PHYSIBLEMDM: ENFORCING INFERENCE-TIME PHYSICAL FEASIBILITY CONSTRAINTS ON MOTION DIFFUSION MODELS VIA NOISE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Motion Diffusion Models (MDMs) can generate diverse and visually appealing human motion; yet, the resulting trajectories frequently violate physical constraints, limiting their downstream use in simulation and control. We present *PhysibleMDM*, a framework that promotes physical feasibility of MDM output by optimizing at *inference time* the diffusion noise *without* retraining the generative model. We introduce a set of surrogate feasibility constraints that can be evaluated without full trajectory simulation. These constraints combine kinematic checks (e.g., ground penetration, slip, and self-collision) with dynamic feasibility tests computed via an inverse-dynamics inner loop formulated as a cone program. We solve the resulting black-box noise optimization problem using the Separable Covariance Matrix Adaptation Evolution Strategy (Sep-CMA-ES), thereby avoiding reliance on computationally expensive and discontinuous gradients. An autoregressive scheme then chains optimized segments to synthesize long motion sequences. Across 100 motion styles on an SMPL humanoid, PhysibleMDM reduces mean ground penetration, self-collision penetration, and ground slip by 75%, 92%, and 74%, respectively, relative to unoptimized diffusion output. Across 11 sequences for a custom bipedal robot, it achieves median reductions of 89.5% in per-sequence mean acceleration-reconstruction error and 79.5% in the mean inverse-dynamics residual.

1 INTRODUCTION

Motion Diffusion Models (MDMs) (Tevet et al., 2023; Zhang et al., 2024; Kim et al., 2023; Rempe et al., 2026) currently achieve state-of-the-art performance in synthesizing human-like motion for physics-based characters and robots. These generators are typically trained on large corpora of human motion capture data and can be conditioned on various signals, including text, style, goals, and semantic tasks (Dabral et al., 2023; Chen et al., 2024a; Huang et al., 2025; Zhao et al., 2026). The generated motion commonly serves as a reference for Reinforcement Learning (RL)-based trackers (Peng et al., 2018; Luo et al., 2023; Serifi et al., 2024b; Liao et al., 2026; Luo et al., 2026). Errors can arise at multiple stages of this pipeline. In particular, generated motions may exhibit *physical artifacts* that hinder their execution. *Kinematic* artifacts include joint-limit violations (Lee et al., 2025), self-collisions (Khazoom et al., 2022), ground penetration (Yuan et al., 2023), undesired ground slip (Pan et al., 2025), and foot placement in non-traversable regions (Nguyen et al., 2016). *Dynamic* artifacts include motions requiring joint torques beyond actuator limits (Herzog et al., 2014) or contact forces outside friction cones (Grandia et al., 2021), as well as trajectories inconsistent with the dynamics of unactuated coordinates, such as the floating base of a legged robot (Herzog et al., 2014; Grizzle et al., 2014). These artifacts degrade downstream tracking performance (Araújo et al., 2026; Zhang et al., 2026), as infeasible reference motions cannot be reliably tracked.

Several methods mitigate these artifacts before or during *training*, for example, by learning from motions tracked in physics simulation (Wu et al., 2025; Watanabe et al., 2025; Xu et al., 2025), or fine-tuning the generator with physics-informed objectives (Serifi et al., 2024a; Zhang et al., 2026). While effective in practice, these approaches can be resource-intensive, particularly when generative models and tracking policies are retrained across refinement iterations, and they do not inherently guarantee that all samples from the final diffusion model satisfy the desired physical constraints.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

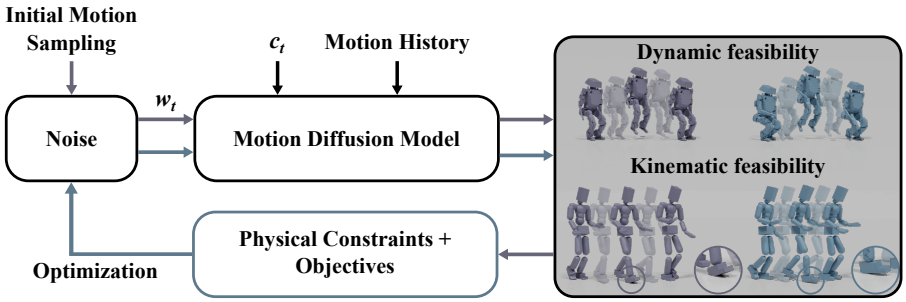


Figure 1: **Overview of PhysibleMDM.** We improve physical feasibility by optimizing the diffusion noise of a frozen motion generator. Kinematic constraint violations and constrained inverse-dynamics residuals serve as physical-feasibility proxies, guiding refinement without simulator rollouts. Initial (purple) and optimized (blue) motions illustrate improvements in dynamic feasibility (top) and kinematic feasibility (bottom). Optimized motion history conditions subsequent generation.

Inference-time optimization incorporates explicit objectives and constraints during generation without retraining the diffusion model. Common approaches steer intermediate denoising states through gradient guidance (Dhariwal & Nichol, 2021; Karunratanakul et al., 2023) or projection onto feasible sets (Yuan et al., 2023; Xiao et al., 2025; Watanabe et al., 2026). Noise optimization offers an alternative in which objectives and constraints are evaluated on fully denoised motions (Karunratanakul et al., 2024; Zhao et al., 2025). Gradient-based noise optimization requires costly backpropagation through both the diffusion sampler and the physical-feasibility evaluation. Contact-rich motion further introduces non-smooth objectives and constraints for which gradients may be poorly informative (Antonova et al., 2023; Paulus et al., 2026). Sampling-based optimization has shown promise for optimal control of contact-rich legged locomotion (Xue et al., 2025; Alvarez-Padilla et al., 2025; Schramm et al., 2026). Recent work also applies sampling-based optimization to diffusion noise for image generation and protein design (Kim et al., 2026; Tan et al., 2026), but does not address the kinematic and dynamic feasibility requirements of robotic motion. Efficient gradient-free optimization of initial and posterior diffusion noise for these requirements, while retaining characteristic motion style, remains unexplored.

To address this gap, we propose PhysibleMDM, a gradient-free method that jointly optimizes initial and posterior diffusion noise at inference time. Parallel forward passes through the frozen model generate a population of motions for evaluating task objectives, kinematic constraints, and constrained inverse-dynamics residuals. These evaluations guide Separable Covariance Matrix Adaptation Evolution Strategy (Sep-CMA-ES) (Ros & Hansen, 2008), whose diagonal covariance supports high-dimensional search, in iteratively refining the noise population. Noise standardization regularizes noise toward its training-time statistics to help limit deviations from the learned motion distribution. An autoregressive scheme chains optimized segments into long motion sequences.

In summary, the contributions of this paper are:

- A physical-feasibility formulation for MDMs that combines kinematic constraints with constrained inverse-dynamics residuals, enabling inference-time refinement without a tracking controller or full trajectory simulation.
- A scalable sampling-based method for high-dimensional noise optimization in MDMs (realized via Sep-CMA-ES), evaluating non-smooth objectives and constraints on fully denoised motions without costly gradient computation.
- A diffusion-noise optimization strategy, applicable to any pretrained MDM without architecture-specific changes, that combines joint initial and posterior noise optimization with noise standardization, balancing optimization flexibility against deviations from the model's learned distribution and supporting motion-style retention in MDMs.

2 Related Work

Inference-Time Optimization Guidance and projection steer diffusion sampling by modifying intermediate denoising states or their clean-sample estimates (Karunratanakul et al., 2023; Christopher et al., 2024; Bodmer et al., 2026). Noise optimization instead evaluates objectives on fully denoised

samples and adjusts the stochastic inputs, using gradients (Karunratanakul et al., 2024; Zhao et al., 2025) or sampling-based updates (Kim et al., 2026; Tan et al., 2026). The latter support non-differentiable objectives without backpropagating through the denoising process. Relative to these sampling-based methods, our approach jointly optimizes initial latent and posterior noise with Sep-CMA-ES, whose diagonal covariance yields linear internal time and memory scaling with the number of noise variables (Ros & Hansen, 2008), making it suitable for this high-dimensional search.

Physical Feasibility of Generated Motion Existing methods incorporate physics at different stages of motion generation. Before training, motion data can be refined through kinematic-constraint-aware retargeting (Yang et al., 2026) or collected through simulated tracking, with the latter sometimes repeated as the generator and tracker improve over successive training rounds (Watanabe et al., 2025; Xu et al., 2025). Physical objectives can also guide learning directly through dynamics-consistency losses (Serifi et al., 2024a; Xing et al., 2026) or preference-based post-training (Zhang et al., 2026). Other approaches introduce physics during generation: PhysDiff projects intermediate denoising states through simulated execution (Yuan et al., 2023), while CLoSD alternates motion planning and execution (Tevet et al., 2025). Alongside this choice of when to incorporate physics, methods differ in what they predict. Whereas most motion diffusion models generate kinematic state trajectories, joint state–action models also generate controls, learning both from trajectories collected with tracking controllers (Huang et al., 2025; Wu et al., 2025). However, incorporating physics before or during training does not ensure feasibility of individual generated samples, particularly under distribution shift. Predicting controls alongside states does not remove this limitation, since the predicted states may differ from the motion produced by executing those controls. Furthermore, approaches that use simulated tracking depend on a trained tracking controller. To address these limitations, our method evaluates kinematic constraints and constrained inverse-dynamics residuals on each fully generated motion and optimizes diffusion noise at inference time, requiring neither generator retraining nor tracking-controller rollouts.

Further related work appears in Appendix A.

3 Preliminaries: Motion Diffusion Models

Motion Representation Let $t \geq 0$ denote the current physical frame, F the number of predicted future frames, and P the number of past frames supplied as conditioning. With inclusive ranges, we write

$$\mathbf{M}_{\mathrm{fut},t} = (\mathbf{m}^{\mathrm{enc}}_{t+1}, \ldots, \mathbf{m}^{\mathrm{enc}}_{t+F}), \qquad \mathbf{M}_{\mathrm{pst},t} = (\mathbf{m}^{\mathrm{enc}}_{t-P+1}, \ldots, \mathbf{m}^{\mathrm{enc}}_{t}). \tag{1}$$

Each $\mathbf{m}^{\mathrm{enc}}_{\ell} \in \mathbb{R}^{n_{\mathrm{enc}}}$ is a full encoded motion feature vector; stacking frames gives $\mathbf{M}_{\mathrm{fut},t} \in \mathbb{R}^{F \times n_{\mathrm{enc}}}$. The local future-frame index is $f = 1, \ldots, F$, with absolute frame $\ell = t + f$.

For a floating-base character, decoding $\mathbf{m}^{\mathrm{enc}}_{\ell}$ gives the motion feature tuple $\mathbf{m}_{\ell} = (\mathbf{x}_{\ell}, \mathbf{r}_{\ell}, \mathbf{J}_{\ell}, \mathbf{X}_{\ell}, \mathbf{R}_{\ell})$, where $\mathbf{x}_{\ell} \in \mathbb{R}^3$ is the root position, $\mathbf{r}_{\ell} \in \mathbb{R}^6$ is its orientation feature, and $\mathbf{J}_{\ell} \in \mathbb{R}^{J \times 6}$ contains the J joint orientations relative to their parents. The link position and orientation features are $\mathbf{X}_{\ell} \in \mathbb{R}^{n_{\mathrm{b}} \times 3}$ and $\mathbf{R}_{\ell} \in \mathbb{R}^{n_{\mathrm{b}} \times 6}$, where n_{b} counts rigid bodies. Orientations use the continuous 6D representation (Zhou et al., 2019). Flattening this tuple gives $n_m = 9 + 6J + 9n_{\mathrm{b}}$ features. The full encoding uses six entries per position or orientation block, giving $n_{\mathrm{enc}} = 6(2 + J + 2n_{\mathrm{b}})$.

Motion Diffusion Models Motion diffusion models generate motion by iteratively denoising a noisy sequence conditioned on past motion, task commands, and style. We consider a frozen Denoising Diffusion Probabilistic Model (DDPM) (Ho et al., 2020) whose denoiser $\mathcal{M}_{\theta}$ predicts clean motion from the noisy sequence, diffusion level, and conditioning $\mathbf{c}_t$. Superscripts $d = 0, \ldots, D$ index diffusion levels from clean motion ($d = 0$) to initial noise ($d = D$); hats denote clean-motion estimates. Sampling starts from $\mathbf{M}^{D}_{\mathrm{fut},t}$ and injects posterior noise $\mathbf{Z}_d$ at reverse steps with noise scale $\sigma^{\mathrm{diff}}_d > 0$. These noise tensors match the future motion's shape and have independent standard-normal entries before optimization. Fixing all injected noise and conditioning makes the complete sampler deterministic, as formalized in Sec. 5. Appendix B gives the forward and reverse processes.

4 OVERVIEW

PhysibleMDM refines generated motion at inference time by optimizing diffusion noise under task objectives and physical-feasibility penalties (Fig. 1). Starting from standard-normal noise, we standardize each candidate, generate a complete motion through the frozen MDM, and evaluate its kinematic violations and constrained inverse-dynamics residuals. Batched Sep-CMA-ES (Ros & Hansen, 2008) updates the candidate population over a fixed iteration budget. We then commit the first part of the best motion and repeat autoregressively, conditioning on the preceding optimized motion and encouraging consistency between overlapping predictions.

5 DIFFUSION NOISE OPTIMIZATION

We jointly optimize initial and posterior diffusion noise to provide additional control over the denoising trajectory, increasing flexibility to reduce constraint violations and improve task performance. To help preserve motion style, we standardize the noise produced by the optimizer to have zero mean and unit variance, matching the corresponding moments of the model's standard-normal sampling distribution. This regularizes the search to help limit deviations from the learned motion distribution.

Joint Initial and Posterior Noise Parameterization For a frozen model and fixed conditioning, the generated sample is fully determined by the specific sequence of injected noise values. We use these noise values as decision variables.

We collect the initial noise and all D allocated posterior-noise tensors into the sampling-noise vector

$$\mathbf{w} = \mathrm{concat}\big(\mathrm{vec}(\mathbf{M}^{D}_{\mathrm{fut},t}), \{\mathrm{vec}(\mathbf{Z}_d)\}_{d=1}^{D}\big) \in \mathbb{R}^{n_w}, \qquad n_w = (D+1)Fn_{\mathrm{enc}}. \tag{2}$$

The sampler injects posterior noise only at steps with $\sigma_d^{\mathrm{diff}} > 0$. The complete reverse sampler is the deterministic map $\mathbf{M}^{0}_{\mathrm{fut},t} = \Phi_\theta(\mathbf{w}, \mathbf{c}_t)$. Here Φ_θ comprises all reverse transitions; $\mathcal{M}_\theta$ denotes a single denoiser evaluation.

Noise Standardization We optimize the unnormalized noise $\mathbf{w}^{\mathrm{raw}} \in \mathbb{R}^{n_w}$. For each candidate, we standardize its initial noise and combined posterior noise separately. Let $b \in \{\mathrm{init}, \mathrm{post}\}$ index these two groups, with empirical mean μ_b and sample standard deviation s_b over all entries of $\mathbf{w}_b^{\mathrm{raw}}$. The sampling noise is

$$\mathbf{w} = \Psi(\mathbf{w}^{\mathrm{raw}}) = \mathrm{concat}_b\left(\frac{\mathbf{w}_b^{\mathrm{raw}} - \mu_b}{\max(s_b, \varepsilon_{\mathrm{norm}})}\right), \qquad \varepsilon_{\mathrm{norm}} > 0. \tag{3}$$

The posterior statistics are shared across all D diffusion steps. Correlations and higher-order statistics may still change during optimization.

Diffusion Noise Optimization Problem We optimize the noise and evaluate objectives and constraints on each fully denoised sample:

$$\begin{aligned} \min_{\mathbf{w}^{\mathrm{raw}}} \quad & \mathcal{J}(\mathbf{M}) \\ \text{s.t.} \quad & \mathbf{M} = \Phi_\theta(\Psi(\mathbf{w}^{\mathrm{raw}}), \mathbf{c}_t), \\ & \mathbf{h}(\mathbf{M}) = \mathbf{0}, \qquad \mathbf{g}(\mathbf{M}) \le \mathbf{0}. \end{aligned} \tag{4}$$

Here, $\mathcal{J}$ is the task objective, and h_j, g_j are the scalar components of the stacked equality and inequality constraints. Positive values of g_j indicate violations. Physical constraints follow in Sec. 6, with task objectives in Appendix C.3.

6 PHYSICAL FEASIBILITY REFINEMENT OF GENERATED MOTION

Under the assumed robot and contact model, physical feasibility requires valid geometry and consistent contact kinematics, together with accelerations realizable by admissible joint torques and contact forces. We encode these requirements as kinematic and dynamic constraints in Eq. (4). An inner Inverse Dynamics (ID) solve enforces actuation and contact-force limits and quantifies the remaining dynamic mismatch in acceleration or generalized-force coordinates for each generated motion at inference time. This provides a proxy for physical feasibility without simulator rollouts or a tracking controller.

6.1 Robot Configuration and Dynamics

At each generated frame ℓ, extracting the physical configuration from $\mathbf{m}_\ell$ gives $\mathbf{s}_\ell = (\mathbf{x}_\ell, \mathbf{R}_{\mathrm{root},\ell}, \mathbf{q}_\ell)$, comprising root position, root orientation, and joint coordinates. Generalized velocities $\boldsymbol{\nu}_\ell = (\mathbf{v}_\ell, \boldsymbol{\omega}_\ell, \mathbf{v}_{\mathrm{joint},\ell})$ and accelerations $\dot{\boldsymbol{\nu}}_\ell$ lie in $\mathbb{R}^{n_\nu}$, where $n_\nu = 6 + n_{\mathrm{joint}}$ and n_{joint} counts joint velocity degrees of freedom. Root positions and velocities use world coordinates; contact geometry follows from forward kinematics.

Of n_{c} candidate contact points, $\mathcal{C}_\ell$ denotes the active set, modeled as sticking: $\mathcal{C}_\ell = \mathcal{C}_\ell^{\mathrm{stick}}$. Joint torques and stacked 3D contact forces are denoted by $\boldsymbol{\tau}$ and $\boldsymbol{\lambda}$, respectively. Appendix C.1 details coordinates, dynamics, and contact forces. Smoothing and derivative estimation follow Appendix C.2.

6.2 Kinematic Feasibility Constraints

We write kinematic constraints as functions of the generated motion $\mathbf{M}$ through its physical configurations and derivatives.

Ground Penetration Let $\delta_{\ell,p} \in \mathbb{R}$ be signed terrain separation at checked contact point p, with negative values indicating penetration. The pointwise nonpenetration condition is $g_{\mathrm{pen},\ell,p}(\mathbf{M}) = -\delta_{\ell,p} \leq 0$ for $p = 1, \ldots, n_{\mathrm{c}}$.

Ground Slip Let $\mathbf{J}_{\mathrm{c},\parallel,p}(\mathbf{s}) \in \mathbb{R}^{2\times n_\nu}$ be the contact point's translational Jacobian projected onto the contact tangent plane. Its tangential velocity defines the pointwise sticking condition

$$\mathbf{h}_{\mathrm{slip},\ell,p}(\mathbf{M}) = \mathbf{v}_{\parallel,p}(\mathbf{s}_\ell, \boldsymbol{\nu}_\ell) = \mathbf{J}_{\mathrm{c},\parallel,p}(\mathbf{s}_\ell)\boldsymbol{\nu}_\ell = \mathbf{0}, \qquad p \in \mathcal{C}_\ell^{\mathrm{stick}}. \tag{5}$$

We measure ground penetration and slip by averaging penetration depths and active-contact tangential speeds over all n_{c} candidate points (Appendix C.4). On flat ground, tangential velocity comprises the horizontal components, and contact activation is $a_{\ell,p} = \mathbf{1}[\delta_{\ell,p} \leq 0]$.

Contact Pattern Preservation Let $a_{\ell,p}^{\mathrm{ref}} \in \{0, 1\}$ denote a reference contact pattern inferred from unoptimized samples and held fixed during each optimization horizon (Appendix C.4). Let $\delta_{\mathrm{pen}} \geq 0$ be the allowed penetration and $\delta_{\mathrm{clear}} \geq 0$ the clearance required for inactive contacts. With $[x]_+ = \max(x, 0)$, the pointwise constraint $g_{\mathrm{contact},\ell,p} \leq 0$ uses

$$g_{\mathrm{contact},\ell,p} = a_{\ell,p}^{\mathrm{ref}}\big([\delta_{\ell,p}]_+ + [-\delta_{\ell,p} - \delta_{\mathrm{pen}}]_+\big) + (1 - a_{\ell,p}^{\mathrm{ref}})[\delta_{\mathrm{clear}} - \delta_{\ell,p}]_+. \tag{6}$$

This penalizes separation and excessive penetration during expected contact, and insufficient clearance during expected flight. Averaging over contact points gives the per-frame violation $v_{\mathrm{contact},\ell}$.

6.3 Dynamic Feasibility Constraints

Dynamic feasibility requires existence of joint torques $\boldsymbol{\tau}$ and contact forces $\boldsymbol{\lambda}$ that realize the prescribed motion under three conditions. *Underactuated dynamics:* the motion must satisfy the equations of motion for the unactuated coordinates, including the floating base; joint torques cannot directly compensate for an imbalance in these equations. *Actuation limits:* joint torques must stay within actuator bounds. *Contact limits:* normal forces must be nonnegative ($\lambda_\perp \geq 0$), and tangential forces must lie within the friction cone for the assumed sticking mode; no-slip kinematics are checked separately. Admissible torques and contact forces alone therefore do not establish that the prescribed acceleration is realizable.

6.3.1 Constrained Inverse Dynamics

For each planned state and its derivatives, constrained ID returns joint torques, contact forces, and reconstructed acceleration as $(\boldsymbol{\tau}_\ell^{\mathrm{ID}}, \boldsymbol{\lambda}_\ell^{\mathrm{ID}}, \dot{\boldsymbol{\nu}}_\ell^{\mathrm{ID}}) = \mathrm{ID}(\mathbf{s}_\ell, \boldsymbol{\nu}_\ell, \dot{\boldsymbol{\nu}}_\ell; \mathcal{C}_\ell)$. Following optimization-based control formulations (Righetti et al., 2011; 2013; Wensing et al., 2023), the solve minimizes acceleration mismatch subject to actuator bounds $\boldsymbol{\tau}_{\mathrm{lb}}(\mathbf{v}_{\mathrm{joint}}) \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}_{\mathrm{ub}}(\mathbf{v}_{\mathrm{joint}})$ and unilateral friction cones at active contacts. It accounts for the unactuated floating base through the equations of motion. Appendices C.5.1 and D detail the formulation, velocity-dependent bounds, and batched solver.

6.3.2 ACCELERATION FEASIBILITY CONSTRAINT

The inner ID solve returns torques and contact forces satisfying its actuator and force-cone constraints. With acceleration residual $\mathbf{e}_{\mathrm{acc},\ell} = \dot{\boldsymbol{\nu}}_\ell^{\mathrm{ID}} - \dot{\boldsymbol{\nu}}_\ell$, acceleration feasibility additionally requires

$$\mathbf{h}_{\mathrm{acc},\ell}(\mathbf{M}) = \mathbf{e}_{\mathrm{acc},\ell} = \mathbf{0}, \qquad \ell = t+1, \ldots, t+F. \tag{7}$$

We measure acceleration mismatch as the mean absolute residual over all generalized acceleration components, including the floating base and joints: $v_{\mathrm{acc},\ell} = \|\mathbf{e}_{\mathrm{acc},\ell}\|_1 / n_\nu$.

6.4 AUTO-REGRESSIVE REFINEMENT AND MOTION CONSISTENCY

We generate a long sequence $\mathbf{M}_{1:T}^\star$, with $T \gg F$, by repeatedly optimizing F future frames conditioned on P history frames, as in CAMDM and A-MDM (Chen et al., 2024a; Shi et al., 2024). A star denotes the best evaluated candidate in each window. We commit its first $F_{\mathrm{keep}} \leq F$ frames before replanning; the motion-consistency objective encourages agreement over the $F_{\mathrm{ov}} = F - F_{\mathrm{keep}}$ overlapping future frames. Initialization, history updates, and the consistency objective are detailed in Appendices B.4 and C.3.

7 DIFFUSION NOISE OPTIMIZATION WITH SEP-CMA-ES

Each candidate noise vector $\mathbf{w}^{\mathrm{raw}}$ is standardized using Eq. (3) and evaluated through the complete sampler $\Phi_\theta(\Psi(\mathbf{w}^{\mathrm{raw}}), \mathbf{c}_t)$.

Constraint Softening and Candidate Evaluation The resulting soft-constraint objective is

$$\mathcal{L}(\mathbf{M}) = \mathcal{J}(\mathbf{M}) + \frac{1}{F}\sum_{f=1}^{F}\left[\sum_{j\in\mathcal{T}_{\mathrm{h}}} \rho_{\mathrm{h},j} v_{j,t+f}^2 + \sum_{j\in\mathcal{T}_{\mathrm{g}}} \rho_{\mathrm{g},j} v_{j,t+f}^2\right], \tag{8}$$

Here, $\mathcal{T}_{\mathrm{h}}$ and $\mathcal{T}_{\mathrm{g}}$ index equality and inequality terms, $v_{j,\ell} \geq 0$ is the aggregated per-frame violation, and $\rho_{\mathrm{h},j}, \rho_{\mathrm{g},j} \geq 0$ weight the squared penalties. See Appendix E.1 for aggregation and fitness details.

Batched Sep-CMA-ES Search Discrete contact decisions and expensive gradients through the diffusion sampler and constrained ID motivate sampling-based optimization. We seek convergence in few iterations despite the high-dimensional noise space. We therefore use Sep-CMA-ES (Ros & Hansen, 2008), whose diagonal covariance keeps computational complexity and memory costs linear in the number of optimization variables for a fixed population size.

We optimize the raw diffusion noise through K population-search iterations:

$$\mathbf{w}^{\mathrm{raw},\star} \leftarrow \mathrm{SepCMAES}_K[\mathbf{w}^{\mathrm{raw}} \mapsto \mathcal{L}(\Phi_\theta(\Psi(\mathbf{w}^{\mathrm{raw}}), \mathbf{c}_t))]\,. \tag{9}$$

Sep-CMA-ES (Ros & Hansen, 2008) samples and ranks candidates, then adapts the search mean, step size, and diagonal covariance. The star denotes the best evaluated candidate after the iteration budget. Appendix E details initialization and updates.

8 EXPERIMENTS

8.1 EXPERIMENTAL DETAILS

We evaluate the Skinned Multi-Person Linear (SMPL) humanoid (Loper et al., 2015) and a custom bipedal robot using MDMs based on Conditional Autoregressive Motion Diffusion Model (CAMDM) (Chen et al., 2024a), trained on retargeted 100STYLE motions (Mason et al., 2022). Generation is conditioned on past motion, a commanded root trajectory, and a style label, with eight denoising steps and autoregressive execution of optimized segments. We evaluate task tracking and physical feasibility from decoded configurations, forward kinematics, and smoothed motion derivatives; evaluation in the dynamic setting additionally uses constrained ID. Joint initial- and posterior-noise optimization uses $78{,}840$ scalar optimization parameters per motion horizon for SMPL and $103{,}680$ for the bipedal robot in the experimental implementation (Appendix E.2). Full implementation details appear in Appendices B–F.

8.2 BASELINES

We compare inference-time refinements of a frozen generator without requiring a pretrained tracking controller. *Unoptimized* generation establishes the starting quality, and *Noise Best-of-N* tests independent sampling against adaptive search. *Motion GD* and *Motion Sep-CMA-ES* directly refine generated trajectories, motivated by constraint-aware retargeting (Yang et al., 2026). *Guided Diffusion* represents objective guidance during denoising (Karunratanakul et al., 2023), while *Noise GD* represents gradient-based diffusion-noise optimization (Karunratanakul et al., 2024). We include *Noise ZeNO* (Kim et al., 2026) as a prior sampling-based noise-optimization method, adapting its zeroth-order updates to joint initial and posterior noise for kinematic feasibility. We adapt these technique families to our common objectives and constraints. Two component ablations, *Lat. Noise Sep-CMA-ES* and *Noise Sep-CMA-ES (NNS)*, test posterior-noise optimization and noise standardization, respectively. The latent-only variant disables posterior-noise optimization, holding sampled posterior noise fixed across candidates and iterations within each horizon. NNS denotes no noise standardization: the normalization in Eq. (3) is omitted (implementations in Appendix E.5).

8.3 PHYSICAL FEASIBILITY OPTIMIZATION SETTINGS

We consider two feasibility optimization settings: **kinematic feasibility optimization**, which penalizes contact-pattern deviations, ground slip, and self-collision; and **dynamic feasibility optimization**, which penalizes acceleration-reconstruction error, ground slip, and self-collision. Both settings also include velocity tracking and autoregressive motion consistency objectives. The kinematic setting is comparatively inexpensive but requires contact-pattern preservation to discourage floating, whereas the dynamic setting evaluates whether accelerations can be supported under actuator bounds and contact-force cones, at the additional cost of dynamics evaluation and inner ID solves. Within each setting, methods share objective & constraint weights (budgets and weights: Appendix F.3; active terms: Tab. 4).

8.4 EVALUATION METRICS

We measure ground penetration, slip, self-collision, acceleration-reconstruction error, and the ID force residual, alongside task tracking, autoregressive consistency, and runtime per iteration and motion horizon. Smoothness is measured by Log Dimensionless Jerk (LDJ) (Balasubramanian et al., 2015), which normalizes jerk energy separately for root translation, root rotation, and joints before averaging and taking the negative logarithm; higher is smoother. For SMPL, a pretrained style classifier provides style accuracy and overall/per-style Fréchet Inception Distance (FID) (Chen et al., 2024a).

Framewise metrics pool committed frames over time and, where applicable, styles. Tables report means with the 90th percentile for minimization metrics (↓) and the 10th for maximization metrics (↑) in brackets. Colors range from red (worse) to green (better); boldface includes means within 3% of the best value. Appendix F.4 defines the metrics and aggregation; classifier and timing details follow in Appendix F.

8.5 KINEMATIC FEASIBILITY OPTIMIZATION RESULTS

Across 100STYLE and both embodiments, the comparisons in Tabs. 1–2 show improved kinematic feasibility with Physible; on SMPL, mean ground penetration, self-collision penetration, and slip decrease by 75%, 92%, and 74% relative to unoptimized generation. Figure 2(a,b) illustrates the qualitative improvements: Physible reduces foot-ground penetration in the *cat* motion and arm interpenetration in the *lawn-mower* motion while retaining the characteristic poses and motion patterns. Direct motion refinement can introduce jitter (i.e., lower LDJ), while Noise GD can degrade motion style. Compared with Noise ZeNO on SMPL, Physible reduces mean ground penetration, self-collision penetration, and slip by 28%, 79%, and 39%, respectively. The component ablations highlight the benefits of standardization for Physible: SMPL style accuracy increases from 57.49% to 87.73%, and per-style FID decreases from 133.39 to 43.95. The remaining gap to the unoptimized style accuracy of 96.03% reflects the tradeoff between physical feasibility and style preservation. Latent-only optimization provides only a slight improvement over unoptimized motion and falls considerably short of Physible (see Appendix G for further comparisons and ablations).

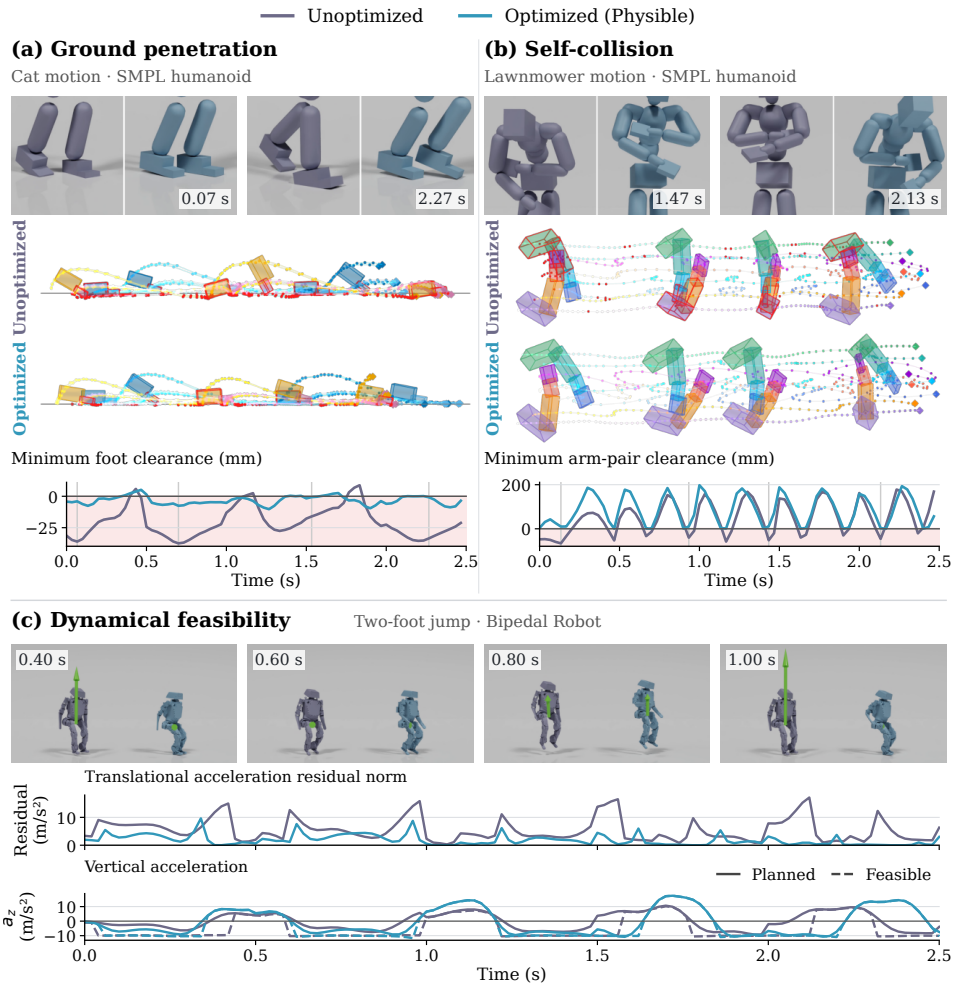


Figure 2: Qualitative comparison of unoptimized motion (gray-purple) and Physible-optimized motion (cyan-blue). (a) Ground-contact behavior for the SMPL humanoid during the *cat* motion, with side-view foot trajectories and minimum foot clearance. (b) Arm self-collision behavior during the *lawn-mower* motion, with top-down arm trajectories and minimum arm-pair clearance. In both panels, negative clearance indicates penetration, and red collision geometries mark active collisions or penetrations. (c) Dynamic feasibility of the bipedal robot's *two-foot jump*, with rendered snapshots, the translational acceleration residual norm, and planned (solid) versus feasible (dashed) vertical acceleration. Physible reduces the illustrated feasibility violations while preserving the characteristic motion patterns.

Table 1: Kinematic feasibility on the SMPL humanoid. Metrics show mean [p90 for ↓; p10 for ↑]. Style Acc., Style FID (Overall) report means only. NNS: no noise standardization. Green indicates better performance; **bold** marks the best results.

Method	Ground Penetr. [mm] ↓	Self-Coll. Penetr. [mm] ↓	Ground Slip [m/s] ↓	Velocity Tracking ↓	Motion Consistency ↓	LDJ ↑	Style Acc. [%] ↑	Style FID (Overall) ↓	Style FID (Per-Style) ↓	Comp. Time ↓ Iter. / Horizon [ms] / [s]
Unoptimized	2.66e+0 [6.21e+0]	1.93e-1 [0.00e+0]	1.85e-1 [4.25e-1]	7.70e-2 [1.10e-1]	2.71e-4 [6.92e-4]	**-6.42 [-7.15]**	**96.03**	**7.52**	**16.16 [44.54]**	N/A / **0.02**
Motion GD	3.69e+0 [9.74e+0]	9.85e-3 [0.00e+0]	5.46e-2 [1.62e-1]	1.97e-2 [5.80e-2]	**1.00e-4 [3.37e-4]**	-8.45 [-9.08]	**95.34**	7.89	17.49 [43.59]	57 / 5.79
Motion Sep-CMA-ES	1.75e+0 [4.46e+0]	2.57e-1 [7.29e-1]	1.58e-1 [3.41e-1]	8.29e-2 [1.30e-1]	1.31e-3 [4.31e-3]	-10.05 [-10.43]	**94.73**	**7.56**	19.82 [46.78]	**23** / 2.41
Guided Diffusion	1.52e+0 [3.38e+0]	1.59e-1 [0.00e+0]	1.44e-1 [3.97e-1]	6.48e-2 [9.90e-2]	2.65e-4 [6.85e-4]	-7.02 [-7.75]	**96.33**	**7.57**	**16.20 [41.30]**	64 / 0.65
Noise Best-of-N	2.03e+0 [5.00e+0]	1.49e-1 [0.00e+0]	1.34e-1 [3.21e-1]	7.19e-2 [1.03e-1]	2.46e-4 [6.35e-4]	**-6.45 [-7.20]**	**95.83**	7.88	16.99 [42.26]	40 / 3.79
Noise GD	2.80e+0 [2.39e+0]	6.84e-3 [0.00e+0]	1.27e-2 [4.89e-2]	**7.86e-3 [2.33e-2]**	3.44e-4 [1.07e-3]	-7.60 [-8.62]	15.41	82.27	274.81 [396.01]	105 / 10.78
Noise ZeNO	9.39e-1 [2.63e+0]	7.19e-2 [9.19e-2]	7.96e-2 [1.97e-1]	5.81e-2 [8.49e-2]	2.14e-4 [5.81e-4]	-7.62 [-8.31]	91.69	9.73	25.99 [81.18]	45 / 4.59
Lat. Noise Sep-CMA-ES	2.62e+0 [6.13e+0]	1.89e-1 [0.00e+0]	1.82e-1 [4.22e-1]	7.62e-2 [1.09e-1]	2.70e-4 [6.83e-4]	**-6.42 [-7.17]**	**96.20**	**7.53**	**16.10 [43.65]**	40 / 3.80
Noise Sep-CMA-ES (NNS)	**2.14e-4 [0.00e+0]**	**4.59e-3 [0.00e+0]**	**9.45e-5 [0.00e+0]**	1.32e-2 [2.95e-2]	1.61e-4 [5.29e-4]	-7.63 [-8.56]	57.49	44.11	133.39 [305.77]	40 / 3.80
Physible (ours)	6.76e-1 [1.94e+0]	1.52e-2 [0.00e+0]	4.84e-2 [1.21e-1]	3.92e-2 [5.91e-2]	1.52e-4 [4.13e-4]	**-6.59 [-7.39]**	87.73	14.41	43.95 [124.04]	40 / 3.80

8.6 Dynamic Feasibility Optimization Results

Across 11 bipedal robot sequences spanning motion styles, slopes, and friction settings, Physible achieves median reductions of 89.5% in mean acceleration-reconstruction error and 79.5% in mean ID residual relative to unoptimized motion (Appendix H.1.2). The two-foot-jump comparison (Tab. 3

Table 2: Kinematic feasibility on the bipedal robot. Metrics show mean [p90 for ↓; p10 for ↑]. NNS: no noise standardization. Green indicates better performance; **bold** marks the best results.

Method	Ground Penetr. [mm] ↓	Self-Coll. Penetr. [mm] ↓	Ground Slip [m/s] ↓	Velocity Tracking ↓	Motion Consistency ↓	LDJ ↑
Unoptimized	0.56 [1.43]	1.25e-2 [0.00e+0]	8.45e-2 [1.68e-1]	0.07 [0.11]	2.70e-4 [9.95e-4]	**-7.52 [-8.08]**
Motion GD	1.09 [2.91]	5.85e-3 [0.00e+0]	2.20e-2 [4.89e-2]	**0.03 [0.09]**	**6.85e-5 [1.83e-4]**	-10.53 [-11.19]
Motion Sep-CMA-ES	0.25 [1.28]	9.99e-3 [0.00e+0]	**3.80e-3 [8.74e-3]**	0.04 [0.08]	2.34e-4 [8.73e-4]	-9.52 [-10.31]
Guided Diffusion	0.74 [1.42]	1.15e-2 [0.00e+0]	1.09e-1 [2.38e-1]	0.07 [0.10]	2.44e-4 [9.27e-4]	-8.35 [-9.29]
Noise Best-of-N	0.49 [1.30]	9.68e-3 [0.00e+0]	5.24e-2 [1.19e-1]	0.06 [0.10]	2.22e-4 [8.34e-4]	**-7.55 [-8.12]**
Noise GD	0.70 [1.53]	2.85e-3 [0.00e+0]	1.15e-1 [2.70e-1]	0.10 [0.18]	9.67e-4 [4.71e-3]	-8.00 [-8.51]
Noise ZeNO	0.52 [1.40]	1.55e-2 [0.00e+0]	5.56e-2 [1.27e-1]	0.07 [0.11]	2.35e-4 [9.12e-4]	-8.25 [-8.79]
Lat. Noise Sep-CMA-ES	0.55 [1.42]	1.18e-2 [0.00e+0]	8.04e-2 [1.59e-1]	0.07 [0.10]	2.59e-4 [9.78e-4]	**-7.52 [-8.08]**
Noise Sep-CMA-ES (NNS)	**0.20 [0.69]**	8.18e-3 [0.00e+0]	2.54e-2 [7.44e-2]	0.05 [0.07]	5.78e-4 [2.52e-3]	-8.70 [-9.20]
Physible (ours)	0.31 [0.85]	**2.36e-3 [0.00e+0]**	2.16e-2 [5.59e-2]	0.04 [0.06]	7.40e-5 [2.63e-4]	-7.75 [-8.49]

and Fig. 25) highlights the tradeoffs among methods. Guided Diffusion produces substantial ground penetration, large dynamic residuals, and degraded motion style. Motion Sep-CMA-ES reduces residuals at the expense of smoothness, while Noise GD achieves the smallest residuals but requires more computation and produces persistent floating instead of the intended jumping motion. Physible retains the characteristic two-foot-jump behavior while reducing the translational acceleration residual and improving agreement between planned and feasible vertical accelerations (Fig. 2(c)). Further quantitative and qualitative results in Appendix H show that these feasibility gains extend to other motion sequences and contact conditions, including uphill and downhill walking at $\pm 25°$.

