.5, 5.0, 7.5\}$kg, and that of the initial robot yaw angle ϕ is $\{0, 0.6, 1.2, -0.6, -1.2\}$rad. A minimal set of collision constraints is added as explained in Sec. 5.1 for S2. The results are shown in Fig. 6b.

5.3.3 Discussion

For both S1 and S2, it can be observed in Fig. 6 that using keyframes helps the success rate of TO in solving the problem, and leads to better solutions with low penetration. This is evident when we



(a) S1 collision data. The y-axis bound of average penetration is 7.5cm.



(b) S2 collision data. The y-axis bound of average penetration is 15cm.

Figure 6: Collision ablation study data. Subtitles are (box mass, robot initial yaw). The upward solid triangles denote that the average penetrations exceed the y-axis bound. Slash hatches on bars mean the optimization failed to converge with the corresponding settings. The legends represent settings of keyframe utilization. The average penetration is calculated as the sum of penetration divided by the number of shooting nodes.

compare the case where AllKeyframe is used (pink, rightmost bars) versus the NoKeyframe case (grey, leftmost bars).

For example, in the (5.0kg, 0rad) test of S1, the AllKeyframe case results in zero penetration, while removing any part of the keyframe either hinders convergence or results in large penetrations. In particular, settings with the warm-start tend to have better convergence and lower penetration. In some cases, we can see that the AllKeyframe case has achieved slightly worse results than other cases. We believe this is due to some details in the trajectory optimization solver.

6 Conclusion

In this work, we presented a novel approach that generates physically consistent trajectories for long-horizon loco-manipulation tasks. We do so by leveraging LDMs to synthesize 2D images demonstrating how a human would accomplish a task. From such demonstrations, we extract the robot configurations and contact locations for a long-horizon high-level plan, which are used to guide a whole-body TO. We evaluated the proposed method in simulation for two challenging scenarios that require long-horizon reasoning, and showed that our proposed pipeline can generate physically plausible trajectories for long-horizon humanoid loco-manipulation tasks.

Future work will focus on evaluating the proposed work on a real humanoid robot. Furthermore, we will relax the assumption of being given the high-level plan, by developing a long-horizon task planner that leverages large language models instead. Finally, we will consider using human videos instead of 2D images due to the current limited capability of LDMs in generating more complex human-object interactions tasks.

References

[1] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[2] K. Bouyarmane and A. Kheddar. Humanoid robot locomotion and manipulation step planning. *Advanced Robotics*, 26(10):1099–1126, 2012.

[3] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi. A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot. *IEEE Robotics & Automation Magazine*, 26(4):73–82, 2019.

[4] W. Thibault, F. J. A. Chavez, and K. Mombaur. A standardized benchmark for humanoid whole-body manipulation. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pages 608–615. IEEE, 2022.

[5] J. Li and Q. Nguyen. Multi-contact mpc for dynamic loco-manipulation on humanoid robots. In *2023 American Control Conference (ACC)*, pages 1215–1220. IEEE, 2023.

[6] A. Settimi, D. Caporale, P. Kryczka, M. Ferrati, and L. Pallottino. Motion primitive based random planning for loco-manipulation tasks. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pages 1059–1066. IEEE, 2016.

[7] P. Ferrari, M. Cognetti, and G. Oriolo. Humanoid whole-body planning for loco-manipulation tasks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4741–4746. IEEE, 2017.

[8] M. Murooka, I. Kumagai, M. Morisawa, F. Kanehiro, and A. Kheddar. Humanoid loco-manipulation planning based on graph search and reachability maps. *IEEE Robotics and Automation Letters*, 6(2):1840–1847, 2021.

[9] H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, and S. Vijayakumar. Roloma: Robust loco-manipulation for quadruped robots with arms. *Autonomous Robots*, 47(8):1463–1481, 2023.

[10] J.-P. Sleiman, F. Farshidian, and M. Hutter. Versatile multicontact planning and control for legged loco-manipulation. *Science Robotics*, 8(81):eadg5014, 2023.

[11] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. *Robotics: Science and Systems Foundation*, 2018.

[12] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[13] S. Jeon, M. Jung, S. Choi, B. Kim, and J. Hwangbo. Learning whole-body manipulation for quadrupedal robot. *arXiv preprint arXiv:2308.16820*, 2023.

[14] T. Portela, G. B. Margolis, Y. Ji, and P. Agrawal. Learning force control for legged manipulation. *arXiv preprint arXiv:2405.01402*, 2024.

[15] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv. Learning-based legged locomotion: State of the art and future perspectives. *The International Journal of Robotics Research*, 44(8):1396–1427, 2025.

[16] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[17] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024.

[18] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[19] F. Liu, Z. Gu, Y. Cai, Z. Zhou, S. Zhao, H. Jung, S. Ha, Y. Chen, D. Xu, and Y. Zhao. Opt2skill: Imitating dynamically-feasible whole-body trajectories for versatile humanoid loco-manipulation. *arXiv preprint arXiv:2409.20514*, 2024.

[20] black forest labs. Flux, 2014. URL https://blackforestlabs.ai/. 2024-07-01.

[21] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, and L. Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024.

[22] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[23] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[24] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

[25] J. Zhang, C. Herrmann, J. Hur, E. Chen, V. Jampani, D. Sun, and M.-H. Yang. Telling left from right: Identifying geometry-aware semantic correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[26] S. Shin, J. Kim, E. Halilaj, and M. J. Black. Wham: Reconstructing world-grounded humans with accurate 3d motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2070–2080, 2024.

[27] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[28] H. Dai, A. Valenzuela, and R. Tedrake. Whole-body motion planning with centroidal dynamics and full kinematics. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 295–302, 2014. doi:10.1109/HUMANOIDS.2014.7041375.