Constrained Style Learning from Imperfect Demonstrations under Task Optimality

Kehan Wen
ETH Zurich
kehwen@ethz.ch

Chenhao Li
ETH AI Center
chenhao.li@ai.ethz.ch

Junzhe He
ETH Zurich
junzhe@ethz.ch

Marco Hutter
ETH Zurich
mahutter@ethz.ch

Abstract: Learning from demonstration has proven effective in robotics for acquiring natural behaviors, such as natural motions and lifelike agility, particularly when explicitly defining style-oriented reward functions is challenging. Synthesizing stylistic motions for real-world tasks usually requires balancing task performance and imitation quality. Existing methods generally depend on expert demonstrations closely aligned with task objectives. However, practical demonstrations are often incomplete or unrealistic, causing current methods to boost style at the expense of task performance. To address this issue, we propose formulating the problem as a constrained Markov Decision Process (CMDP). Specifically, we optimize a style-imitation objective with constraints to maintain near-optimal task performance. We introduce an adaptively adjustable Lagrangian multiplier to guide the agent to imitate demonstrations selectively, capturing stylistic nuances without compromising task performance. We validate our approach across multiple robotic platforms and tasks, demonstrating both robust task performance and high-fidelity style learning. On ANYmal-D hardware we show a 14.5% drop in mechanical energy and a more agile gait pattern, showcasing real-world benefits.

Keywords: Constrained Markov Decision Process, Imitation Learning, Legged Robots

1 Introduction

Reinforcement learning (RL) [1] has demonstrated significant potential to achieve robust and adaptive control of legged robots, due to its capability to handle uncertainties in real-world applications [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Despite this success, integrating fine-grained stylistic behaviors, such as agile locomotion or expressive motions, remains challenging. Although RL excels at optimizing clear, task-specific goals (e.g., velocity tracking, goal-reaching), crafting reward functions for nuanced, high-dimensional stylistic behaviors is inherently difficult [3, 7, 13].

Learning from Demonstration (LfD) [14] has emerged as a powerful technique that embeds stylistic imitation in robotic behaviors. Traditional LfD techniques include motion-clip tracking [15] and adversarial imitation learning [16, 17, 18]. However, obtaining high-quality reference motions that perfectly align with task objectives is often labor intensive (e.g., motion capture). Most practical demonstrations contain imperfections, such as incomplete data collection or unrealistic motions from biological counterparts with different morphologies [19]. For example, human motion data collected on flat terrain may poorly generalize to tasks involving complex, uneven terrains, causing suboptimal task performance. Strict adherence to these demonstrations often leads to suboptimal outcomes, emphasizing the critical trade-off between stylistic fidelity and task effectiveness.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

A common approach is to manually tune fixed weights or craft a customized curriculum, a tedious process that provides **no guarantee** of near-optimal task reward. Inspired by work from safe reinforcement learning domain—where constrained Markov decision processes (CMDPs) are commonly used to enforce safety constraints [20, 21]—we propose to apply a similar formulation to "safely" learning from imperfect demonstrations. Accordingly, we introduce **ConsMimic**, a CMDP-based policy optimization framework that adaptively optimizes a stylistic imitation objective while rigorously keeping task performance above a user-set optimality threshold.

ConsMimic employs a multi-critic architecture [22] to independently estimate task-specific and style-specific values. These separate reward signals are adaptively combined using a self-adjustable Lagrangian multiplier, dynamically adjusting in response to observed violations of task optimality constraints [20]. Additionally, we propose a symmetry-augmented style reward formulation to mitigate mode collapse that often plagues adversarial imitation learning, particularly when demonstrations are poorly aligned with task objectives. We validate our approach in three increasingly complex settings: (1) goal-reaching tasks on a Franka arm; (2) velocity tracking on a quadruped; and (3) velocity tracking over challenging terrains on a humanoid. In each scenario, demonstration data imperfectly match task conditions, yet still provide essential stylistic cues, such as agile quadruped gaits or coordinated arm-leg movements for humanoid locomotion. Our experiments demonstrate that ConsMimic allows robots to automatically determine when and how to utilize partial demonstrations, effectively preserving stylistic behaviors without compromising overall task performance. Supplementary videos for this work are available at our website.

In summary, our main contributions are:

1. A CMDP-based policy optimization framework with a self-adjustable Lagrangian multiplier that explicitly enforces task optimality, enabling "safe" incorporation of stylistic cues from imperfect demonstrations.
2. A novel symmetry-augmented style reward formulation that effectively counteracts the mode collapse commonly induced by task-demo misalignments.
3. Comprehensive empirical validation on various robots through simulations and on ANYmal-D hardware, demonstrating effectiveness and generalization across various tasks and robotic platforms.

2 Related Work

Deep reinforcement learning has brought great advances in the quadruped locomotion domain, where agents trained with parallel sampling [23] and domain randomization [24] are capable of transferring the learned policies from the simulation to the real world without finetuning. These policies often focus on optimizing task-specific objectives, such as speed or stability. However, such objectives alone can lead to unnatural, jerky movements that reduce robustness and realism.

To address these limitations, researchers have turned to more expressive reward designs that encode motion priors. By shaping the reward to discourage undesirable behaviors and guide agents toward plausible gait patterns, methods such as those in [3, 5, 25] achieve more stable motion. Nevertheless, such priors are typically handcrafted and closely tied to the target task [26, 27, 28], which limits their generalizability. A natural step forward is to move beyond manually crafted priors by leveraging demonstrations. Using retargeted motion capture data, researchers have trained policies to mimic skills exhibited by humans or animals. These skills can then be composed or reused in downstream tasks [29, 30]. In particular, adversarial imitation frameworks such as AMP [16] and its successor ASE [31] have proven effective in combining motion imitation with task execution, allowing smooth, agile behaviors in quadrupeds [32, 33] and humanoids [34]. Further work [17] demonstrated that even highly dynamic motions, such as backflips, can be achieved with adversarial objectives, powered by frameworks like Wasserstein GANs [35].