Table 3: Dynamic feasibility for the bipedal robot during a two-foot jump. Metrics show mean [p90 for ↓; p10 for ↑]. Green indicates better performance; **bold** marks the best results.

Method	Acceleration Reconstruction ↓	Inverse Dynamics Residual ↓	LDJ ↑	Comp. Time ↓ Iter. / Horizon [s] / [s]
Unoptimized	0.454 [0.954]	8.44e+1 [1.87e+2]	**-7.71 [-8.11]**	N/A / **< 1**
Motion Sep-CMA-ES	0.032 [0.056]	7.34e+0 [1.29e+1]	-11.67 [-11.83]	**1.66** / 4145
Guided Diffusion	19.163 [34.988]	1.86e+3 [3.32e+3]	-9.91 [-10.37]	17.65 / 141
Noise GD	**0.013 [0.022]**	**1.94e+0 [4.57e+0]**	-10.03 [-10.69]	5.93 / 14828
Physible (ours)	0.081 [0.174]	2.65e+1 [6.27e+1]	-8.51 [-8.83]	1.81 / 4515

8.7 Training RL Controller on Optimized vs. Unoptimized Motion

DeepMimic-style policies (Peng et al., 2018) trained on the optimized bipedal two-foot jump reduce body position, body orientation, and joint position tracking errors by 16.4%, 11.9%, and 7.4%, respectively, relative to training on the unoptimized reference. Root position and orientation errors increase by 39.6% and 41.3%, revealing a tradeoff between root-relative body tracking and accumulated global root-pose drift. Appendices F.7 and H.3 detail the setup and results, respectively.

9 Conclusion

Physible refines frozen motion diffusion models through batched, gradient-free optimization of initial and posterior noise, combining kinematic penalties with constrained inverse-dynamics evaluation. Across 100 motion styles on SMPL, it reduces mean ground penetration, self-collision penetration, and slip by 75%, 92%, and 74% relative to unoptimized generation. Across 11 bipedal robot sequences, median reductions in per-sequence mean acceleration-reconstruction error and inverse-dynamics residual are 89.5% and 79.5%, respectively, supporting inference-time noise optimization as a means of improving physical feasibility without retraining the generator or requiring tracking-controller rollouts.

Limitations & Future Work Our soft penalties do not guarantee strict physical feasibility, and optimized noise can drift from the learned motion distribution and degrade style. Noise standardization favors style preservation in this tradeoff at the cost of optimization expressivity, a restriction partly alleviated by jointly optimizing initial and posterior noise rather than initial latent noise alone. Evaluation is limited to bipedal embodiments with assumed dynamics and sticking contacts, while repeated feasibility evaluations and inverse-dynamics solves can be costly for real-time use. We have not yet systematically quantified the variability of optimization results across optimization seeds. Appendix I expands on limitations and future work.

AI DISCLOSURE

Generative AI tools were used for limited implementation and editorial assistance. Coding assistants were used to generate draft software code, and AI tools were used to assist with manuscript formatting, readability improvements, consistency checks, identification of potentially relevant literature, and reference formatting. All AI-assisted outputs were independently reviewed, verified, and, where appropriate, modified by the authors. Generative AI tools did not contribute to the conception or development of the research ideas, hypotheses, methods, experimental design, analyses, interpretation of results, conclusions, or other substantive scientific or technical contributions reported in this work. All such contributions were made by the human authors, who retain full responsibility for the content of the manuscript and all associated artifacts.

REPRODUCIBILITY STATEMENT

We support reproducibility through detailed descriptions of our methodology, implementation, and experimental protocol. Appendix B documents data preparation, motion retargeting, diffusion-model training, and autoregressive motion generation. Appendix C specifies the task objectives and physical feasibility constraints, while Appendix D describes the numerical conditioning and efficient batched solution of the inverse-dynamics optimization. Appendix E provides optimization hyperparameters and baseline implementation details. Appendix F specifies the robot models, experimental configurations, evaluation metrics, and computational environment. Appendices G and H provide supplementary kinematic and dynamic feasibility results, including convergence comparisons, ablations, and detailed motion and feasibility diagnostics covering all robot degrees of freedom for the illustrated sequences. Together, these descriptions and results support independent implementation of the method and examination of the behavior underlying the aggregate metrics and examples presented in the main text.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019a. doi: 10.5555/3454287.3455145. URL https://dl.acm.org/doi/10.5555/3454287.3455145.

Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M. Moursi. Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, 1(2):107–115, 2019b. doi: 10.23952/jano.1.2019.2.02. URL https://jano.biemdas.com/index.php/home/article/view/17.

Juan Alvarez-Padilla, John Z. Zhang, Sofia Kwok, John M. Dolan, and Zachary Manchester. Real-Time Whole-Body Control of Legged Robots with Model-Predictive Path Integral Control. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14721–14727, 2025. doi: 10.1109/ICRA55743.2025.11128271. URL https://doi.org/10.1109/ICRA55743.2025.11128271.

Rika Antonova, Jingyun Yang, Krishna Murthy Jatavallabhula, and Jeannette Bohg. Rethinking optimization with differentiable simulation from a global perspective. In *Conference on robot learning*, pp. 276–286. PMLR, 2023. URL https://proceedings.mlr.press/v205/antonova23a.html.

João Pedro Araújo, Yanjie Ze, Pei Xu, Jiajun Wu, and C Karen Liu. Retargeting Matters: General Motion Retargeting for Humanoid Motion Tracking. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://jaraujo98.github.io/retargeting_matters/.

Sivakumar Balasubramanian, Alejandro Melendez-Calderon, Agnes Roby-Brami, and Etienne Burdet. On the analysis of movement smoothness. *Journal of NeuroEngineering and Rehabilitation*, 12:112, 2015. doi: 10.1186/s12984-015-0090-9.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-Flow: differentiating through flows for controlled generation. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 3462–3483, 2024. URL https://proceedings.mlr.press/v235/ben-hamu24a.html.

William Bittle. SAT (separating axis theorem). https://dyn4j.org/2010/01/sat/, 2010. URL https://dyn4j.org/2010/01/sat/. Accessed: 2026-09-22.

Sabrina Bodmer, René Zurbrügg, Tifanny Portela, Hao Ma, Alexandre Didier, Marco Hutter, Colin Jones, and Melanie Zeilinger. Grounding Generative Policies in Physics: Optimization-Guided Diffusion for Robot Control. *arXiv preprint arXiv:2606.24208*, 2026. URL https://arxiv.org/abs/2606.24208.

Rui Chen, Mingyi Shi, Shaoli Huang, Ping Tan, Taku Komura, and Xuelin Chen. Taming diffusion probabilistic models for character control. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–10, 2024a. doi: 10.1145/3641519.3657440. URL https://dl.acm.org/doi/10.1145/3641519.3657440.

Sherry X. Chen, Yaron Vaxman, Elad Ben Baruch, David Asulin, Aviad Moreshet, Kuo-Chin Lien, Misha Sra, and Pradeep Sen. TiNO-Edit: Timestep and Noise Optimization for Robust Diffusion-Based Image Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. doi: 10.1109/CVPR52733.2024.00606. URL https://www.computer.org/csdl/proceedings-article/cvpr/2024/00606000/20hSMs2QRzi.

Xiaoyuan Cheng, Xiaohang Tang, and Yiming Yang. Safe and Stable Control via Lyapunov-Guided Diffusion Models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=aY97JGello.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025. doi: 10.1177/02783649241273668. URL https://journals.sagepub.com/doi/10.1177/02783649241273668.

Hanbyel Cho, Sang-Hun Kim, Jeonguk Kang, and Donghan Koo. SafeFlow: Real-Time Text-Driven Humanoid Whole-Body Control via Physics-Guided Rectified Flow and Selective Safety Gating. *arXiv preprint arXiv:2603.23983*, 2026. URL https://arxiv.org/abs/2603.23983.

Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. *Advances in Neural Information Processing Systems*, 37:89307–89333, 2024. doi: 10.5555/3737916.3740750. URL https://dl.acm.org/doi/10.5555/3737916.3740750.

Rishabh Dabral, Muhammad Hamza Mughal, Vladislav Golyanik, and Christian Theobalt. MoFusion: A framework for denoising-diffusion-based motion synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9760–9770, 2023. doi: 10.1109/CVPR52729.2023.00941. URL https://openaccess.thecvf.com/content/CVPR2023/html/Dabral_Mofusion_A_Framework_for_Denoising-Diffusion-Based_Motion_Synthesis_CVPR_2023_paper.html.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Victor Dhedin, Ilyass Taouil, Shafeef Omar, Dian Yu, Kun Tao, Angela Dai, and Majid Khadiv. DynaRetarget: Dynamically-Feasible Retargeting using Sampling-Based Trajectory Optimization. *arXiv preprint arXiv:2602.06827*, 2026. URL https://arxiv.org/abs/2602.06827.

Paul H. C. Eilers. A perfect smoother. *Analytical Chemistry*, 75(14):3631–3636, 2003. doi: 10.1021/ac034173t. URL https://doi.org/10.1021/ac034173t.

Luca Eyring, Shyamgopal Karthik, Karsten Roth, Alexey Dosovitskiy, and Zeynep Akata. ReNO: Enhancing one-step text-to-image models through reward-based noise optimization. *Advances in Neural Information Processing Systems*, 37:125487–125519, 2024. doi: 10.52202/079017-3987. URL https://doi.org/10.52202/079017-3987.

Luca Eyring, Shyamgopal Karthik, Alexey Dosovitskiy, Nataniel Ruiz, and Zeynep Akata. Noise Hypernetworks: Amortizing Test-Time Compute in Diffusion Models. *Advances in neural information processing systems*, 2025. doi: 10.52202/085713-3190. URL https://doi.org/10.52202/085713-3190.

Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, 2008. doi: 10.1007/978-1-4899-7560-7. URL https://link.springer.com/book/10.1007/978-1-4899-7560-7.

Paul J Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives. *arXiv preprint arXiv:2405.12762*, 2024. URL https://arxiv.org/abs/2405.12762.

Ruben Grandia, Andrew J Taylor, Aaron D Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8352–8358. IEEE, 2021. doi: 10.1109/ICRA48506.2021.9561510. URL https://doi.org/10.1109/ICRA48506.2021.9561510.

Ruben Grandia, Espen Knoop, Michael A. Hopkins, Georg Wiedebach, Jared Bishop, Steven Pickles, David Müller, and Moritz Bächer. Design and control of a bipedal robotic character. In *Robotics: Science and Systems (RSS)*, 2024. doi: 10.15607/RSS.2024.XX.103. URL https://doi.org/10.15607/RSS.2024.XX.103.

Jessy W Grizzle, Christine Chevallereau, Ryan W Sinnet, and Aaron D Ames. Models, feedback control, and open problems of 3D bipedal robotic walking. *Automatica*, 50(8):1955–1988, 2014. doi: 10.1016/j.automatica.2014.04.021. URL https://www.sciencedirect.com/science/article/pii/S0005109814001654.

Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001. doi: 10.1162/106365601750190398. URL https://doi.org/10.1162/106365601750190398.

Alexander Herzog, Ludovic Righetti, Felix Grimminger, Peter Pastor, and Stefan Schaal. Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 981–988. IEEE, 2014. doi: 10.1109/IROS.2014.6942678. URL https://doi.org/10.1109/IROS.2014.6942678.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. Diffuse-CLoC: Guided diffusion for physics-based character look-ahead control. *ACM Transactions on Graphics (TOG)*, 44(4):1–12, 2025. doi: 10.1145/3731206. URL https://doi.org/10.1145/3731206.

Xiaofei Hui, Bo Yan, Haoxuan Qu, Hossein Rahmani, and Jun Liu. Training-free Controllable Human Motion Generation under Heterogeneous Constraints. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2607.01990.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2151–2162, 2023. doi: 10.1109/ICCV51070.2023.00205. URL https://doi.org/10.1109/ICCV51070.2023.00205.

Korrawe Karunratanakul, Konpat Preechakul, Emre Aksan, Thabo Beeler, Supasorn Suwajanakorn, and Siyu Tang. Optimizing diffusion noise can serve as universal motion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1334–1345. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00133. URL https://openaccess.thecvf.com/content/CVPR2024/html/Karunratanakul_Optimizing_Diffusion_Noise_Can_Serve_As_Universal_Motion_Priors_CVPR_2024_paper.html.

Charles Khazoom, Daniel Gonzalez-Diaz, Yanran Ding, and Sangbae Kim. Humanoid self-collision avoidance using whole-body control with control barrier functions. In *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pp. 558–565. IEEE, 2022. doi: 10.1109/Humanoids53995.2022.10000235. URL https://doi.org/10.1109/Humanoids53995.2022.10000235.

Jeongsol Kim, Hongeun Kim, Jian Wang, and Jong Chul Ye. Gradient-Free Noise Optimization for Reward Alignment in Generative Models. *arXiv preprint arXiv:2605.11347*, 2026. URL https://arxiv.org/abs/2605.11347.

Jihoon Kim, Jiseob Kim, and Sungjoon Choi. FLAME: Free-form language-based motion synthesis & editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8255–8263, 2023. doi: 10.1609/aaai.v37i7.25996. URL https://ojs.aaai.org/index.php/AAAI/article/view/25996.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL https://arxiv.org/abs/1412.6980.

Kyungmin Lee, Sibeen Kim, Youngdo Lee, Minho Park, Hyunseung Kim, Dongyoon Hwang, Donghu Kim, Hojoon Lee, and Jaegul Choo. PHUMA: Physically Reliable Humanoid Locomotion Dataset. *arXiv preprint arXiv:2510.26236*, 2025. URL https://arxiv.org/abs/2510.26236.

Giuseppe L'Erario, Silvio Traversaro, Carlotta Sartore, Riccardo Grieco, Stefano Dafarra, Giulio Romualdi, Filippo Luca Ferretti, and Daniele Pucci. adam, 2026. URL https://github.com/gbionics/adam/releases/tag/v0.6.2. Version 0.6.2.

Zeyang Li, Kaveh Alim, and Navid Azizan. HardFlow: Hard-Constrained Sampling for Flow-Matching Models via Trajectory Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3708824. URL https://doi.org/10.1109/TPAMI.2026.3708824.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11(117):eadx8924, 2026. doi: 10.1126/scirobotics.adx8924. URL https://doi.org/10.1126/scirobotics.adx8924.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics*, 34(6), 2015. doi: 10.1145/2816795.2818013. URL https://dl.acm.org/doi/10.1145/2816795.2818013. Proceedings of SIGGRAPH Asia 2015.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10895–10904, 2023. doi: 10.1109/ICCV51070.2023.01000. URL https://doi.org/10.1109/ICCV51070.2023.01000.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117): eaed4592, 2026. doi: 10.1126/scirobotics.aed4592. URL https://doi.org/10.1126/scirobotics.aed4592.

Hao Ma, Sabrina Bodmer, Andrea Carron, Melanie Zeilinger, and Michael Muehlebach. Constraint-Aware Diffusion Guidance for Robotics: Real-Time Obstacle Avoidance for Autonomous Racing. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1756–1776. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/ma25a.html.

Ian Mason, Sebastian Starke, and Taku Komura. Real-Time Style Modelling of Human Locomotion via Feature-Wise Transformations and Local Motion Phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1), may 2022. doi: 10.1145/3522618. URL https://dl.acm.org/doi/10.1145/3522618.

AJ Miller, Fangzhou Yu, Michael Brauckmann, and Farbod Farshidian. High-Performance Reinforcement Learning on Spot: Optimizing Simulation Parameters with Distributional Measures. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9981–9988, 2025. doi: 10.1109/ICRA55743.2025.11128575. URL https://doi.org/10.1109/ICRA55743.2025.11128575.

David Müller, Agon Serifi, Sammy Christen, Ruben Grandia, Espen Knoop, and Moritz Bächer. ReActor: Reinforcement learning for physics-aware motion retargeting. *ACM Trans. Graph.*, 45(4), July 2026. ISSN 0730-0301. doi: 10.1145/3811378. URL https://doi.org/10.1145/3811378.

Quan Nguyen, Ayonga Hereid, Jessy W Grizzle, Aaron D Ames, and Koushil Sreenath. 3D dynamic walking on stepping stones with control barrier functions. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 827–834. IEEE, 2016. doi: 10.1109/CDC.2016.7798370. URL https://ieeexplore.ieee.org/document/7798370.

Francesco Nori, Silvio Traversaro, Jorhabib Eljaik, Francesco Romano, Andrea Del Prete, and Daniele Pucci. iCub Whole-Body Control through Force Regulation on Rigid Non-Coplanar Contacts. *Frontiers in Robotics and AI*, 2(6), 2015. ISSN 2296-9144. doi: 10.3389/frobt.2015.00006. URL https://doi.org/10.3389/frobt.2015.00006.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. SPIDER: Scalable Physics-Informed Dexterous Retargeting. *arXiv preprint arXiv:2511.09484*, 2025. URL https://arxiv.org/abs/2511.09484.

Anselm Paulus, Andreas René Geist, Pierre Schumacher, Vít Musil, Simon Rappenecker, and Georg Martius. Differentiable Simulation of Hard Contacts with Soft Gradients for Learning and Control. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/44039e59aaf6a41b16f1fc5b27bcd409-Abstract-Conference.html.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018. doi: 10.1145/3197517.3201311. URL https://dl.acm.org/doi/10.1145/3197517.3201311.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pp. 1049–1065. PMLR, 2021. URL https://proceedings.mlr.press/v155/pinneri21a.html.

Yujiang Pu and Yu Kong. NoisEasier: Test-Time Noise Optimization for Text-to-Video Generation. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2608.30194.

Davis Rempe, Mathis Petrovich, Ye Yuan, Haotian Zhang, Xue Bin Peng, Yifeng Jiang, Tingwu Wang, Umar Iqbal, David Minor, Michael de Ruyter, Jiefeng Li, Chen Tessler, Edy Lim, Eugene Jeong, Sam Wu, Ehsan Hassani, Michael Huang, Jin-Bey Yu, Chaeyeon Chung, Lina Song, Olivier Dionne, Jan Kautz, Simon Yuen, and Sanja Fidler.

Kimodo: Scaling Controllable Human Motion Generation. Technical report, NVIDIA, March 2026. URL https://research.nvidia.com/labs/sil/projects/kimodo/assets/kimodo_tech_report.pdf.

Ludovic Righetti, Jonas Buchli, Michael Mistry, and Stefan Schaal. Inverse Dynamics Control of Floating-Base Robots with External Constraints: A Unified View. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1085–1090, 2011. doi: 10.1109/ICRA.2011.5980156. URL https://ieeexplore.ieee.org/abstract/document/5980156/.

Ludovic Righetti, Jonas Buchli, Michael Mistry, Mrinal Kalakrishnan, and Stefan Schaal. Optimal distribution of contact forces with inverse-dynamics control. *The International Journal of Robotics Research*, 32(3):280–298, 2013. doi: 10.1177/0278364912469821. URL https://dl.acm.org/doi/10.1177/0278364912469821.

Rokoko Electronics. Rokoko. https://www.rokoko.com/, 2026. URL https://www.rokoko.com/.

Raymond Ros and Nikolaus Hansen. A simple modification in CMA-ES achieving linear time and space complexity. In *International conference on parallel problem solving from nature*, pp. 296–305. Springer, 2008. doi: 10.1007/978-3-540-87700-4_30. URL https://link.springer.com/chapter/10.1007/978-3-540-87700-4_30.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143. URL https://doi.org/10.1023/A:1010091220143.

Fabian Schramm, Pierre Fabre, Nicolas Perrin-Gilbert, and Justin Carpentier. Reference-Free Sampling-Based Model Predictive Control. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://arxiv.org/abs/2511.19204.

Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. Robot motion diffusion model: Motion generation for robotic characters. In *SIGGRAPH asia 2024 conference papers*, pp. 1–9, 2024a. doi: 10.1145/3680528.3687626. URL https://doi.org/10.1145/3680528.3687626.

Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. VMP: Versatile motion priors for robustly tracking motion on physical characters. In *Computer graphics forum*, volume 43, pp. e15175. Wiley Online Library, 2024b. doi: 10.1111/cgf.15175. URL https://doi.org/10.1111/cgf.15175.

Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. Interactive Character Control with Auto-Regressive Motion Diffusion Models. *ACM Transactions on Graphics*, 43(4), 2024. doi: 10.1145/3658140. URL https://doi.org/10.1145/3658140.

Xun Su and Hiroyuki Kasai. Noise is All You Need: Solving Linear Inverse Problems by Noise Combination Sampling with Diffusion Models. *arXiv preprint arXiv:2510.23633*, 2025. URL https://arxiv.org/abs/2510.23633.

Yuzheng Tan, Yuan He, Yao Zhu, Tianlin Huo, Huanqian Yan, Hang Su, Shuxin Zhang, and Guangneng Hu. Dual-Seed Evolutionary Algorithm for Noise Optimization in Diffusion Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9341–9349, 2026. doi: 10.1609/aaai.v40i11.37893. URL https://doi.org/10.1609/aaai.v40i11.37893.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-Or, and Amit Haim Bermano. Human Motion Diffusion Model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SJ1kSyO2jwu.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Haim Bermano, and Michiel van de Panne. CLoSD: Closing the Loop between Simulation and Diffusion for multi-task character control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZISppZSTv.

Takara Everest Truong, Michael Piseno, Zhaoming Xie, and Karen Liu. PDP: Physics-based character animation via diffusion policy. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–10, 2024. doi: 10.1145/3680528.3687683. URL https://doi.org/10.1145/3680528.3687683.

Masatoshi Uehara, Yulai Zhao, Chenyu Wang, Xiner Li, Aviv Regev, Sergey Levine, and Tommaso Biancalani. Inference-time alignment in diffusion models with reward-guided generation: Tutorial and review. *arXiv preprint arXiv:2501.09685*, 2025. URL https://arxiv.org/abs/2501.09685.

Utkarsh Utkarsh, Pengfei Cai, Alan Edelman, Rafael Gomez-Bombarelli, and Christopher Vincent Rackauckas. Physics-Constrained Flow Matching: Sampling Generative Models with Hard Constraints. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5354. URL https://doi.org/10.52202/085713-5354.

Akihisa Watanabe, Jiawei Ren, Li Siyao, Yichen Peng, Erwin Wu, and Edgar Simo-Serra. SimDiff: Simulator-constrained Diffusion Model for Physically Plausible Motion Generation. *arXiv preprint arXiv:2509.20927*, 2025. URL https://arxiv.org/abs/2509.20927.

Akihisa Watanabe, Qing Yu, Edgar Simo-Serra, and Kent Fujiwara. ProjFlow: Projection Sampling with Flow Matching for Zero-Shot Exact Spatial Motion Control. *arXiv preprint arXiv:2602.22742*, 2026. URL https://arxiv.org/abs/2602.22742.

Patrick M. Wensing, Michael Posa, Yue Hu, Adrien Escande, Nicolas Mansard, and Andrea Del Prete. Optimization-Based Control for Dynamic Legged Robots. *IEEE Transactions on Robotics*, 2023. doi: 10.1109/TRO.2023.3324580. URL https://doi.org/10.1109/TRO.2023.3324580.

Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. UniPhys: Unified Planner and Controller with Diffusion for Flexible Physics-Based Character Control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Wu_UniPhys_Unified_Planner_and_Controller_with_Diffusion_for_Flexible_Physics-Based_ICCV_2025_paper.html.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. SafeDiffuser: Safe planning with diffusion probabilistic models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ig2wk7kK9J.

Chaoyue Xing, Wei Mao, and Miaomiao Liu. InterPhys: Physics-aware Human Motion Synthesis in a Dynamic Scene. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xing_InterPhys_Physics-aware_Human_Motion_Synthesis_in_a_Dynamic_Scene_CVPR_2026_paper.html.

Michael Xu, Yi Shi, KangKang Yin, and Xue Bin Peng. PARC: Physics-based augmentation with reinforcement learning for character controllers. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025. doi: 10.1145/3721238.3730616. URL https://dl.acm.org/doi/10.1145/3721238.3730616.

Haoru Xue, Chaoyi Pan, Zeji Yi, Guannan Qu, and Guanya Shi. Full-Order Sampling-Based MPC for Torque-Level Locomotion Control via Diffusion-Style Annealing. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4974–4981, 2025. doi: 10.1109/ICRA55743.2025.11127320. URL https://doi.org/10.1109/ICRA55743.2025.11127320.

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C Karen Liu, Rocky Duan, and Guanya Shi. OmniRetarget: Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026. URL https://omniretarget.github.io/.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-guided human motion diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16010–16021. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01467. URL https://doi.org/10.1109/ICCV51070.2023.01467.

Stefano Zampini, Jacob K Christopher, Luca Oneto, Davide Anguita, and Ferdinando Fioretto. Training-Free Constrained Generation With Stable Diffusion Models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=TrNB08KuHK.

Yanjie Ze, Zixuan Chen, João Pedro Araújo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. TWIST: Teleoperated Whole-Body Imitation System. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2143–2154. PMLR, 2025. URL https://proceedings.mlr.press/v305/ze25a.html.

Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained Diffusers for Safe Planning and Control. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tahkGZjjWA.

Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. MotionDiffuse: Text-driven human motion generation with diffusion model. *IEEE transactions on pattern analysis and machine intelligence*, 46(6):4115–4128, 2024. doi: 10.1109/TPAMI.2024.3355414. URL https://doi.org/10.1109/TPAMI.2024.3355414.

Yangsong Zhang, Anujith Muraleedharan, Rikhat Akizhanov, Abdul Ahad Butt, Gül Varol, Pascal Fua, Fabio Pizzati, and Ivan Laptev. PhysMoDPO: Physically-Plausible Humanoid Motion with Preference Optimization. *arXiv preprint arXiv:2603.13228*, 2026. URL https://arxiv.org/abs/2603.13228.

Yunbo Zhang, Deepak Gopinath, Yuting Ye, Jessica Hodgins, Greg Turk, and Jungdam Won. Simulation and retargeting of complex multi-character interactions. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023. doi: 10.1145/3588432.3591491. URL https://doi.org/10.1145/3588432.3591491.

Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A Diffusion-Based Autoregressive Motion Model for Real-Time Text-Driven Motion Control. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.05260.

Kaifeng Zhao, Mathis Petrovich, Haotian Zhang, Tingwu Wang, Siyu Tang, and Davis Rempe. ARDY: Autoregressive Diffusion with Hybrid Representation for Interactive Human Motion Generation. *ACM Transactions on Graphics (TOG)*, 45(4), 2026. doi: 10.1145/3811284. URL https://doi.org/10.1145/3811284.

Lei Zhong, Yiming Xie, Varun Jampani, Deqing Sun, and Huaizu Jiang. SMooDi: Stylized Motion Diffusion Model. In *ECCV*, 2024. doi: 10.1007/978-3-031-73232-4_23. URL https://doi.org/10.1007/978-3-031-73232-4_23.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5745–5753, 2019. doi: 10.1109/CVPR.2019.00589. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

A Extended Related Work

A.1 Motion Diffusion Models

Diffusion models (Ho et al., 2020) serve as strong generative priors for humanoid motion, enabling high-quality synthesis under rich conditioning (Tevet et al., 2023; Chen et al., 2024a; Dabral et al., 2023) and interactive control via guided diffusion (Karunratanakul et al., 2023; Huang et al., 2025; Wu et al., 2025; Liao et al., 2026). Motion diffusion models commonly generate kinematic trajectories for a downstream tracking controller (Serifi et al., 2024a; Yuan et al., 2023; Truong et al., 2024; Watanabe et al., 2025; Tevet et al., 2025). Diffusion policies instead generate action sequences (Chi et al., 2025), while joint state–action diffusion models predict both motion and control, allowing objectives on predicted states to guide action generation (Huang et al., 2025; Wu et al., 2025). While these methods generate visually compelling humanoid motions, ensuring strict kinematic validity and contact-rich dynamical feasibility remains a core challenge. Our method optimizes the stochastic inputs of a kinematic motion generator and evaluates physical feasibility before downstream tracking.

A.2 Inference-Time Optimization of Diffusion Models

A major practical advantage of diffusion models is that sampling can be modified *at inference time* to satisfy additional objectives or constraints without retraining the base model. A common approach is *gradient-based guidance*, which biases the denoising process using gradients of an auxiliary signal. Building on classifier guidance (Dhariwal & Nichol, 2021), prior work in robotics has explored constraint-aware guidance for kinematic constraints (Ma et al., 2025; Cho et al., 2026) as well as Lyapunov-based objectives to ensure safety and stability (Cheng et al., 2025; Zhang et al., 2025). Gradient-based guidance requires differentiable objectives, while sampling-based guidance can also incorporate non-differentiable constraint evaluations (Hui et al., 2026). Both modify the intermediate sampling dynamics and can bias samples away from the model distribution.

Alternatively, *projection* methods enforce constraints by mapping intermediate denoising states or their clean-sample estimates onto feasible sets (Christopher et al., 2024; Zampini et al., 2025). Related strategies have also been explored in flow-matching frameworks, including exact linear spatial constraints for motion control (Li et al., 2026; Utkarsh et al., 2025; Watanabe et al., 2026). While these approaches can be effective when well-defined projection operators exist, their success depends on the informativeness of intermediate sample estimates and the degree to which the projection conflicts with the learned prior.

More recently, *noise optimization* has emerged, which directly optimizes the initial diffusion noise and, for stochastic samplers, noise injected along the sampling trajectory, to improve the generated output under a given objective (Chen et al., 2024b; Karunratanakul et al., 2024; Eyring et al., 2024; Uehara et al., 2025; Ben-Hamu et al., 2024; Pu & Kong, 2026). This approach evaluates objectives on generated outputs and can regularize the noise to limit deviations from the prior. Gradient-based approaches update the noise by backpropagating through the generator (Ben-Hamu et al., 2024; Karunratanakul et al., 2024; Pu & Kong, 2026; Zhao et al., 2025). Recent work introduces sampling-based noise updates, including evolutionary search, using only objective evaluations (Kim et al., 2026; Tan et al., 2026). For example, ZeNO uses a path-integral update of the latent noise, whereas our Sep-CMA-ES strategy adapts a diagonal search covariance and step size using ranked candidates for both latent and posterior noise.

Guidance and projection also require choosing how strongly constraints influence each diffusion step: intermediate trajectories may give incomplete feasibility information, while strong corrections can conflict with the learned prior and compromise motion style. Noise optimization instead evaluates fully denoised outputs and permits regularization of noise statistics (Ben-Hamu et al., 2024; Karunratanakul et al., 2024; Su & Kasai, 2025; Eyring et al., 2025). Gradient-based noise updates require backpropagation through both the denoising graph and the objective and constraint evaluations. This can be expensive, and contact decisions introduce non-smoothness that can make gradients poorly informative (Antonova et al., 2023; Paulus et al., 2026).

Efficient gradient-free optimization of high-dimensional (motion) diffusion model noise for non-smooth objectives and constraints remains underexplored. We address this gap by jointly optimizing initial and posterior noise with Sep-CMA-ES.

A.3 Physical Feasibility of Generated Robotic Character Motion

Physically infeasible references hinder downstream motion tracking and can distort the style and semantics of generated motion (Araújo et al., 2026). Approaches to improve feasibility generally focus on (i) enhancing the training data and (ii) enforcing constraints at *inference-time* during motion generation.

Physics-Informed Training A common strategy is to retarget motion capture data to new morphologies while enforcing kinematic and contact constraints via constraint-aware optimization (Luo et al., 2023; Zhang et al., 2023; Ze et al., 2025; Yang et al., 2026; Lee et al., 2025; Pan et al., 2025; Dhedin et al., 2026). SimDiff (Watanabe et al., 2025) constructs simulator-augmented motion data using a pretrained tracking policy and trains adapters around a frozen diffusion backbone, enabling inference without repeated simulator calls. PARC (Xu et al., 2025) instead iteratively trains a motion generator and tracking controller: the generator proposes motions, the tracker executes them in simulation, and successful executions augment the dataset for subsequent training rounds. Both use physics-based execution to improve the training pipeline. Joint state–action diffusion models, such as Diffuse-CLoC and UniPhys (Huang et al., 2025; Wu et al., 2025), similarly use tracking controllers to collect simulated state trajectories and associated actions for training. They learn to predict motion and control together. InterPhys trains motion generation for consistency with predicted torques and contact parameters supervised by offline dynamics fitting (Xing et al., 2026). SafeFlow distills samples guided by kinematic penalties on joint-limit violations, self-collisions, and joint and center-of-mass velocities and accelerations (Cho et al., 2026). Preference-based post-training methods such as PhysMoDPO (Zhang et al., 2026) also rely on a physics-based tracking controller. Physics-based corrections can introduce style drift (Zhang et al., 2026), and incorporating physics into the training pipeline does not guarantee feasible generation at inference time. For joint state–action models, this limitation also manifests as potential disagreement between predicted states and the motion produced by executing the generated actions, particularly under out-of-distribution conditioning. Our refinement evaluates explicit constraints on generated motion at inference time without training additional generator components or requiring a tracking controller.

Motion Diffusion Model Optimization Finally, physical feasibility of MDMs can be enforced *at inference time* by modifying diffusion sampling. *Gradient-based guidance* steers sampling using differentiable penalties for kinematic constraints (Cho et al., 2026). PhysDiff (Yuan et al., 2023) injects physics through *intermittent simulator projection*: at selected denoising steps, the partially denoised motion is tracked in simulation and denoising resumes from the resulting closed-loop trajectory. CLoSD (Tevet et al., 2025) adopts a related approach in an *autoregressive* setting, where short motion segments are repeatedly planned and executed. The motion actually executed by the simulated character becomes the prefix for the next diffusion plan, closing the loop through the tracking controller; our autoregressive procedure instead conditions on the preceding optimized motion. While both PhysDiff and CLoSD use physics-based tracking to project kinematic into dynamically consistent motions, PhysDiff applies projection within the diffusion trajectory, whereas CLoSD applies it between successive planning steps.

Improving both kinematic and contact-rich dynamical feasibility under non-smooth constraints while preserving motion quality, style, and diversity remains challenging, particularly without retraining the generator or relying on a physics-based tracking controller. Our approach combines gradient-free noise optimization with kinematic penalties and constrained inverse-dynamics evaluations on fully generated motions.

B Motion Diffusion Model: Data, Training, and Generation

B.1 Motion Data and Preprocessing

We train on the *100STYLE* dataset (Mason et al., 2022), which contains locomotion sequences performed in 100 distinct styles. The human motion capture is retargeted to the SMPL skeleton and subsequently to the bipedal robot.

This pipeline follows SMooDi (Zhong et al., 2024), using Rokoko (Rokoko Electronics, 2026) to map the source motion capture onto the SMPL skeleton and obtain retargeted motions for all 100 styles. SMPL's human-like proportions allow this mapping to be performed kinematically.

The bipedal robot differs substantially from the human body in size, proportions, and kinematic structure. We therefore retarget the SMPL motions using ReActor (Müller et al., 2026). The simulated retargeting controller tracks scaled SMPL motion. A fixed correspondence maps the pelvis, hips, knees, feet, shoulders, elbows, neck, and head between the two embodiments. The reference motion is scaled by 0.5; the controller tracks root pose, root velocity, and corresponding body orientations, with auxiliary root forces and torques penalized in the retargeting reward. The resulting robot-motion files provide training data for the bipedal robot. This retargeting stage is distinct from the downstream tracking comparison in Appendix F.7.

The motion loader resamples the SMPL and bipedal robot clips to $30\,\mathrm{Hz}$ and $50\,\mathrm{Hz}$, respectively. It constructs every contiguous window of $P + F$ frames with stride one and omits clips shorter than this window. The past and future parts are expressed in the heading frame associated with the history. Position features retain metric units; each three-component position occupies a six-entry feature block with three zero-padding entries. Root, joint, and body orientations use the 6D representation described below. Style labels are obtained from the motion-file style identifiers.

B.2 Motion Representation, Architecture, and Conditioning

Our MDM builds on the CAMDM transformer architecture (Chen et al., 2024a) with latent width 256 (4 layers, 4 heads, feed-forward size 1024, dropout 0.2) and predicts motions over a horizon of $F = 20$ frames at $30\,\mathrm{Hz}$ for the SMPL humanoid and $F = 30$ at $50\,\mathrm{Hz}$ for the bipedal robot. It is trained on the *100STYLE* dataset (Mason et al., 2022). The model is conditioned on (i) the past-motion window $\mathbf{M}_{\mathrm{pst},t}$, with $P = 10$ frames for SMPL and $P = 15$ for the bipedal robot (expressed in the current heading frame $\mathcal{H}_t$), (ii) a desired root pose sequence $\mathbf{C}_{\mathrm{fut},t}$ in the same frame derived from velocity commands (local x–y translational velocity and yaw rate), (iii) a discrete motion-style label ℓ_{syl}, and (iv) scalar attention biases $b_{\mathrm{pst}}, b_{\mathrm{syl}}$ that modulate the influence of past-motion and style conditioning. We collect these inputs as $\mathbf{c}_t = (\mathbf{M}_{\mathrm{pst},t}, \mathbf{C}_{\mathrm{fut},t}, \ell_{\mathrm{syl}}, b_{\mathrm{pst}}, b_{\mathrm{syl}})$. Conditioning is encoded in a tokenized representation: at each denoising step, we embed a timestep token, a global style/condition token, and per-frame tokens for the commanded root trajectory and the past-motion context, concatenate them with the per-frame future-motion tokens to be denoised, add positional encodings, and process the resulting sequence with a Transformer encoder; the scalar conditioning weights enter as learned attention-mask biases on the corresponding token blocks, and Classifier-Free Guidance (CFG) is implemented by randomly masking the past-motion condition during training (mask probability 0.15).

Orientations use the continuous 6D rotation representation (Zhou et al., 2019). The conversion from generated motion features to physical robot configurations is described in Appendix C.1.

The two models share the transformer dimensions above. SMPL uses $J = 23$ and $n_{\mathrm{b}} = 24$, whereas the bipedal robot uses $J = 20$ and $n_{\mathrm{b}} = 21$. Consequently, $n_{\mathrm{enc}} = 438$ for SMPL and $n_{\mathrm{enc}} = 384$ for the bipedal robot. Although the bipedal robot has revolute joints, the denoiser represents each joint orientation with six features.

B.3 Training Procedure and Hyperparameters

Forward Diffusion The forward diffusion process gradually perturbs a clean motion sequence $\mathbf{M}^0_{\mathrm{fut},t}$ over D diffusion steps into noise $\mathbf{M}^D_{\mathrm{fut},t}$ via

$$\mathbf{M}^d_{\mathrm{fut},t} \;=\; \sqrt{\bar{\alpha}_d}\,\mathbf{M}^0_{\mathrm{fut},t} \;+\; \sqrt{1-\bar{\alpha}_d}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{10}$$

with variance schedule $\{\beta_d\}_{d=1}^{D}$, $\alpha_d = 1 - \beta_d$, and $\bar{\alpha}_d := \prod_{j=1}^{d} \alpha_j$. Gaussian matrix noise denotes independent standard-normal entries, or equivalently a standard-normal distribution after vectorization.

We train on 100STYLE motions retargeted to the target embodiment. The training objective combines MSE reconstruction of clean motion features with losses on consecutive-frame feature differences, body positions obtained by forward kinematics, and consecutive-frame differences of these body positions. Following the geometric-loss construction of Tevet et al. (2023), a foot-contact term penalizes predicted foot displacement at frames where the reference foot displacement falls below a contact threshold.

B.4 Diffusion Sampling and Autoregressive Motion Generation

The learned reverse process uses an ancestral DDPM sampler with an x_0-prediction parameterization. Concretely, we initialize with latent noise $\mathbf{M}_{\mathrm{fut},t}^{D} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and for $d = D, D-1, \ldots, 1$ perform

$$\widehat{\mathbf{M}}_{\mathrm{fut},t|d}^{0} = \mathcal{M}_\theta\big(\mathbf{M}_{\mathrm{fut},t}^{d},\, d,\, \mathbf{c}_t\big)\,, \tag{11}$$

$$\mathbf{M}_{\mathrm{fut},t}^{d-1} = \boldsymbol{\mu}_d^{\mathrm{diff}}\Big(\mathbf{M}_{\mathrm{fut},t}^{d}, \widehat{\mathbf{M}}_{\mathrm{fut},t|d}^{0}\Big) + \sigma_d^{\mathrm{diff}}\,\mathbf{Z}_d, \quad \mathbf{Z}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{12}$$

where $\mathcal{M}_\theta$ is the denoising neural network predicting the clean sequence $\widehat{\mathbf{M}}_{\mathrm{fut},t|d}^{0}$. The reverse-transition mean $\boldsymbol{\mu}_d^{\mathrm{diff}}$ is given in closed form by the diffusion posterior,

$$\boldsymbol{\mu}_d^{\mathrm{diff}} \;=\; a_d\,\widehat{\mathbf{M}}_{\mathrm{fut},t|d}^{0} \;+\; b_d\,\mathbf{M}_{\mathrm{fut},t}^{d}, \tag{13}$$

with scalar schedule coefficients $\{a_d, b_d\}$, and $(\sigma_d^{\mathrm{diff}})^2$ set by the (clipped) posterior variance schedule (Ho et al., 2020). In many implementations, the final step returns the mean (i.e., no noise is added for $d = 1$). The conditioning $\mathbf{c}_t$ is intentionally kept general and may include, for example, past motion, task descriptions, the current robot state, sensor measurements, style commands, or any combination thereof.

We use a DDPM (Ho et al., 2020) with a cosine variance schedule β_d and $D = 8$ diffusion steps. The forward process perturbs $\mathbf{M}_{\mathrm{fut},t}^{0}$ according to Eq. (10). Reverse sampling starts from $\mathbf{M}_{\mathrm{fut},t}^{D}$ and proceeds through $d = D, \ldots, 1$, using a denoiser trained to predict $\widehat{\mathbf{M}}_{\mathrm{fut},t|d}^{0}$ and the corresponding posterior with fixed "small" variance.

At each autoregressive planning call at frame t, the sampler receives $\mathbf{M}_{\mathrm{pst},t}$ and predicts F future frames. We commit the first F_{keep} frames from the best evaluated candidate before replanning, leaving $F_{\mathrm{ov}} = F - F_{\mathrm{keep}}$ overlapping future frames. The next history is $\mathbf{M}_{t+F_{\mathrm{keep}}-P+1:t+F_{\mathrm{keep}}}$, using the supplied initial history where needed. At $t = 0$, the procedure accepts an initial history or the model's neutral history condition. In our experiments, we set $F_{\mathrm{keep}} = 15$ for the SMPL humanoid and $F_{\mathrm{keep}} = 25$ for the bipedal robot, and use a diffusion inference batch size of 64 for all optimization methods. The experimental history initialization is specified below.

We use full conditional sampling with a classifier-free guidance multiplier of one. The attention biases b_{pst} and b_{syl} are specified in Appendix F.3; constraint guidance is a separate baseline described in Appendix E.5. At the first horizon, $\mathbf{M}_{\mathrm{pst},0}$ is an all-zero encoded history condition. Subsequent calls take the last P committed frames as history. The zero feature condition is an input to the denoiser, rather than a prescribed physical joint configuration.

At each planning call, the planar velocity command is integrated over the prediction horizon in the current heading frame. Yaw is accumulated from ω_z^{cmd}, and the commanded planar displacements are rotated by the evolving yaw before accumulation. The command is held fixed within that horizon. After committing F_{keep} frames, the heading frame is rebuilt from the last committed root pose and the reference trajectory is constructed again. The remaining F_{ov} frames of the previous prediction supply the overlap-consistency target.

C Task Objectives and Physical Feasibility Constraints

This appendix formalizes all relevant optimization objectives and physical feasibility constraints in depth, unless this has already been done in the main manuscript. Table 4 reports the active optimization terms for the two physical feasibility optimization settings considered in this paper: (1) kinematic feasibility optimization, and (2) dynamic feasibility optimization.

Table 4: Objectives and constraints actively included in the optimization loss for the two feasibility optimization settings. A checkmark indicates a nonzero loss weight; terms without a checkmark may still be evaluated for reporting.

Term	Kinematic	Dynamic
Objectives		
Velocity tracking $\mathcal{J}_{\text{track}}$	✓	✓
Autoregressive motion consistency $\mathcal{J}_{\text{cons}}$	✓	✓
Equality constraints		
Ground slip $\mathbf{h}_{\text{slip}}$	✓	✓
Acceleration reconstruction $\mathbf{h}_{\text{acc}}$	✗	✓
Inequality constraints		
Ground penetration g_{pen}	✗	✗
Contact-pattern preservation g_{contact}	✓	✗
Self-collision g_{self}	✓	✓
Joint-torque limits g_{torque}	✗	✗
Friction cone g_{fric}	✗	✗
Unilateral contact g_{uni}	✗	✗
Diagnostics (not optimized)		
Smoothness diagnostics (LDJ and jerk)	✗	✗
Reported inverse-dynamics residual E_{ID}	✗	✗

C.1 Robot Configuration and Dynamics

For physical-feasibility evaluation, we decode the generated motion features $\mathbf{m}_\ell$ into the robot configuration $\mathbf{s}_\ell = \mathcal{P}(\mathbf{m}_\ell)$:

$$\mathbf{s}_\ell = (\mathbf{x}_\ell, \mathbf{R}_{\text{root},\ell}, \mathbf{q}_\ell), \qquad \mathbf{R}_{\text{root},\ell} \in SO(3), \quad \mathbf{q}_\ell \in \mathcal{Q}_{\text{joint}}. \tag{14}$$

Here $\mathbf{x}_\ell$ and $\mathbf{R}_{\text{root},\ell}$ describe the root pose, and $\mathbf{q}_\ell$ collects the joint configurations: 23 spherical-joint rotations for SMPL and 20 revolute-joint angles for the bipedal robot. The decoder $\mathcal{P}$ recovers rotations from the generated 6D features. For the bipedal robot, it extracts each scalar joint angle from the decoded axis-angle vector, assigning its magnitude the sign of its dot product with the configured joint axis.

Generated root poses are expressed in the heading frame $\mathcal{H}_t$ of the last history frame. Its origin is the ground-projected root position and its yaw is aligned with the root; it remains fixed across the feature window. The decoder transforms the root poses into the world frame $\mathcal{W}$, in which physical positions are expressed. The rotation $\mathbf{R}_{\text{root},\ell}$ maps root-frame vectors into $\mathcal{W}$. Physical-feasibility evaluation uses link and contact geometry computed by forward kinematics from $\mathbf{s}_\ell$. The predicted per-body position and orientation features enter the motion-consistency objective.

We use generalized velocities and accelerations (Featherstone, 2008)

$$\boldsymbol{\nu}_\ell = (\mathbf{v}_\ell, \boldsymbol{\omega}_\ell, \mathbf{v}_{\text{joint},\ell}) \in \mathbb{R}^{n_\nu}, \qquad \dot{\boldsymbol{\nu}}_\ell \in \mathbb{R}^{n_\nu}, \qquad n_\nu = 6 + n_{\text{joint}}. \tag{15}$$

Here $\mathbf{v}_{\text{joint},\ell} \in \mathbb{R}^{n_{\text{joint}}}$ collects the joint generalized velocities, and n_{joint} is the number of joint velocity degrees of freedom: 69 for SMPL and 20 for the bipedal robot. Root linear and angular velocities are expressed in world coordinates. Feasibility evaluation uses the decoded configurations together with the velocity and acceleration estimates described in Appendix C.2.

Rigid Body Dynamics For the per-frame dynamics equations below, we omit the time subscript ℓ for readability. The floating-base equations of motion are (Featherstone, 2008)

$$\mathbf{B}(\mathbf{s})\dot{\boldsymbol{\nu}} + \mathbf{n}(\mathbf{s}, \boldsymbol{\nu}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_{\mathrm{c}}(\mathbf{s})^\top \boldsymbol{\lambda}. \tag{16}$$

The mass matrix is $\mathbf{B} \in \mathbb{R}^{n_\nu \times n_\nu}$, and $\mathbf{n} \in \mathbb{R}^{n_\nu}$ contains gravity, Coriolis, centrifugal, and any other modeled bias forces. Assuming all joint velocity degrees of freedom are actuated, $\mathbf{S} = [\mathbf{0}\ \mathbf{I}_{n_{\mathrm{joint}}}] \in \mathbb{R}^{n_{\mathrm{joint}} \times n_\nu}$ selects joint torques $\boldsymbol{\tau} \in \mathbb{R}^{n_{\mathrm{joint}}}$.

At frame ℓ, $\mathcal{C}_\ell \subseteq \{1, \ldots, n_{\mathrm{c}}\}$ is the active contact set, with $n_{\lambda,\ell} = 3|\mathcal{C}_\ell|$. The stacked contact Jacobian has dimension $n_{\lambda,\ell} \times n_\nu$, and $\boldsymbol{\lambda}$ stacks 3D point-contact forces. Force components and Jacobian rows use the same contact-local frames. For each active contact, impose the unilateral friction cone

$$\mathcal{K}_p = \{\boldsymbol{\lambda}_p : \lambda_{\perp,p} \geq 0, \quad \|\boldsymbol{\lambda}_{\parallel,p}\|_2 \leq \mu_{\mathrm{c},p}\lambda_{\perp,p}\}, \qquad \boldsymbol{\lambda}_p \in \mathcal{K}_p. \tag{17}$$

Here $\lambda_{\perp,p} \in \mathbb{R}$ and $\boldsymbol{\lambda}_{\parallel,p} \in \mathbb{R}^2$ are normal and tangential force components, and $\mu_{\mathrm{c},p}$ is the friction coefficient. We model all active contacts as sticking, $\mathcal{C}_\ell = \mathcal{C}_\ell^{\mathrm{stick}}$, $\mathcal{C}_\ell^{\mathrm{slide}} = \emptyset$. Sticking additionally requires contact kinematics, e.g., $\mathbf{J}_{\mathrm{c},p}\boldsymbol{\nu} = \mathbf{0}$; its acceleration-level condition is $\mathbf{J}_{\mathrm{c},p}\dot{\boldsymbol{\nu}} + \dot{\mathbf{J}}_{\mathrm{c},p}\boldsymbol{\nu} = \mathbf{0}$.

C.2 Motion Smoothing and Derivative Estimation

Whittaker–Eilers Smoothing We apply Whittaker–Eilers (WE) filtering (Eilers, 2003) to the coordinate signals used for derivative estimation. Root positions are filtered directly, and revolute joint angles are unwrapped before filtering. For each root or spherical-joint orientation sequence $\mathbf{R}_\ell$, we form the cumulative incremental rotation-vector signal

$$\boldsymbol{\phi}_{t+1} = \mathbf{0}, \qquad \boldsymbol{\phi}_\ell = \sum_{k=t+2}^{\ell} \mathrm{Log}\left(\mathbf{R}_k \mathbf{R}_{k-1}^\top\right), \qquad \ell = t+2, \ldots, t+F,$$

where $\mathrm{Log} : SO(3) \to \mathbb{R}^3$ returns the rotation vector. For the root, $\mathbf{R}_\ell = \mathbf{R}_{\mathrm{root},\ell}$ and the increments use world coordinates; for spherical joints, $\mathbf{R}_\ell$ is the parent-relative joint rotation and the increments use parent-frame coordinates. The cumulative sum is a signal for derivative estimation, rather than the logarithm of the total relative rotation. Let $\boldsymbol{\Xi} = \boldsymbol{\Xi}_{t+1:t+F} \in \mathbb{R}^{F \times n_\Xi}$ stack the position, unwrapped-angle, and cumulative rotation-vector signals over the predicted frames, with n_Ξ coordinate channels. The filter gives

$$\boldsymbol{\Xi}^{\mathrm{sm}} = \mathcal{S}_{\mathrm{WE}}\boldsymbol{\Xi}, \qquad \mathcal{S}_{\mathrm{WE}} = (\mathbf{I}_F + \kappa_{\mathrm{WE}}\mathbf{D}_2^\top \mathbf{D}_2)^{-1}. \tag{18}$$

The superscript sm denotes the smoothed signals, and $\mathbf{D}_2 \in \mathbb{R}^{(F-2)\times F}$ is the second-difference matrix with row stencil $(1, -2, 1)$. The parameter $\kappa_{\mathrm{WE}} > 0$ controls smoothing strength. The filtered signals provide the basis for velocity and acceleration estimation.

Motion Derivatives Estimation We estimate the coordinate-signal derivatives by repeated application of WE filtering and finite differences:

$$\dot{\boldsymbol{\Xi}} = \mathbf{D}_1 \boldsymbol{\Xi}^{\mathrm{sm}}, \qquad \ddot{\boldsymbol{\Xi}} = \mathbf{D}_1 \mathcal{S}_{\mathrm{WE}} \dot{\boldsymbol{\Xi}}. \tag{19}$$

The differentiation matrix $\mathbf{D}_1 \in \mathbb{R}^{F \times F}$ uses the central stencil $(-1, 0, 1)/(2\Delta t)$ at interior frames and the stencil $(-1, 1)/\Delta t$ on the first and last pairs of frames for the boundary rows. Here, $\Delta t = 1/f_s$, with $f_s = 30\,\mathrm{Hz}$ for SMPL and $f_s = 50\,\mathrm{Hz}$ for the bipedal robot. We assemble the resulting root and joint components into the generalized velocities $\boldsymbol{\nu}_\ell$ and accelerations $\dot{\boldsymbol{\nu}}_\ell$.

Smoothing is used to estimate derivatives; it does not replace the configurations used for geometric evaluation. Ground separation, contact-pattern preservation, and self-collision use forward kinematics of the decoded $\mathbf{s}_\ell$. The tracking and autoregressive-consistency objectives compare the generated motion features directly. Ground slip uses the estimated generalized velocity $\boldsymbol{\nu}_\ell$, and inverse dynamics uses $\mathbf{s}_\ell$, $\boldsymbol{\nu}_\ell$, and $\dot{\boldsymbol{\nu}}_\ell$, including the velocity-dependent torque bounds. Jerk diagnostics use a separate estimator with unsmoothed finite differences, since smoothing would attenuate rapid acceleration changes and obscure the jerk present in the generated motion.

Each optimization evaluation estimates derivatives over its F predicted frames, with the boundary stencils stated above. The previous horizon's overlap enters $\mathcal{J}_{\mathrm{cons}}$ but is not prepended to the derivative-estimation window. The committed per-frame diagnostics are concatenated across horizons.

C.3 Task Objectives

In our experiments, the task objective is $\mathcal{J} = \rho_{\text{track}}\mathcal{J}_{\text{track}} + \rho_{\text{cons}}\mathcal{J}_{\text{cons}}$, combining velocity tracking with the motion-consistency objective in Eq. (21).

Velocity Tracking Objective We track the planar root-velocity command $\mathbf{v}^{\text{cmd}} = (v_x^{\text{cmd}}, v_y^{\text{cmd}}, \omega_z^{\text{cmd}})$. We integrate the commanded planar velocity and yaw rate to obtain a reference root trajectory. Each candidate is evaluated in the heading frame of its first predicted frame, denoted here by $\mathcal{H} = \mathcal{H}_{t+1}$. The trajectory-tracking objective combines errors in horizontal root position and 6D root orientation:

$$\mathcal{J}_{\text{track}}(\mathbf{M}) = \frac{1}{F}\sum_{f=1}^{F}\left[\frac{\|{}^{\mathcal{H}}\mathbf{x}_{\text{xy},t+f} - {}^{\mathcal{H}}\mathbf{x}^{\text{ref}}_{\text{xy},t+f}\|_2}{\sqrt{2}} + \frac{\|{}^{\mathcal{H}}\mathbf{r}_{t+f} - {}^{\mathcal{H}}\mathbf{r}^{\text{ref}}_{t+f}\|_2}{\sqrt{6}}\right]. \tag{20}$$

The left superscript $\mathcal{H}$ indicates the frame in which a feature is expressed. Here, ${}^{\mathcal{H}}\mathbf{x}^{\text{ref}}_{\text{xy},\ell}$ and ${}^{\mathcal{H}}\mathbf{r}^{\text{ref}}_{\ell}$ are the command-derived reference position and orientation features, expressed in $\mathcal{H}$. The subscript xy selects the two horizontal position components. Reference positions accumulate the commanded planar displacements in the evolving heading frame, and reference orientations encode the integrated yaw. Position features are expressed in meters, and orientation features are dimensionless.

Auto-Regressive Motion Consistency Objective We encourage agreement between successive predictions over their overlapping future frames. We express the full encoded motion features in the common world frame. With $\mathbf{m}_{\ell}^{\text{enc,prev}}$ denoting the prediction from the preceding horizon, we define

$$\mathcal{J}_{\text{cons}}(\mathbf{M}) = \frac{1}{F n_{\text{enc}}}\sum_{f=1}^{F_{\text{ov}}}\|\mathbf{m}^{\text{enc}}_{t+f} - \mathbf{m}^{\text{enc,prev}}_{t+f}\|_2^2, \qquad F_{\text{ov}} = F - F_{\text{keep}}. \tag{21}$$

The objective measures the feature discrepancy over the overlap, averaged over the full prediction horizon. We set $\mathcal{J}_{\text{cons}} = 0$ for the first horizon and whenever $F_{\text{ov}} = 0$.

Objective weights are given in Appendix F.3.

C.4 Kinematic Feasibility Constraints

Ground Penetration and Slip We measure ground penetration and slip by averaging their magnitudes over the n_{c} candidate contact points. With $[x]_+ = \max(x, 0)$, the per-frame violations are

$$v_{\text{pen},\ell} = \frac{1}{n_{\text{c}}}\sum_{p=1}^{n_{\text{c}}}[-\delta_{\ell,p}]_+, \qquad v_{\text{slip},\ell} = \frac{1}{n_{\text{c}}}\sum_{p=1}^{n_{\text{c}}} a_{\ell,p}\|\mathbf{v}_{\parallel,\ell,p}\|_2. \tag{22}$$

Here, $\mathbf{v}_{\parallel,\ell,p} = \mathbf{v}_{\parallel,p}(\mathbf{s}_\ell, \boldsymbol{\nu}_\ell)$ and $a_{\ell,p}$ indicates an active contact. For flat ground, $a_{\ell,p} = \mathbf{1}[\delta_{\ell,p} \le 0]$, where $\mathbf{1}[\cdot]$ denotes the indicator function. Tangential velocity comprises the horizontal world-frame components.

Contact Pattern Preservation We infer a reference contact pattern by a strict majority vote over an initial population of N_{pop} unoptimized motion samples:

$$a^{\text{ref}}_{\ell,p} = \mathbf{1}\left[\sum_{i=1}^{N_{\text{pop}}}\mathbf{1}[\delta^{(i)}_{\ell,p} \le \delta_{\text{vote}}] > \frac{N_{\text{pop}}}{2}\right], \tag{23}$$

where $\delta^{(i)}_{\ell,p}$ is the signed terrain separation in sample i, and δ_{vote} is the separation threshold used for pattern identification. The reference pattern is held fixed during optimization of each horizon and inferred again at the next autoregressive step. The thresholds δ_{vote}, δ_{pen}, and δ_{clear} govern reference-pattern identification and preservation; they do not change the physical contact-activation rule $a_{\ell,p} = \mathbf{1}[\delta_{\ell,p} \le 0]$ defined in Sec. 6.2. The indicator $\mathbf{1}[\cdot]$ equals one when its argument is true and zero otherwise. The pointwise penalty is defined in Eq. (6). The per-frame violation is $v_{\text{contact},\ell} = \frac{1}{n_{\text{c}}}\sum_{p=1}^{n_{\text{c}}} g_{\text{contact},\ell,p}$, averaging these nonnegative components over all contact points.

An example of the contact pattern identification based on the initial motion and subsequent enforcement during optimization, including the thresholds, can be found in Figs. 17-18 of the appendix.

Self-Collision For self-collision, let $\mathcal{B}_{\text{col}}$ be the set of checked body pairs and $\delta^{\text{self}}_{\ell,bb'}$ their signed separation. The pointwise condition is $g_{\text{self},\ell,bb'} = -\delta^{\text{self}}_{\ell,bb'} \leq 0$ for $(b,b') \in \mathcal{B}_{\text{col}}$. Each checked body is approximated by an oriented bounding box. Native boxes retain their dimensions; a sphere is enclosed by a cube of side twice its radius, and a cylinder by a box spanning its diameter and length. Mesh geometry is enclosed by a box aligned with principal axes computed from its convex hull. The box pose includes the geometry's local offset and orientation and the body's forward-kinematic transform. We apply Separating Axis Theorem (SAT) (Bittle, 2010) to each pair in $\mathcal{B}_{\text{col}}$ and use the minimum overlap of the tested axes as penetration depth when the boxes intersect. The per-frame mean self-collision violation is

$$v_{\text{self},\ell} = \frac{1}{|\mathcal{B}_{\text{col}}|} \sum_{(b,b')\in\mathcal{B}_{\text{col}}} [-\delta^{\text{self}}_{\ell,bb'}]_+. \tag{24}$$

The maximum over checked pairs is also available as a diagnostic. These box approximations and the selected pair set define the collision measure; it is not a mesh-intersection volume.

C.5 Dynamic Feasibility Constraints

C.5.1 Nominal Constrained Inverse Dynamics

Following the procedure established for optimization-based control of legged robots (Righetti et al., 2011; 2013; Wensing et al., 2023), we estimate for each generated motion state with associated time derivatives $(\mathbf{s}, \boldsymbol{\nu}, \dot{\boldsymbol{\nu}})$, joint torques and contact forces approximating the motion-implied dynamics by solving floating-base ID as a constrained least-squares problem based on the tuple of generalized coordinates, velocities and accelerations. Let $\mathbf{y} := [\boldsymbol{\tau}^\top, \boldsymbol{\lambda}^\top]^\top$ stack the joint torques $\boldsymbol{\tau} \in \mathbb{R}^{n_{\text{joint}}}$ and the 3D contact forces $\boldsymbol{\lambda} \in \mathbb{R}^{3|\mathcal{C}|}$ (one force per active contact point $p \in \mathcal{C}$). Using a root-first row ordering, we rewrite the Equations of Motion (EOM) in the linear form

$$\mathbf{A}_{\text{ID}}\,\mathbf{y} = \mathbf{b}_{\text{ID}}, \quad \mathbf{A}_{\text{ID}} = \begin{bmatrix} \mathbf{0} & \mathbf{J}^\top_{\text{c,root}}(\mathbf{s}) \\ \mathbf{I}_{n_{\text{joint}}} & \mathbf{J}^\top_{\text{c,joint}}(\mathbf{s}) \end{bmatrix}, \quad \mathbf{b}_{\text{ID}} = \mathbf{B}(\mathbf{s})\,\dot{\boldsymbol{\nu}} + \mathbf{n}(\mathbf{s},\boldsymbol{\nu}), \tag{25}$$

where $\mathbf{J}_{\text{c,root}} \in \mathbb{R}^{n_\lambda \times 6}$ and $\mathbf{J}_{\text{c,joint}} \in \mathbb{R}^{n_\lambda \times n_{\text{joint}}}$ denote the contact Jacobian columns for the root and joint coordinates, respectively. Equivalently, $\mathbf{A}_{\text{ID}} = [\mathbf{S}^\top\ \mathbf{J}_{\text{c}}^\top]$. We solve for $\mathbf{y}$ by minimizing the squared generalized acceleration mismatch

$$\begin{aligned} \mathbf{y}^{\text{ID}} \in \arg\min_{\mathbf{y}} \quad & \|\mathbf{B}^{-1}(\mathbf{s})\,(\mathbf{A}_{\text{ID}}\,\mathbf{y} - \mathbf{b}_{\text{ID}})\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\tau}_{\text{lb}}(\mathbf{v}_{\text{joint}}) \ \leq\ \boldsymbol{\tau} \ \leq\ \boldsymbol{\tau}_{\text{ub}}(\mathbf{v}_{\text{joint}}), \\ & \boldsymbol{\lambda}_p \in \mathcal{K}_p, \qquad \forall p \in \mathcal{C}. \end{aligned} \tag{26}$$

The inner inverse-dynamics solution $\mathbf{y}^{\text{ID}}$ supplies joint torques and contact forces subject to the above constraints. The generalized force residual compares the forces supplied by this solution with those required by the generated motion. Applying the inverse mass matrix converts this residual into the corresponding acceleration mismatch:

$$\mathbf{e}_{\text{dyn}} = \underbrace{\mathbf{A}_{\text{ID}}\mathbf{y}^{\text{ID}}}_{\text{ID generalized force}} - \underbrace{\mathbf{b}_{\text{ID}}}_{\text{required generalized force}}, \tag{27}$$

$$\dot{\boldsymbol{\nu}}^{\text{ID}} = \mathbf{B}(\mathbf{s})^{-1}(\mathbf{A}_{\text{ID}}\mathbf{y}^{\text{ID}} - \mathbf{n}), \tag{28}$$

$$\mathbf{e}_{\text{acc}} = \mathbf{B}(\mathbf{s})^{-1}\mathbf{e}_{\text{dyn}} = \underbrace{\dot{\boldsymbol{\nu}}^{\text{ID}}}_{\text{ID acceleration}} - \underbrace{\dot{\boldsymbol{\nu}}}_{\text{motion acceleration}}. \tag{29}$$

Here, $\dot{\boldsymbol{\nu}}^{\text{ID}}$ is the acceleration induced by the returned torques and contact forces at the same state $(\mathbf{s}, \boldsymbol{\nu})$, whereas $\dot{\boldsymbol{\nu}}$ is the acceleration of the generated motion. The superscript ID denotes quantities obtained from the inner inverse-dynamics solve.

Joint Torque Limits The joint torque limits are either static bounds $[\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}]$ or velocity-dependent bounds with joint-specific slopes $\boldsymbol{\kappa}_\tau \in \mathbb{R}^{n_{\text{joint}}}$, offsets $\mathbf{o}_\tau \in \mathbb{R}^{n_{\text{joint}}}_{\geq 0}$ and saturation to the interval $[\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}]$

$$\boldsymbol{\tau}_{\text{lb}}(\mathbf{v}_{\text{joint}}) = \max\big(\boldsymbol{\kappa}_\tau \odot \mathbf{v}_{\text{joint}} - \mathbf{o}_\tau, -\boldsymbol{\tau}_{\max}\big), \quad \boldsymbol{\tau}_{\text{ub}}(\mathbf{v}_{\text{joint}}) = \min\big(\boldsymbol{\kappa}_\tau \odot \mathbf{v}_{\text{joint}} + \mathbf{o}_\tau, \boldsymbol{\tau}_{\max}\big), \tag{30}$$

where the displayed saturation assumes symmetric static bounds $\boldsymbol{\tau}_{\min} = -\boldsymbol{\tau}_{\max}$, $\odot$ denotes componentwise multiplication, and $\max$ and $\min$ act componentwise. The slopes and offsets can be determined via actuator system identification (Grandia et al., 2024; Miller et al., 2025).

C.5.2 Relaxed Inverse Dynamics with a Virtual Wrench

For analysis, we also consider a relaxed inverse-dynamics problem that allows a virtual force and torque to act directly on the floating base. This removes the restriction imposed by floating-base underactuation while retaining the joint actuation limits and contact-force constraints. The formulation therefore lets us examine the remaining dynamical feasibility conditions when direct root support is available. Its objective is to find the smallest virtual wrench compatible with these constraints and the planned motion, quantifying the additional root support required. We use this formulation only for analysis and diagnostics; physical-feasibility optimization uses the nominal inverse-dynamics problem in Eq. (26).

Let $\boldsymbol{\xi}_{\mathrm{virt}} \in \mathbb{R}^6$ denote the virtual floating-base wrench, and define

$$\mathbf{y}^{\mathrm{virt}} = [\boldsymbol{\tau}^\top, \boldsymbol{\lambda}^\top, \boldsymbol{\xi}_{\mathrm{virt}}^\top]^\top, \qquad \mathbf{A}_{\mathrm{ID}}^{\mathrm{virt}} = \begin{bmatrix} \mathbf{0} & \mathbf{J}_{\mathrm{c,root}}^\top(\mathbf{s}) & \mathbf{I}_6 \\ \mathbf{I}_{n_{\mathrm{joint}}} & \mathbf{J}_{\mathrm{c,joint}}^\top(\mathbf{s}) & \mathbf{0} \end{bmatrix}. \tag{31}$$

The virtual wrench acts on the root rows of the equations of motion. Solve

$$\begin{aligned} \mathbf{y}^{\mathrm{virt,ID}} \in \arg\min_{\mathbf{y}^{\mathrm{virt}}} \quad & \|\boldsymbol{\xi}_{\mathrm{virt}}\|^2_{\mathbf{W}_{\mathrm{virt}}} \\ \text{s.t.} \quad & \mathbf{A}_{\mathrm{ID}}^{\mathrm{virt}} \mathbf{y}^{\mathrm{virt}} = \mathbf{b}_{\mathrm{ID}}, \\ & \boldsymbol{\tau}_{\mathrm{lb}}(\mathbf{v}_{\mathrm{joint}}) \le \boldsymbol{\tau} \le \boldsymbol{\tau}_{\mathrm{ub}}(\mathbf{v}_{\mathrm{joint}}), \\ & \boldsymbol{\lambda}_p \in \mathcal{K}_p, \qquad p \in \mathcal{C}. \end{aligned} \tag{32}$$

Here $\mathbf{W}_{\mathrm{virt}} \succeq 0$ is a diagonal weighting matrix and $\|\boldsymbol{\xi}\|^2_{\mathbf{W}} = \boldsymbol{\xi}^\top \mathbf{W} \boldsymbol{\xi}$. For a feasible solution, denote its wrench by $\boldsymbol{\xi}_{\mathrm{virt}}^{\mathrm{ID}}$. The zero-wrench constraint and its mean-absolute violation are

$$\begin{aligned} \mathbf{h}_{\mathrm{virt},\ell}(\mathbf{M}) &= \boldsymbol{\xi}_{\mathrm{virt},\ell}^{\mathrm{ID}} = \mathbf{0}, \\ v_{\mathrm{virt},\ell} &= \frac{1}{6} \|\mathbf{h}_{\mathrm{virt},\ell}(\mathbf{M})\|_1, \qquad \ell = t+1, \ldots, t+F. \end{aligned} \tag{33}$$

The horizon mean $F^{-1} \sum_{\ell=t+1}^{t+F} v_{\mathrm{virt},\ell}$ is zero exactly when the returned virtual wrench vanishes at every evaluated frame.

D Numerical Implementation of Batched Inverse Dynamics

D.1 Solver Formulation and Numerical Conditioning

The differentiable floating-base dynamics in Eq. (16) are implemented in PyTorch using a custom version of `adam` (Nori et al., 2015; L'Erario et al., 2026) that supports spherical joints. Each spherical joint has four quaternion position coordinates and three angular-velocity coordinates; the implementation accounts for these different position and velocity dimensions in the rigid-body dynamics.

In the ID formulation, we assume that $\mathcal{C}_\ell = \mathcal{C}_\ell^{\text{stick}}$, $\mathcal{C}_\ell^{\text{slide}} = \emptyset$. Equation (26) has a convex quadratic objective and second-order cone constraints (friction cones). We solve it as an inner-loop Second-Order Cone Program (SOCP) with Clarabel (Goulart & Chen, 2024), using DiffCP (Agrawal et al., 2019b) and the differentiable CVXPYlayers PyTorch interface (Agrawal et al., 2019a).

Motion generation, forward kinematics, and dynamics tensors use single precision. The CVXPYlayers problem data are converted to double precision for the CPU solve, and the returned solution is converted back to the dynamics tensor precision and device. Acceleration reconstruction uses a linear solve with $\mathbf{B}$.

D.2 Batching and Computational Efficiency

At each diffusion noise optimization step, we group the SOCPs instances (i.e., per individual in the population and per motion-plan time step) by the tuple of active contact-point identities. For each group, we compile a dedicated CVXPY problem and then solve it for all samples in the batch in parallel on the CPU. This approach avoids recompilation as stances change, while also sidestepping the numerical issues that can arise from "Big-M" reformulations of disjunctive contact constraints.

The CVXPY layer is cached for each active-contact tuple and reused when that tuple recurs. All candidate/frame instances with the same active-contact tuple are stacked into one layer call. The CPU backend uses eight forward workers by default, with an environment-variable override available.

D.3 Differentiation for Gradient-Based Baselines

Gradient-based methods differentiate the scalar batch-mean loss through the optimized motion or noise, rotation decoding, forward kinematics, derivative estimation, and, when active, the CVXPYlayers/DiffCP inverse-dynamics solve. Denoiser parameters remain fixed. Hard contact masks and the grouping of active contacts are discrete selections; derivatives follow the selected problem and do not differentiate a change of contact set. Absolute values, clamping, vector norms, and separating-axis minima use the piecewise derivatives provided by PyTorch.

E Physical Feasibility Optimization and Baselines

E.1 Constraint Softening and Candidate Evaluation

The soft objective in Eq. (8) adds weighted squared feasibility violations to the task objective $\mathcal{J}$. The sets $\mathcal{T}_{\text{h}}$ and $\mathcal{T}_{\text{g}}$ contain the equality- and inequality-constraint term labels, respectively, including $\text{slip}, \text{acc} \in \mathcal{T}_{\text{h}}$ and $\text{pen}, \text{contact}, \text{self} \in \mathcal{T}_{\text{g}}$. Each $v_{j,\ell}$ is a nonnegative per-frame violation aggregated over the relevant contacts, collision pairs, or generalized coordinates. We square these scalar violations and average over future frames $f = 1, \ldots, F$, with absolute frame $\ell = t + f$. The nonnegative weights $\rho_{\text{h},j}$ and $\rho_{\text{g},j}$ determine each term's contribution. The fitness of unnormalized noise is

$$f(\mathbf{w}^{\text{raw}}; \mathbf{c}_t) = \mathcal{L}(\Phi_\theta(\Psi(\mathbf{w}^{\text{raw}}), \mathbf{c}_t)). \tag{34}$$

E.2 Noise Parameterization and Optimizer Initialization

The noise vector in Sec. 5 contains $(D+1)Fn_{\text{enc}}$ scalar optimization parameters per candidate. With $D = 8$, this gives $9 \cdot 20 \cdot 438 = 78{,}840$ for SMPL and $9 \cdot 30 \cdot 384 = 103{,}680$ for the bipedal robot.

The sampler directly injects $D-1$ posterior-noise tensors, since its final transition returns the mean, but all D allocated tensors enter posterior-noise standardization.

For each candidate, Eq. (3) standardizes the initial-noise tensor and the combined posterior-noise tensor separately, using their empirical means and sample standard deviations. The denominator floor $\varepsilon_{\mathrm{norm}}$ is the machine epsilon of the noise dtype.

For each run (Sec. 7), initialize the optimizer mean with independent standard-normal entries and set the remaining state to

$$\begin{aligned} \boldsymbol{\mu}_0^{\mathrm{ES}} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_w}), & \sigma_0^{\mathrm{ES}} &= 1, \\ \mathbf{a}_0^{\mathrm{ES}} &= \mathbf{1}, & \mathbf{C}_0^{\mathrm{ES}} &= \mathbf{I}_{n_w}, \\ \mathbf{p}_{\sigma,0} &= \mathbf{0}, & \mathbf{p}_{C,0} &= \mathbf{0}. \end{aligned} \tag{35}$$

Use N_{pop} candidates, $N_{\mathrm{sel}} = N_{\mathrm{pop}}/2$ parents, and the recombination weights and adaptation hyperparameters in Appendix E.3. The experimental population size is specified in Appendix F.3.

The methods draw latent and posterior noise from the model's standard-normal distributions within each run. Method-specific handling of these noise samples is described in Appendix E.5.

E.3 Sep-CMA-ES Updates

We minimize f with Sep-CMA-ES (Ros & Hansen, 2008), which restricts the CMA-ES covariance (Hansen & Ostermeier, 2001) to a diagonal matrix. At iteration $k = 0, \ldots, K-1$, the optimizer maintains mean $\boldsymbol{\mu}_k^{\mathrm{ES}} \in \mathbb{R}^{n_w}$, global step size $\sigma_k^{\mathrm{ES}} > 0$, and covariance $\mathbf{C}_k^{\mathrm{ES}} = \mathrm{diag}((\mathbf{a}_k^{\mathrm{ES}})^{\odot 2})$, with positive diagonal scales $\mathbf{a}_k^{\mathrm{ES}}$. It samples N_{pop} candidates:

$$\boldsymbol{\zeta}_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{w}_{k,i}^{\mathrm{raw}} = \boldsymbol{\mu}_k^{\mathrm{ES}} + \sigma_k^{\mathrm{ES}} \mathbf{a}_k^{\mathrm{ES}} \odot \boldsymbol{\zeta}_{k,i}, \qquad i = 1, \ldots, N_{\mathrm{pop}}. \tag{36}$$

Here $\odot$ and $(\cdot)^{\odot 2}$ denote entrywise multiplication and squaring. Rank candidates by $f(\mathbf{w}_{k,i}^{\mathrm{raw}}; \mathbf{c}_t)$ and recombine the best N_{sel}:

$$\boldsymbol{\mu}_{k+1}^{\mathrm{ES}} = \sum_{i=1}^{N_{\mathrm{sel}}} \eta_i \mathbf{w}_{k,i:N_{\mathrm{pop}}}^{\mathrm{raw}}, \qquad \eta_i \propto \log(N_{\mathrm{sel}} + \tfrac{1}{2}) - \log i, \qquad \sum_i \eta_i = 1. \tag{37}$$

The index $i : N_{\mathrm{pop}}$ denotes rank within the population, and $\mu_{\mathrm{eff}} = (\sum_i \eta_i^2)^{-1}$ is the effective number of parents. Define the normalized mean shift and step-size evolution path by

$$\mathbf{u}_k^{\mathrm{ES}} = (\boldsymbol{\mu}_{k+1}^{\mathrm{ES}} - \boldsymbol{\mu}_k^{\mathrm{ES}})/\sigma_k^{\mathrm{ES}}, \tag{38}$$

$$\mathbf{p}_{\sigma,k+1} = (1 - c_\sigma)\mathbf{p}_{\sigma,k} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\mathrm{eff}}}\, (\mathbf{C}_k^{\mathrm{ES}})^{-1/2} \mathbf{u}_k^{\mathrm{ES}}. \tag{39}$$

The step size is updated using cumulative step-size adaptation, with $\chi_{n_w} = \sqrt{n_w}(1 - 1/(4n_w) + 1/(21n_w^2))$:

$$\sigma_{k+1}^{\mathrm{ES}} = \sigma_k^{\mathrm{ES}} \exp\left[\frac{c_\sigma}{d_\sigma}\left(\frac{\|\mathbf{p}_{\sigma,k+1}\|_2}{\chi_{n_w}} - 1\right)\right]. \tag{40}$$

For the diagonal covariance update, let $\widetilde{\mathbf{u}}_{k,i}^{\mathrm{ES}} = (\mathbf{w}_{k,i:N_{\mathrm{pop}}}^{\mathrm{raw}} - \boldsymbol{\mu}_k^{\mathrm{ES}})/\sigma_{k+1}^{\mathrm{ES}}$ and define the path gate

$$h_{\sigma,k} = \mathbf{1}\left[\|\mathbf{p}_{\sigma,k+1}\|_2 < \left(1.4 + \frac{2}{n_w + 1}\right)\chi_{n_w}\sqrt{1 - (1 - c_\sigma)^{2(k+1)}}\right].$$

With $\delta_k = (1 - h_{\sigma,k})c_c(2 - c_c)$, the updates are

$$\mathbf{p}_{C,k+1} = (1 - c_c)\mathbf{p}_{C,k} + h_{\sigma,k}\sqrt{c_c(2 - c_c)\mu_{\mathrm{eff}}}\, \mathbf{u}_k^{\mathrm{ES}}, \tag{41}$$

$$(\mathbf{a}_{k+1}^{\mathrm{ES}})^{\odot 2} = (1 - c_1 - c_\mu + \delta_k)(\mathbf{a}_k^{\mathrm{ES}})^{\odot 2} + c_1 \mathbf{p}_{C,k+1}^{\odot 2} + c_\mu \sum_{i=1}^{N_{\mathrm{sel}}} \eta_i (\widetilde{\mathbf{u}}_{k,i}^{\mathrm{ES}})^{\odot 2}. \tag{42}$$

ALGORITHM 1: Autoregressive motion diffusion noise optimization with Sep-CMA-ES

Input: Frozen sampler Φ_θ, target length T, history length P, horizon F, commitment F_{keep}, iteration budget K

Initialize time and zero encoded history
$t \leftarrow 0$, $\mathbf{M}_{\text{pst},0} \leftarrow \mathbf{0}$;
while $t < T$ (autoregressive motion generation) **do**
  Construct the heading frame $\mathcal{H}_t$ and conditioning from the committed history and current command
  $\mathbf{c}_t \leftarrow (\mathbf{M}_{\text{pst},t}, \mathbf{C}_{\text{fut},t}, \ell_{\text{syl}}, b_{\text{pst}}, b_{\text{syl}})$;
  Initialize the Sep-CMA-ES state using Eq. (35);
  Generate an initial population; infer the reference contacts when contact-pattern preservation is used
  Generate initial motions with $\Phi_\theta \circ \Psi$ at fixed $\mathbf{c}_t$; compute $a^{\text{ref}}_{\ell,p}$ using Eq. (23);
  Hold the inferred contact pattern and previous prediction's overlap fixed within this horizon;
  for $k = 0, \ldots, K-1$ (Sep-CMA-ES optimization) **do**
    foreach $i = 1, \ldots, N_{\text{pop}}$ (population evaluation in parallel) **do**
      Sample raw noise
      $\boldsymbol{\zeta}_{k,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_w}), \quad \mathbf{w}^{\text{raw}}_{k,i} \leftarrow \boldsymbol{\mu}^{\text{ES}}_k + \sigma^{\text{ES}}_k \mathbf{a}^{\text{ES}}_k \odot \boldsymbol{\zeta}_{k,i}$;
      Standardize the noise
      $\mathbf{w}_{k,i} \leftarrow \Psi(\mathbf{w}^{\text{raw}}_{k,i})$;
      Generate motion
      $\mathbf{M}^{0,(k,i)}_{\text{fut},t} \leftarrow \Phi_\theta(\mathbf{w}_{k,i}, \mathbf{c}_t)$;
      Evaluate the fitness
      $f(\mathbf{w}^{\text{raw}}_{k,i}; \mathbf{c}_t) \leftarrow \mathcal{L}(\mathbf{M}^{0,(k,i)}_{\text{fut},t})$;
    end
    Update the Sep-CMA-ES state using Eqs. (37)–(42);
    Retain the best candidate's noise and already evaluated motion across iterations
    $(s^\star, i^\star) \leftarrow \underset{0 \le s \le k,\, 1 \le i \le N_{\text{pop}}}{\arg\min} f(\mathbf{w}^{\text{raw}}_{s,i}; \mathbf{c}_t)$;
    $\mathbf{w}^{\text{raw},\star} \leftarrow \mathbf{w}^{\text{raw}}_{s^\star,i^\star}, \quad \mathbf{M}^{0,\star}_{\text{fut},t} \leftarrow \mathbf{M}^{0,(s^\star,i^\star)}_{\text{fut},t}$;
  end
  Commit its first F_{keep} frames and retain the remaining F_{ov} frames as the next overlap target;
  Advance time and update the history from the last P committed frames
  $t \leftarrow t + F_{\text{keep}}, \quad \mathbf{M}_{\text{pst},t} \leftarrow \mathbf{M}^\star_{t-P+1:t}$;
end

The next scale vector is the entrywise square root. All search vectors and evolution paths have dimension n_w. Writing $n = n_w$ and $m = \mu_{\text{eff}}$, the adaptation hyperparameters are

$$c_\sigma = \frac{m+2}{n+m+5}, \qquad d_\sigma = 1 + 2\max\left(0, \sqrt{\frac{m-1}{n+1}} - 1\right) + c_\sigma,$$

$$c_c = \frac{4 + m/n}{n + 4 + 2m/n}, \qquad c_1 = \min\left(1, \frac{2n}{(n+1.3)^2 + m}\right),$$

$$\bar{c}_\mu = \min\left(1 - c_1, \frac{2(m - 2 + 1/m)}{(n+2)^2 + m}\right), \quad c_\mu = \min(1 - c_1, n\bar{c}_\mu).$$

E.4 Optimization Procedure and Autoregressive Integration

Each population is evaluated with $f(\mathbf{w}^{\text{raw}}; \mathbf{c}_t) = \mathcal{L}(\Phi_\theta(\Psi(\mathbf{w}^{\text{raw}}), \mathbf{c}_t))$, using the generator in Appendix B, the objectives and constraints in Appendix C, and the numerical evaluation in Appendix D. The updates above are repeated for the fixed iteration budget. The best evaluated candidate supplies the committed motion frames and the next history window according to the autoregressive procedure in the main text.