Despite their success, these approaches implicitly assume that demonstration data will always help or at least not hinder task performance. In practice, this assumption does not hold: high-performing

policies often require generalization to diverse terrains and command ranges [24], while demonstrations may be limited, misaligned, or collected under drastically different conditions. Blindly following such data can negatively impact performance, especially when task requirements conflict with stylistic cues. Our work builds upon these insights and proposes a principled framework to reconcile this conflict. By adopting a CMDP-based policy optimization framework and using a self-adjustable Lagrangian multiplier to adjust imitation weight, we allow the agent to selectively learn from imperfect demonstrations without compromising its ability to learn robust task policies.

3 Preliminary

Crafting explicit reward functions for stylistic behaviors in robotics is often challenging due to complexity or sparsity, making direct imitation from expert demonstrations a practical alternative. One widely used technique is motion clip tracking, as introduced by Peng et al. [15]. This method rewards the agent based on how closely its trajectory aligns with a provided reference motion clip at each timestep:

$$r^s_{\text{track}} = \exp\left(-\sum_i w_i (s_i - \hat{s}_i)^2\right), \tag{1}$$

where s_i and $\hat{s}_i$ are subgroups of the agent and demonstration states, respectively, and w_i are weighting coefficients.

Although effective in structured settings, tracking-based methods struggle when demonstrations are unstructured or not synchronized with tasks. To improve generalization, adversarial methods like Adversarial Motion Priors (AMP) [16] employ a discriminator D_ϕ parameterized by ϕ trained to differentiate transitions from expert demonstrations and agent-generated transitions. The adversarial style reward is defined as:

$$r^s_{\text{adv}}(s_t, s_{t+1}) = \max\left(0, 1 - 0.25\left(D_\phi(\Phi(s_t), \Phi(s_{t+1})) - 1\right)^2\right), \tag{2}$$

where Φ denotes a feature selector and the discriminator parameters ϕ are optimized by minimizing the loss:

$$\begin{aligned}\arg\min_\phi \quad & \mathbb{E}_{(s,s')\sim d^M}\left[(D_\phi(\Phi(s), \Phi(s')) - 1)^2\right] + \mathbb{E}_{(s,s')\sim d^\pi}\left[(D_\phi(\Phi(s), \Phi(s')) + 1)^2\right] \\ & + \frac{w_{\text{gp}}}{2}\mathbb{E}_{(s,s')\sim d^M}\left[\|\nabla_\phi D_\phi(\phi)\|^2\big|_{\phi=(\Phi(s),\Phi(s'))}\right],\end{aligned} \tag{3}$$

where d^M and d^π denote distributions of demonstration and policy-generated data, respectively. The last term is a gradient penalty weighted by w_{gp} used to stabilize the training process. This adversarial framework promotes generalization to partial or imperfect demonstrations. In our work, we selectively apply tracking-based methods for manipulation tasks while adversarial imitation methods for locomotion tasks.

4 Approach

In this section, we present **ConsMimic**—a CMDP-based training pipeline for learning style-aware behaviors under task optimality constraints. An overview of the framework is illustrated in Fig. 1, and the complete pseudocode is provided in Algorithm 1.

Constrained Style Learning under Task Optimality. In robotic tasks that require simultaneous task completion and stylistic imitation, effectively balancing these objectives is crucial but challenging. Typically, we categorize the rewards into two distinct groups: a manually designed demonstration-independent task reward group r^g, and a demonstration-driven style reward group r^s. Formally, we model this scenario as an extended Markov Decision Process (MDP): $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R^g, R^s, \mu, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ denotes the action space, $P(s'|s,a)$ represents state transition probabilities, and reward functions $R^g, R^s : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ output scalar task and style rewards, respectively. μ defines the initial state distribution, and $\gamma \in (0,1)$

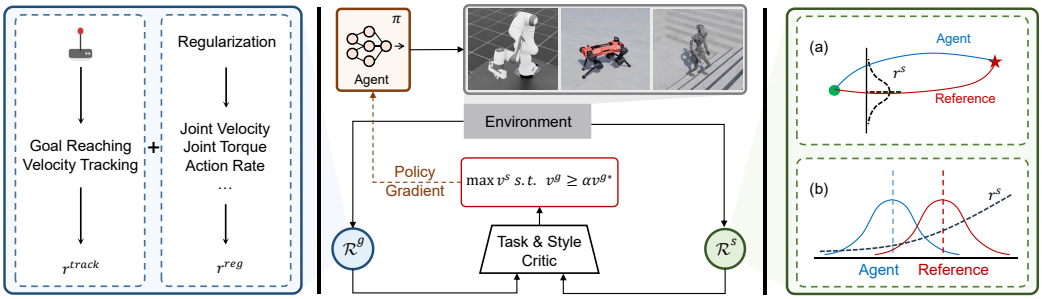

Figure 1: **ConsMimic Overview.** Given a reference dataset, ConsMimic calculates style reward using either (a) motion clip tracking or (b) adversarial imitation learning methods. Such style reward (R^s) are combined with task reward (R^g) within a constrained optimization framework illustrated in the red frame. Separate critic networks estimate task and style advantages, which are subsequently weighted by a self-adjustable Lagrangian multiplier and finally used to optimize the policy.

is the discount factor. The objective is to find a policy π maximizing the expected cumulative discounted rewards from both reward groups:

$$J_\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(r^g(s_t, a_t, s_{t+1}) + r^s(s_t, a_t, s_{t+1}) \right) \right]. \tag{4}$$

Previous work typically employs hand-tuned weighted combinations of task and style rewards as the optimization objective. However, selecting appropriate weights can be time-consuming and provides no explicit guarantee of near-optimal task performance. To address this, we propose a CMDP-based policy optimization framework that guides the agent in determining *when and how much* to extract stylistic cues from demonstration data while maintaining strong task performance. By explicitly incorporating a task optimality constraint, our approach ensures that the learned policy remains near-optimal. Formally, we define the CMDP as follows:

$$\max_\theta \; v^s(\pi_\theta) \quad \text{subject to} \quad v^g(\pi) \geq \alpha v^{g\star}, \tag{5}$$

where $v^s(\pi_\theta)$ is the expected style value under policy π_θ, $v^g(\pi_\theta)$ is the expected task value, and $v^{g\star}$ denotes the optimal achievable task performance. The parameter $\alpha \in [0, 1]$ specifies the threshold for acceptable task performance relative to optimality.