Optimization uses the fixed K-iteration budget. The initial population establishes the reference contact pattern and initial diagnostics; the retained solution is updated by the subsequent optimization iterations. Since Φ_θ is deterministic for fixed noise and conditioning, Alg. 1 retains the winning motion together with its noise, avoiding an additional sampling pass.

E.5 Inference-Time Optimization Baselines and Comparison Budgets

We compare inference-time methods that improve the physical feasibility of generated motion while keeping the MDM fixed. Within each experimental setting, the optimization methods use the same task objectives and feasibility constraints. Approaches that require retraining or finetuning the motion model (Xu et al., 2025; Watanabe et al., 2025), or a pretrained tracking controller for closed-loop simulator rollouts (Yuan et al., 2023; Watanabe et al., 2025; Tevet et al., 2025), fall outside this comparison and are, therefore, out-of-scope for this paper. Appendix F.3 specifies the experimental settings.

E.5.1 Baseline Methods

Unoptimized Generation We generate a batch of stochastic motions from the trained MDM and select the sample whose aggregated loss is closest to the batch median as a representative sample by loss. For an even batch size, the median is the average of the two middle losses.

Motion-Space Optimization We directly post-process the generated motion sequence, analogous in spirit to constraint-aware, physics-informed retargeting of motion capture to robot embodiments (Luo et al., 2023; Zhang et al., 2023; Ze et al., 2025; Yang et al., 2026; Lee et al., 2025; Pan et al., 2025; Dhedin et al., 2026). *Motion GD* updates the generated batch of motions to minimize the aggregated loss and returns the lowest-loss motion after its final update. *Motion Sep-CMA-ES* initializes its search mean with the lowest-loss generated motion, optimizes in motion space, and returns and evaluates the final search mean.

Guided Diffusion We use gradients of the aggregated loss to guide diffusion sampling (Dhariwal & Nichol, 2021; Karunratanakul et al., 2023; Ma et al., 2025; Cheng et al., 2025; Zhang et al., 2025; Wu et al., 2025; Liao et al., 2026). At each denoising step, we evaluate the loss on the predicted clean sample $\widehat{\mathbf{M}}^0_{\mathrm{fut},t|d}$ and shift the posterior mean:

$$\boldsymbol{\mu}_d^{\mathrm{diff}} \;=\; a_d\,\widehat{\mathbf{M}}^0_{\mathrm{fut},t|d} + b_d\,\mathbf{M}^d_{\mathrm{fut},t} - \eta_d^{\mathrm{guide}}\,\nabla_{\widehat{\mathbf{M}}^0_{\mathrm{fut},t|d}}\mathcal{L}\left(\widehat{\mathbf{M}}^0_{\mathrm{fut},t|d}\right). \tag{43}$$

Here η_d^{guide} combines the fixed guidance scale s^{guide}, the clipped posterior variance at step d, and the batch-mean reduction of the loss used for differentiation. Guidance is applied throughout the D denoising steps, and the final sample with the lowest aggregated loss is returned.

Noise Best-of-N We sample successive populations of diffusion noise tensors from the nominal training distribution and rank their generated motions by the aggregated loss. The method retains the best noise sampled during its optimization iterations.

Noise GD We optimize the batch of diffusion noise tensors by differentiating through the sampler, following prior work on diffusion noise optimization (Chen et al., 2024b; Karunratanakul et al., 2024; Eyring et al., 2024; Uehara et al., 2025; Ben-Hamu et al., 2024). Learning rates and Adam parameters are specified in Appendix F.3.

Noise ZeNO ZeNO (Kim et al., 2026) optimizes a frozen generator's initial noise using only rewards evaluated on generated outputs. Its update is derived from path-integral control with an Ornstein–Uhlenbeck reference process whose stationary distribution is standard Gaussian. Centering rewards and linearizing their exponential weights under a small-variance approximation yields a control estimate that favors perturbations producing above-average rewards. Each iteration evaluates perturbed noise candidates, combines this estimate with an independent Gaussian perturbation, and projects the updated state onto a sphere.

We adapt this strategy, originally evaluated for image generation and protein design, to the physical-feasibility constraint formulation and task objectives of Physible, extending the optimization variables to both initial and posterior diffusion noise. In the kinematic setting, the reward is the negative of our shared loss, combining contact-pattern preservation, ground slip, and self-collision penalties with velocity tracking and autoregressive motion consistency under the same weights as the other methods.

Let $\mathbf{z}_k \in \mathbb{R}^{n_z}$ collect the active noise coordinates, $\mathcal{L}_{k,i}$ denote the loss of candidate i, and $\overline{\mathcal{L}}_k$ its population mean. The update is

$$\begin{aligned}
\mathbf{z}_{k,i} &= \sqrt{1-\beta^{\mathrm{Z}}}\,\mathbf{z}_k + \sqrt{\beta^{\mathrm{Z}}}\,\boldsymbol{\epsilon}_{k,i} && \text{(candidate sampling)},\\
\mathbf{u}_k &= -\frac{1}{N_{\mathrm{pop}}}\sum_{i=1}^{N_{\mathrm{pop}}}(\mathcal{L}_{k,i}-\overline{\mathcal{L}}_k)\,\boldsymbol{\epsilon}_{k,i} && \text{(control estimate)},\\
\widetilde{\mathbf{z}}_{k+1} &= \sqrt{1-\beta^{\mathrm{Z}}}\,\mathbf{z}_k + \sqrt{\beta^{\mathrm{Z}}}\,\boldsymbol{\xi}_k + \eta^{\mathrm{Z}}\mathbf{u}_k && \text{(state update)},\\
\mathbf{z}_{k+1} &= \sqrt{n_z}\,\frac{\widetilde{\mathbf{z}}_{k+1}}{\|\widetilde{\mathbf{z}}_{k+1}\|_2} && \text{(spherical projection)}.
\end{aligned} \tag{44}$$

where $\boldsymbol{\epsilon}_{k,i}$ and $\boldsymbol{\xi}_k$ are independent standard Gaussian vectors. We initialize $\mathbf{z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_z})$, evaluate candidates without projection, and retain the lowest-loss evaluated motion across initialization, candidate populations, and the terminal state. The Langevin interpretation of this practical update is approximate.

The active coordinates comprise the initial noise and all seven consumed posterior-noise tensors, omitting the unused final posterior slot; thus, $n_z = 8 \cdot 20 \cdot 438 = 70{,}080$ for SMPL and $n_z = 8 \cdot 30 \cdot 384 = 92{,}160$ for the bipedal robot, compared with $78{,}840$ and $103{,}680$ allocated coordinates, respectively, in our standardized formulation (Appendix E.2). We omit Eq. (3): ZeNO's spherical projection acts on the combined active vector, whereas Physible standardizes initial and posterior noise separately. Hyperparameters are specified in Appendix F.3.

E.5.2 Ablations of the Proposed Method

Both ablations use the Sep-CMA-ES updates and autoregressive integration described above. Their experimental implementations retain the best sampled noise and regenerate its motion after optimization.

Latent Noise Only In *Lat. Noise Sep-CMA-ES*, we optimize only the latent noise $\mathbf{M}^{D}_{\mathrm{fut},t}$. One posterior-noise tensor is sampled and held fixed across candidates and optimization iterations within each horizon.

No Noise Standardization (NNS) In *Noise Sep-CMA-ES (NNS)*, NNS denotes *no noise standardization*: we omit the noise normalization in Eq. (3) and pass $\mathbf{w} = \mathbf{w}^{\mathrm{raw}}$ directly to the sampler, without centering or rescaling either the initial or posterior noise.

E.5.3 Computational Budgets

Tab. 5 counts full batched reverse-sampling passes per motion horizon. Motion-space methods generate their initial batch once and then perform K updates without further denoiser calls. Guided Diffusion uses one pass with guidance at each denoising step. Noise GD uses one pass per update and one final pass to regenerate the motion after the last noise update, giving $K+1$ passes. We report the same $K+1$ sampling-pass budget for all sampling-based noise methods: Noise Best-of-N, Noise ZeNO, Noise Sep-CMA-ES, and its two ablations.

Runtime measurement is reported in Appendix F.6.

F Experimental Setup and Evaluation Protocol

F.1 Robot Models and Contact Specifications

Robotic Characters We consider two robotic characters in this work: the SMPL humanoid (Loper et al., 2015) and a custom bipedal robot. The SMPL humanoid has 75 Degrees of Freedoms (DOFs) in total, comprising six root DOFs and 23 spherical joints. We assume these spherical joints are actuated by simulated Unitree actuators with symmetric torque limits $\tau_{\min} = -\tau_{\max}$, where $\tau_{\max} = 500\,\mathrm{Nm}$, and with a velocity-dependent torque bound parameterized by $\kappa_\tau = -6.25\,\mathrm{Nms}$ and offset $o_\tau = 625\,\mathrm{Nm}$. The bipedal robot has 20 revolute joints actuated by 10 Unitree A1 motors

Table 5: Number of full MDM reverse-sampling passes per horizon, each over the inference batch. Each pass contains D batched denoiser evaluations. Numerical budgets are specified in Appendix F.3. Motion-space objective evaluations require no additional denoiser calls.

Method	Batched sampling passes	Additional evaluation structure
Unoptimized	1	Score generated batch
Motion GD	1	K motion updates and final scoring
Motion Sep-CMA-ES	1	K populations and final mean scoring
Guided Diffusion	1	Score/guide at each step; score final motion
Noise GD	$K+1$	Regenerate after final noise update
Sampling-based noise methods	$K+1$	Initial population, K populations

(symmetric torque limits $\tau_{\max} = 34\,\mathrm{Nm}$, $\kappa_\tau = -2.70\,\mathrm{Nms}$, $o_\tau = 53.97\,\mathrm{Nm}$) and 10 Dynamixel XH540-V150 motors (symmetric torque limits $\tau_{\max} = 4.8\,\mathrm{Nm}$, $\kappa_\tau = -0.71\,\mathrm{Nms}$, $o_\tau = 4.94\,\mathrm{Nm}$).

Contact Points For both characters, contact-related constraints are evaluated with a differentiable contact-point module using hard contact masks. For the SMPL character, we model contacts at the four corners of the bottom faces of the *ankle* and *toe* rigid bodies on each leg, yielding 16 possible contact points in total. For the bipedal robot, we consider four candidate contact points per foot: two at the toe edge and two at the heel edge, identified from the foot-geometry mesh.

The SMPL model has 24 bodies and 23 spherical joints; the bipedal robot model has 21 bodies and 20 revolute joints. For the bipedal robot, the ten hip, knee, and ankle joints use Unitree A1 actuators; the six shoulder joints, two elbows, and two neck joints use Dynamixel XH540-V150 actuators. Unless specified otherwise for the slope and friction experiments, the ground is flat, with gravity $(0, 0, -9.80665)$ in meters per second squared, and the dynamic solve uses friction coefficient $\mu_{\mathrm{c},p} = 0.5$. Joint damping is zero.

For SMPL, the set of body pairs checked for self-collision $\mathcal{B}_{\mathrm{col}}$ contains all 16 left–right combinations of hip, knee, ankle, and toe bodies, and all 16 left–right combinations of shoulder, elbow, wrist, and hand bodies. For the bipedal robot, it contains all 16 left–right combinations of hip-roll, hip-pitch, knee-pitch, and ankle-pitch bodies, and all 16 left–right combinations of the three shoulder bodies and the elbow. Each pair is checked once. No within-limb or limb–torso pairs are added to these sets.

F.2 Experimental Details for the Optimization Settings

Kinematic Feasibility Optimization We present a quantitative evaluation in the kinematic setting on an optimized set of 1500 time steps per style from the *100STYLE* dataset, corresponding to a total optimized motion duration of $83.3\,\mathrm{min}$ for the SMPL humanoid and $50\,\mathrm{min}$ for the bipedal robot. We resample a velocity command every third motion horizon, drawing from $v_x^{\mathrm{cmd}} \in [0.5, 2.0]$ (forward velocity), $v_y^{\mathrm{cmd}} = 0$ (sideway velocity), and $\omega_z^{\mathrm{cmd}} \in [-2.0, 2.0]$ (yaw angular velocity).

Dynamic Feasibility Optimization For the quantitative evaluation in the dynamic setting, we compare Physible and unoptimized generation on 11 bipedal robot sequences spanning six flat-ground motion styles, two slopes, and three friction settings (Appendix H.1.2). The *two-foot jump* sequence contains 250 time steps (5 s); the other ten sequences each contain 150 time steps (3 s), all sampled at 50 Hz. The comparison across optimization methods uses the two-foot-jump sequence. Unlike in the kinematic setting, we do not resample the root-reference velocity command, but use the fixed command $(v_x^{\mathrm{cmd}}, v_y^{\mathrm{cmd}}, \omega_z^{\mathrm{cmd}}) = (1\,\mathrm{m\,s^{-1}}, 0\,\mathrm{m\,s^{-1}}, 0\,\mathrm{rad\,s^{-1}})$ throughout each sequence.

Selected Kinematic Experiments The targeted self-collision experiment for the SMPL *lawn-mower* motion and the neutral-walking contact-preservation ablation use attention-bias weights $b_{\mathrm{syl}} = 1.5$ and $b_{\mathrm{pst}} = -1$, with WE smoothing strength $\kappa_{\mathrm{WE}} = 1$. The self-collision experiment optimizes only the self-collision constraint for 100 iterations. Penetration depths are evaluated using SAT tests on predefined body pairs belonging to different leaves of the kinematic tree, with spheroid bodies approximated by enclosing cuboids. The contact-preservation ablation compares three selected constant-command runs: unoptimized generation, optimization with a ground-penetration loss, and optimization with contact pattern preservation.

Slope Experiments The uphill and downhill experiments use neutral walking at slope angles $\theta = \pm 25^\circ$, with 150 frames at 50 Hz and the fixed command $(v_x^{\mathrm{cmd}}, v_y^{\mathrm{cmd}}, \omega_z^{\mathrm{cmd}}) = (1\,\mathrm{m\,s^{-1}}, 0\,\mathrm{m\,s^{-1}}, 0\,\mathrm{rad\,s^{-1}})$. In the slope-aligned coordinate frame, inverse dynamics uses $\mathbf{g} = (-g\sin\theta, 0, -g\cos\theta)^\top$, with $g = 9.806\,65\,\mathrm{m\,s^{-2}}$. Both slopes use friction coefficient $\mu_{\mathrm{c},p} = 1.0$, attention-bias weights $b_{\mathrm{syl}} = 2.5$ and $b_{\mathrm{pst}} = -1$, and WE smoothing strength $\kappa_{\mathrm{WE}} = 1$. Physible uses Sep-CMA-ES with 2500 iterations per horizon, a population of 64, and the dynamic loss weights in Appendix F.3.

F.3 Optimization Hyperparameters and Objective Weights

The active objective and constraint terms are summarized in Tab. 4, and the shared numerical settings are collected in Tab. 6. We use an inference batch of $N_{\mathrm{pop}} = 64$ samples, matching the population size of the sampling-based methods, and $D = 8$ denoising steps for every method. The Gradient Descent (GD)-based methods update their batch of motion or noise candidates independently, and Guided Diffusion guides each sample in the batch. For *kinematic feasibility optimization*, we optimize each motion horizon for $K = 100$ iterations unless stated otherwise. Since *dynamic feasibility optimization* is substantially more challenging and convergence is slower, we run it for $K = 2500$ iterations per horizon. Within each scenario, all iterative sampling-based and GD-based methods use the same outer iteration budget, and all optimization methods use the same loss/fitness weights. Guided Diffusion applies guidance during its single D-step sampling pass. Method-specific learning rates and guidance scales are given below.

Gradient-Based Baselines For all GD optimization approaches, we employ an Adam (Kingma & Ba, 2014) optimizer with $\beta_1^{\mathrm{Adam}} = 0.9$ and $\beta_2^{\mathrm{Adam}} = 0.999$. For *Noise GD*, we use a learning rate of 10^{-1} (SMPL) and 10^{-2} (bipedal robot). For *Motion GD*, we use a learning rate of 10^{-3}.

Guided Diffusion We use a fixed guidance scale $s^{\mathrm{guide}} = 10$ in the guidance coefficient η_d^{guide} defined in Appendix E.5.

Noise ZeNO For kinematic feasibility optimization, we use $\beta^{\mathrm{Z}} = 0.03$ and $\eta^{\mathrm{Z}} = 50$, selected in preliminary 25-update kinematic tests, with $N_{\mathrm{pop}} = 64$, $K = 100$, and the shared loss weights and generation settings.

Kinematic Feasibility Optimization Unless specified otherwise for an individual experiment, we use attention-bias weights $b_{\mathrm{syl}} = 1.0$ and $b_{\mathrm{pst}} = -0.5$ for SMPL, and $b_{\mathrm{syl}} = 2.5$ and $b_{\mathrm{pst}} = -1.0$ for the bipedal robot. The corresponding default WE smoothing strengths are $\kappa_{\mathrm{WE}} = 20$ and $\kappa_{\mathrm{WE}} = 1$, respectively. We use a weighted-sum loss with no loss-weight curriculum and no dynamic-feasibility objectives or constraints. The nonzero objective weights are $\rho_{\mathrm{track}} = 5$ and $\rho_{\mathrm{cons}} = 300$ for SMPL, and $\rho_{\mathrm{track}} = 1$ and $\rho_{\mathrm{cons}} = 300$ for the bipedal robot. The nonzero equality-constraint weight is $\rho_{\mathrm{h,slip}} = 10\,\mathrm{s^2\,m^{-2}}$ for both humanoids, and the nonzero inequality-constraint weights are $\rho_{\mathrm{g,contact}} = 5 \times 10^4\,\mathrm{m^{-2}}$ for contact-pattern preservation and $\rho_{\mathrm{g,self}} = 3 \times 10^4\,\mathrm{m^{-2}}$ for self-collision. The penetration and max-violation terms are evaluated for reporting but have zero loss weight. Contact-pattern preservation uses majority-vote contact schedule inference with $\delta_{\mathrm{clear}} = 0$; for SMPL we set $\delta_{\mathrm{vote}} = 0.008\,\mathrm{m}$ and $\delta_{\mathrm{pen}} = 0.005\,\mathrm{m}$, while for the bipedal robot we use $\delta_{\mathrm{vote}} = 0.004\,\mathrm{m}$ and $\delta_{\mathrm{pen}} = 0.002\,\mathrm{m}$.

Dynamic Feasibility Optimization Unless specified otherwise for an individual experiment, we use attention-bias weights $b_{\mathrm{syl}} = 1.5$ and $b_{\mathrm{pst}} = -0.5$. The loss is a weighted sum without a loss-weight curriculum. The nonzero objective weights are $\rho_{\mathrm{track}} = 1$ and $\rho_{\mathrm{cons}} = 300$. The nonzero equality-constraint weights are $\rho_{\mathrm{h,slip}} = 10\,\mathrm{s^2\,m^{-2}}$ and $\rho_{\mathrm{h,acc}} = 50$, and the nonzero inequality-constraint weight is $\rho_{\mathrm{g,self}} = 3 \times 10^4\,\mathrm{m^{-2}}$. Contact-pattern preservation uses the same settings for the bipedal robot as in the kinematic setting, and dynamics evaluation uses a static friction coefficient $\mu_{\mathrm{c},p} = 0.5$ at every contact with acceleration-space residuals. For the two-foot-jump benchmark, motion derivatives are estimated using WE smoothing with $\kappa_{\mathrm{WE}} = 1$.

Table 6: Default optimization hyperparameters and objective weights for each embodiment and feasibility optimization setting. Experiment-specific settings are given in the corresponding subsections. Zero weights mean that a term does not contribute to the optimized loss.

Parameter	SMPL (kinematic)	bipedal robot (kinematic)	bipedal robot (dynamic)
P, F	$10, 20$	$15, 30$	$15, 30$
$F_{\text{keep}}, F_{\text{ov}}$	$15, 5$	$25, 5$	$25, 5$
K	100	100	2500
N_{pop}	64	64	64
D	8	8	8
$b_{\text{syl}}, b_{\text{pst}}$	$1, -0.5$	$2.5, -1$	$1.5, -0.5$
$\rho_{\text{track}}, \rho_{\text{cons}}$	$5, 300$	$1, 300$	$1, 300$
$\rho_{\text{h,slip}}$	10	10	10
$\rho_{\text{h,acc}}$	0	0	50
$\rho_{\text{g,contact}}$	5×10^4	5×10^4	0
$\rho_{\text{g,self}}$	3×10^4	3×10^4	3×10^4
δ_{vote} (mm)	8	4	4
δ_{pen} (mm)	5	2	2
δ_{clear} (mm)	0	0	0

F.4 Evaluation Metrics

Physical Feasibility Metrics We quantify violations of physical feasibility at each motion frame using the metrics $E_{j,\ell} = v_{j,\ell}$ and also report the task-objective values defined above. Ground penetration and ground slip are averaged over all candidate contact points, and self-collision penetration is averaged over all checked body pairs. Acceleration reconstruction is the mean absolute acceleration residual over the floating base and joints. The inverse-dynamics residual is the Euclidean norm of the generalized force imbalance:

$$E_{\text{acc},\ell} = \frac{1}{n_\nu}\|\mathbf{e}_{\text{acc},\ell}\|_1, \qquad E_{\text{ID},\ell} = \|\mathbf{e}_{\text{dyn},\ell}\|_2. \tag{45}$$

Log Dimensionless Jerk We report group-weighted LDJ (Balasubramanian et al., 2015) $E_{\text{LDJ},t}$ as a smoothness diagnostic for an evaluated window at autoregressive step t. Let $F_{\text{eval}} \geq 5$ be its number of frames and $T_{\text{eval}} = (F_{\text{eval}} - 1)\Delta t$ its duration, using the sampling interval Δt from Appendix C.2. Following the generalized-velocity decomposition $\boldsymbol{\nu}_\ell = (\mathbf{v}_\ell, \boldsymbol{\omega}_\ell, \mathbf{v}_{\text{joint},\ell})$, we denote the corresponding unsmoothed groups by $\boldsymbol{\nu}^{\text{raw}}_{g,\ell}$, with $g \in \mathcal{G} = \{\text{lin}, \text{ang}, \text{joint}\}$, and their jerk vectors by $\mathbf{j}^{\text{raw}}_{g,\ell}$. The group dimensions are 3, 3, and n_{joint}; each spherical joint contributes three angular components.

Velocities and jerks use finite differences of the generated trajectories, without the WE smoothing used for dynamics. Angular velocities use quaternion differences for root and spherical-joint orientations; angular jerk uses their cumulative incremental rotation vectors. Revolute-joint angles are unwrapped before differentiation. We retain the estimator's endpoint samples and integrate squared jerk with the trapezoidal rule:

$$\begin{aligned} V_g^2 &= \max_{1\leq f\leq F_{\text{eval}}} \|\boldsymbol{\nu}^{\text{raw}}_{g,t+f}\|_2^2, \\ I_g &= \Delta t\left(\tfrac{1}{2}\|\mathbf{j}^{\text{raw}}_{g,t+1}\|_2^2 + \sum_{f=2}^{F_{\text{eval}}-1}\|\mathbf{j}^{\text{raw}}_{g,t+f}\|_2^2 + \tfrac{1}{2}\|\mathbf{j}^{\text{raw}}_{g,t+F_{\text{eval}}}\|_2^2\right), \qquad Q_g = \frac{T_{\text{eval}}^3 I_g}{V_g^2}. \end{aligned} \tag{46}$$

Each Q_g is dimensionless. The active set $\mathcal{G}_{\text{act}} \subseteq \mathcal{G}$ contains groups with peak speed $V_g > 10^{-8}$ in the corresponding velocity units. We give each active group equal weight and apply the negative natural logarithm after averaging its normalized energy:

$$E_{\text{LDJ},t} = -\ln\left[\frac{1}{|\mathcal{G}_{\text{act}}|}\sum_{g\in\mathcal{G}_{\text{act}}} Q_g\right]. \tag{47}$$

Thus, the joint group has the same weight as each active root group, irrespective of its number of degrees of freedom. Higher $E_{\text{LDJ},t}$ indicates smoother motion.

F.5 Style Classifier and FID Evaluation

Following Chen et al. (2024a), we evaluate whether generated motions preserve the commanded style using a style classifier trained on 15-timestep motion chunks (0.5 s) from the 100STYLE dataset retargeted to the SMPL humanoid (Loper et al., 2015). We segment each generated sequence (with a fixed style command) into overlapping 15-timestep windows, run the classifier, and report: (i) window-level style accuracy and (ii) overall FID pooled across styles and per-style FID computed from the classifier activations against the corresponding reference data.

The style classifier embeds each frame's complete encoded motion into 128 features and prepends a learned register token. After positional encoding, a four-layer transformer encoder with four heads, feed-forward width 1024, GELU activation, and dropout 0.2 produces the register representation. This representation, before the final GELU and linear classification head, supplies the features for FID. The head predicts one of the 100 styles.

Classifier inputs are expressed relative to the root position and heading of the window's first frame. Its training script uses 15-frame windows at stride ten, cross-entropy loss, and Adam with learning rate 10^{-3}. Evaluation uses every sliding 15-frame window at stride one. For a 1500-frame sequence of one style this gives 1486 evaluated windows. The generated features are compared with the stored per-style reference means and covariances; the final style accuracy and FID are evaluated over the full committed sequence.

F.6 Computational Environment and Runtime Measurement

We benchmark the runtime of each optimization method in two ways: (i) time per iteration (e.g., a single GD update, rolling out and scoring a population of noise samples, or one guided diffusion step) and (ii) total wall-clock time to optimize one motion horizon. For the total-time comparison, we use the same number of iterations for all sampling-based methods (e.g., best-of-n, Sep-CMA-ES) and GD-based methods. The optimization experiments and runtime measurements are run on a desktop workstation with a 12th Gen 16-core Intel i9-12900K CPU, an NVIDIA RTX 4090 GPU, and 64 GB of RAM.

The per-iteration timers bracket the optimizer's sampling/generation, objective and constraint evaluation, and update. They exclude optimizer initialization and the final regeneration or reevaluation after the last update. Horizon timing brackets the complete optimization call and therefore includes that setup and final evaluation, including CPU transfers and cone-layer construction when first required. For unoptimized generation, the horizon timer brackets sampling; the subsequent feasibility scoring is outside that timer.

F.7 Downstream RL Tracking Protocol

Following prior work (Yang et al., 2026; Müller et al., 2026), we assess whether improved physical feasibility translates into improved downstream control by training motion-tracking policies on the generated robot motions. We use tracking performance as a proxy for motion quality. We train a separate DeepMimic-style policy (Peng et al., 2018) for each of the unoptimized and Physible-optimized bipedal two-foot-jump motions from Tab. 3. Each trajectory contains 250 frames at 50 Hz (5 s) and is converted directly into a robot reference without further motion correction. The reference contains the root state and 20 joint states.

Simulation and policy. We simulate the robot in NVIDIA Isaac Sim 4.2 using 4,096 parallel environments on a single NVIDIA RTX 5090 GPU. The policy runs at 50 Hz. The action $\mathbf{a}_t \in \mathbb{R}^{20}$ specifies joint-position targets for joint-level PD controllers. The leg actuators use $k_p = 15.0$ and $k_d = 0.6$, while the arm and neck actuators use $k_p = 5.0$ and $k_d = 0.2$.

The actor and critic are three-layer MLPs with 512 ELU units per layer. The Gaussian actor is initialized with standard deviation 1.0.

Following VMP (Serifi et al., 2024b), let $\mathcal{H}_t$ denote the robot's motion frame. We write $\mathbf{p}_t^{\mathcal{H}}$, $\mathbf{R}_t^{\mathcal{H}}$, $\mathbf{v}_t^{\mathcal{H}}$, and $\boldsymbol{\omega}_t^{\mathcal{H}}$ for the root position, orientation, linear velocity, and angular velocity expressed in this frame, respectively, and let $\rho_6(\cdot)$ denote the 6D rotation representation. The actor observation is

$$\mathbf{o}_t = \left(\mathbf{p}_t^{\mathcal{H}}, \rho_6(\mathbf{R}_t^{\mathcal{H}}), \mathbf{v}_t^{\mathcal{H}}, \boldsymbol{\omega}_t^{\mathcal{H}}, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, \mathbf{a}_{t-2}, \hat{\mathbf{g}}_t\right), \tag{48}$$

Table 7: **Downstream RL reward.** Regularization terms are nonnegative costs with negative weights.

Name	**Term** $r_{i,t}$	**Weight** w_i
	Motion tracking	
Root position	$1 - \|\mathbf{p}_t^{\mathcal{H}} - \hat{\mathbf{p}}_t^{\mathcal{H}}\|_2^2$	1.0
Root orientation	$1 - \|\rho_6(\mathbf{R}_t^{\mathcal{H}}) - \rho_6(\hat{\mathbf{R}}_t^{\mathcal{H}})\|_2^2$	1.0
Root linear velocity	$1 - \|\mathbf{v}_t^{\mathcal{H}} - \hat{\mathbf{v}}_t^{\mathcal{H}}\|_2^2$	1.0
Root angular velocity	$1 - \|\boldsymbol{\omega}_t^{\mathcal{H}} - \hat{\boldsymbol{\omega}}_t^{\mathcal{H}}\|_2^2$	1.0
Rigid-body positions	$1 - \sum_b \|\mathbf{p}_{b,t}^{\mathcal{H}} - \hat{\mathbf{p}}_{b,t}^{\mathcal{H}}\|_2^2$	1.0
Rigid-body orientations	$1 - \sum_b \|\rho_6(\mathbf{R}_{b,t}^{\mathcal{H}}) - \rho_6(\hat{\mathbf{R}}_{b,t}^{\mathcal{H}})\|_2^2$	1.0
Joint positions	$1 - \|\mathbf{q}_t - \hat{\mathbf{q}}_t\|_2^2$	1.0
Joint velocities	$1 - \|\dot{\mathbf{q}}_t - \hat{\dot{\mathbf{q}}}_t\|_2^2$	0
Survival	1	10.0
	Regularization	
Joint torques	$\|\boldsymbol{\tau}_t\|_2^2$	$-1.0 \cdot 10^{-3}$
Joint accelerations	$\|\ddot{\mathbf{q}}_t\|_2^2$	$-2.5 \cdot 10^{-6}$
Action rate	$\|\mathbf{a}_t - \mathbf{a}_{t-1}\|_2^2$	-1.5
Action acceleration	$\|\mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2}\|_2^2$	-0.45

where the motion reference is

$$\hat{\mathbf{g}}_t = \left(\hat{\mathbf{p}}_t^{\mathcal{H}}, \rho_6(\hat{\mathbf{R}}_t^{\mathcal{H}}), \hat{\mathbf{v}}_t^{\mathcal{H}}, \hat{\boldsymbol{\omega}}_t^{\mathcal{H}}, \hat{\mathbf{q}}_t, \hat{\dot{\mathbf{q}}}_t\right). \tag{49}$$

The critic additionally receives the privileged root pose, noise-free root and joint velocities, and joint positions.

Reward. The reward terms and their weights are summarized in Tab. 7. Here, $\mathbf{p}_{b,t}^{\mathcal{H}}$ denotes the root-relative position of rigid body b, and hats denote reference quantities.

Training. We optimize both policies with PPO using $10\,\mathrm{s}$ episodes. Each iteration collects 24 policy steps from every environment, and training runs for at most 100,000 iterations. PPO uses five epochs, four minibatches, an adaptive learning rate initialized at 10^{-3}, clipping parameter 0.2, discount factor 0.99, and GAE parameter 0.95.

During training, the default bipedal motion-tracking task randomizes body masses by a factor in $[0.9, 1.1]$ and foot friction in $[0.4, 0.8]$. We additionally apply uniform measurement noise of $\pm 0.02\,\mathrm{rad}$ to root roll and pitch, $\pm 0.1\,\mathrm{m\,s^{-1}}$ to root linear velocity, and $\pm 0.1\,\mathrm{rad\,s^{-1}}$ to root angular velocity.

Evaluation. For evaluation, we run the mean action of the Gaussian policy in a single environment using the same observation model and flat terrain. External disturbances are disabled, masses are fixed to their nominal values, and foot friction is fixed at 0.5. We evaluate both policies under identical conditions and report root and rigid-body position and orientation errors, together with joint-position tracking error, over the complete reference trajectory.

G SUPPLEMENTARY KINEMATIC FEASIBILITY RESULTS

This appendix presents quantitative baseline comparisons, optimization convergence, component ablations, and motion case studies. Experimental settings and metric definitions are given in Appendix F. The case studies pair motion visualizations with the corresponding trajectory and contact diagnostics.

G.1 QUANTITATIVE COMPARISON WITH BASELINES

Tab. 1 and Tab. 2 compare the proposed diffusion-noise optimization approach against the baseline methods for the SMPL humanoid and the bipedal robot, respectively.

Overall, the proposed method best addresses the multi-objective problem, consistently reducing kinematic constraint violations and objective costs relative to the unoptimized motion while preserving the motion style. In contrast, the baselines—despite using identical loss/fitness weights—tend to improve only a subset of constraints and objectives while degrading others. Gradient-based methods (e.g., GD, *Guided Diffusion*) particularly struggle with the *contact pattern preservation* constraint due to its discontinuous gradients. Sampling-based methods handle this constraint more robustly, with Sep-CMA-ES performing best; however, simple *Best-of-*N sampling is insufficient in our high-dimensional search space ($78{,}840$ variables for SMPL and $103{,}680$ for the bipedal robot).

While direct motion optimization can be effective, it often introduces visible jitter, which is reflected in lower LDJ values. *Guided Diffusion* is computationally efficient because it avoids iterative inference through (or backpropagation through) the full MDM and instead only differentiates the loss terms, but its effectiveness is limited—likely due to discontinuous gradients and the small number of denoising steps. Somewhat surprisingly, GD-based noise optimization—despite its popularity (Karunratanakul et al., 2024; Eyring et al., 2025)—performs poorly in our setting: it is computationally expensive (requiring backpropagation through both the objectives and the entire MDM at each iteration), is sensitive to discontinuous gradient information, and often yields severe ground penetration and degraded motion quality and style.

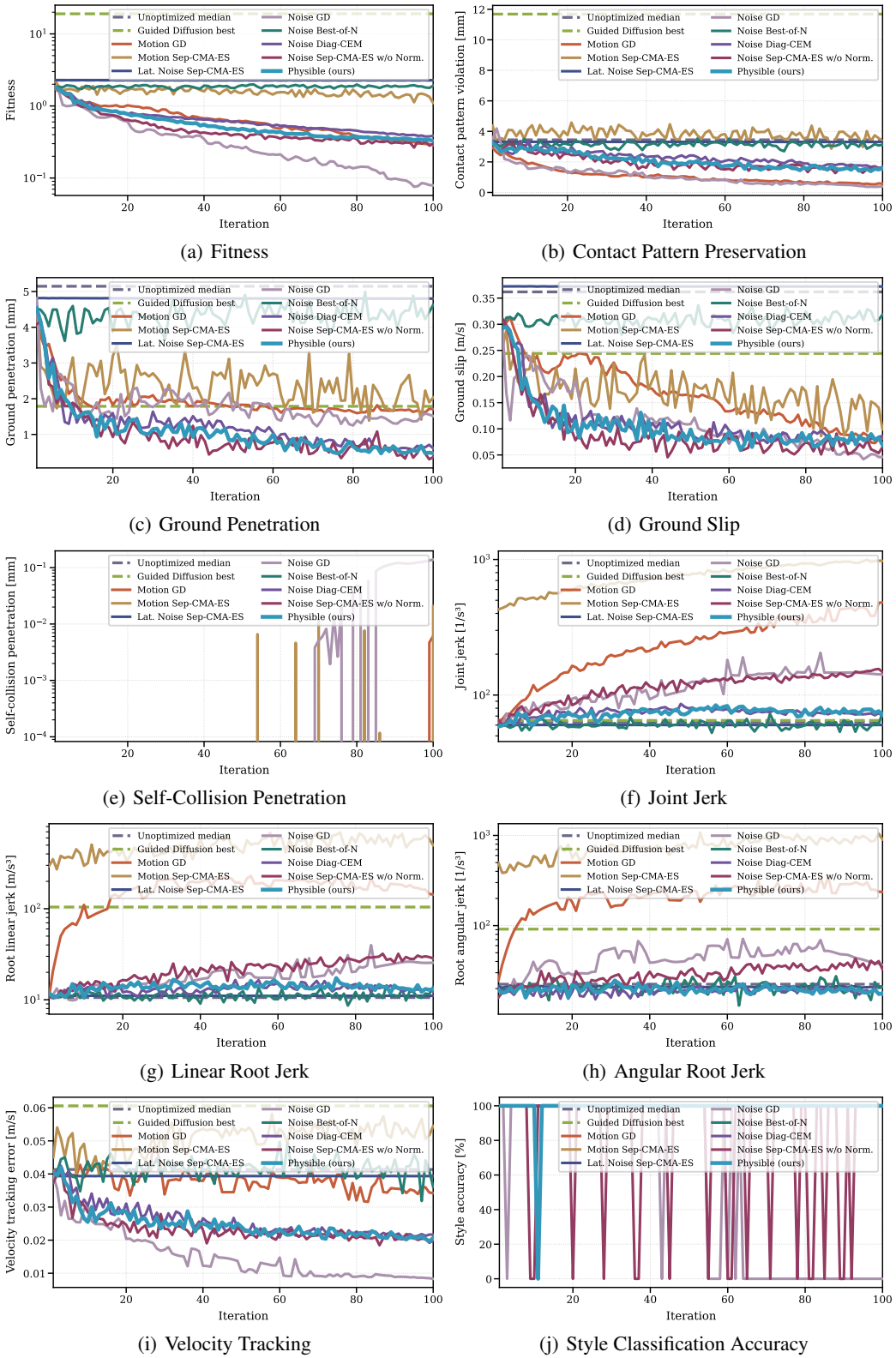
Under the same physical-feasibility constraint formulation and loss weights, Physible achieves lower mean ground penetration (0.676 vs. $0.939\,\mathrm{mm}$), self-collision penetration (0.0152 vs. $0.0719\,\mathrm{mm}$), and ground slip (0.0484 vs. $0.0796\,\mathrm{m\,s^{-1}}$) than Noise ZeNO on SMPL, although ZeNO retains slightly higher style accuracy (91.69% vs. 87.73%; Tab. 1). On the bipedal robot, Physible reduces mean ground penetration, self-collision penetration, and slip by 40%, 85%, and 61% relative to Noise ZeNO, respectively (Tab. 2).

G.2 CONVERGENCE

G.2.1 OPTIMIZATION CONVERGENCE

In Fig. 3, we compare the convergence behavior of the different methods for kinematic feasibility optimization. We optimize a *single* horizon (with the autoregressive motion-consistency objective disabled) for the SMPL humanoid under a velocity command $\mathbf{v}^{\mathrm{cmd}} = (1, 0, 1)$ (forward walking) and the *neutral* style. All methods in this comparison use WE smoothing with $\kappa_{\mathrm{WE}} = 1$. We report metrics for the median sample (by loss/fitness) from a batch of 64 unoptimized motions, the best sample from 64 motions optimized in parallel with *Guided Diffusion*, and the best sample per iteration for all remaining methods.

The results show that *Noise GD* reduces the fitness fastest, but it does not decrease ground penetration as effectively as the other approaches and the motion style degrades as iterations increase. Notably, despite identical loss weights, *Noise GD* appears to emphasize velocity tracking. While it performs well on contact pattern preservation, it still falls short of the best sampling-based noise optimization methods in terms of reducing ground penetration. *Diag-CEM* (Pinneri et al., 2021) (Cross-Entropy Method (CEM) (Rubinstein, 1999) with diagonal Gaussian covariance) achieves performance comparable to *Sep-CMA-ES*, albeit with slightly slower convergence. Direct motion optimization, in contrast, quickly increases jerk, whereas jerk grows more gradually for the noise optimization methods. Overall, the proposed *Physible* approach shows strong convergence across objectives and constraints while only modestly increasing jerk relative to the unoptimized motion.



(a) Fitness (b) Contact Pattern Preservation

(c) Ground Penetration (d) Ground Slip

(e) Self-Collision Penetration (f) Joint Jerk

(g) Linear Root Jerk (h) Angular Root Jerk

(i) Velocity Tracking (j) Style Classification Accuracy

Figure 3: Convergence of kinematic feasibility optimization methods for a single motion horizon and constant forward reference velocity on the SMPL humanoid with *neutral* walking style with velocity command $\mathbf{v}^{\text{cmd}} = (1, 0, 0)$.

G.3 Ablation Studies

G.3.1 Initial and Posterior Noise Optimization

The *Lat. Noise Sep-CMA-ES* ablation in Tabs. 1 and 2 optimizes only the initial/latent noise, holding posterior noise fixed. This restriction yields only modest gains over the unoptimized baseline and falls short of joint initial and posterior noise optimization.

G.3.2 Noise Standardization

Although noise standardization constrains the optimization and often leads to larger kinematic constraint violations and higher objective costs after optimization, it greatly helps preserve the original motion style. As shown in the style accuracy and style FID columns of Tab. 1, for the same optimizer (Sep-CMA-ES) and identical hyperparameters, *Noise Sep-CMA-ES (NNS)*, which omits Eq. (3), achieves a style accuracy of $57.49\,\%$ and a per-style FID of 133.39, whereas the method with noise standardization achieves a style accuracy of $87.73\,\%$ and a per-style FID of 43.95.

G.3.3 Contact Pattern Preservation

One aspect of kinematic feasibility is avoiding (excessive) ground penetrations, which are physically infeasible. A natural question is why we favor the *Contact Pattern Preservation* constraint (Sec. 6.2) over directly minimizing the ground penetration distance. The key reason is to prevent *floating* characters, i.e., motions that exhibit extended time intervals without ground contact.[1] When we co-optimize dynamical feasibility, floating is implicitly discouraged because, without contacts, the accelerations encoded in the motion cannot be supported due to the absence of contact forces. In contrast, under purely kinematic feasibility optimization, this implicit safeguard is absent: optimizing only a ground penetration constraint can yield "seemingly" valid solutions that simply avoid penetration by lifting the character off the ground for long durations. We therefore employ the contact pattern preservation constraint, which enforces (i) contact—while limiting penetration—during phases where contact should occur (thereby preventing floating), and (ii) separation during phases where contact should not occur.

To substantiate this reasoning, we perform an ablation study that compares the contact behavior obtained when using the *Ground Penetration* constraint versus the proposed *Contact Pattern Preservation* constraint. The three conditions use the settings in Appendix F.2.

Figure 4 compares unoptimized motion and the two losses for the SMPL humanoid in the *neutral* walking style, using matched foot close-ups, signed ground clearance, and intended versus observed foot contacts. Over the 240-frame ($8\,\mathrm{s}$) clips, the ground penetration distance loss reduces mean penetration from $3.826\,\mathrm{mm}$ to $0.002\,\mathrm{mm}$, but leaves all checked contact points above ground in 91.7% of frames, with a longest contact loss of $2.20\,\mathrm{s}$. Contact pattern preservation reduces mean penetration to $1.166\,\mathrm{mm}$, while limiting frames without contact to 4.2% and the longest contact loss to $0.13\,\mathrm{s}$ (unoptimized: 3.3% and $0.07\,\mathrm{s}$). Thus, it reduces penetration while avoiding the prolonged floating induced by the ground penetration distance loss, although some penetrations still exceed the $5\,\mathrm{mm}$ allowance. Figs. 5 and 6 provide the corresponding contact-point ground separation plots.

[1] Short no-contact phases can be feasible or even desired (e.g., jumping or running). However, if a motion lacks ground contact for long periods, it will generally not be dynamically feasible.

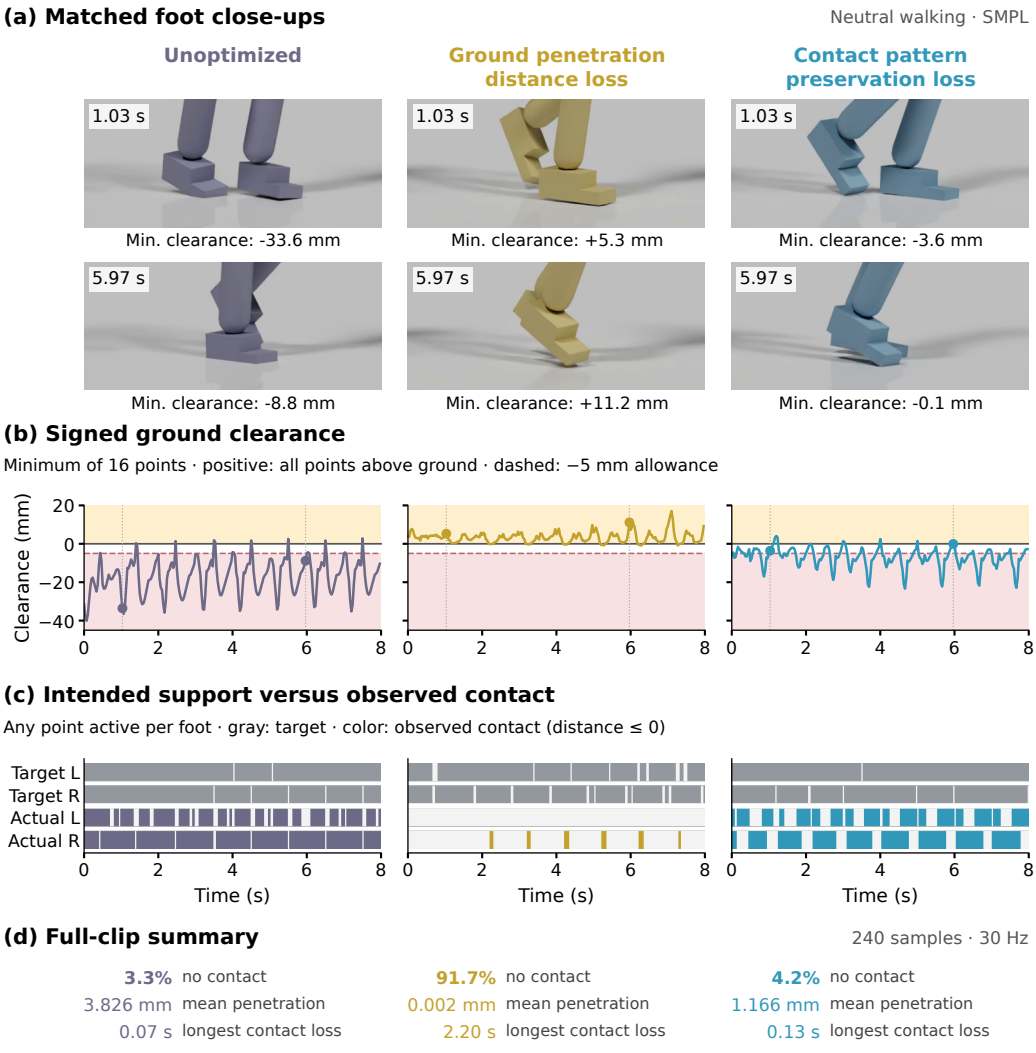


Figure 4: Ablation of the contact pattern preservation loss for *neutral* walking on the SMPL humanoid. Columns compare unoptimized motion, ground penetration distance loss, and contact pattern preservation loss. (a) Matched foot close-ups at 1.03 and 5.97 s, with minimum signed ground clearance. (b) Minimum clearance over 16 checked points; positive values indicate that all points are above ground, and the dashed line marks the -5 mm penetration allowance. (c) Target and observed left/right foot contacts; a foot is in contact when any checked point has nonpositive separation. (d) Summary over 240 samples at 30 Hz, reporting the fraction of frames without contact, mean penetration, and longest continuous contact loss.

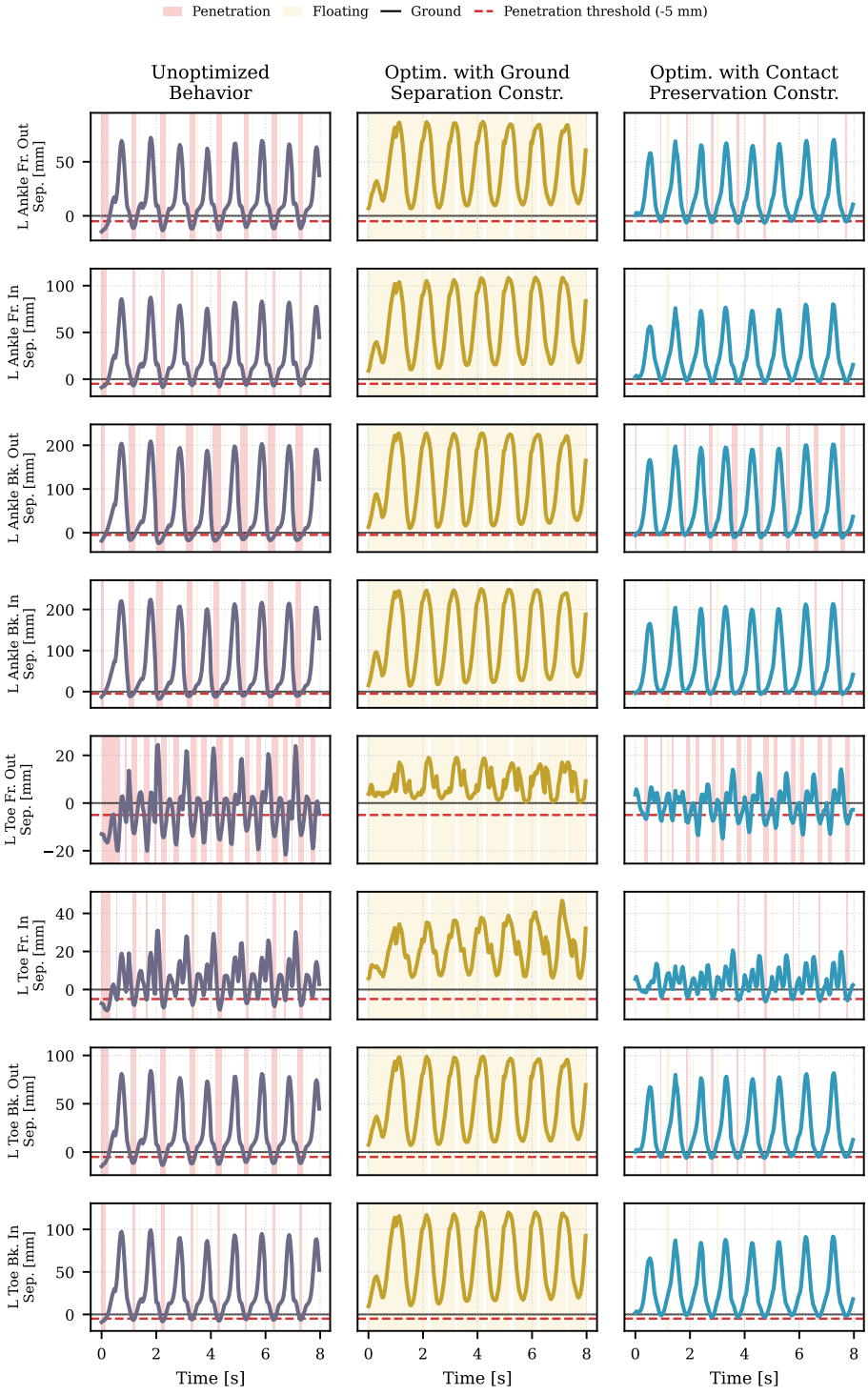


Figure 5: Part 1 of the ablation study compares two constraints for limiting (excessive) ground penetration during kinematic feasibility optimization on the SMPL humanoid for the *neutral* motion style: the baseline *ground penetration* constraint versus the proposed *contact pattern preservation* constraint. For each contact point, the left column shows the ground separation distance of the median unoptimized motion, the center column shows the best motion optimized with the *ground penetration* constraint, and the right column shows the best motion optimized with the *contact pattern preservation* constraint. Red shaded intervals indicate excessive-contact periods where penetration exceeds the threshold (as used by the *contact pattern preservation* constraint), while yellow shaded intervals indicate phases where none of the robot's contact points are in contact with the ground.

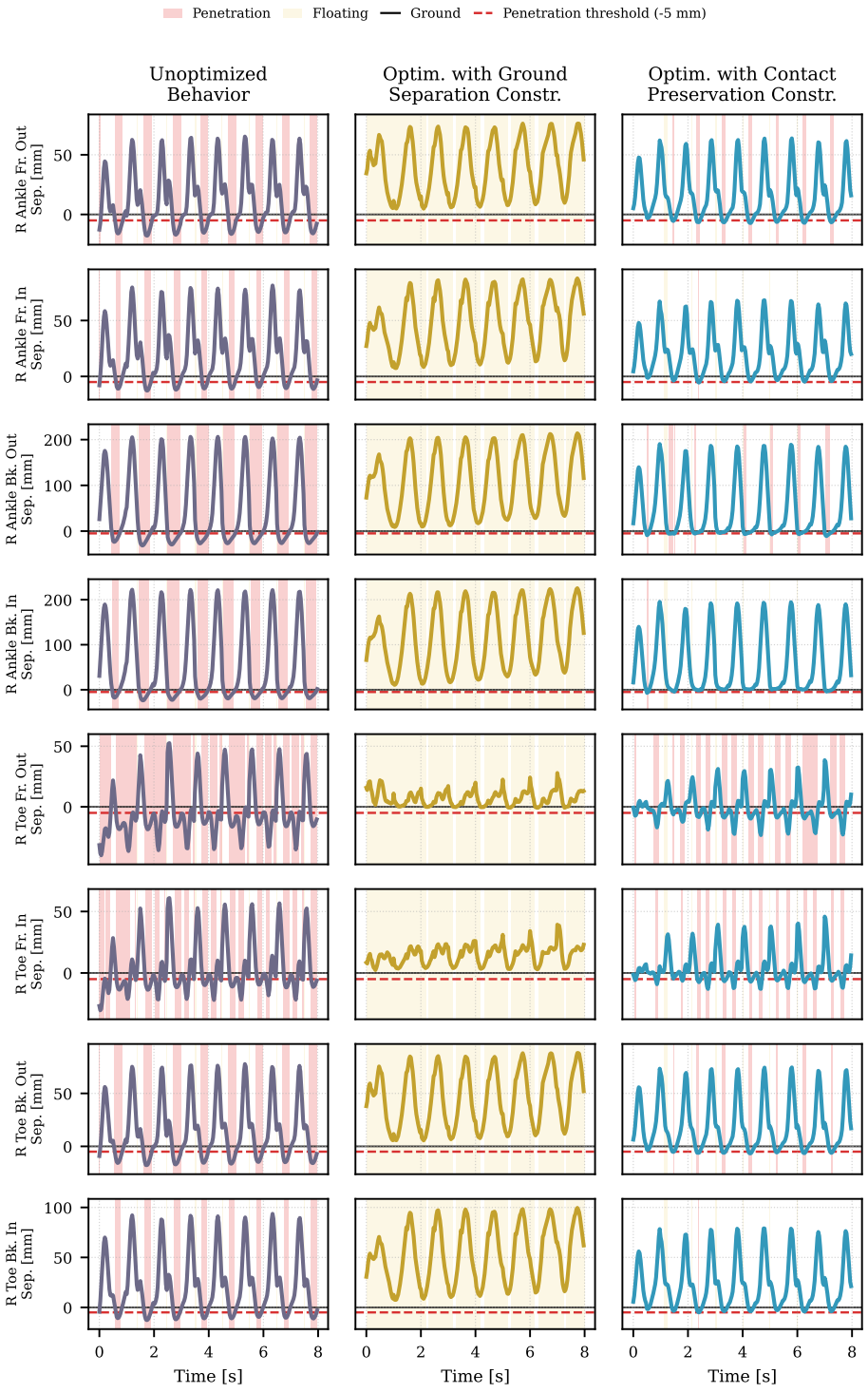


Figure 6: Part 2 of the ablation study compares two constraints for limiting (excessive) ground penetration during kinematic feasibility optimization on the SMPL humanoid for the *neutral* motion style: the baseline *ground penetration* constraint versus the proposed *contact pattern preservation* constraint. For each contact point, the left column shows the ground separation distance of the median unoptimized motion, the center column shows the best motion optimized with the *ground penetration* constraint, and the right column shows the best motion optimized with the *contact pattern preservation* constraint. Red shaded intervals indicate excessive-contact periods where penetration exceeds the threshold (as used by the *contact pattern preservation* constraint), while yellow shaded intervals indicate phases where none of the robot's contact points are in contact with the ground.

G.4 Motion Case Studies for Qualitative Analysis

G.4.1 Neutral Walking: Comparison of Optimization Methods

In Figs. 8–19, we compare the kinematic-feasibility-optimized behavior over 150 time steps for the SMPL humanoid in the *neutral* style. We include root and joint trajectories, the resulting SE(2) motion relative to the integrated velocity reference, ground separation, and slip velocity per contact point, contact-pattern preservation, and penetration distances for each self-collision pair. For clarity, we show a subset of methods: the unoptimized baseline, direct motion optimization with Sep-CMA-ES, *Guided Diffusion*, noise GD, and the proposed *Physible* noise optimization with Sep-CMA-ES. Figure 7 summarizes this comparison through matched-time snapshots and ground-clearance, contact-slip, self-collision, and joint-jerk diagnostics.

The plots again highlight the *jittery* artifacts introduced by direct motion optimization (*Motion Sep-CMA-ES*). For velocity tracking, the proposed method is the only approach that follows the integrated command almost perfectly, with negligible drift over the full 5 s horizon; the other methods (including the unoptimized motion) begin to drift from the reference around 1.5 s. Figs. 17 and 18 further show that *Physible* largely preserves the contact pattern, maintaining contact whenever the unoptimized motion is in contact. Crucially, it also substantially reduces ground penetration—especially the deepest events—so that the penetration threshold is exceeded only rarely and by a small margin. On a related note, the ground slip velocities are with *Physible* also significantly reduced compared to the unoptimized motion and to a larger extent for the other optimization methods. Only for a single contact point (*Left Ankle Back Out*), the slip velocity remains slightly higher at certain time steps. For the *neutral* style, self-collision is not a major issue in the unoptimized motion; however, methods that noticeably degrade motion quality and deviate from the style, such as *Motion Sep-CMA-ES* and *Noise GD*, can introduce leg–leg self-collisions.

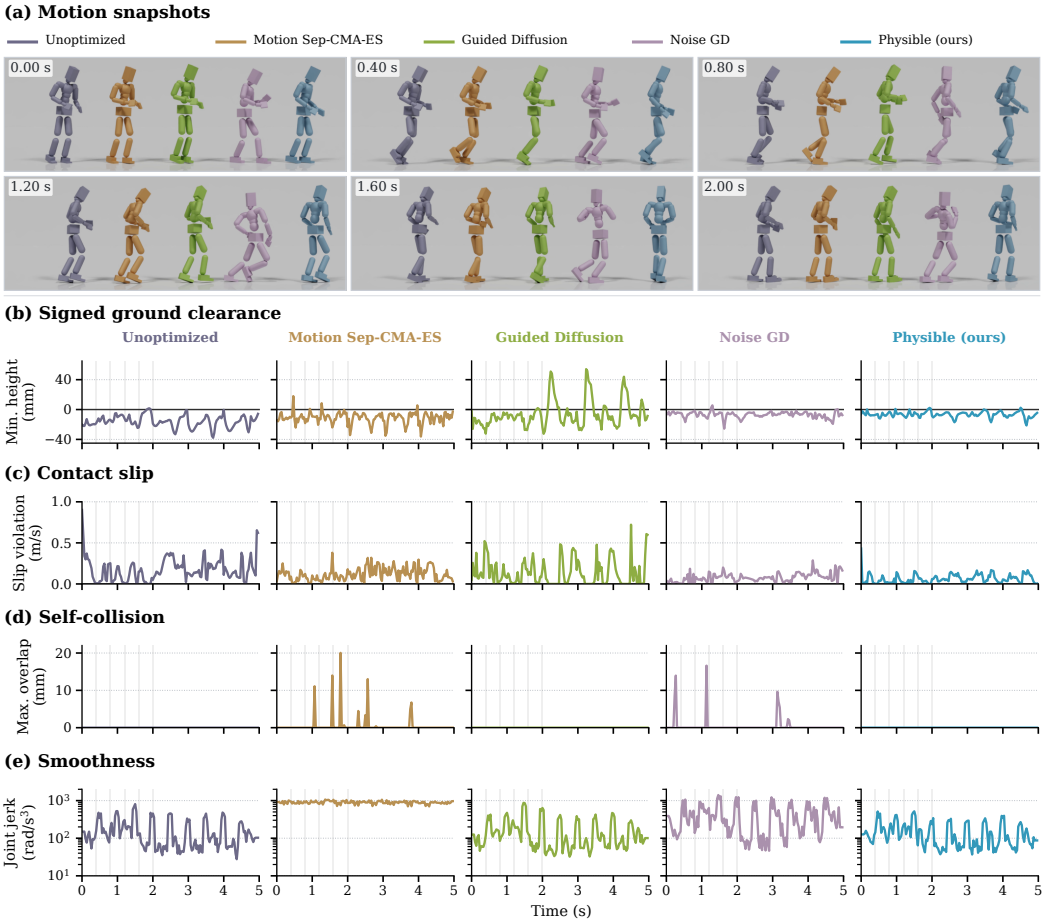


Figure 7: Kinematic feasibility optimization methods comparison for *neutral* walking on the SMPL humanoid. **(a)** Matched-time snapshots of unoptimized motion, Motion Sep-CMA-ES, Guided Diffusion, Noise GD, and Physible. **(b)** Minimum signed height across contact points; negative values indicate ground penetration, while positive values indicate that all contact points are above ground. **(c)** Contact slip violation, averaged over all contact points after contact masking and velocity-threshold subtraction. **(d)** Maximum self-collision overlap across evaluated body pairs. **(e)** Root-mean-square joint jerk on a logarithmic scale. Diagnostics cover the full 5 s clip, with a shared scale within each row; gray bands mark snapshot times. Physible reduces penetration and slip while retaining jerk closer to the unoptimized motion, although some ground penetration remains.

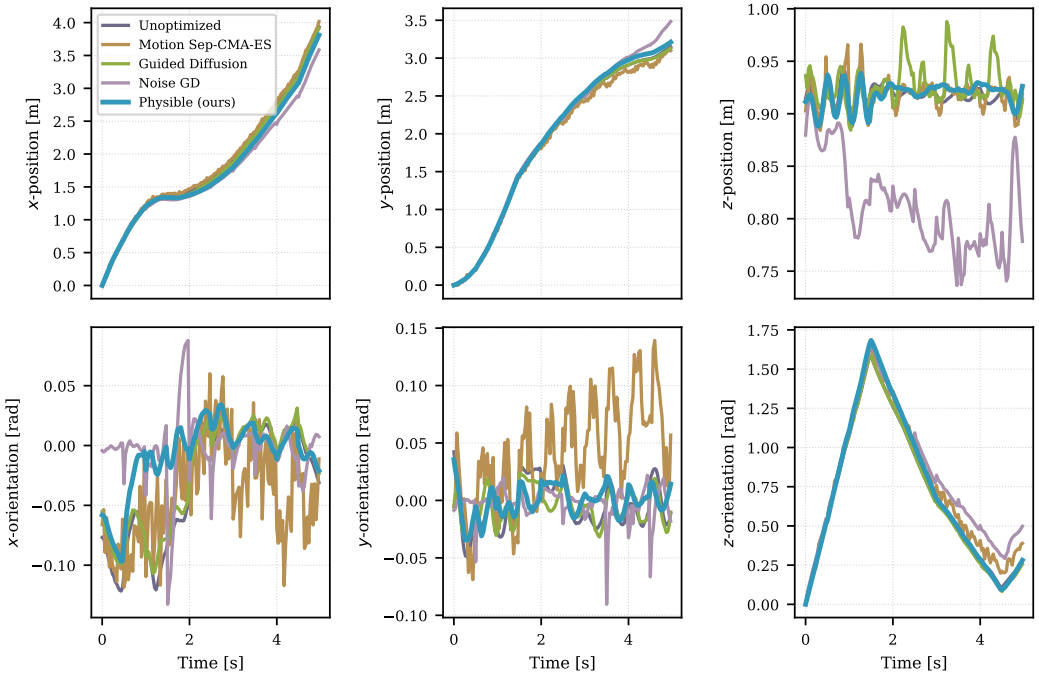


Figure 8: Behavior comparison of the *root motion* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character.

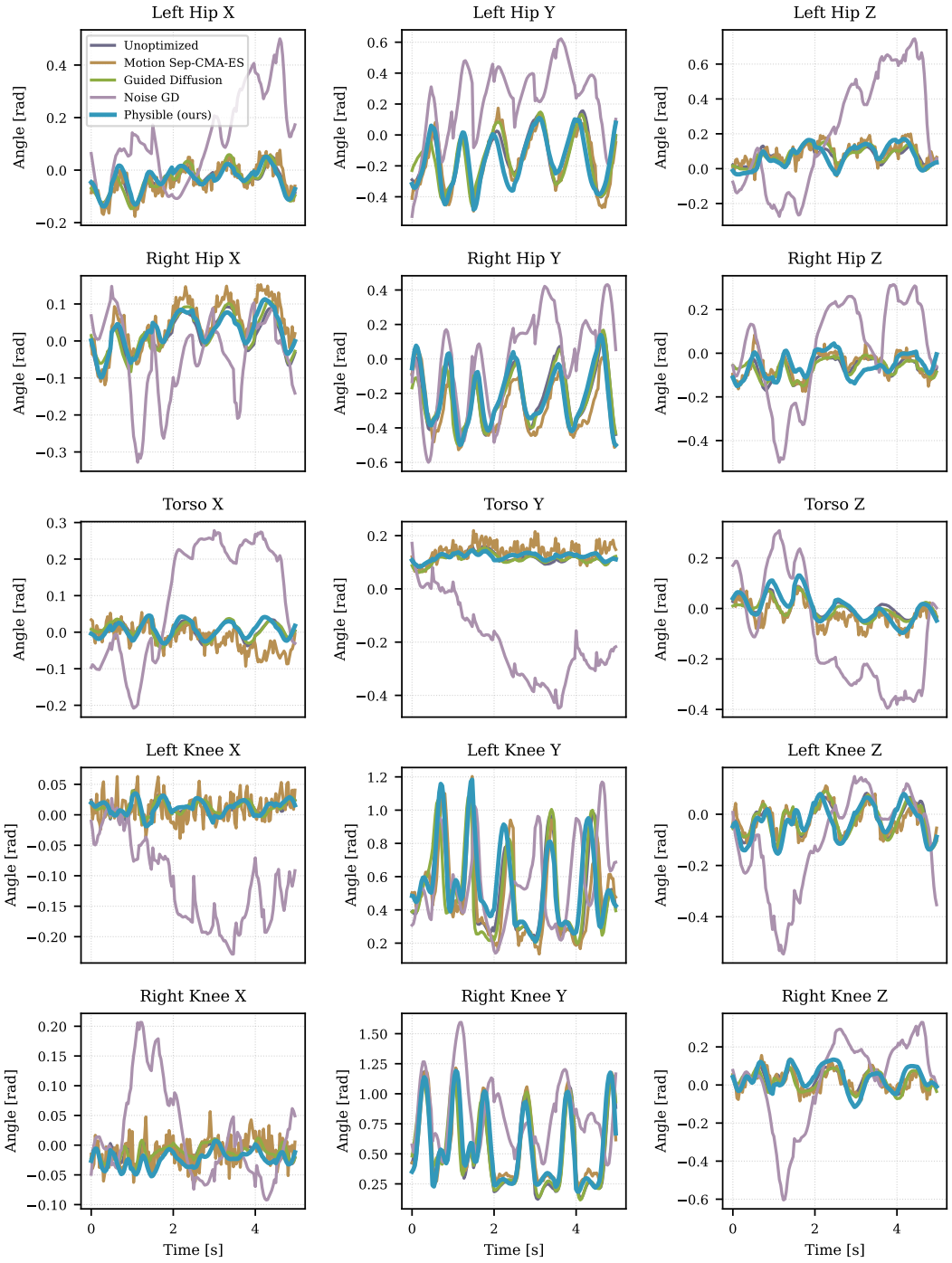


Figure 9: Behavior comparison of the *joint motion* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character - part 1.

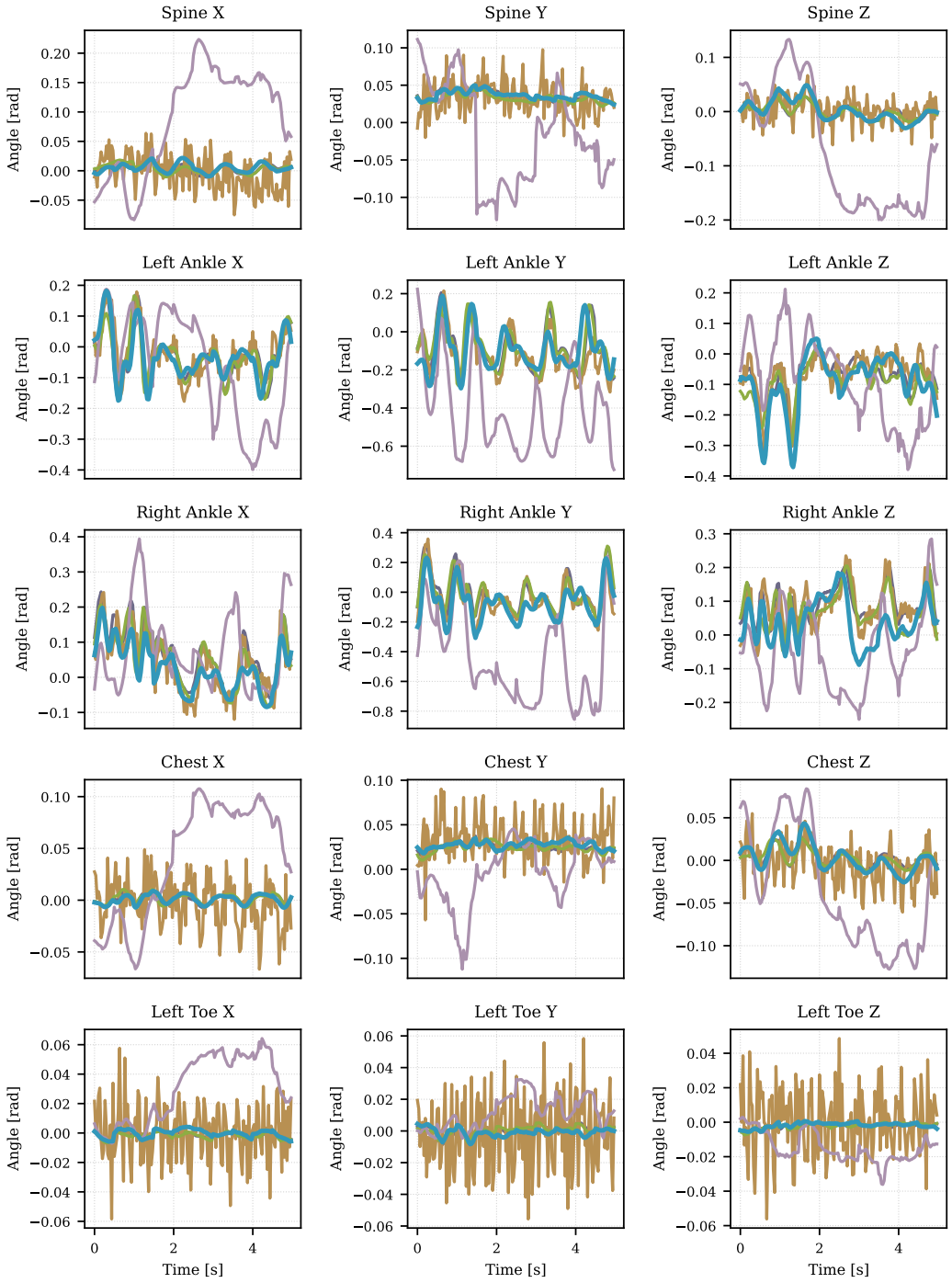


Figure 10: Behavior comparison of the *joint motion* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character - part 2.

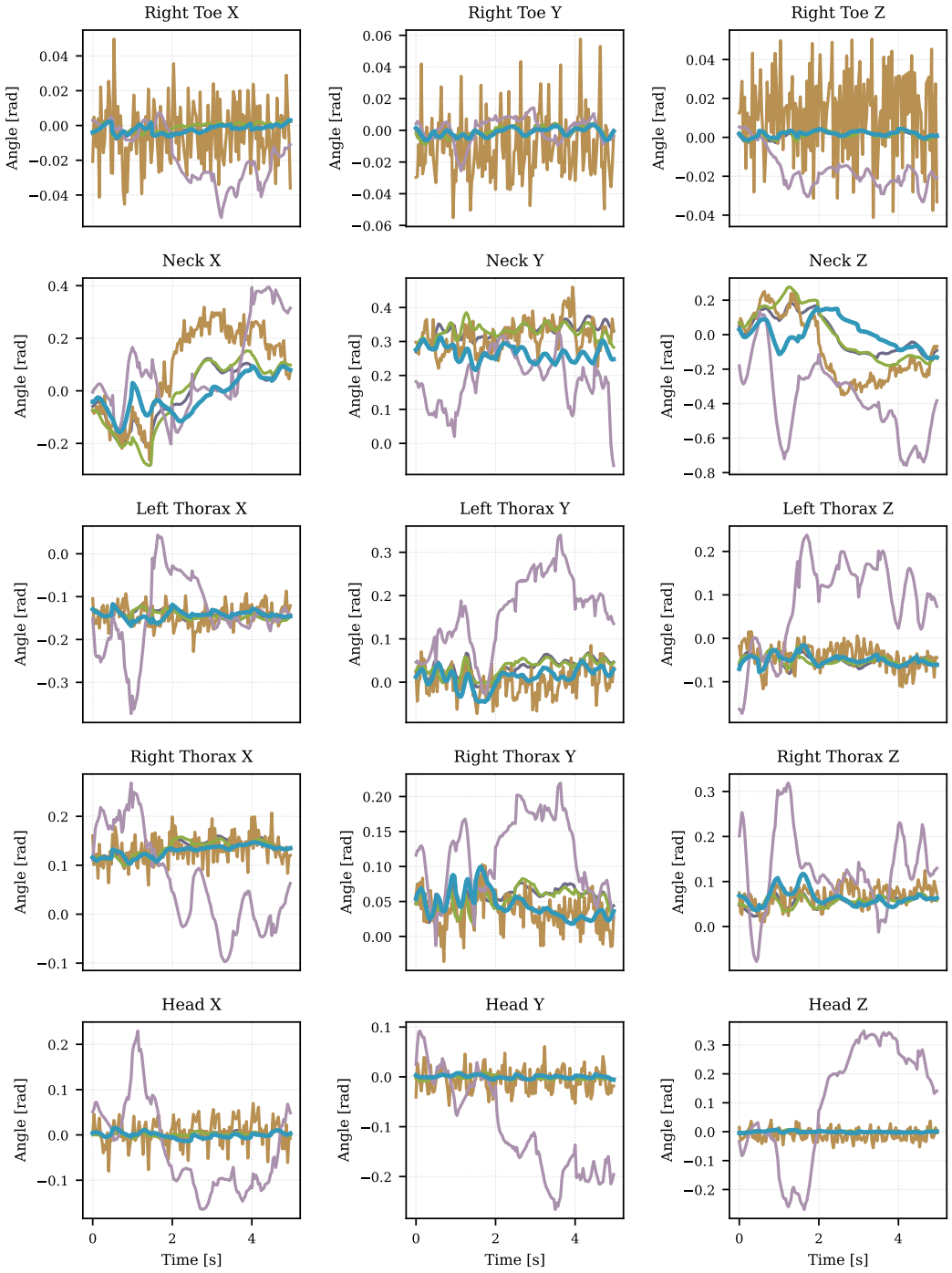


Figure 11: Behavior comparison of the *joint motion* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character - part 3.

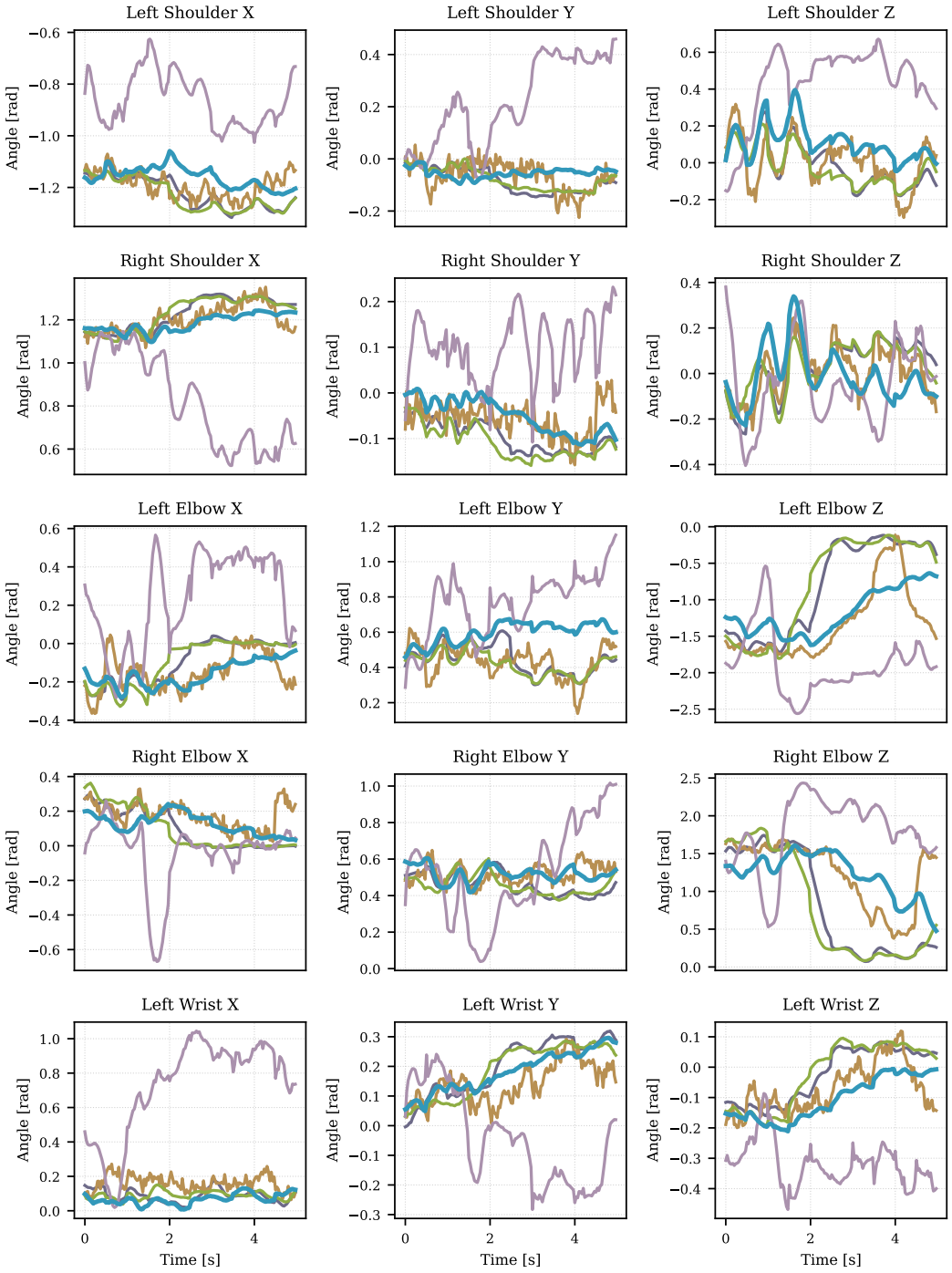


Figure 12: Behavior comparison of the *joint motion* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character - part 4.

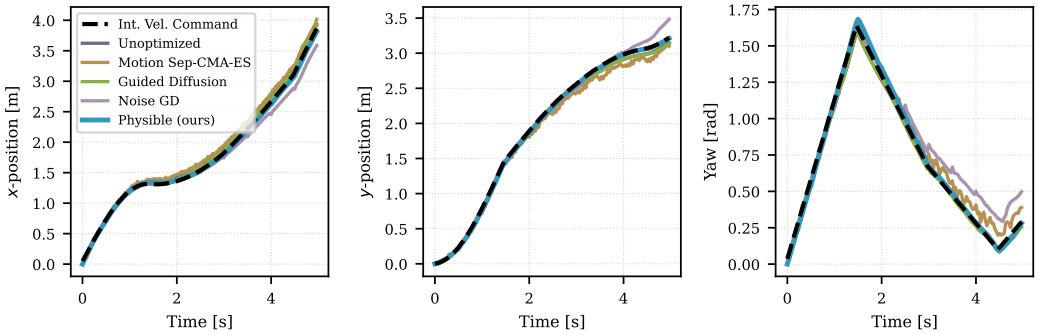


Figure 13: Behavior comparison of the *root motion* in the x-y plane with integrated velocity command-based reference for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character.

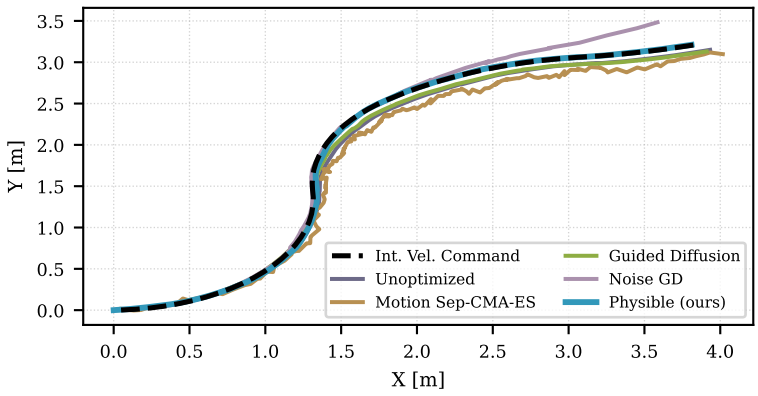


Figure 14: Behavior comparison of the *positional root motion* in the x-y plane with integrated velocity command-based reference for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character.