In practice, we reduce the CMDP in Eq. (5) to a regular MDP using Lagrangian method [36, 37], resulting in the dual problem given by:

$$\min_{\lambda \geq 0} \max_\theta \mathcal{L}(\theta, \lambda) = v^s(\pi_\theta) + \lambda \left(v^g(\pi_\theta) - \alpha v^{g\star} \right), \tag{6}$$

where λ is the Lagrangian multiplier adaptively balancing task and style learning. To solve Eq. (6), we alternatively optimize θ and λ. The optimization with respect to λ can be expressed as

$$\min_{\lambda \geq 0} \lambda \left(v^g(\pi) - \alpha v^{g\star} \right). \tag{7}$$

Intuitively, if task performance falls below the specified threshold, λ increases, thereby emphasizing task objectives. In contrast, when task performance meets or exceeds the threshold, λ decreases, allowing greater focus on style learning. Following previous work [37, 38] that trades exact saddle-point optimality for empirical stability, we introduce a bounded multiplier via a sigmoid activation on normalized advantages to stabilize Proximal Policy Optimization (PPO) [39] process, corresponding to the optimization of policy parameters θ:

$$A = \sigma(\lambda) \tilde{A}^g + (1 - \sigma(\lambda)) \, \tilde{A}^s, \tag{8}$$

where $\sigma(\cdot)$ denotes the sigmoid function, and $\tilde{A}^g$, $\tilde{A}^s$ represent normalized task and style advantages, respectively, computed via separate critic networks with Generalized Advantage Estimation

(GAE) [40]. This bounded multiplier approach ensures stable training dynamics and adaptively balances the trade-off between task and stylistic objectives throughout the training process.

Online Update of the Task Constraint. Although we can balance task and style objectives via a CMDP formulation, specifying an "oracle" optimal baseline $v^{g\star}$ beforehand is nontrivial. An overly optimistic oracle makes the constraint infeasible, while an underestimate yields suboptimal final performance. To avoid this, we introduce a *warm-up* phase with the imitation weight set to zero, letting the policy optimize only the task reward; the statistical average task value of the converged warm-up policy seeds an initial v_g^*. During subsequent joint training, we monotonically update $v^{g\star}$ based on the best statistical task value $v^g(\pi)$ ever seen along the training process:

$$v^{g\star} \leftarrow \max\Big(v^{g\star},\ v^g(\pi)\Big). \tag{9}$$

Since $v^{g\star}$ reflects empirically demonstrated performance, constraint $v^g(\pi) \geq \alpha\, v^{g\star}$ remains feasible throughout the training. Moreover, once the policy exceeds the baseline, it must maintain at least a fraction α of the new best value of the task. This ensures near-optimal task performance is preserved when style imitation has positive influence over the task performance.

Symmetric Augmented Style Learning. Symmetry is fundamental in robotic tasks, such as balanced locomotion with coordinated arm movements of humanoids, as it ensures harmonious and robust motion. However, adversarial imitation learning often struggles to capture these symmetric patterns due to the common issue of mode collapse in GAN training [35]. This problem becomes particularly severe when the demonstration data are not well aligned with the task objectives, causing the discriminator to dominate, reducing the informativeness of its feedback. In our framework, this issue is primarily manifested by the policy repeatedly reproducing only a sub-segment of the periodic locomotion cycle, rather than capturing the full range of motion typically observed in natural locomotion patterns. The symmetry-augmentation method at policy optimization level proposed by Mittal et al. [41] cannot effectively prevent this in this particular case due to the inherent asymmetry coming from the reward function. To address this, we inject symmetry directly into the reward. We define robot-specific symmetry transformation operators L_g, augmenting both demonstration and policy-generated data during training:

$$\mathcal{B}_{\text{sym}} = \mathcal{B} \cup \bigcup_{g \in G} L_g(\mathcal{B}), \tag{10}$$

where G is the set of symmetry transformations specific to the robot morphology, and $L_g(\mathcal{B}) = \{L_g(s, s') \mid (s, s') \in \mathcal{B}\}$. To reinforce symmetry in the learned policy explicitly, we compute the symmetry-augmented style reward by averaging discriminator outputs over all mirrored transitions:

$$r^s_{\text{sym}}(s_t, s_{t+1}) = \frac{1}{|G|+1}\left[r^s_{\text{adv}}(s_t, s_{t+1}) + \sum_{g \in G} r^s_{\text{adv}}(L_g(s_t, s_{t+1}))\right]. \tag{11}$$

This generalized symmetry-aware formulation directly embeds symmetry constraints into the adversarial training objective, effectively counteracting biases resulting from partial, asymmetric demonstrations. By explicitly guiding the discriminator and the associated reward signal toward recognizing and enforcing symmetric behaviors, our method significantly enhances the robustness and efficiency of style generalization, leading to improved imitation quality even under challenging demonstration conditions.

5 Experiments and Results

Our experiments aim to answer whether ConsMimic (i) achieves higher imitation quality while ensuring task optimality compared to task-only training and two fixed-weight baselines, (ii) improves motion symmetry using symmetry augmented style reward formulation, (iii) allows the task optimality threshold α to effectively control style–task trade-off, (iv) is beneficial to real-world applications.