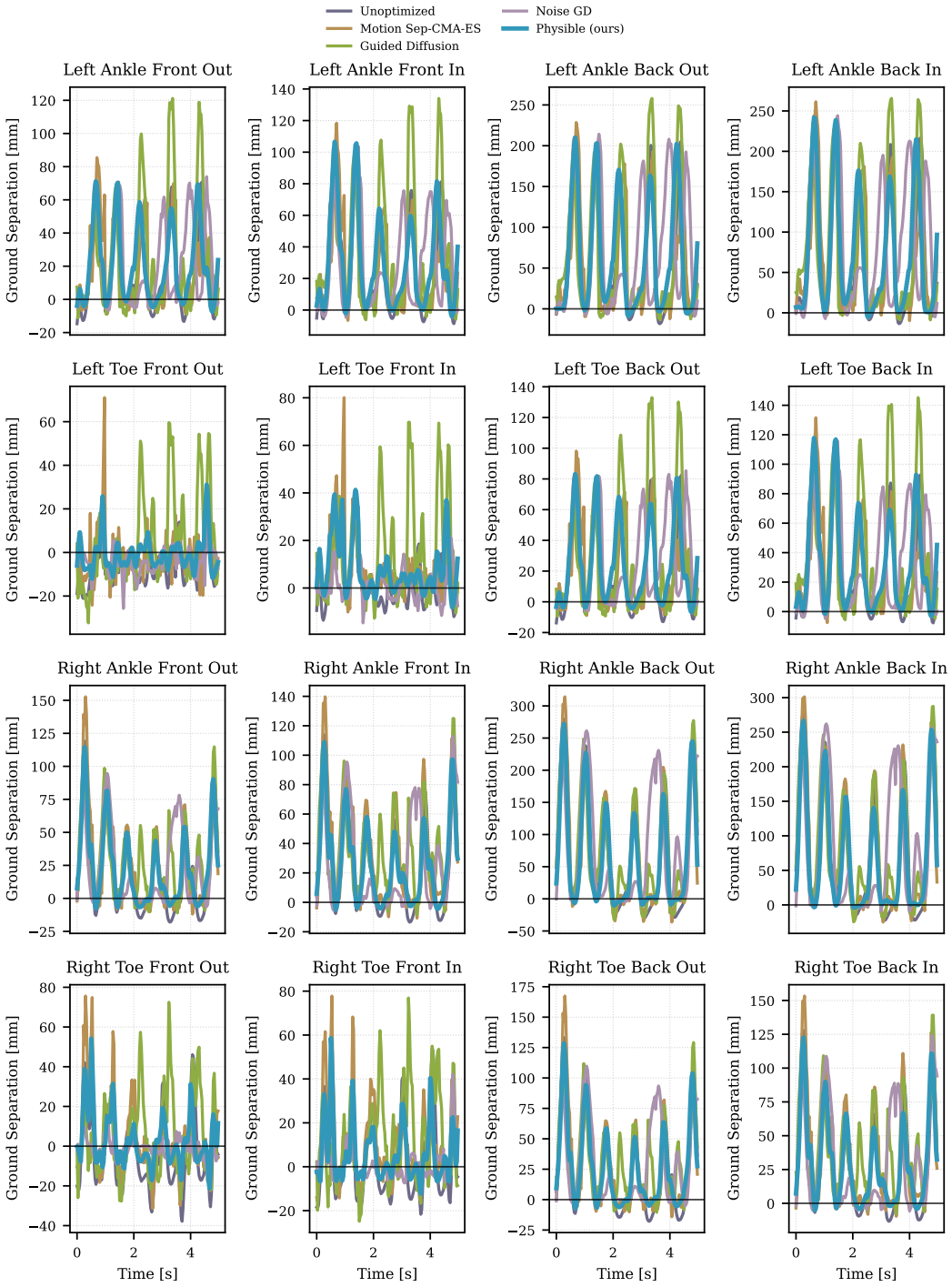


Figure 15: Behavior comparison of the *ground separation* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character.

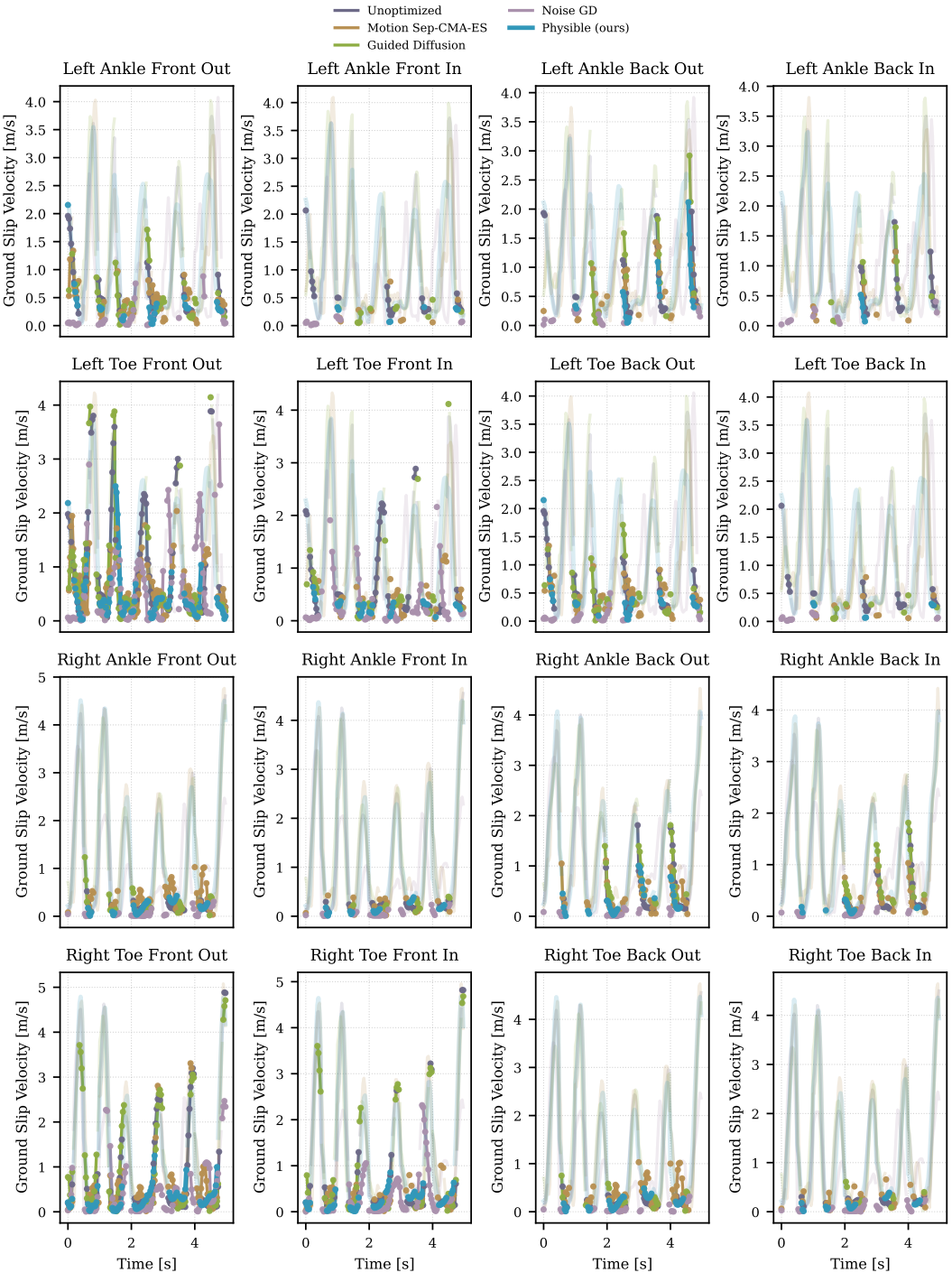


Figure 16: Behavior comparison of the *ground slip velocity* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character. The full-opacity and markers denoted active contacts (i.e., with a semi-negative ground separation distance). The transparent lines denote the tangential velocities of the contact points when they are not in contact and separated from the ground.

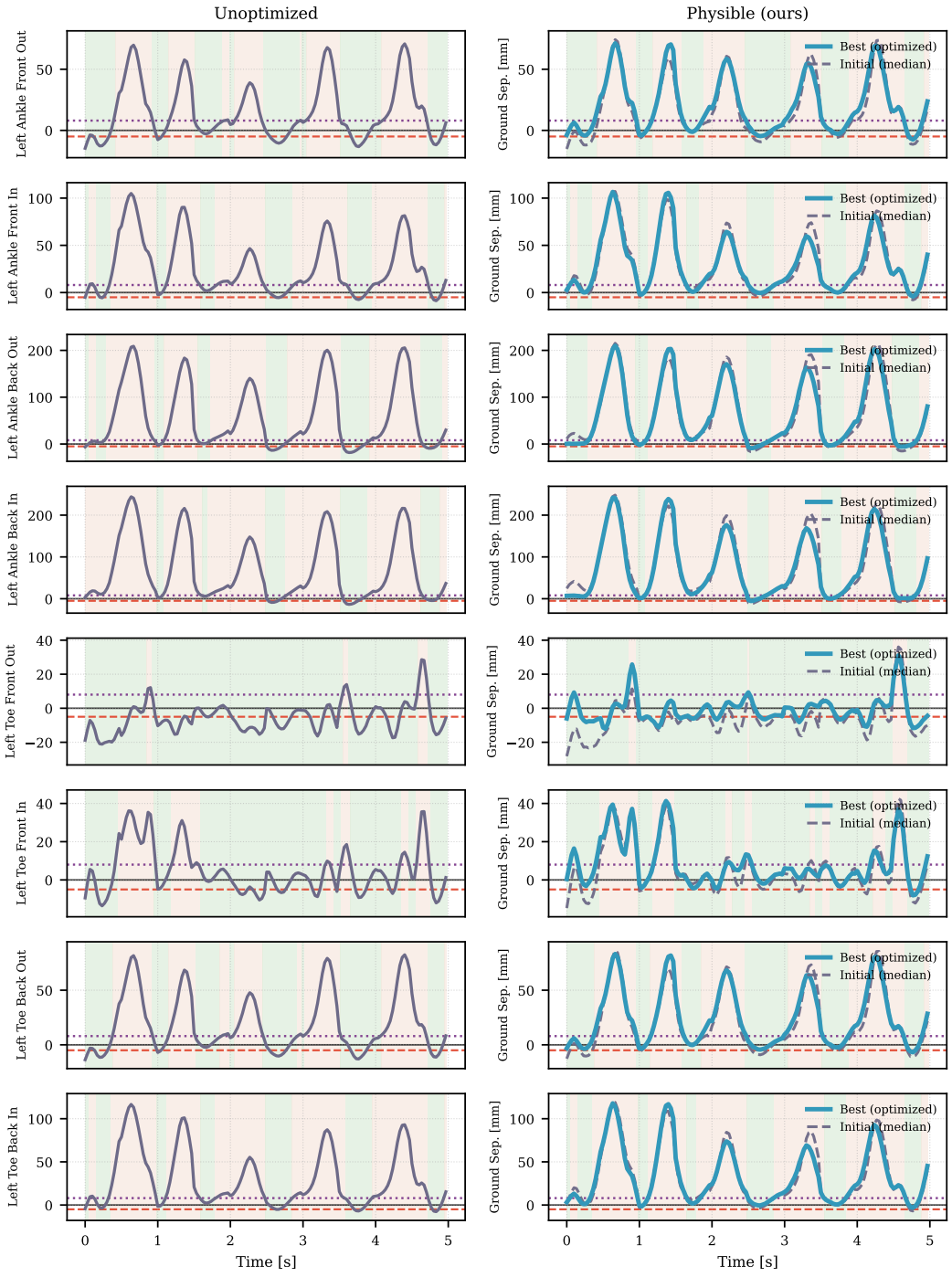


Figure 17: Behavior comparison of the *contact pattern preservation* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character. The soft green and warm tints visualize time periods where the respective contact point should be in contact and *not* in contact with the ground, based on the contact pattern identified via a majority vote using the height threshold (purple-dashed line) using a batch of diffused, unoptimized motions. The red-dashed lines denotes the maximum-allowed ground penetration - Part 1.

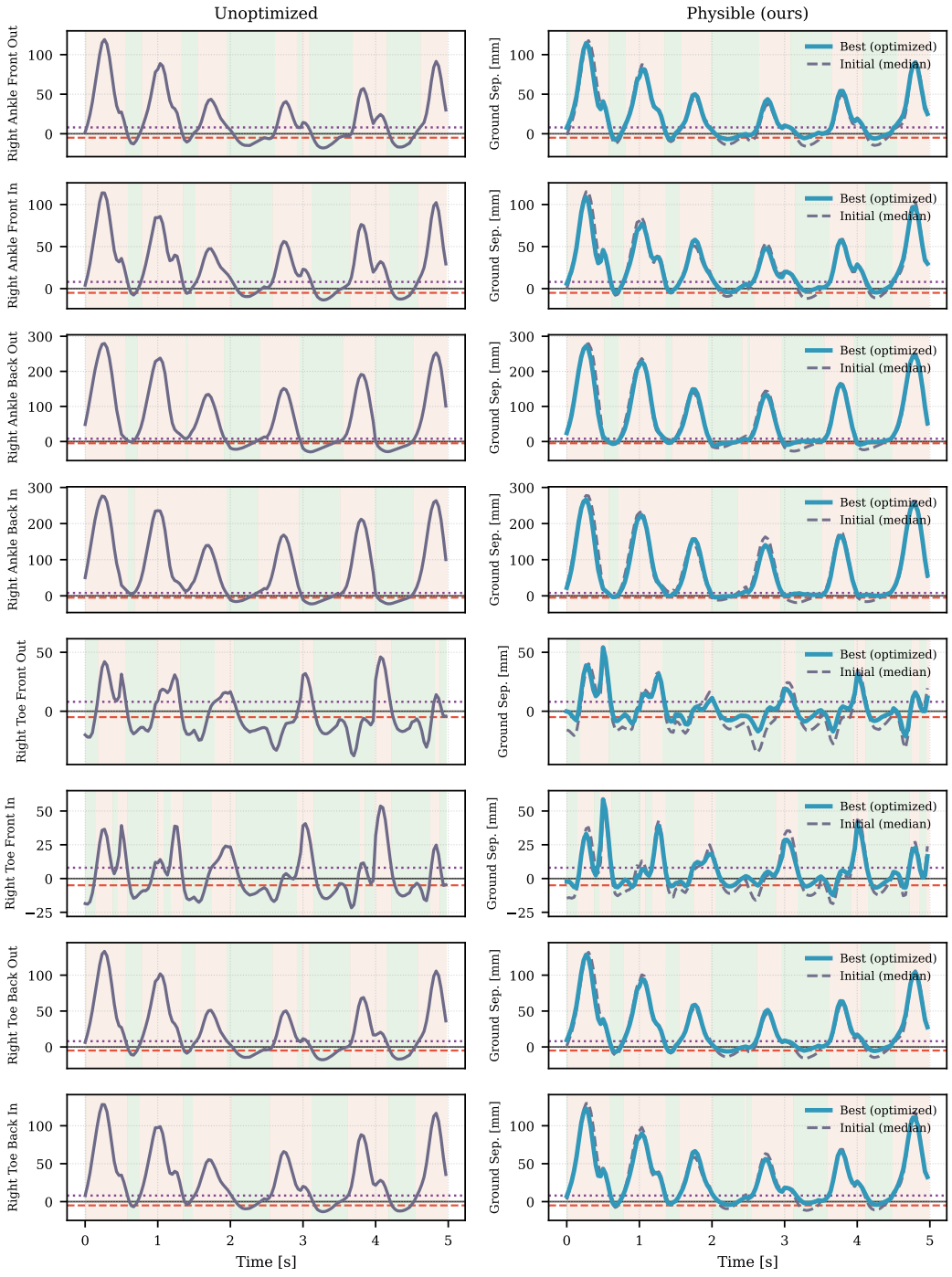


Figure 18: Behavior comparison of the *contact pattern preservation* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character. The soft green and warm tints visualize time periods where the respective contact point should be in contact and *not* in contact with the ground, based on the contact pattern identified via a majority vote using the height threshold (purple-dashed line) using a batch of diffused, unoptimized motions. The red-dashed lines denote the maximum-allowed ground penetration - Part 2.

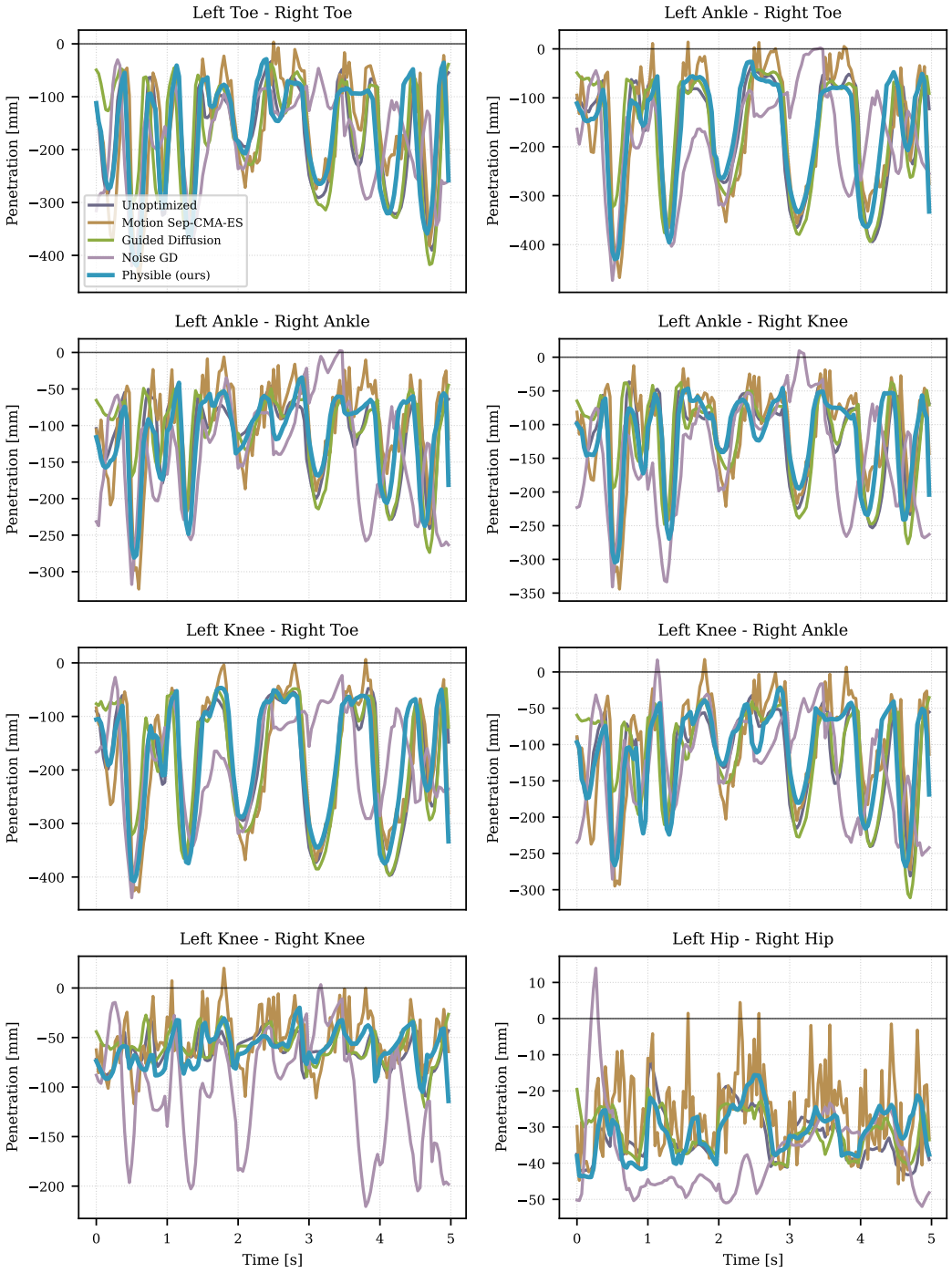


Figure 19: Behavior comparison of the *self-collision penetration* for the kinematic feasibility optimization for 150 time steps of the *neutral* walking style on the SMPL humanoid character.

G.4.2 CAT MOTION: GROUND-PENETRATION REDUCTION

Figure 20 compares unoptimized and Physible-optimized *cat* motion on the SMPL humanoid. The matched snapshots and foot trajectories show reduced ground penetration while retaining the characteristic poses. The ground-distance traces connect the visible foot-ground interaction to the remaining penetration events.

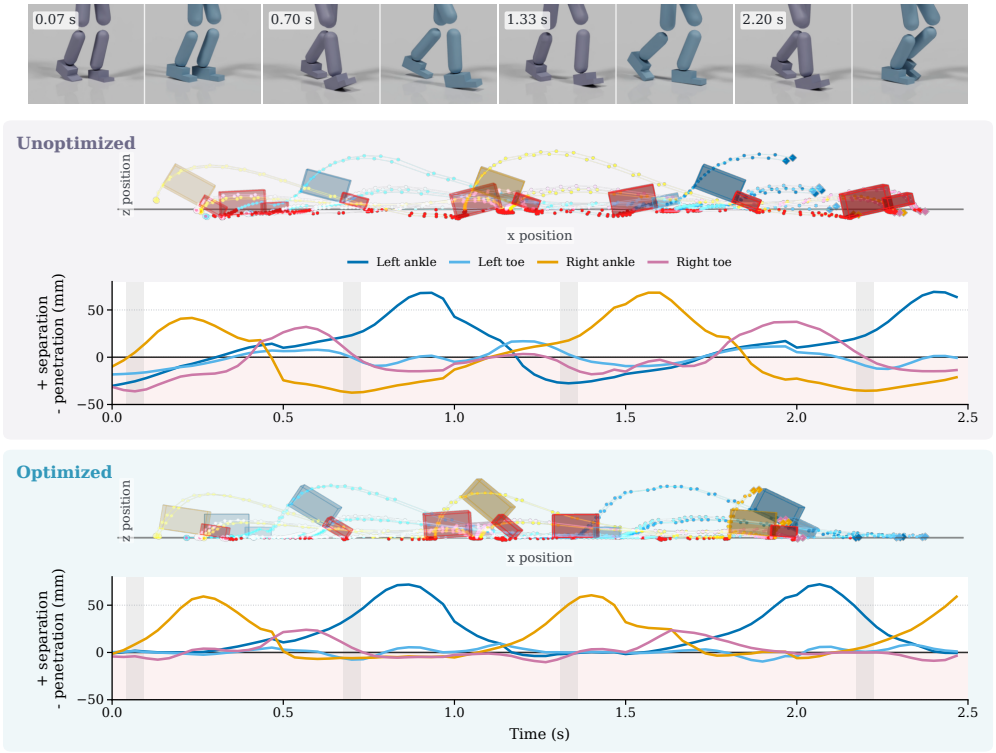


Figure 20: Ground-contact behavior for the SMPL humanoid during the *cat* motion. The top row shows rendered snapshots at time instants with visible foot-ground interaction for the unoptimized gray-purple character and optimized cyan-blue character. The lower panels compare unoptimized and optimized motion with matching side-view contact trajectories in the x-z plane and grouped foot-body ground distances for the ankles and toes. Positive values indicate separation above the ground plane, while negative values indicate penetration below the ground plane. The optimized motion significantly reduces foot-ground penetration while preserving the overall motion style.

G.4.3 LAWN-MOWER MOTION: SELF-COLLISION AVOIDANCE

We evaluate self-collision avoidance on a 150-frame *lawn-mower* sequence of the SMPL humanoid, whose unoptimized motion exhibits arm interpenetration. In this targeted experiment, Physible optimizes only the self-collision constraint for 100 iterations. The experiment settings are given in Appendix F.2.

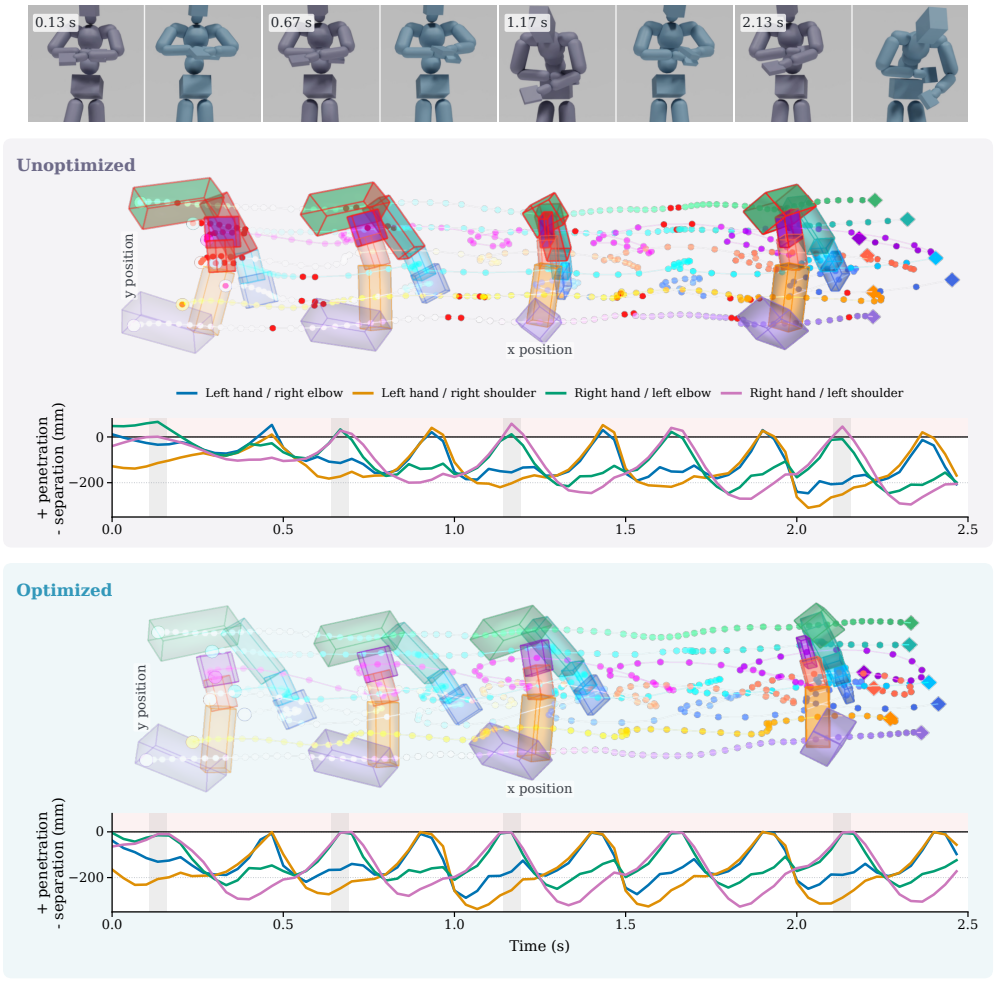


Figure 21: Arm self-collision behavior for the SMPL humanoid during a *lawn-mower* motion. The top row shows rendered snapshots at collision-relevant time instants for the unoptimized gray-purple character and optimized cyan-blue character. The lower panels compare unoptimized and optimized motion with matching top-down arm trajectory views and selected hand-arm collision distances. Red-shaded and red-bordered collision geometries and red trajectory markers emphasize time steps for which collisions for the respective bodies take place. Positive signed distances denote penetration, while negative values denote separation. The optimized motion reduces arm interpenetration while preserving the overall motion style.

The rendered snapshots and arm trajectories in Fig. 21 show how optimization separates the arms at collision-relevant times while retaining the characteristic lawn-mower motion. The corresponding collision-pair traces in Figs. 22 and 23 show that penetration is reduced to negligible levels for the evaluated body pairs. Together, these results demonstrate effective self-collision avoidance in this sequence while preserving the overall arm-motion style.

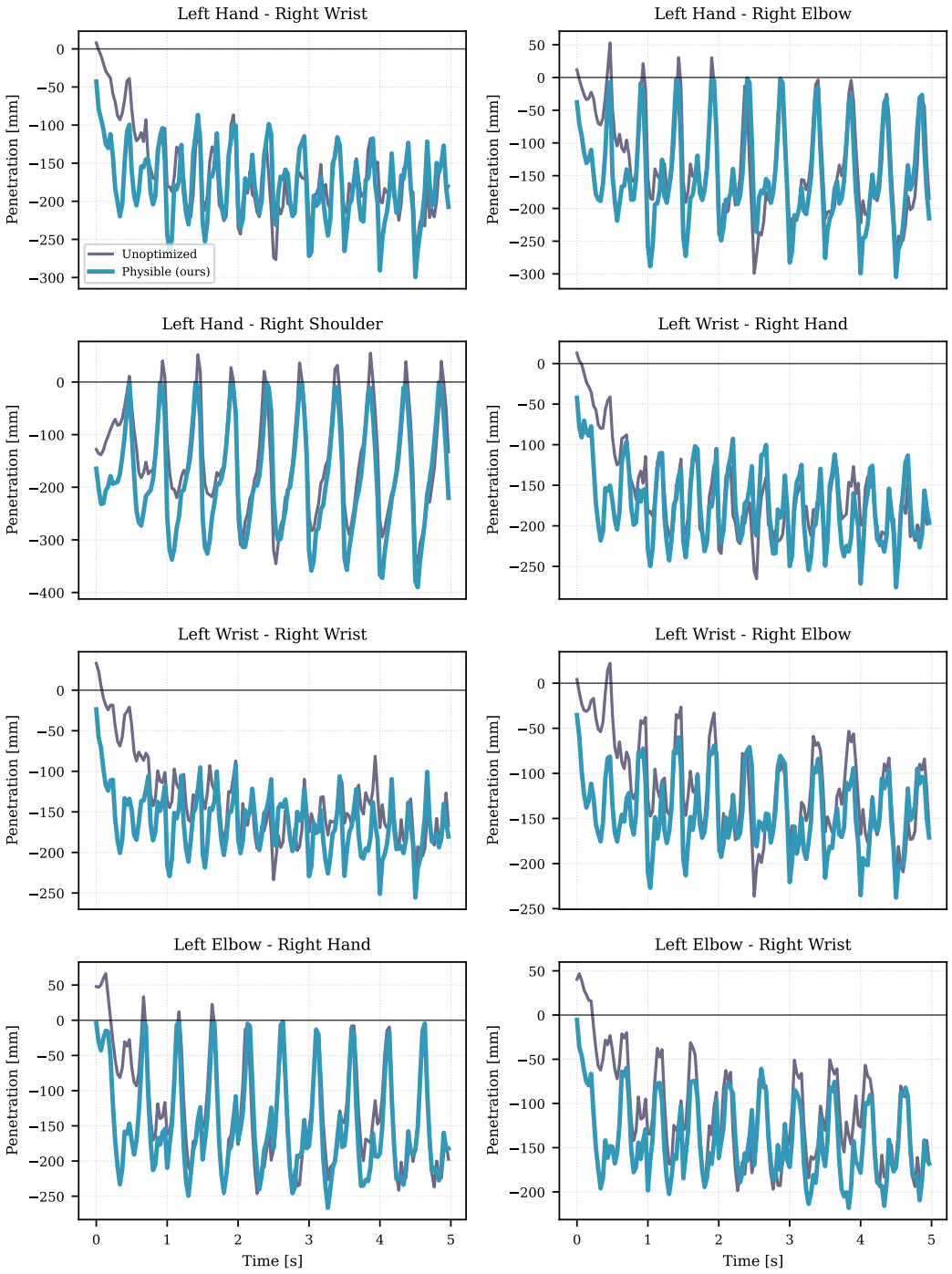


Figure 22: Part 1 of the comparison of the unoptimized vs. optimized self-collision penetration distance when used as a constraint during the kinematic feasibility optimization for 150 time steps of the *lawn mower* walking style on the SMPL humanoid character. The penetration/separation distances are determined via an SAT algorithm applied to pre-defined collision pairs for bodies approximated as cuboids along separate leaves of the kinematic tree.

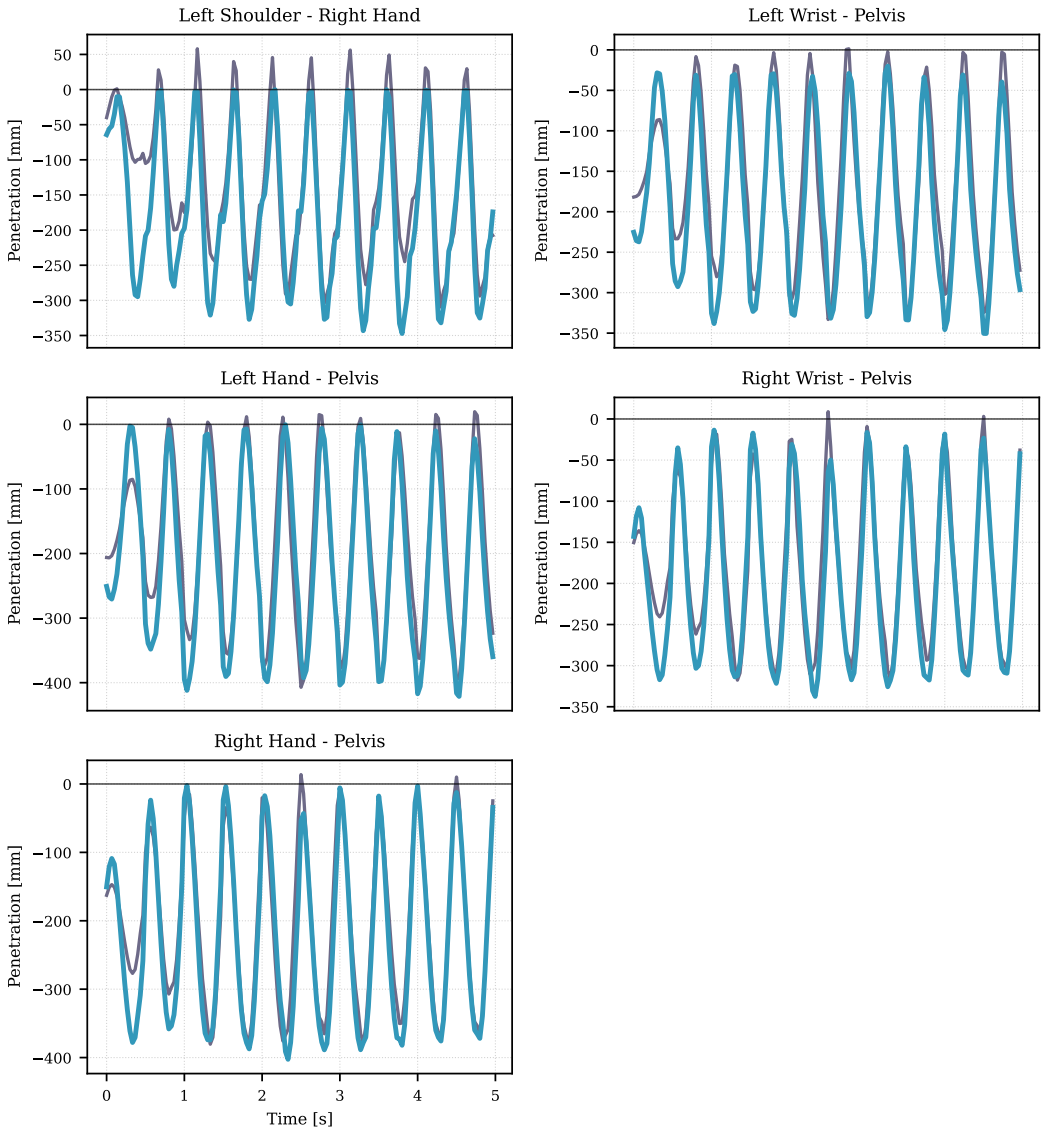


Figure 23: Part 2 of the comparison of the unoptimized vs. optimized self-collision penetration distance when used as a constraint during the kinematic feasibility optimization for 150 time steps of the *lawn mower* walking style on the SMPL humanoid character. The penetration/separation distances are determined via an SAT algorithm applied to pre-defined collision pairs for bodies approximated as cuboids along separate leaves of the kinematic tree.

G.4.4 Style Preservation across Selected Motions

Figure 24 illustrates style preservation for *cat*, *cross-over*, and *duck-foot*. Matched-time snapshots compare characteristic poses, while sliding-window classification over each full 8 s clip shows how style preservation varies over time. These examples complement the aggregate style accuracy and FID results in Tab. 1.

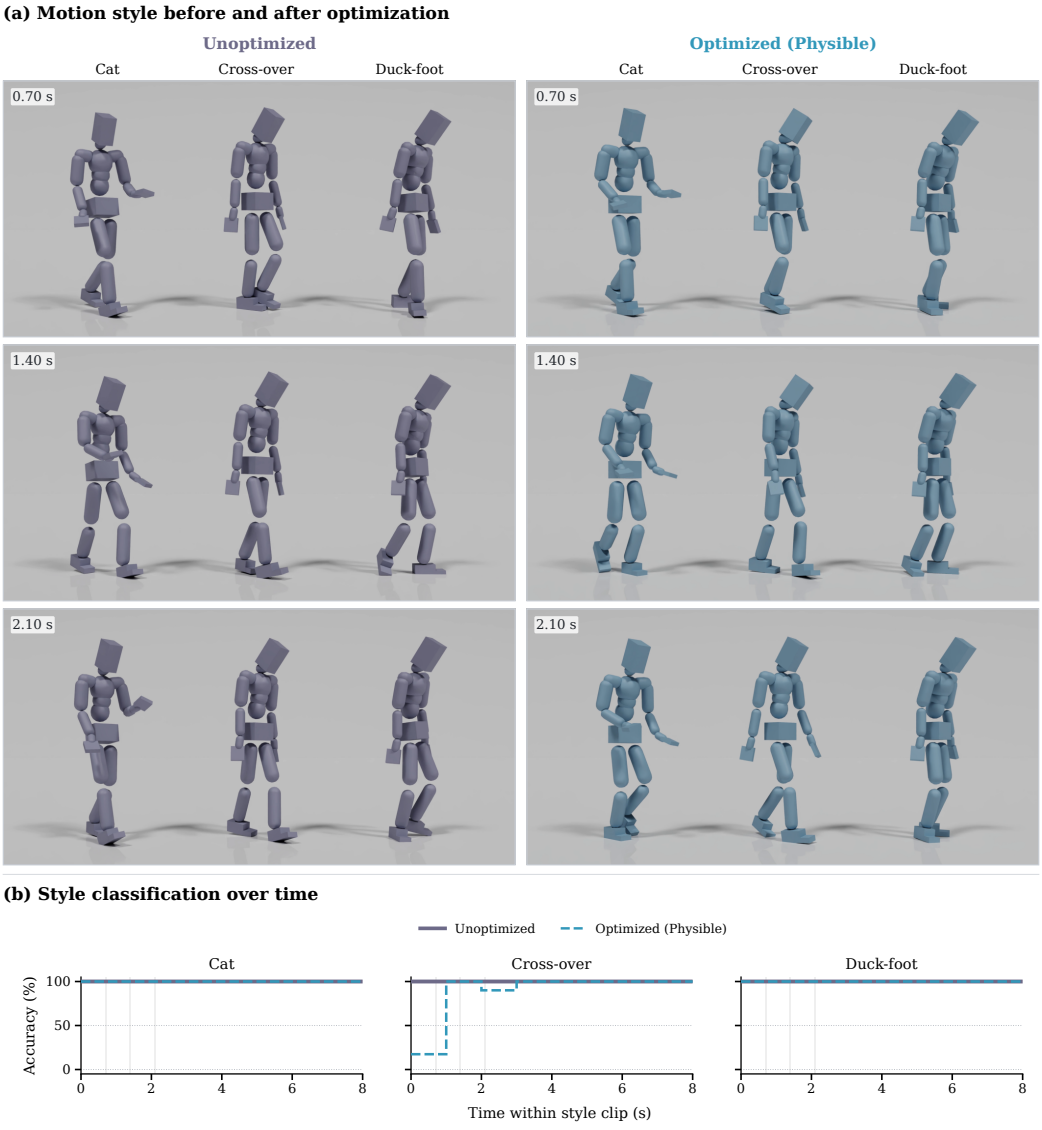


Figure 24: Motion-style preservation after kinematic feasibility optimization with Physible on the SMPL humanoid. **(a)** Selected, matched-time snapshots of the *cat*, *cross-over*, and *duck-foot* styles at 0.70 s, 1.40 s, and 2.10 s, comparing unoptimized motion (left) with Physible (right). **(b)** Style classification accuracy over each full 8 s clip, computed from 15-frame sliding windows and averaged in 1 s bins; vertical markers indicate snapshot times. Cat and duck-foot retain 100% accuracy, while cross-over exhibits a temporary decrease after optimization.

H SUPPLEMENTARY DYNAMIC FEASIBILITY RESULTS

This appendix first compares methods and evaluates results across motion styles and contact conditions, then analyzes selected motions and downstream RL tracking. Experimental settings and metric definitions are given in Appendix F.

H.1 QUANTITATIVE EVALUATION

H.1.1 COMPARISON WITH BASELINES FOR TWO-FOOT-JUMP

The quantitative results in Tab. 3 show that most optimization methods reduce the dynamical infeasibility metric relative to the unoptimized motion, but with different trade-offs. *Noise GD* obtains the lowest acceleration-reconstruction error and ID residual (0.013 and 1.94, respectively), but requires by far the longest wall-clock time per horizon and reduces LDJ from -7.71 to -10.03. Direct *Motion Sep-CMA-ES* also strongly reduces the two dynamical metrics, but has the lowest LDJ (-11.67) among the compared methods. In contrast, Physible reduces acceleration reconstruction from 0.454 to 0.081 and the ID residual from 84.4 to 26.5, while attaining the highest LDJ among the optimized methods (-8.51). *Guided Diffusion* is comparatively fast in total horizon time, but fails in this setting, increasing both dynamical errors and reducing LDJ to -9.91. Physible trades larger remaining residuals for smoother motion and lower computational cost than *Noise GD*. Appendix H.2.1 examines the associated changes in motion style and contact behavior.