Q1. Style Learning Quality. To rigorously validate ConsMimic, we designed comprehensive experiments across various robotic platforms and tasks that exhibit different degrees of misalignment

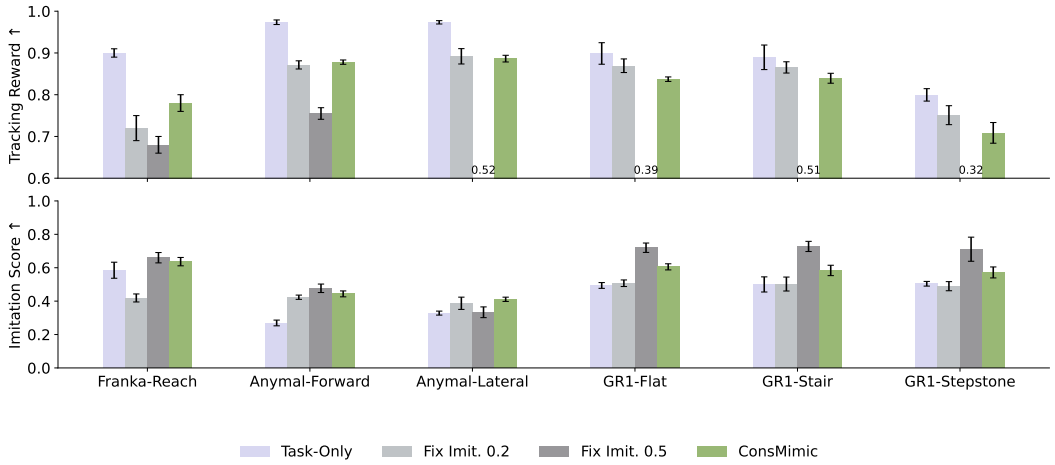

Figure 2: **Visualization Results across Tasks.** We report the mean and standard deviation over 5 seeds. The top row shows the tracking reward for each method. ConsMimic achieves task rewards comparable to the task-only baseline (ω_0^s), demonstrating its ability to enforce near-optimal task performance. In contrast, the baseline with an aggressive imitation weight ($\omega_{0.5}^s$) struggles to learn how to complete the task. The bottom row presents the imitation scores. ConsMimic consistently outperforms all baselines that are capable of solving the tasks and only trails behind $\omega_{0.5}^s$, which achieves higher imitation at the cost of degraded task performance.

between the demonstration tasks. Specifically, we evaluated performance on: (1) Franka-Reach, where the agent must reach a goal efficiently, while demonstrations follow stylistically sinusoidal trajectories; (2) Anymal-Forward and Anymal-Lateral tasks, where the ANYmal-D quadruped robot tracks forward or lateral velocities despite demonstrations consisting primarily of forward-trotting motions; and (3) GR1-Flat, GR1-Stair, and GR1-Stepstone, where a whole body humanoid robot GR1 is commanded to track velocities over diverse terrains while the reference motion is collected on flat ground.

We compare ConsMimic (ω_{adapt}^s) against two main baselines: (a) a task only baseline (ω_0^s), which completely ignores stylistic imitation; and (b) fixed-weight baselines ($\omega_{0.5}^s$, $\omega_{0.2}^s$), that indicate that the style reward contribute 0.5 and 0.2 times the total reward group weight, respectively. We report detailed task formulations and task reward compositions in Appendix A.

We report both achieved task performance and imitation quality across all six tasks in Fig. 2. For clarity and comparability, imitation scores are defined as:

$$S_{\text{imit}} = \max\{0, 1 - \text{DTW}(\tau^\pi, \tau^M)/\eta\}, \tag{12}$$

where DTW represents dynamic time warping [42], typically measuring distances between temporal sequences. τ^π and τ^M denote policy-generated and demonstration trajectories, respectively, and η is a task-dependent normalization constant (e.g., $\eta = 20$ for manipulation, $\eta = 100$ for locomotion tasks). This formulation ensures that the scores are in the range $[0, 1]$.

As shown in Fig. 2, ConsMimic consistently achieves an effective balance between task performance and imitation quality. Specifically, in Franka-Reach, ConsMimic obtains a high task reward (0.78) and demonstrates significantly better imitation quality compared to the task-only baseline. In quadruped tasks, particularly Anymal-Lateral where style-task misalignment is prominent, ConsMimic achieves high task performance close to the task-only baseline while substantially surpassing fixed-weight baselines in imitation quality, highlighting its capacity to effectively leverage stylistic cues without compromising task objectives. For humanoid tasks, fixed-weight baselines either sacrifice task completion capability or experience a significant decline in stylistic imitation. For instance, $\omega_{0.5}^s$ demonstrates strong imitation but fails in complex scenarios like GR1-Stair and GR1-Stepstone, whereas ω_0^s maintains task performance but consistently scores lower in imitation quality. ConsMimic's adaptive strategy consistently delivers better generalization by dynamically balancing

style imitation and task requirements, enabling stable training and robust performance under realistic task-demo misalignments. We provide details for DTW calculation in Appendix C.

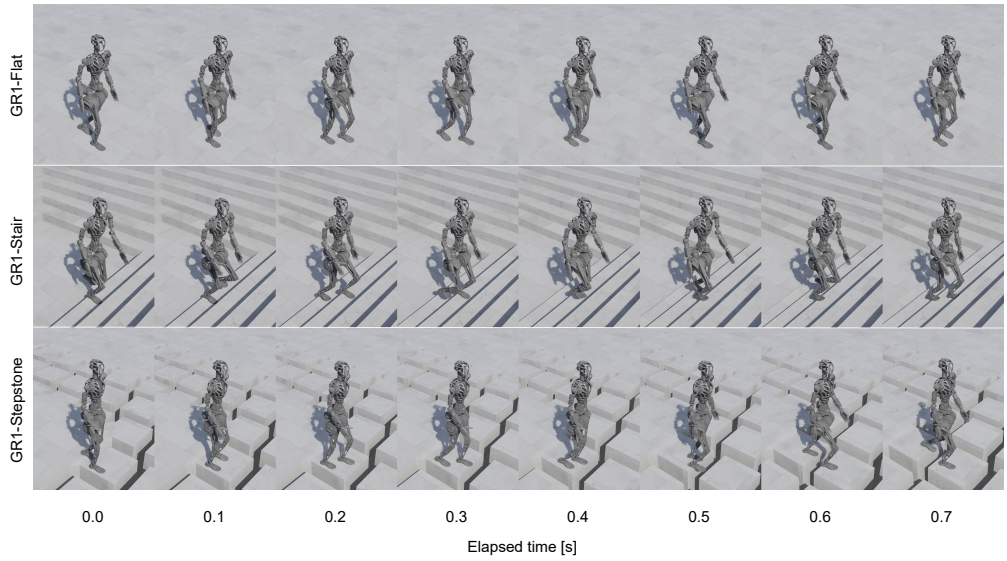

Figure 3: **Visualization Results of ConsMimic on GR1.** With symmetric augmented style learning, GR1 achieved symmetric and natural motion on both flat ground (in distribution) and stair & stone ground (out of distribution).

Q2. Symmetry-Augmented Style Learning. To assess whether our symmetry-augmented reward improves motion symmetry, we measure a symmetry score calculated based on DTW distances between trajectories and their mirrored counterparts:

$$S_{\text{sym}} = \max\{0, 1 - \frac{1}{|G|} \sum_{g \in G} \frac{\text{DTW}(\tau^\pi, L_g(\tau^\pi))}{\eta}\}, \tag{13}$$

where G is the set of predefined symmetry transformations, L_g applies transformation g to trajectory τ^π, and η is again a task-dependent normalization constant.

As shown in Table 1, incorporating our symmetry-augmented reward significantly improves the symmetry score across all GR1 locomotion tasks. This shows the effectiveness of our method on symmetric policy learning even from non-expert demonstrations. Our formulation enables the policy to generalize symmetric motion patterns across various terrains while preserving task optimality. We present visualization results of ConsMimic in GR1 tasks as Fig. 3.

Task	Ours (w/o sym aug)	Ours (w/ sym aug)
GR1-Flat	$0.779_{\pm 0.021}$	$0.814_{\pm 0.018}$
GR1-Stair	$0.741_{\pm 0.025}$	$0.811_{\pm 0.020}$
GR1-Stepstone	$0.642_{\pm 0.030}$	$0.722_{\pm 0.022}$

Table 1: **Symmetry Analysis.** Symmetry scores S_{sym} calculated by Eq. (13) with $\eta = 100$ over 5 seeds. ConsMimic consistently improves symmetry by a large margin.

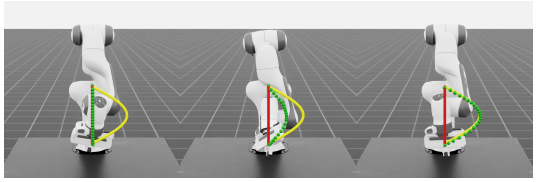

Figure 4: **Visualization of α's Effect on Franka-Reach.** Shown are trajectories for $\alpha = 1.0$ (left), $\alpha = 0.9$ (middle), and $\alpha = 0.8$ (right). The red line indicates the optimal task trajectory, the yellow line is the demonstration trajectory, and the green line shows our policy's trajectory.

Q3. Effectiveness of α. To assess whether α can effectively control the level of task optimality, we conducted experiments on the Franka-Reach task with α set to 0.8, 0.9, and 1.0. As shown in Fig. 4,

Figure 5: **Visualization of ANYmal-D's Locomotion in the Real World.** The top row show motions produced by policy trained by conventional task rewards while the bottom row trained by ConsMimic. ConsMimic enables the robot to achieve a more natural, agile trotting gait pattern.

when $\alpha = 1.0$, the policy completely disregards the demonstration's influence, strictly enforcing near-optimal task performance at the expense of stylistic cues. With $\alpha = 0.9$, the policy achieves high task performance while still incorporating essential stylistic features. When $\alpha = 0.8$, the agent fully assimilates the demonstration, yet still meets the required task constraints. These results demonstrate that α is an effective parameter for modulating the balance between task execution and style imitation.

Q4. ConsMimic on Real-World Tasks. We further validate our framework on the ANYmal-D quadruped. The policy, trained on the Anymal-Forward task, is deployed on the ANYmal-D hardware in a zero-shot manner. The robot is commanded to move forward at a speed of 2 m/s and then return at the same speed. It completes 8 rounds with identical distances.

We analyze the motion styles produced by ConsMimic relative to the provided reference motion dataset. As illustrated in Figure 5, policies trained within the ConsMimic framework achieve more natural and agile trotting gaits, a style difficult to learn solely from pure RL task rewards.

To quantitatively evaluate the quality of the motion, we measure two key metrics: the mechanical work (W_{mech}) done by the robot (computed as $\sum \tau \cdot \dot{\theta}$ per episode), and the average foot-air time (FAT, T_{air}) per step. As shown in Table 2, the ConsMimic-trained controller demonstrates superior energy efficiency and greater dynamism, characterized by lower mechanical energy consumption and increased foot-air time. The quantification results confirm that ConsMimic effectively converts stylistic motion imitation into tangible real-world performance enhancements, highlighting its practical applicability and robustness in real-world robotic locomotion tasks.

Metric	Task-Only	ConsMimic
W_{mech} (J)	1337 ± 515	1143 ± 450
T_{air} (s)	0.28 ± 0.02	0.37 ± 0.04

Table 2: **Motion Analysis.** Policy trained with ConsMimic demonstrates lower energy usage and more dynamic motion.

6 Conclusion

In this work, we introduced ConsMimic, a novel CMDP-based style learning framework designed to explicitly enforce task optimality while learning from imperfect demonstrations. Our method introduces a self-adjustable Lagrangian multiplier to automatically balance the trade-off between style learning and task learning and leverage symmetry augmented style reward formulation to extract symmetric patterns from motion reference. We validate our methods across robot platforms in simulator and real world. The experimental results show ConsMimic's potential as a practical and generalizable approach for real-world robotic style synthesizing tasks especially when expert demonstration is hard to access.

7 Limitation

Despite the promising results, our approach has limitations. Specifically, the framework does not explicitly discriminate between beneficial and detrimental features within demonstrations. Even when overall demonstrations are flawed or misaligned, there might still be valuable stylistic cues worth extracting. Future work should explore techniques to selectively identify and leverage beneficial demonstration features, potentially incorporating mechanisms such as attention or feature weighting to enhance the robustness and adaptability of imitation learning.