H.1.2 RESULTS ACROSS MOTION STYLES AND CONTACT CONDITIONS

Table 8 compares Physible with unoptimized generation on 11 bipedal robot sequences: six flat-ground motion styles, neutral walking on uphill and downhill slopes of 25° with $\mu = 1.0$, and neutral walking on flat ground with $\mu \in \{0.50, 0.10, 0.05\}$. Each condition contributes one sequence per method. The two-foot jump spans $5\,\mathrm{s}$; all other sequences span $3\,\mathrm{s}$. We report temporal means and percentiles of the metrics defined in Appendix F.4; these characterize variation within each sequence. They do not measure variability across optimization seeds, which remains to be evaluated systematically. For each dynamical metric, we summarize the relative improvement as the median of $100(1 - \bar{E}^{\mathrm{Physible}}/\bar{E}^{\mathrm{Unoptimized}})$ across the 11 sequences, where $\bar{E}$ denotes the sequence mean. This gives each sequence equal weight regardless of duration.

Physible lowers both mean dynamical residuals and their 90th percentiles in every evaluated sequence. The median reductions in mean acceleration-reconstruction error and mean ID residual are approximately 89.5% and 79.5%, respectively. Improvements vary by motion: left hop and kick reduce both means by about 98%, whereas bouncy right reduces acceleration-reconstruction error by 68% and the ID residual by 26%. The slope results show larger reductions downhill (92% and 87%) than uphill (71% and 57%). Under reduced friction, Physible continues to improve both metrics, although the remaining mean ID residual rises from 7.22 at $\mu = 0.50$ to 17.39 at $\mu = 0.10$ and 30.63 at $\mu = 0.05$. These results support improved dynamic consistency within the evaluated robot model across the tested styles and contact conditions, while leaving appreciable infeasibility in some sequences.

Evaluated over 25-frame retained windows for all sequences, mean LDJ is lower after optimization in every sequence, indicating a smoothness tradeoff. The largest decrease occurs at $\mu = 0.50$, from -7.63 to -8.97.

Table 8: Dynamical feasibility across bipedal-robot sequences. Each sequence is reported separately for Unoptimized and Physible. Entries are mean [p90] for acceleration reconstruction and inverse-dynamics residual, and mean [p10] for LDJ. Bold marks the better mean within each sequence and metric.

Sequence	Frames / Duration [s]	Acceleration reconstruction ↓		Inverse-dynamics residual ↓		LDJ ↑	
		Unoptimized	Physible	Unoptimized	Physible	Unoptimized	Physible
Flat-ground motion styles							
Two-foot jump	250 / 5	0.454 [0.954]	**0.081 [0.174]**	84.40 [186.93]	**26.48 [62.70]**	**-7.71 [-8.11]**	-8.51 [-8.83]
Left hop	150 / 3	0.511 [0.721]	**0.009 [0.014]**	106.43 [167.33]	**1.91 [3.19]**	**-8.57 [-8.96]**	-8.83 [-9.42]
Bouncy right	150 / 3	0.505 [1.014]	**0.161 [0.271]**	74.11 [147.12]	**54.61 [95.67]**	**-7.59 [-7.80]**	-8.27 [-8.56]
Skip	150 / 3	0.800 [0.927]	**0.059 [0.175]**	93.73 [166.87]	**19.20 [64.97]**	**-7.69 [-7.95]**	-8.31 [-8.73]
Kick	150 / 3	0.893 [1.450]	**0.011 [0.021]**	116.91 [187.61]	**2.70 [6.05]**	**-7.92 [-8.75]**	-8.31 [-8.62]
Morris	150 / 3	0.438 [0.779]	**0.046 [0.149]**	78.25 [157.96]	**15.26 [56.44]**	**-7.93 [-8.41]**	-8.50 [-8.92]
Terrain transfer: neutral style, $\mu = 1.0$							
Uphill 25°	150 / 3	0.451 [0.952]	**0.131 [0.275]**	89.31 [133.85]	**38.65 [74.45]**	**-7.79 [-8.13]**	-8.29 [-8.71]
Downhill 25°	150 / 3	0.387 [0.987]	**0.032 [0.091]**	66.04 [138.51]	**8.58 [24.20]**	**-7.60 [-7.84]**	-8.11 [-8.40]
Friction variation: neutral style, flat ground							
$\mu = 0.50$	150 / 3	0.330 [0.837]	**0.025 [0.058]**	49.98 [132.39]	**7.22 [15.54]**	**-7.63 [-7.77]**	-8.97 [-10.16]
$\mu = 0.10$	150 / 3	0.487 [0.932]	**0.054 [0.161]**	57.82 [140.16]	**17.39 [62.26]**	**-7.69 [-7.97]**	-8.18 [-8.49]
$\mu = 0.05$	150 / 3	0.601 [0.935]	**0.098 [0.299]**	70.10 [143.81]	**30.63 [109.14]**	**-7.63 [-7.96]**	-8.34 [-8.94]

H.2 Motion Case Studies for Qualitative Analysis

The following cases pair motion behavior with time-resolved diagnostics to allow for a qualitative analysis of the optimized motion. Acceleration, torque, and friction diagnostics use the nominal inverse-dynamics solution. Signed acceleration residuals are plotted as planned minus reconstructed acceleration, corresponding to $-\mathbf{e}_{\mathrm{acc}}$. Virtual-wrench diagnostics use the separate relaxed solve in Eq. (32). Across the method comparisons, lower dynamic residuals can coincide with degraded style, reduced smoothness, or loss of ground contact; the diagnostics are therefore interpreted together.

H.2.1 Two-Foot Jump

Comparison against Baselines We compare optimization methods on a 250-frame (5 s) two-foot-jump sequence of the bipedal robot. Figure 25 shows unoptimized generation, Motion Sep-CMA-ES, Guided Diffusion, Noise GD, and Physible, corresponding to the methods evaluated in Tab. 3. Motion snapshots and time histories of ground clearance, vertical base acceleration error, and joint jerk illustrate the tradeoffs between dynamic consistency, contact behavior, and motion quality. *Motion Sep-CMA-ES* introduces substantial jitter, reflected in rapidly varying joint accelerations and torques (Figs. 31–33 and Figs. 35–36), while *Guided Diffusion* increases vertical base acceleration errors and produces both unrealistic floating and ground penetration. *Noise GD* reduces acceleration error but leaves the character floating throughout the sequence, suggesting that optimization can settle on a physically implausible solution despite achieving a small residual. The ground-separation and slip traces show that its low slip coincides with loss of contact (Figs. 37 and 38). *Physible* achieves a more balanced tradeoff: it reduces vertical base acceleration error while retaining the characteristic jumping motion, repeated ground contacts, and comparatively smooth trajectories.

The full-sequence diagnostics compare unoptimized generation, Motion Sep-CMA-ES, Noise GD, and Physible. Guided Diffusion is omitted from these plots because its much larger errors would obscure the differences among the remaining methods.

Unoptimized vs. Optimized Physible improves dynamic consistency while retaining the characteristic jump trajectory and support phases. In Fig. 26, the rendered poses and side-view root trajectories show that the optimized motion follows a similar jump arc to the unoptimized reference. The smaller virtual root forces and acceleration-residual arrows illustrate the reduction in nonphysical residual support and acceleration mismatch, particularly around takeoff and landing. The same improvement is visible over the full sequence in the virtual-wrench and root-acceleration-residual traces (Figs. 27 and 28). The planned-versus-feasible acceleration curves in Fig. 26 and Figs. 29–33 show closer agreement after optimization for the root and joints while preserving the temporal structure of the jump.

These reductions in dynamic mismatch accompany repeated ground contacts during the support phases, as shown by the motion snapshots in Fig. 26 and the ground-separation traces in Fig. 37. The slip and contact-force plots provide complementary views of the ground interaction (Figs. 38 and 34), with contact forces remaining within the friction cones. Residual acceleration mismatch remains, but Physible retains the intended jumping behavior and support phases without an explicit contact-pattern objective.

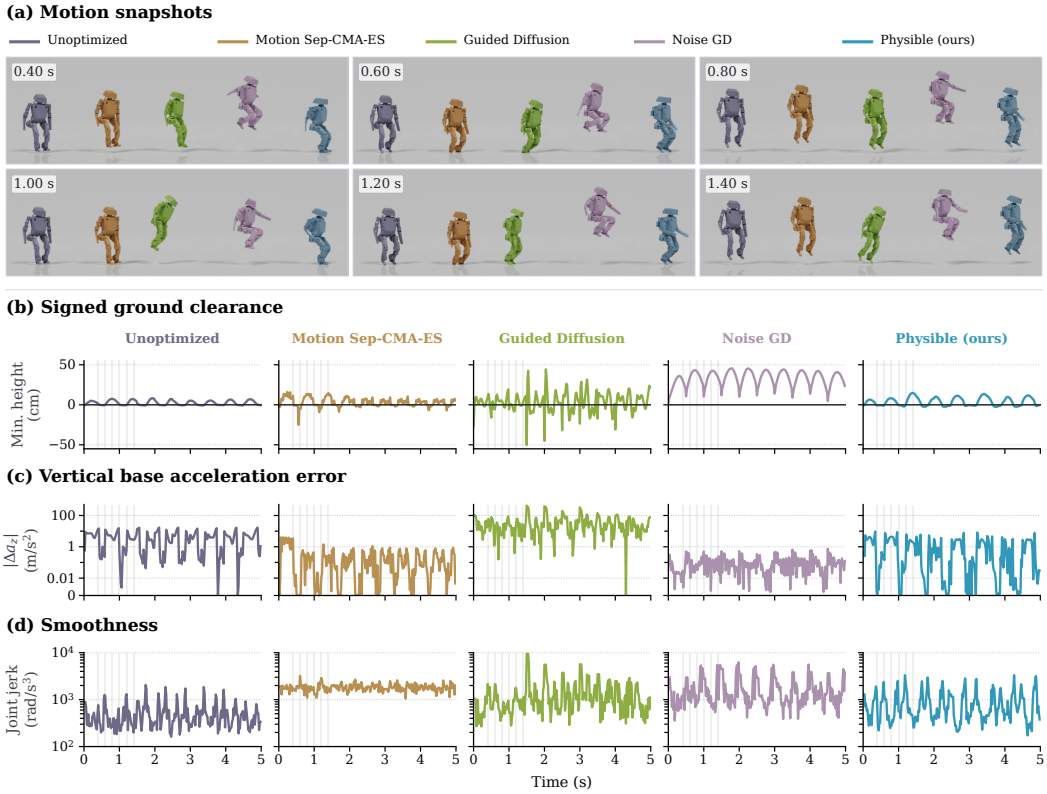


Figure 25: Comparison of dynamic feasibility optimization methods for the bipedal robot's *two-foot-jump* motion: unoptimized generation, Motion Sep-CMA-ES, Guided Diffusion, Noise GD, and Physible. (a) Motion snapshots at the indicated times. The plots show (b) signed ground clearance, with negative values indicating ground penetration; (c) the absolute difference between planned and dynamically feasible vertical base acceleration, $|\Delta a_z|$; and (d) joint jerk over a 5 s sequence. Gray vertical bands mark the snapshot times. The comparison illustrates trade-offs between acceleration consistency, ground contact, and motion smoothness: Noise GD has low acceleration error but remains above the ground, whereas Physible retains repeated ground contacts and the characteristic jumping motion.

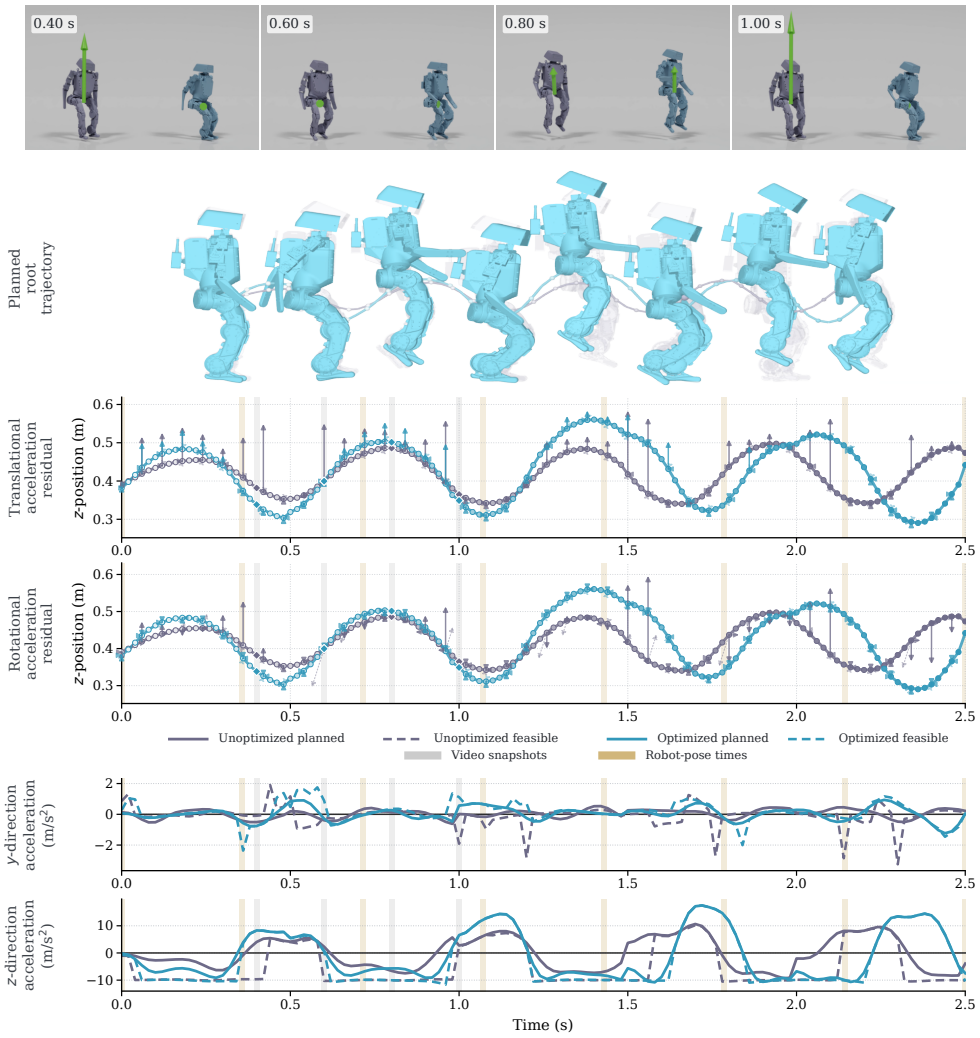


Figure 26: Dynamical feasibility optimization of the *two-foot-jump* motion on the bipedal robot over a 2.5 s horizon. The top row shows rendered video snapshots of the unoptimized and optimized motions, including the virtual root force visualization used during dynamical feasibility analysis. The second row shows the planned root trajectories from a side view: the unoptimized motion is rendered transparently in the background, while the optimized motion is rendered in the foreground. Robot configurations are shown at separate robot-pose times distributed across the full horizon, whereas the video snapshots correspond to the selected first-row rendering times. The gray vertical bands in the quantitative plots mark the video snapshot times, and the amber bands mark the robot-pose times used for the second-row configuration renderings. The translational and rotational acceleration residual plots visualize the planned root trajectory together with arrows denoting the residual acceleration, computed as planned minus dynamically feasible acceleration. Smaller residual arrows indicate improved dynamical consistency. The bottom rows compare planned and feasible root accelerations in the y- and z-directions; solid lines denote planned accelerations and dashed lines denote feasible accelerations, with separate colors for unoptimized and optimized motions.

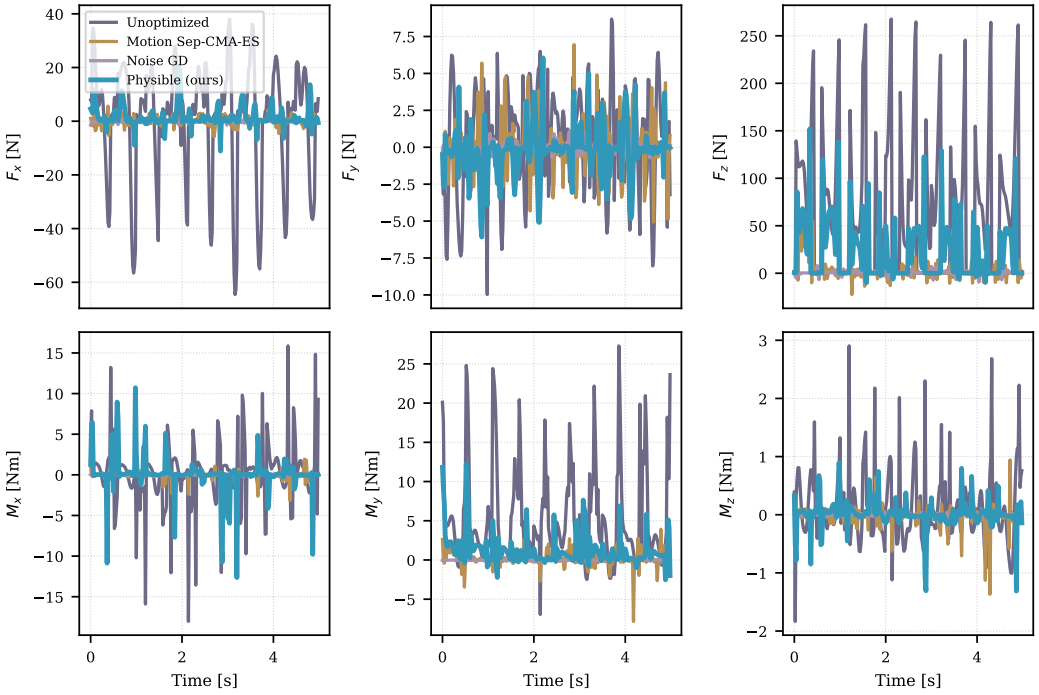


Figure 27: Behavior comparison of the *virtual wrench* resulting from the relaxed ID for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

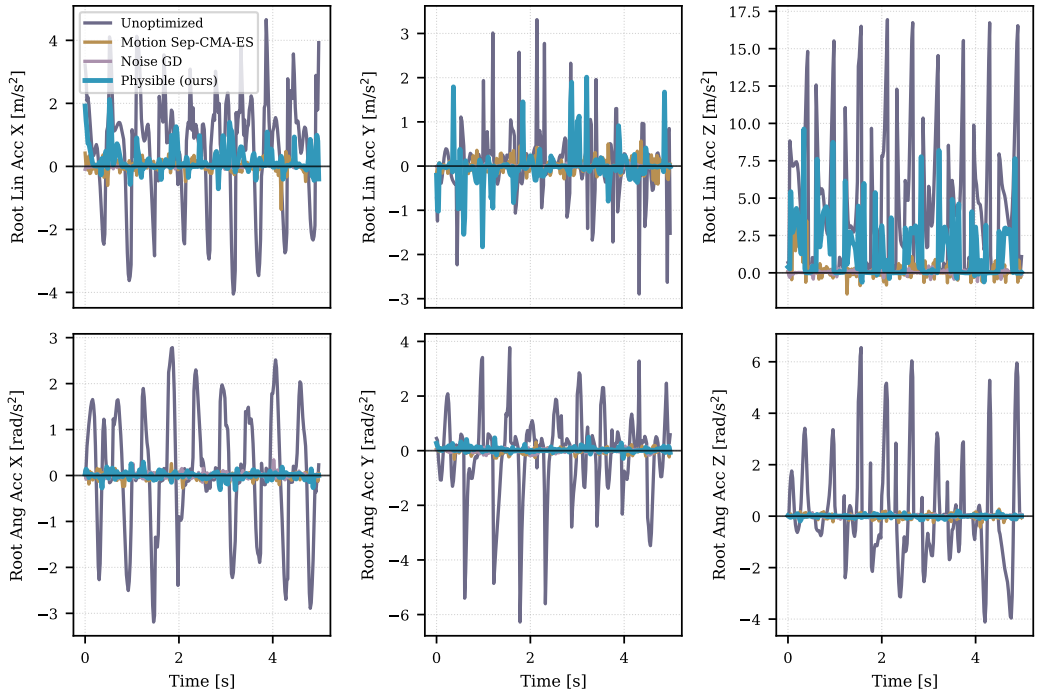


Figure 28: Behavior comparison of the *root acceleration residual* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

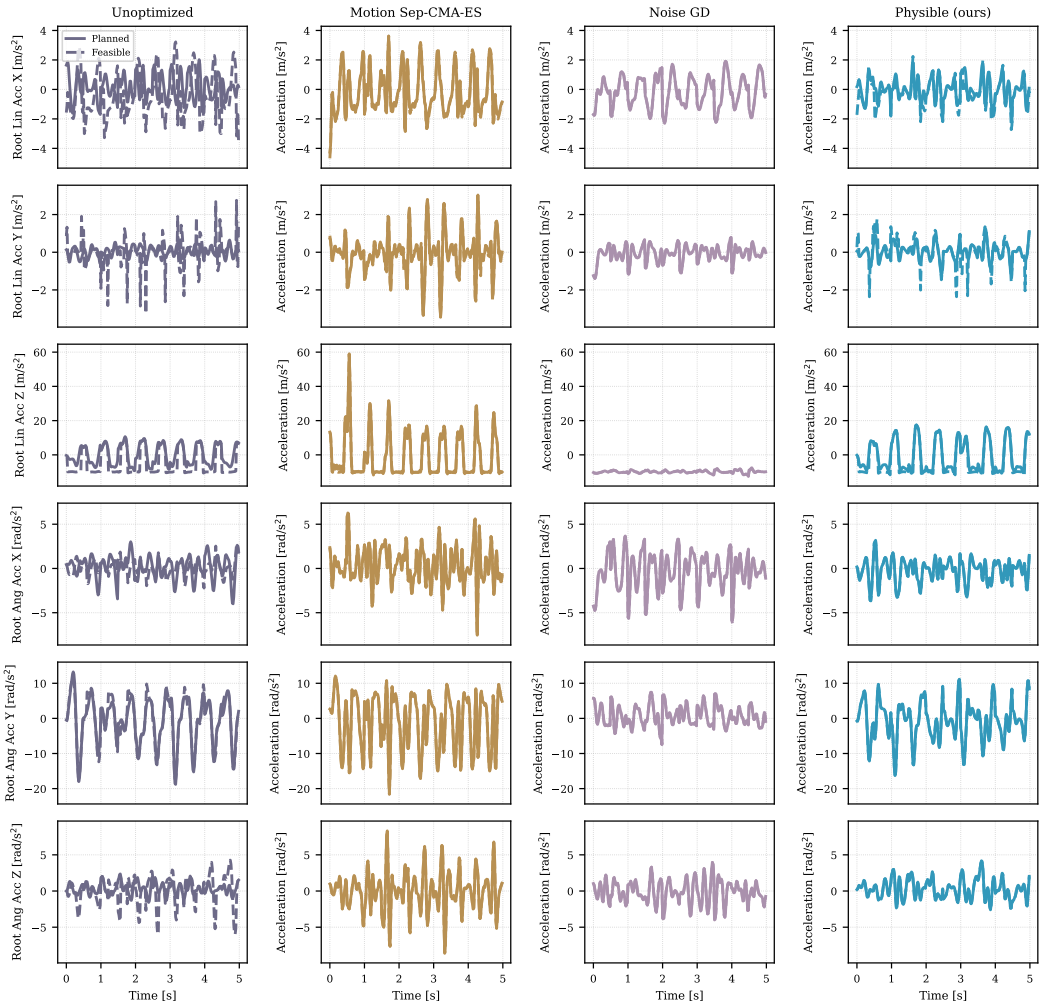


Figure 29: Behavior comparison of the *root acceleration planned vs. feasible* by method for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

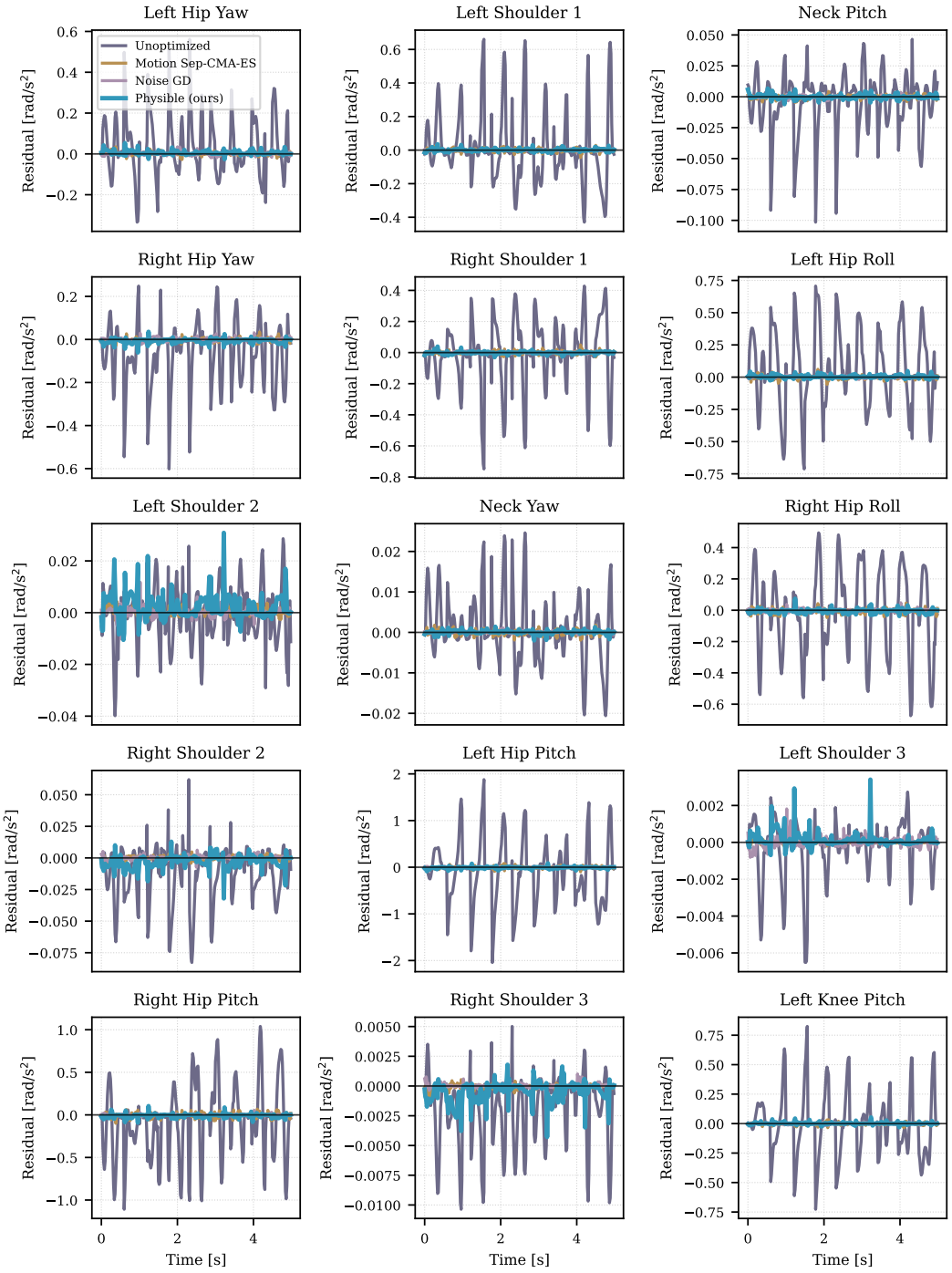


Figure 30: Behavior comparison of the *joint acceleration residual* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

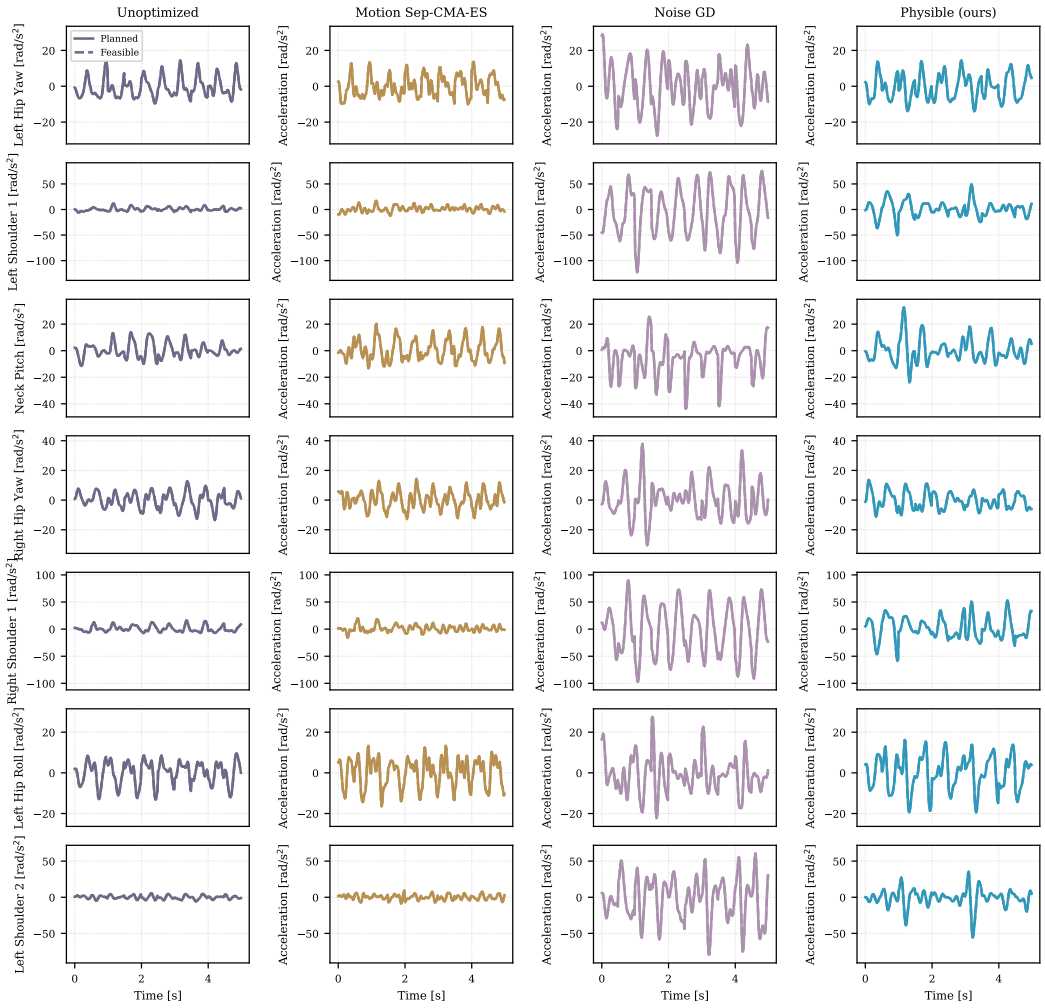


Figure 31: Behavior comparison of the *joint acceleration planned vs. feasible* by method for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot – Part 1.

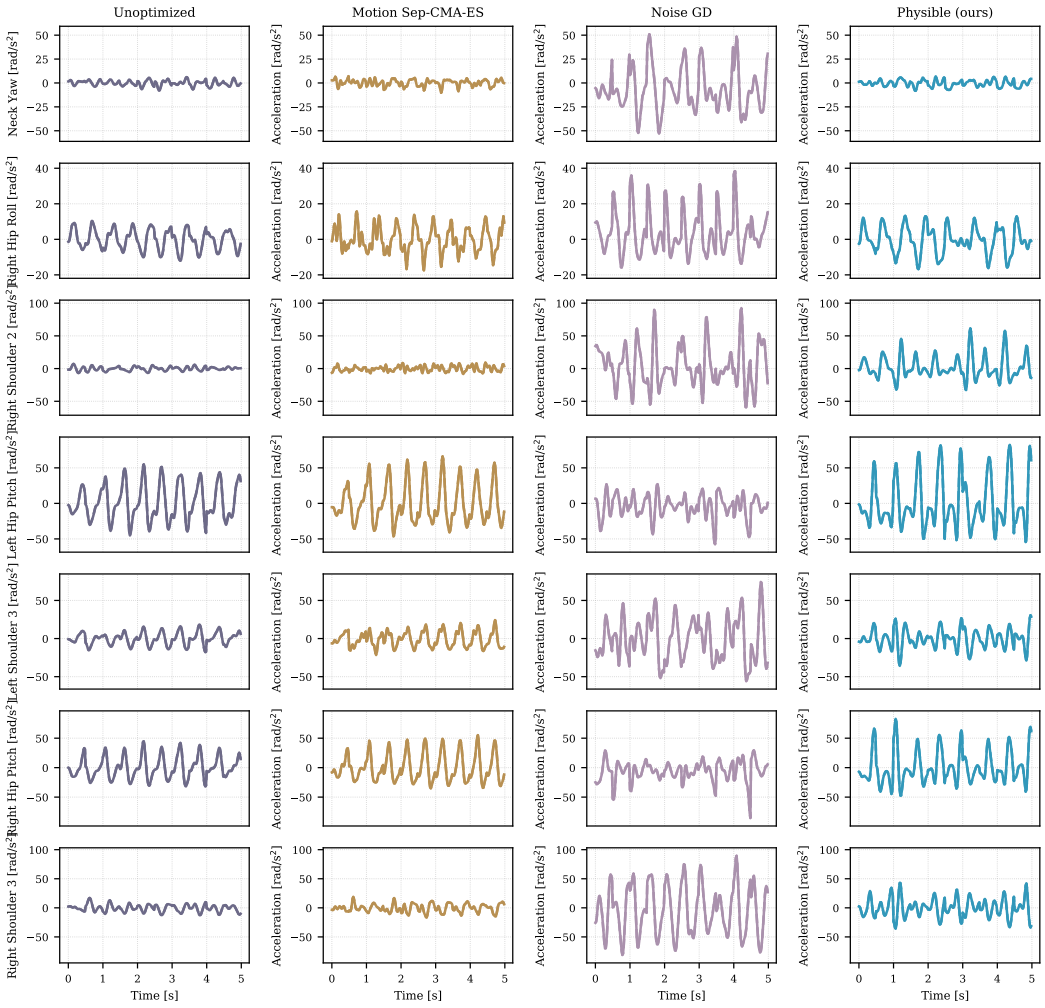


Figure 32: Behavior comparison of the *joint acceleration planned vs. feasible* by method for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot – Part 2.

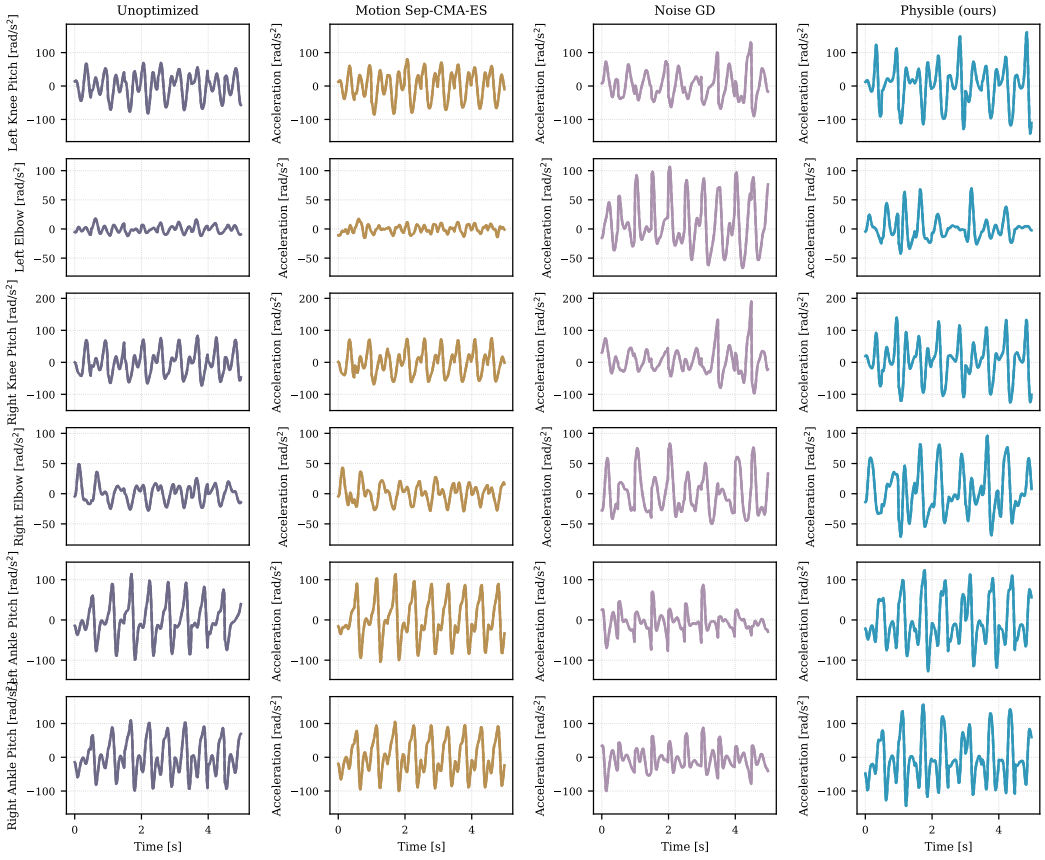


Figure 33: Behavior comparison of the *joint acceleration planned vs. feasible* by method for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot – Part 3.

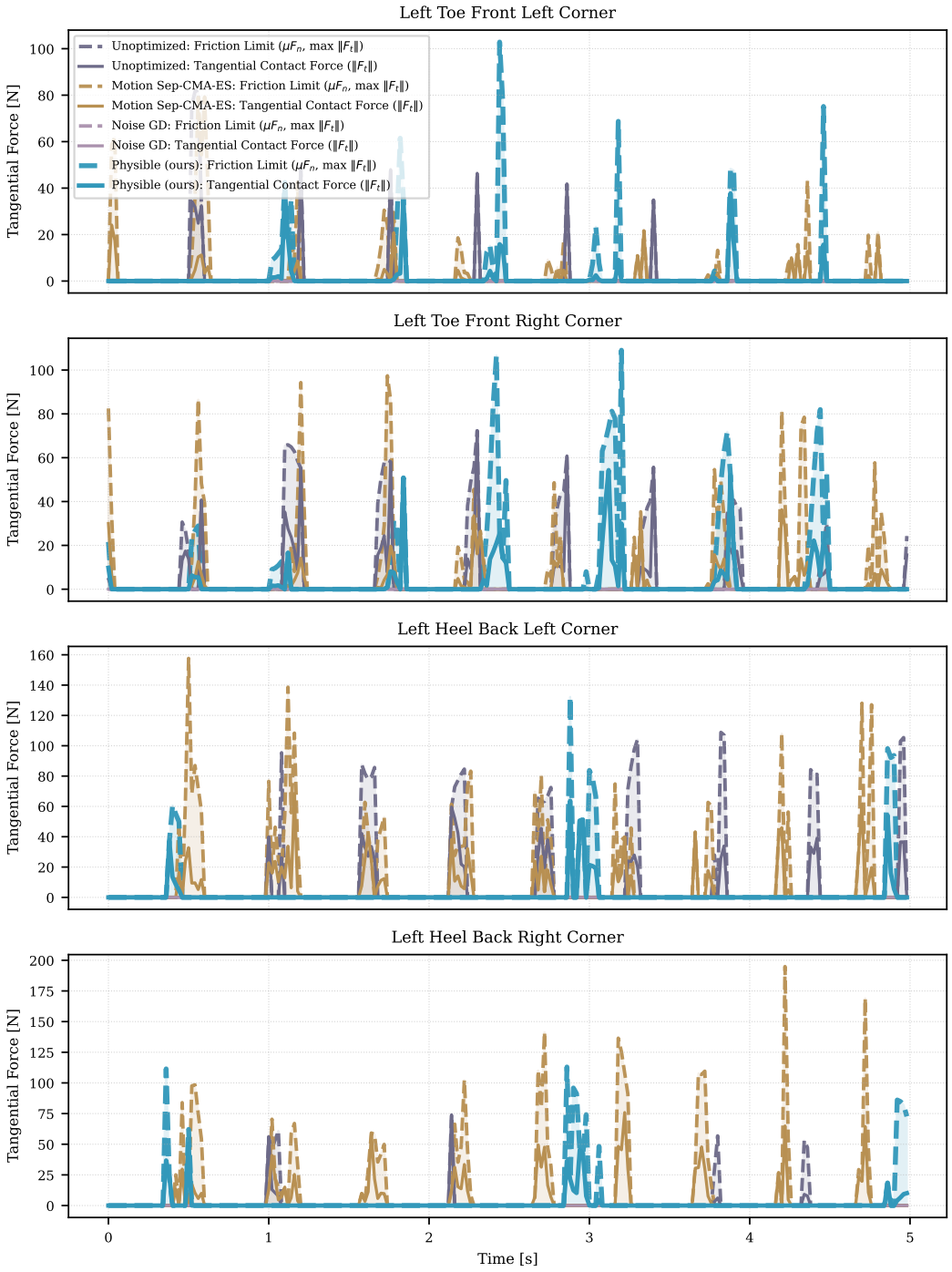


Figure 34: Behavior comparison of the *tangential ground contact forces and the friction cone* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

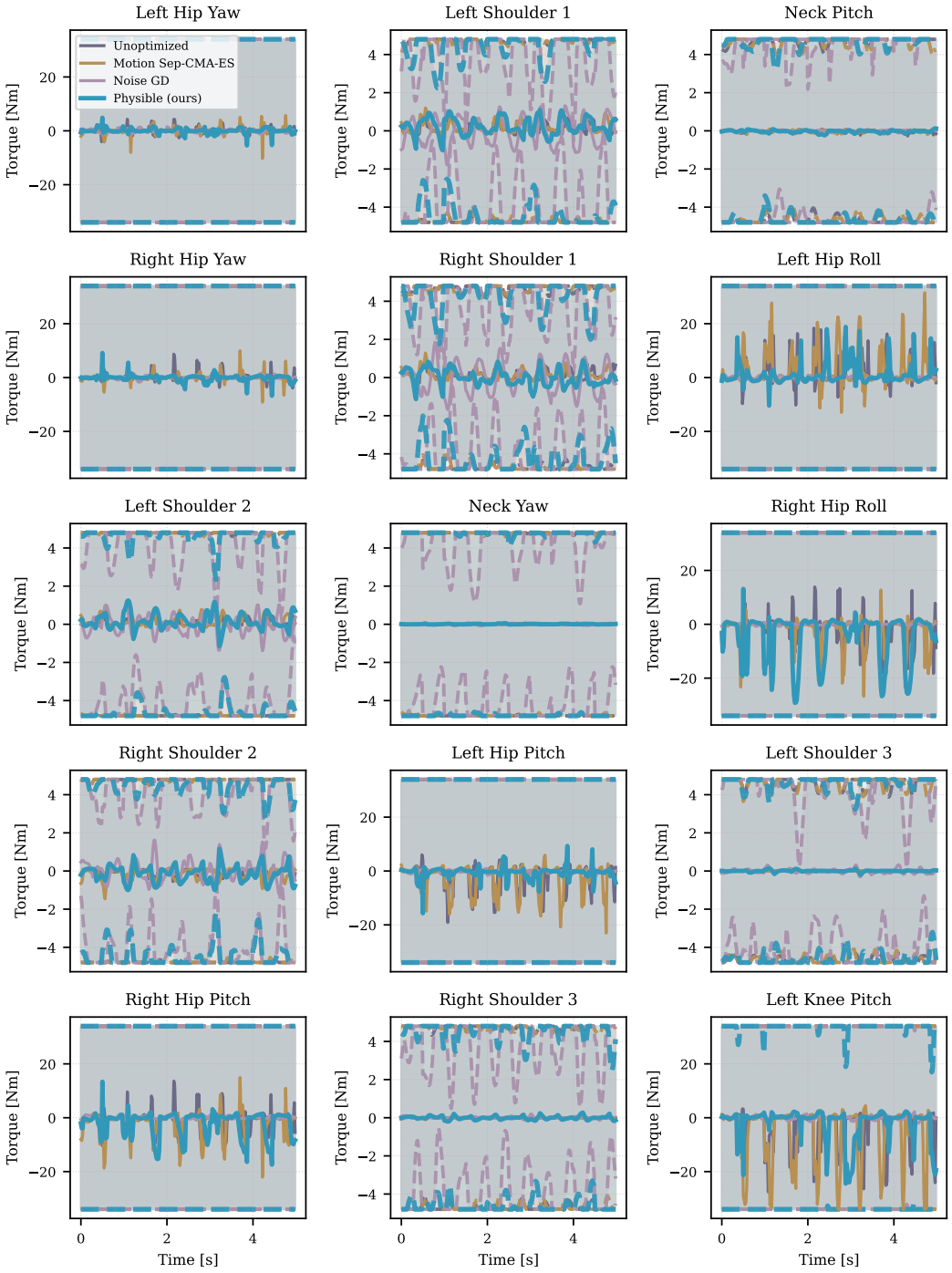


Figure 35: Behavior comparison of the *joint torques (including limits)* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot – Part 1.