Acknowledgments

This research was supported by the ETH AI Center and the Swiss National Science Foundation through the National Centre of Competence in Automation (NCCR automation).

References

[1] R. S. Sutton, A. G. Barto, et al. Reinforcement learning. *Journal of Cognitive Neuroscience*, 11(1):126–134, 1999.

[2] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

[3] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[4] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[5] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62):eabk2822, 2022.

[6] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[7] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal. Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research*, 43(4):572–587, 2024.

[8] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[9] F. Jenelten, J. He, F. Farshidian, and M. Hutter. Dtc: Deep tracking control. *Science Robotics*, 9(86):eadh5401, 2024.

[10] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, and M. Panne. Learning locomotion skills for cassie: Iterative design and sim-to-real. In *Conference on Robot Learning*, pages 317–329. PMLR, 2020.

[11] J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning. *arXiv preprint arXiv:2105.08328*, 2021.

[12] H. Duan, B. Pandit, M. S. Gadde, B. Van Marum, J. Dao, C. Kim, and A. Fern. Learning vision-based bipedal locomotion for challenging terrain. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 56–62. IEEE, 2024.

[13] J. Siekmann, Y. Godse, A. Fern, and J. Hurst. Sim-to-real learning of all common bipedal gaits via periodic reward composition. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7309–7315. IEEE, 2021.

[14] S. Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

[15] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

[16] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4): 1–20, 2021.

[17] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius. Learning agile skills via adversarial imitation of rough partial demonstrations. In *Conference on Robot Learning*, pages 342–352. PMLR, 2023.

[18] C. Li, S. Blaes, P. Kolev, M. Vlastelica, J. Frey, and G. Martius. Versatile skill control via self-supervised adversarial imitation of unlabeled mixed motions. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2944–2950. IEEE, 2023.

[19] C. Li, E. Stanger-Jones, S. Heim, and S. Kim. Fld: Fourier latent dynamics for structured motion representation and learning. *arXiv preprint arXiv:2402.13820*, 2024.

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

[21] J. Garcıa and F. Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[22] S. Mysore, G. Cheng, Y. Zhao, K. Saenko, and M. Wu. Multi-critic actor learning: Teaching rl policies to act with style. In *International Conference on Learning Representations*, 2022.

[23] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[24] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[25] Y. Shao, Y. Jin, X. Liu, W. He, H. Wang, and W. Yang. Learning free gait transition for quadruped robots via phase-guided controller. *IEEE Robotics and Automation Letters*, 7(2): 1230–1237, 2021.

[26] G. B. Margolis and P. Agrawal. Walk these ways: Tuning robot control for generalization with multiplicity of behavior. In *Conference on Robot Learning*, pages 22–31. PMLR, 2023.

[27] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li. Multi-expert learning of adaptive legged locomotion. *Science Robotics*, 5(49):eabb2174, 2020.

[28] Y. Hu, K. Wen, and F. Yu. Dexdribbler: Learning dexterous soccer manipulation via dynamic supervision. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12910–12917. IEEE, 2024.

[29] S. Bohez, S. Tunyasuvunakool, P. Brakel, F. Sadeghi, L. Hasenclever, Y. Tassa, E. Parisotto, J. Humplik, T. Haarnoja, R. Hafner, et al. Imitate and repurpose: Learning reusable robot movement skills from human and animal behaviors. *arXiv preprint arXiv:2203.17138*, 2022.

[30] L. Han, Q. Zhu, J. Sheng, C. Zhang, T. Li, Y. Zhang, H. Zhang, Y. Liu, C. Zhou, R. Zhao, et al. Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models. *Nature Machine Intelligence*, 6(7):787–798, 2024.

[31] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

[32] A. Escontrela, X. B. Peng, W. Yu, T. Zhang, A. Iscen, K. Goldberg, and P. Abbeel. Adversarial motion priors make good substitutes for complex reward functions. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 25–32. IEEE, 2022.

[33] J. Wu, G. Xin, C. Qi, and Y. Xue. Learning robust and agile legged locomotion using adversarial motion priors. *IEEE Robotics and Automation Letters*, 8(8):4975–4982, 2023.

[34] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba. Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13107–13114. IEEE, 2024.

[35] J. Adler and S. Lunz. Banach wasserstein gan. *Advances in neural information processing systems*, 31, 2018.

[36] V. S. Borkar. An actor-critic algorithm for constrained markov decision processes. *Systems & control letters*, 54(3):207–213, 2005.

[37] T. Zahavy, Y. Schroecker, F. Behbahani, K. Baumli, S. Flennerhag, S. Hou, and S. Singh. Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521*, 2022.

[38] J. Cheng, M. Vlastelica, P. Kolev, C. Li, and G. Martius. Learning diverse skills for local navigation under multi-constraint optimality. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5083–5089. IEEE, 2024.

[39] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[40] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[41] M. Mittal, N. Rudin, V. Klemm, A. Allshire, and M. Hutter. Symmetry considerations for learning task symmetric robot policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7433–7439. IEEE, 2024.

[42] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370, 1994.

A Task Representation

A.1 Franka

Observation and Action Spaces. The observation space for Franka consists of joint positions, joint velocities, the end-effector's target pose, and the policy's previous actions. The action space comprises the target joint positions. The demonstration trajectory is represented by end-effector poses, as detailed in Table 3.

Table 3: Observation, action, and demonstration spaces for the Franka arm

Category	Entry	Symbol	Dimension
Observation	Joint positions	q	7
	Joint velocities	$\dot{q}$	7
	End-effector position target	p^*	3
	End-effector orientation target	θ^*	4
	Policy's last actions	a_{t-1}	7
Action	Target joint positions	q^*	7
Demonstration	End-effector poses	x_t	7

Reward Formulation. The task reward includes end-effector position and orientation tracking terms, action smoothness and regularization penalties, and a style imitation term. The style reward is defined as a demonstration trajectory tracking loss (see Eq. 1). Table 4 summarizes the weights for each term.

Table 4: Task reward terms for the Franka arm

Term	Function	Weight
EE tracking (coarse)	$\|\|x - x^*\|\|_2$	-0.5
EE tracking (fine-grained)	$1 - \tanh(\|\|x - x^*\|\|_2)$	1.0
EE orientation tracking	$\|\|\theta - \theta^*\|\|_{\text{quat}}$	-0.1
Action rate penalty	$\|\|a_t - a_{t-1}\|\|_2^2$	-0.01
Joint velocity penalty	$\|\|\dot{q}\|\|_2^2$	-0.01

Task Definition. In the Franka-Reach task, the robot's end effector is commanded to reach a target pose where the position lies within $x \in [0.35, 0.45]$, $y \in [-0.05, 0.05]$, $z \in [0.20, 0.30]$ and the pitch angle satisfies $\theta_{\text{pitch}} \in [-\pi, \pi]$, all defined in the local frame.

A.2 Anymal

Observation and Action Spaces. The observation space for Anymal includes base velocities in the local frame, gravity projection, command inputs, joint positions, joint velocities, and past actions. The action space consists of target joint positions. The demonstration state space includes base velocities (local), gravity projection, joint positions, and joint velocities. These are detailed in Table 5.

Table 5: Observation, action, and demonstration spaces for Anymal

Category	Entry	Symbol	Dimension
Observation	Base linear velocity (local)	$v_{\text{lin}}^{\text{base}}$	3
	Base angular velocity (local)	$v_{\text{ang}}^{\text{base}}$	3
	Projected gravity	g_{proj}	3
	Velocity commands	v_{cmd}	3
	Joint positions	q	12
	Joint velocities	$\dot{q}$	12
	Previous actions	a_{t-1}	12
Action	Target joint positions	q^*	12
Demonstration	Base linear velocity (local)	$v_{\text{lin}}^{\text{base}}$	3
	Base angular velocity (local)	$v_{\text{ang}}^{\text{base}}$	3
	Projected gravity	g_{proj}	3
	Joint positions and velocities	$[q, \dot{q}]$	24

Reward Formulation. The task reward for Anymal includes tracking of commanded base velocities, as well as penalties on vertical motion, joint effort, energy consumption, flatness of base orientation, joint limit violations, and undesired contacts. The style reward is predicted according to Eq. 11. All reward terms and their weights are listed in Table 6.

Table 6: Reward terms for Anymal velocity tracking

Term	Function	Weight
Track linear velocity (xy)	$\exp\left(-\frac{\|\|v_{xy}-v_{xy}^*\|\|^2}{\sigma^2}\right)$	1.0
Track angular velocity (z)	$\exp\left(-\frac{\|\|\omega_z-\omega_z^*\|\|^2}{\sigma^2}\right)$	0.5
Vertical linear velocity penalty	$\|\|v_z\|\|_2^2$	-2.0
Angular velocity penalty (xy)	$\|\|\omega_{xy}\|\|_2^2$	-0.05
Joint torque penalty	$\|\|\tau\|\|_2^2$	$-2.5\text{e}{-5}$
Joint acceleration penalty	$\|\|\ddot{q}\|\|_2^2$	$-2.5\text{e}{-7}$
Action rate penalty	$\|\|a_t - a_{t-1}\|\|_2^2$	-0.01
Power consumption	$\sum \tau \cdot \dot{q}$	$-5\text{e}{-5}$
Feet air time reward	$\mathbb{1}_{t_{\text{air}}>0.5}$	0.125
Undesired contacts (thigh)	$\mathbb{1}_{\text{contact}}$	-1.0
Flat orientation penalty	$\|\|g_{b,xy}\|\|_2^2$	-5.0
Joint limit violation penalty	$\sum_i [\max(0, q_i - q_{\max,i}, q_{\min,i} - q_i)]$	-1.0

Task Definition. In the Anymal-Forward task, the ANYmal robot is commanded to follow linear velocity commands in the body frame with $v_x \in [-3.0, 3.0]$ and angular velocity $\omega_z \in [-1.0, 1.0]$. In the Anymal-Lateral task, the robot is commanded to follow $v_y \in [-2.0, 2.0]$ and $\omega_z \in [-1.0, 1.0]$, also in the body frame.

A.3 GR1

Observation and Action Spaces. The observation space for GR1 includes base velocities in the local frame, projected gravity, command inputs, joint positions, joint velocities, past actions, and exteroceptive height scanning. The action space consists of target joint positions across legs, torso, shoulders, and elbows. The AMP demonstration state includes base motion, joint states, and foot positions. Details are summarized in Table 7.

Table 7: Observation, action, and demonstration spaces for GR1

Category	Entry	Symbol	Dimension
Observation	Base linear velocity	$v_{\text{lin}}^{\text{base}}$	3
	Base angular velocity	$v_{\text{ang}}^{\text{base}}$	3
	Projected gravity	g_{proj}	3
	Velocity commands	v_{cmd}	3
	Joint positions	q_{rel}	23
	Joint velocities	$\dot{q}_{\text{rel}}$	23
	Previous actions	a_{t-1}	23
	Height scan	h_{scan}	173
Action	Target joint positions	q^*	23
Demonstration	Base linear velocity	$v_{\text{lin}}^{\text{base}}$	3
	Base angular velocity (local)	$v_{\text{ang}}^{\text{base}}$	3
	Joint positions	q	23
	Joint velocities	$\dot{q}$	23
	Foot positions in local frame	p_{foot}	12

Reward Formulation. GR1's reward function combines task-level tracking, joint-level regularization, physical constraints, and biped-specific behavior shaping. It includes linear and angular velocity tracking, penalties on torque and joint deviation, foot-ground interaction shaping, and termination penalties. Table 8 lists the main reward terms and weights.

Table 8: Reward terms for GR1 rough terrain locomotion

Term	Function	Weight
Termination penalty	$\mathbb{1}_{\text{terminate}}$	-200.0
Track linear velocity (xy)	$\exp(-\|\|v_{xy} - v_{xy}^*\|\|^2/\sigma^2)$	5.0
Track angular velocity (z)	$\exp(-\|\|\omega_z - \omega_z^*\|\|^2/\sigma^2)$	3.0
Action rate (arms/legs)	$\|\|a_t - a_{t-1}\|\|_2^2$	-0.01
Action rate (2nd order)	$\|\|a_t - 2a_{t-1} + a_{t-2}\|\|_2^2$	-0.005
Joint torque penalty	$\|\|\tau\|\|_2^2$	$-1\text{e}{-4}$
Torque limit violation	$\|\tau - \tau_{\text{applied}}\|$	-0.002
Joint deviation penalty	$\|\|q - q_{\text{ref}}\|\|_2^2$	-0.5
Feet air time reward	$\mathbb{1}_{t_{\text{air}}>0.4}$	1.0
Zero action (ankle roll)	$\mathbb{1}_{\|a\|>\epsilon} \cdot a^2$	-0.5
Joint limit violation	$\mathbb{1}_{q \notin [q_{\min}, q_{\max}]}$	-10.0
Power consumption	$\sum \tau \cdot \dot{q}$	$-5\text{e}{-6}$
Base angular velocity (xy)	$\|\|\omega_{xy}\|\|_2^2$	-0.05
Feet slide penalty	$\|\|v_{\text{slip}}\|\|$ when in contact	-1.0
No-fly penalty	$\mathbb{1}_{\text{both feet airborne}}$	-5.0
Pelvis orientation	$\|\|g_{\text{pelvis},xy}\|\|_2^2$	-5.0
Torso orientation	$\|\|g_{\text{torso},xy}\|\|_2^2$	-5.0

Task Definition. In all GR1 task, the GR1 robot is commanded to follow linear velocity commands in the body frame with $v_x \in [0.5, 2.0]$.

B Training Details

B.1 Training Pipeline

We detail our training pipeline in the following algorithm:

Algorithm 1 ConsMimic Training Pipeline

Require: Policy π, task critic v^g, style critic v^s, discriminator D_ϕ, Lagrange multiplier λ, demonstrations $\mathcal{D}$, symmetry mappings G, threshold coefficient α, Learning iterations N, Constraint update intervals I_c
Initialize networks and rollout buffer $\mathcal{B}$
Set optimal task value v^g as the initial guess v^{g*}
for learning iteration $i = 1, 2, \cdots, N$ **do**
  for time step $t = 1, 2, \ldots, T$ **do**
    Collect transition $(s_t, a_t, s_{t+1}, r_t^g)$ using current policy π
    Compute symmetry-augmented style reward $r^s_{\text{sym},t}$ using Eq. (11)
    Store $(s_t, a_t, s_{t+1}, r_t^g, r^s_{\text{sym},t})$ in rollout buffer $\mathcal{B}$
  end for
  Compute TD targets for value updates
  Compute task advantage A^g and style advantage A^s using GAE
  Compute combined advantage using Eq. (8) ($\sigma(\lambda)$ is set to 1 during *warmup* phase)
  for learning epoch $= 1, 2, \ldots, K$ **do**
    Sample mini-batches $b \sim \mathcal{B}$
    Update policy π, task critic v^g, and style critic v^s using PPO
    Update discriminator D_ϕ using symmetry-augmented mini-batch b_{sym} via Eq. (3)
    Update Lagrange multiplier λ using Eq. (6)
    if $i \bmod I_c = 0$ **then**
      Update constraint using Eq. 9
    end if
  end for
end for

B.2 Network Architecture

The policy network, value network, discriminator network all consist of MLP layers, which is detailed in Table 9

Table 9: Network architectures used for each task

Task	Policy	Value	Discriminator
Franka	[64, 64]	[64, 64]	–
Anymal	[512, 256, 128]	[512, 256, 128]	[1024, 512]
GR1	[512, 256, 128]	[512, 256, 128]	[1024, 512]

B.3 Training Parameters

Table 10: ConsMimic training parameters.

Parameter	Value	Parameter	Value
Num Steps per Environment	24	Training Iterations	20000
clip range	0.2	entropy coef	0.005
mini batches	4	learning rate	1e-3
discount factor	0.99	α for Franka	0.9
α for Anymal	0.7	α for GR1	0.9

C Evaluation Details

C.1 Terrain for Evaluation

We evaluate GR1 locomotion performance under two challenging terrain settings: *Stairs* and *Stepping Stones*, each defined using custom terrain generator configurations in Isaac Lab. Key parameters and sub-terrain definitions are summarized below.

Stairs. The stair terrain consists of pyramid-style inverted stairs with varying step heights. The step height ranges from 0.05 to 0.27 meters; step width ranges from 0.30 to 0.40 meters.

Stepping Stones. This terrain contains high-frequency stepping-stone terrain with variable widths and distances. Stone height is up to 0.01 meters, width ranges from 0.55 to 1.0 meters; distance ranges from 0.1 to 0.2 meters.

C.2 Imitation & Symmetry Score Calculation

We use dynamic time warping (DTW) to evaluate the distance between the policy-generated trajectory and the demonstration trajectory. The standard DTW implementation requires alignment of the start and end of two trajectories, which is impratical in our settings. We relax this constraint in our implementation as specified in Algorithm 2

Algorithm 2 Relaxed DTW Distance

Require: Two sequences: $\text{seq}_1 \in \mathbb{R}^{n \times d}$ and $\text{seq}_2 \in \mathbb{R}^{m \times d}$
Initialize DTW matrix: $D \in \mathbb{R}^{(n+1)\times(m+1)}$ with $D[0,:] \leftarrow 0$, $D[:,0] \leftarrow \infty$
for $i = 1$ to n **do**
  for $j = 1$ to m **do**
    $c \leftarrow \|\text{seq}_1[i] - \text{seq}_2[j]\|_2$
    $D[i,j] \leftarrow c + \min\{D[i-1,j], D[i,j-1], D[i-1,j-1]\}$
  end for
end for
return $\min_j D[n,j]$ {Relax end-alignment by taking minimal cost across final row}

Note that we use end effector trajectory for Franka task while joint position trajectory for Anymal and GR1 tasks.