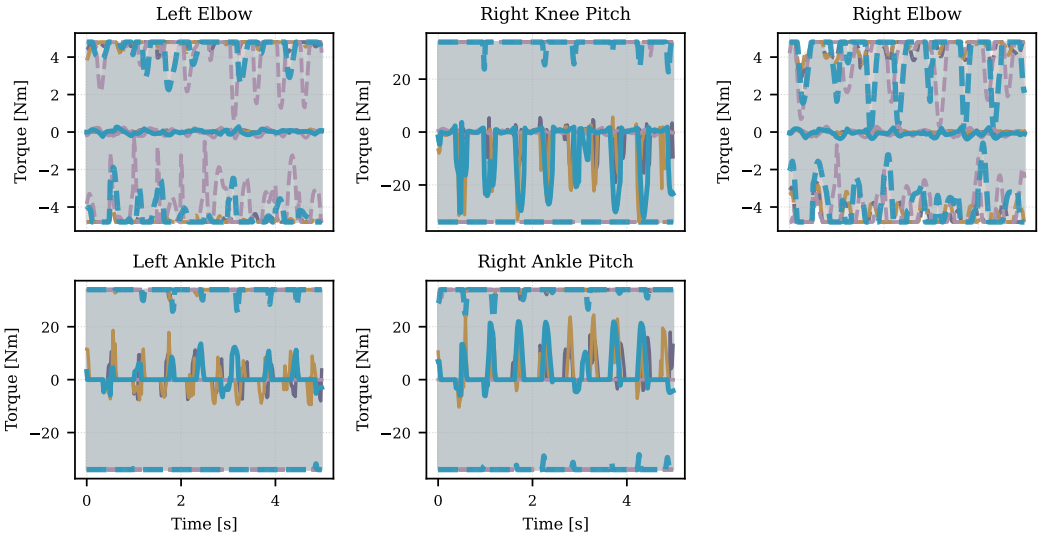


Figure 36: Behavior comparison of the *joint torques (including limits)* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot – Part 2.

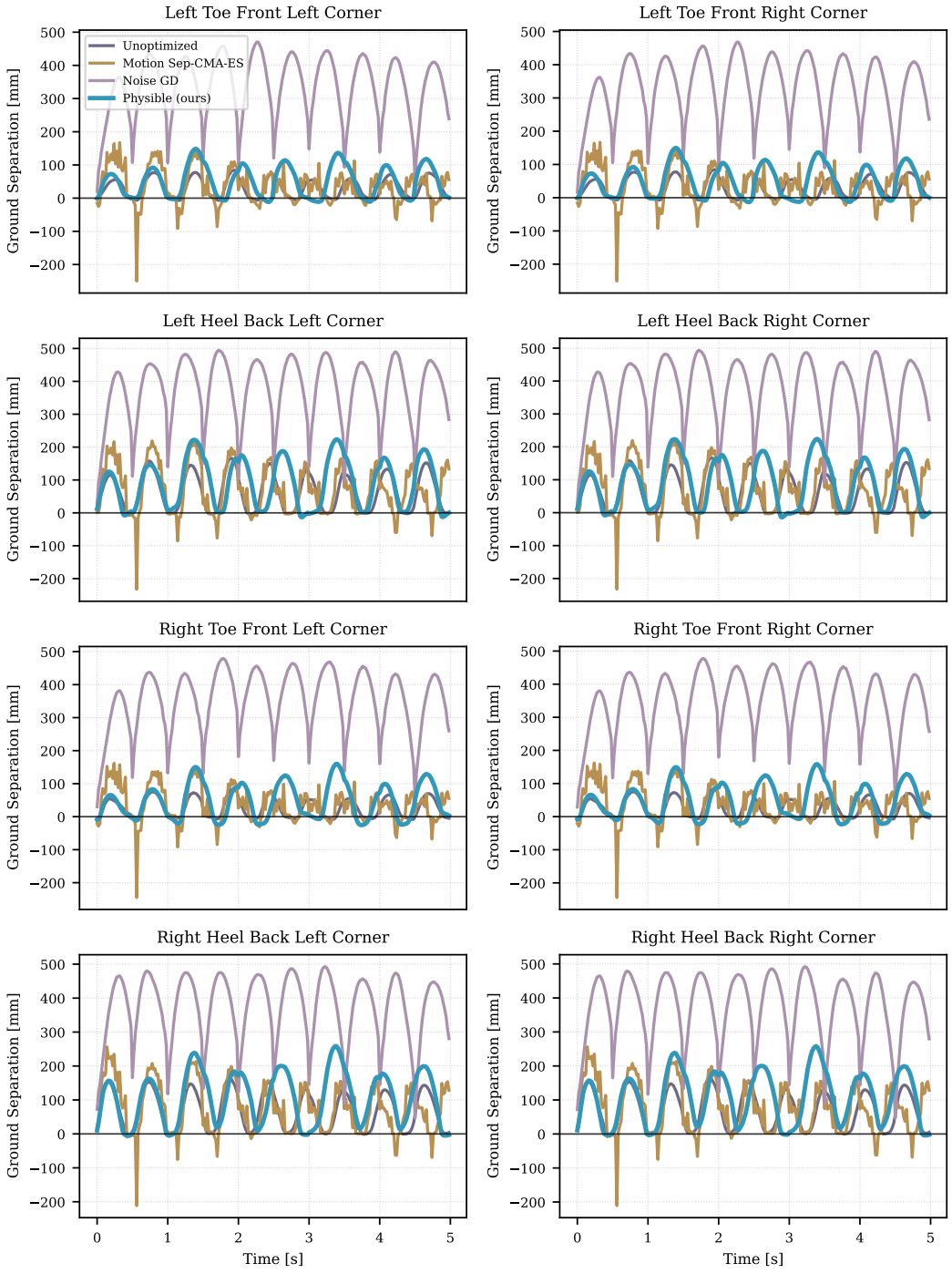


Figure 37: Behavior comparison of the *ground separation* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

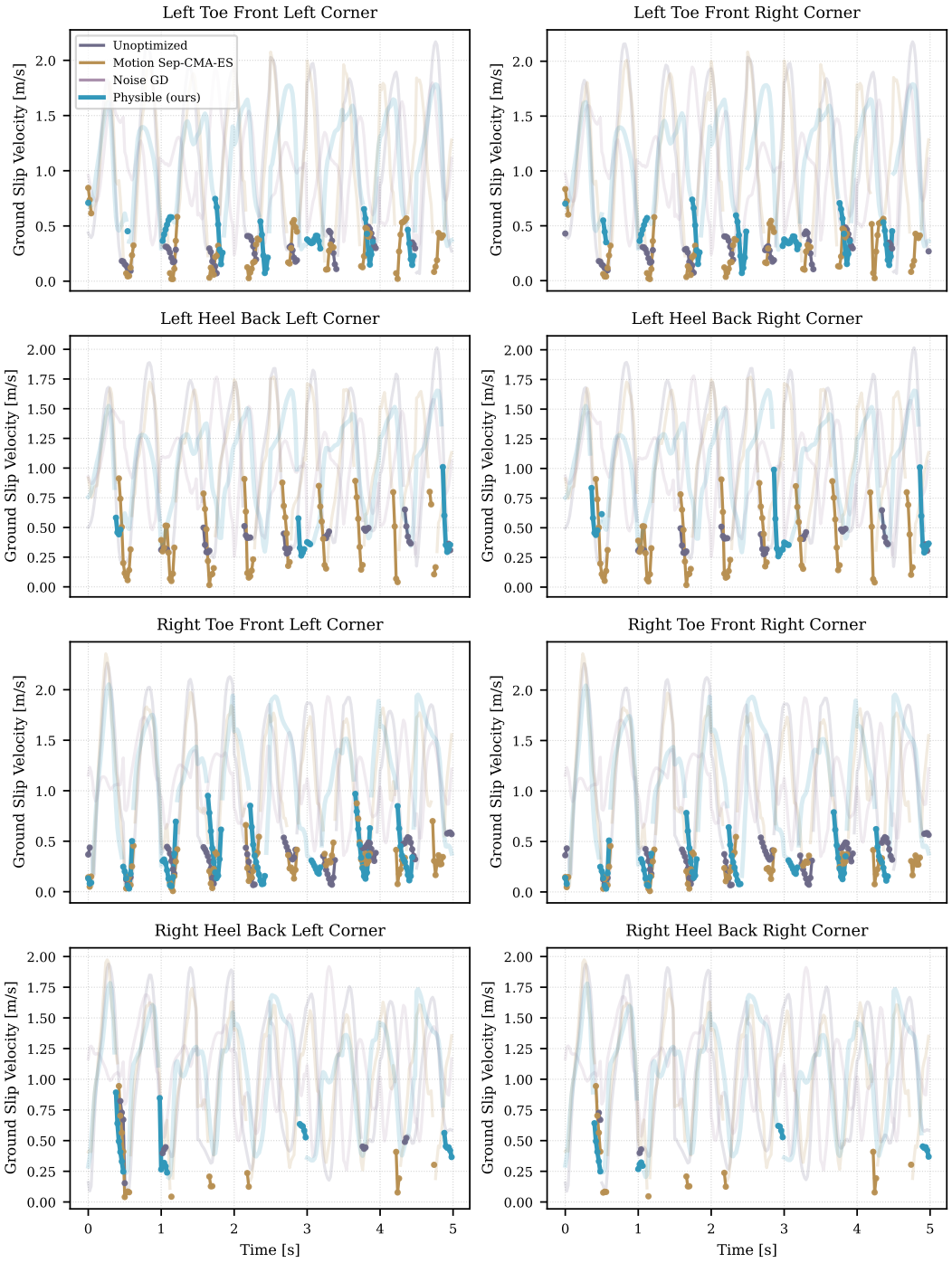


Figure 38: Behavior comparison of the *ground slip velocity* for the dynamic feasibility optimization for 250 time steps of the *two-foot-jump* motion style on the bipedal robot.

H.2.2 LEFT HOP: UNOPTIMIZED VS. PHYSIBLE

The left-hop comparison covers 150 frames (3 s) on the bipedal robot and compares unoptimized generation with Physible under asymmetric support. In Tab. 8, mean acceleration-reconstruction error decreases from 0.511 to 0.009, and the mean ID residual decreases from 106.43 to 1.91, corresponding to reductions of approximately 98% in both metrics. This qualitative analysis reveal how these improvements arise and which aspects of the original motion change during refinement.

A prominent source of infeasibility in the unoptimized motion is the vertical root acceleration. During repeated intervals, the nominal inverse-dynamics solve reconstructs an acceleration close to free fall while the planned acceleration remains much higher (Fig. 41). The resulting vertical acceleration residual repeatedly reaches approximately $10\,\mathrm{m\,s^{-2}}$ or more (Fig. 40). Consistently, the separate relaxed solve requires a large virtual upward force, often of order 100 N, together with substantial corrective moments (Fig. 39). After optimization, the planned and reconstructed root accelerations nearly coincide, and the virtual force and moment traces remain close to zero over most of the sequence, apart from short transients. The joint-acceleration residuals also decrease markedly (Fig. 42); the planned-versus-feasible joint plots retain oscillatory motion while showing changes in its amplitude and timing (Figs. 43–45).

During sustained toe support, the reconstructed tangential force remains well below its friction limit, and the joint torques remain within the displayed actuator bounds (Fig. 46 and Figs. 47–48). The optimized left hip-roll and ankle-pitch torques become more sustained, consistent with the longer support intervals.

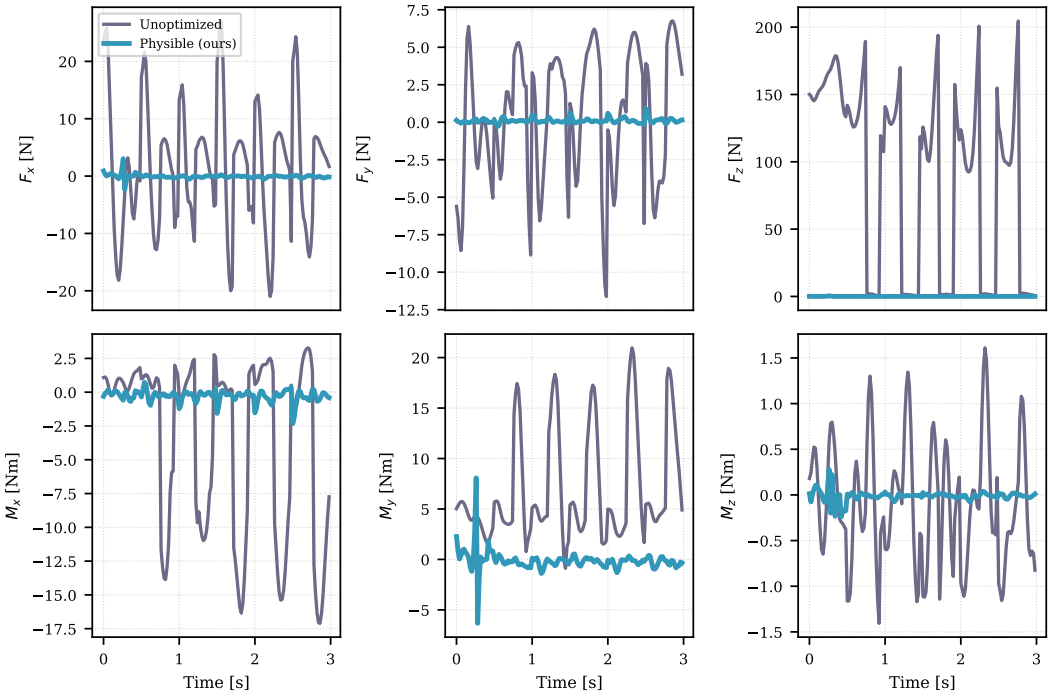


Figure 39: Behavior comparison of the *virtual wrench* resulting from the relaxed ID for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

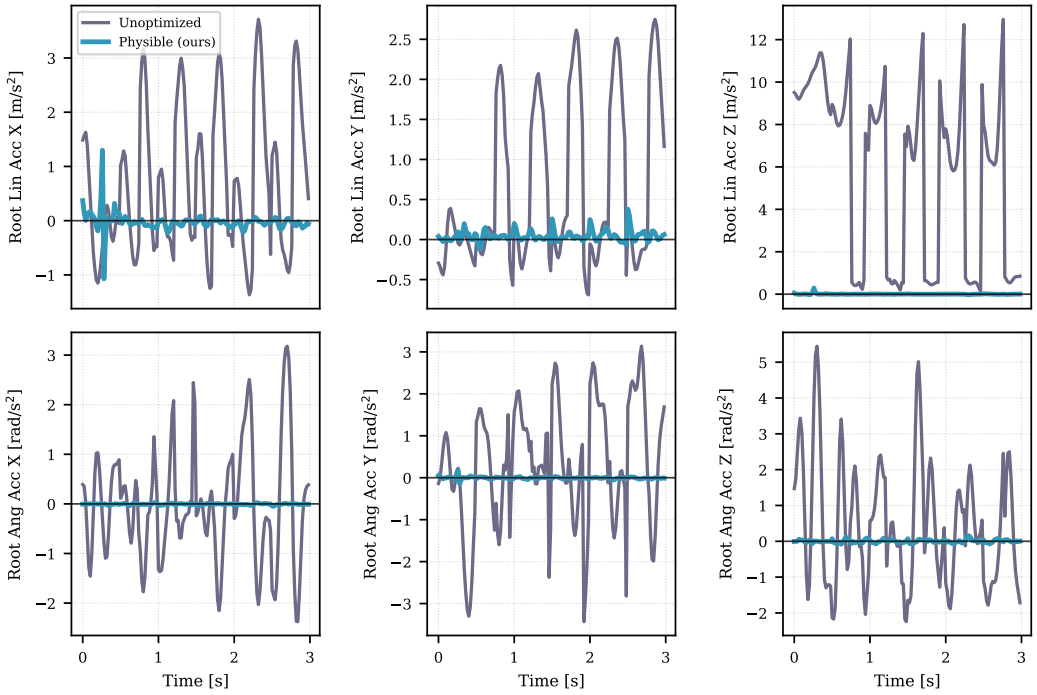


Figure 40: Behavior comparison of the *root acceleration residual* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

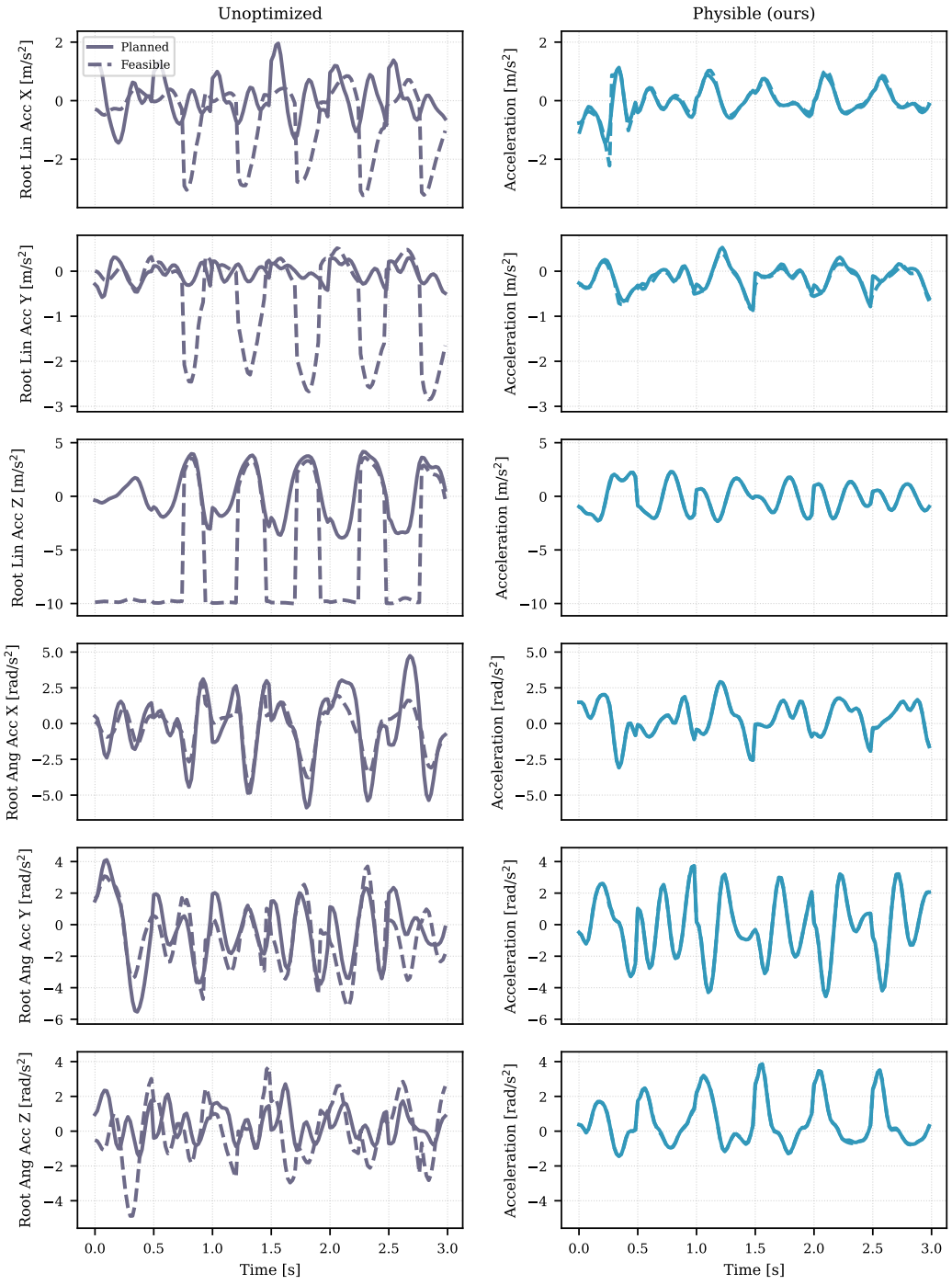


Figure 41: Behavior comparison of the *root acceleration planned vs. feasible* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

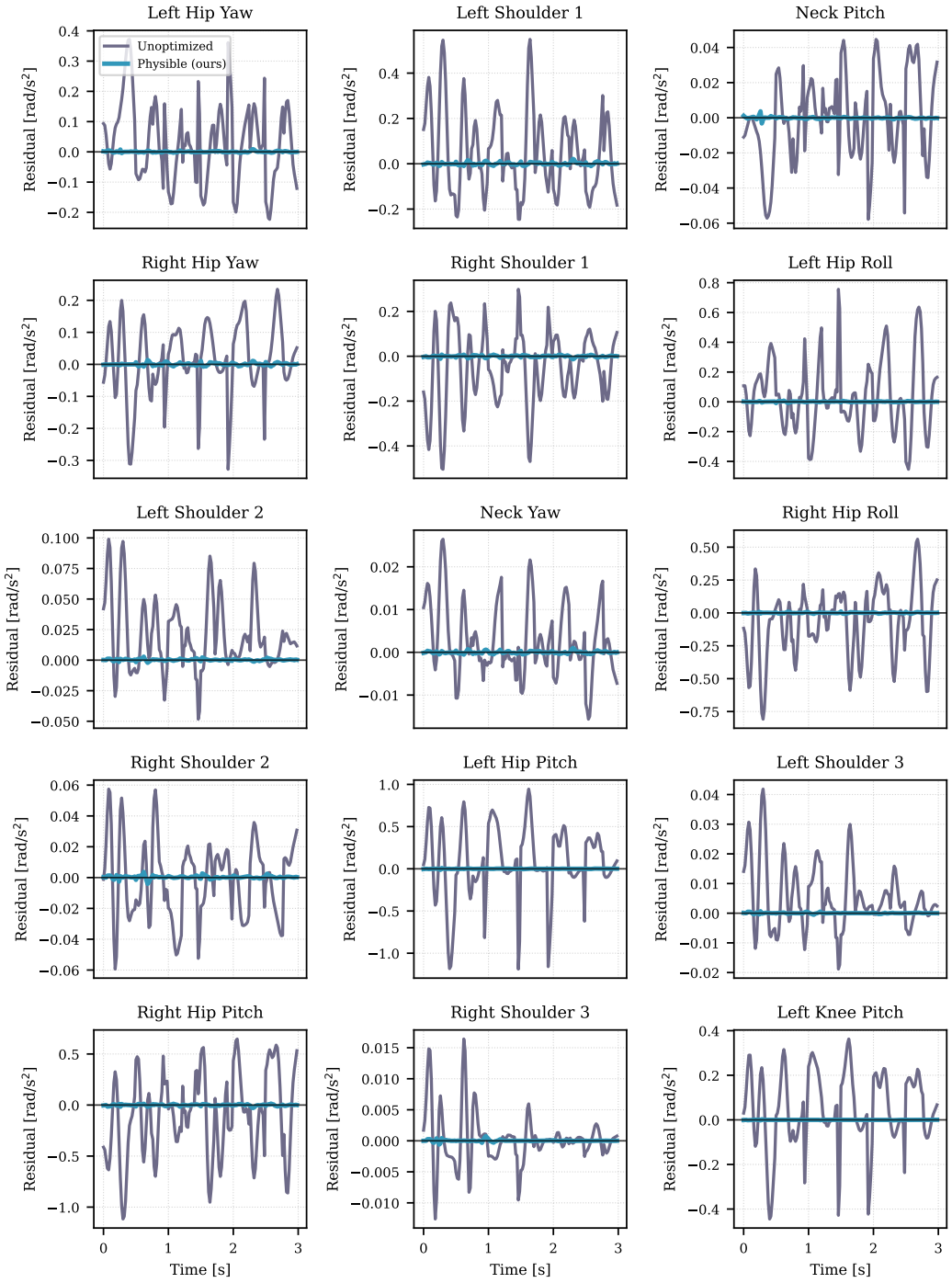


Figure 42: Behavior comparison of the *joint acceleration residual* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

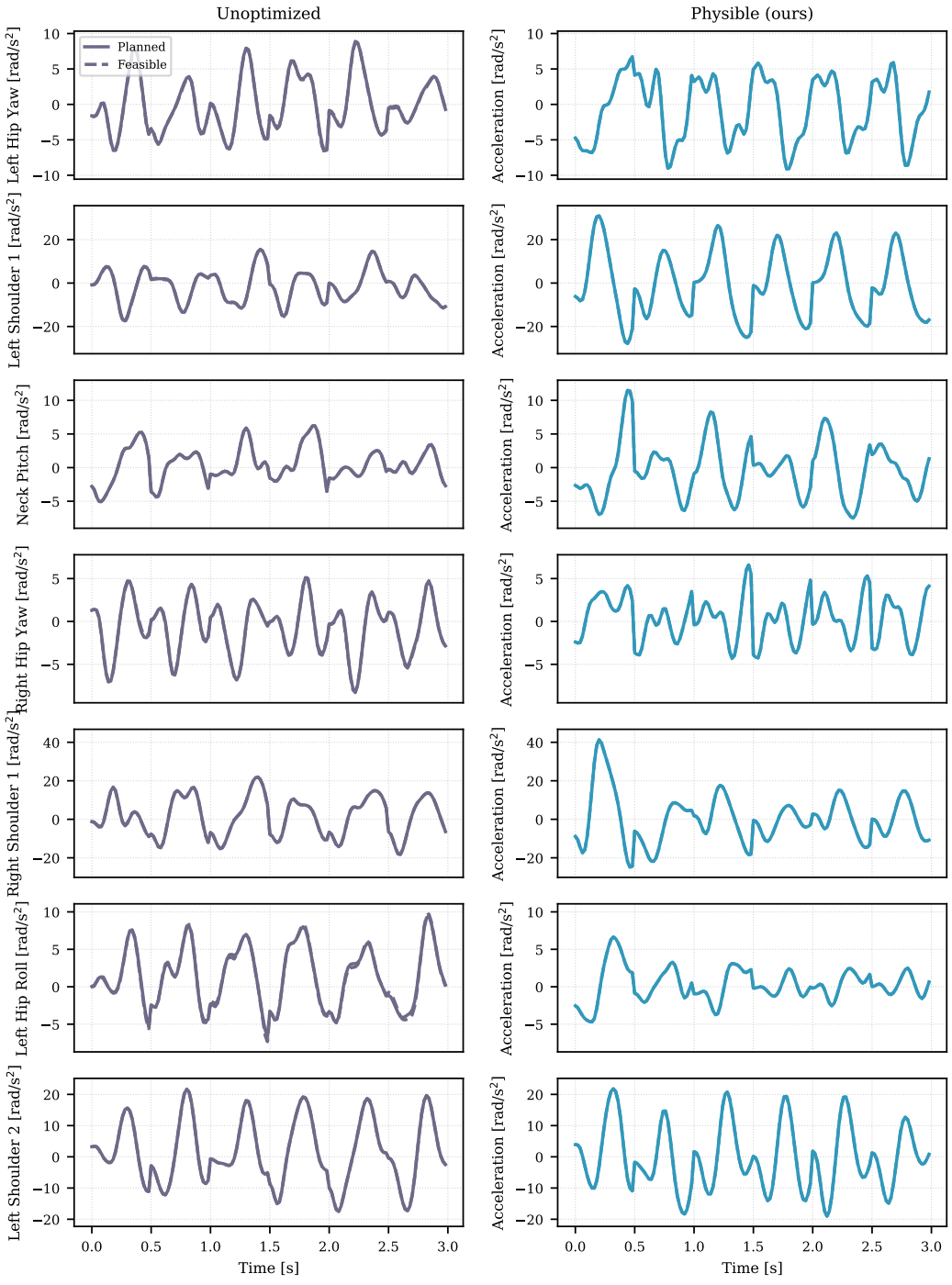


Figure 43: Behavior comparison of the *joint acceleration planned vs. feasible* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot - Part 1.

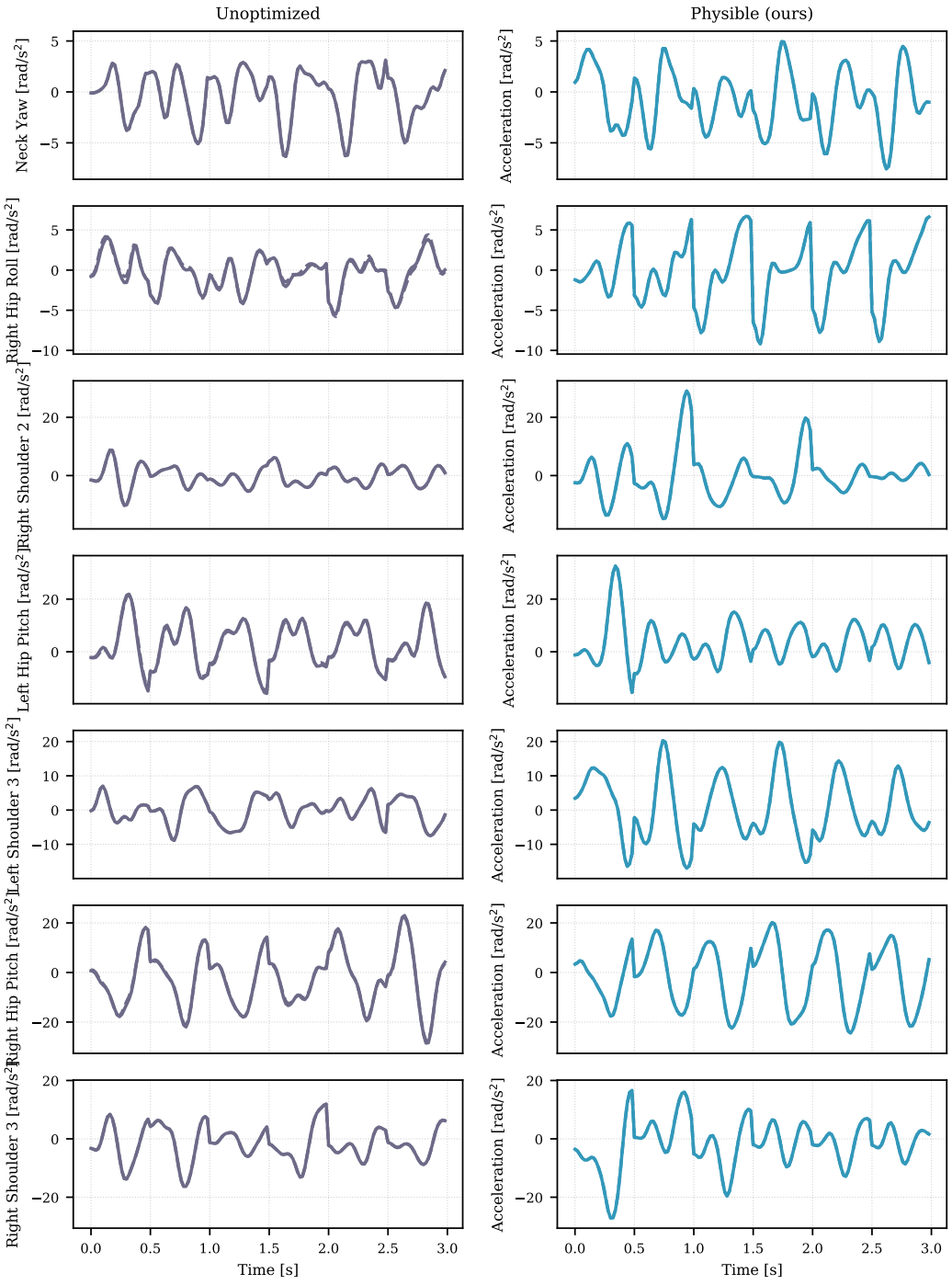


Figure 44: Behavior comparison of the *joint acceleration planned vs. feasible* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot - Part 2.

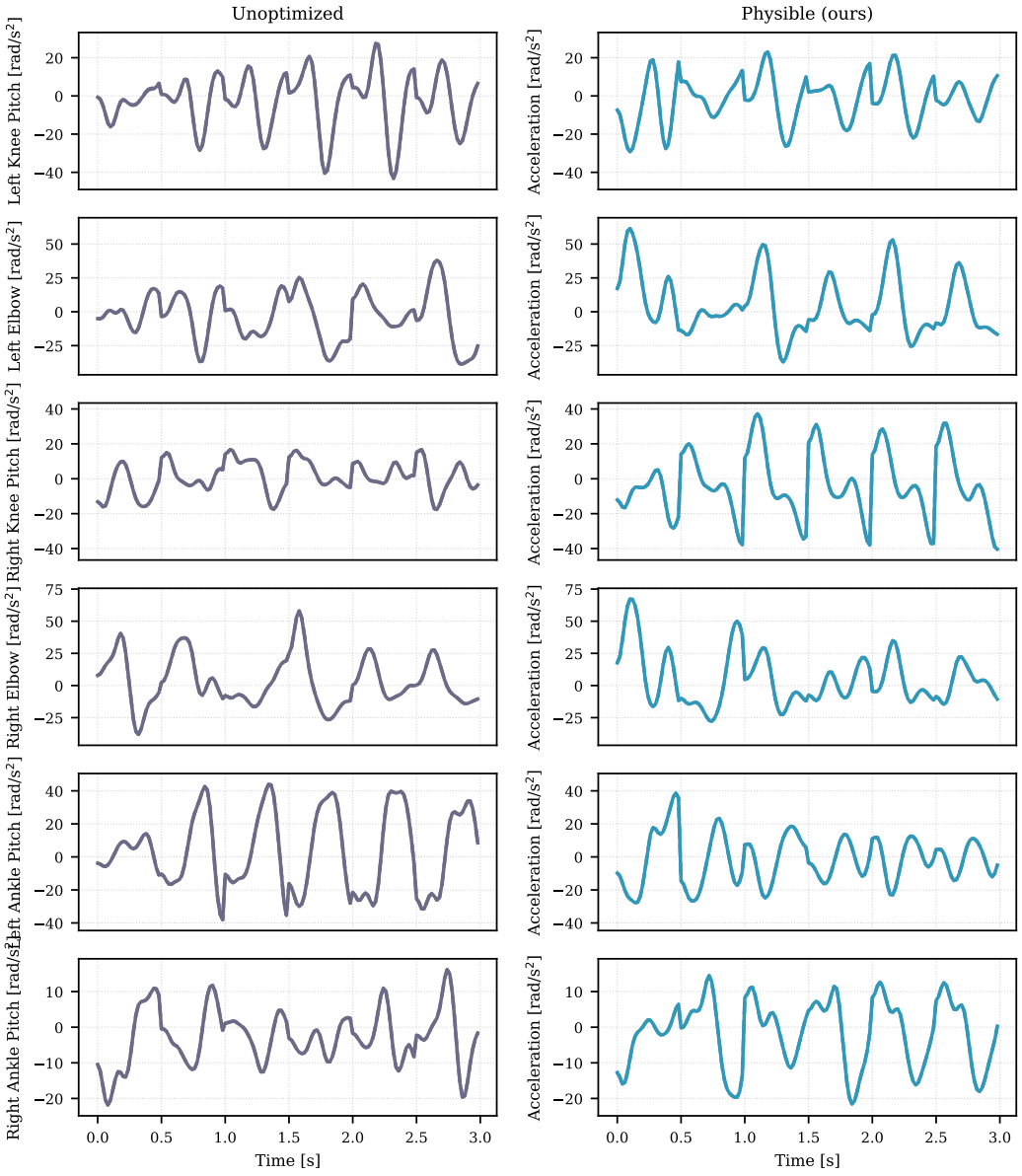


Figure 45: Behavior comparison of the *joint acceleration planned vs. feasible* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot - Part 3.

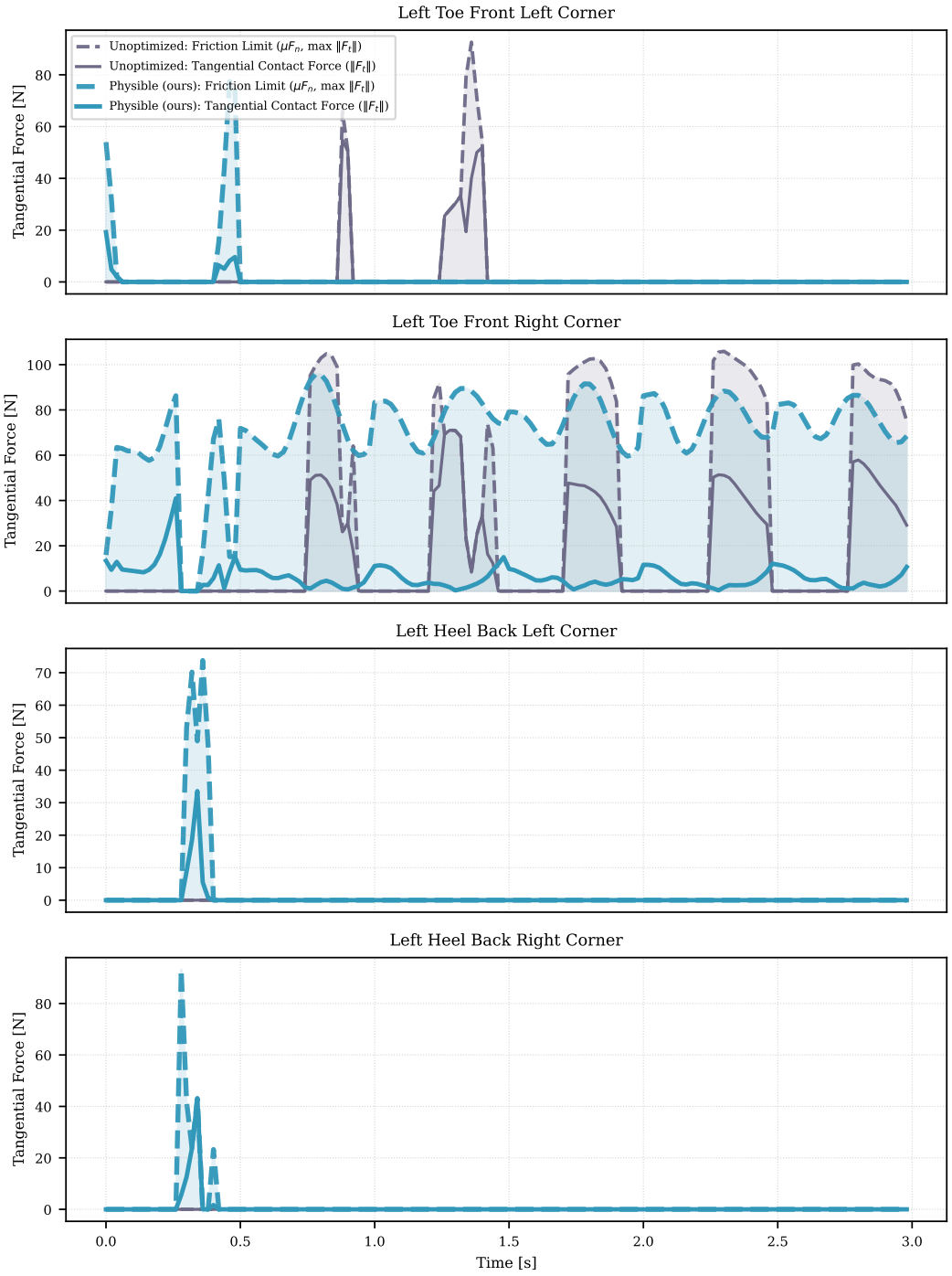


Figure 46: Behavior comparison of the *tangential ground contact forces and the friction cone* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

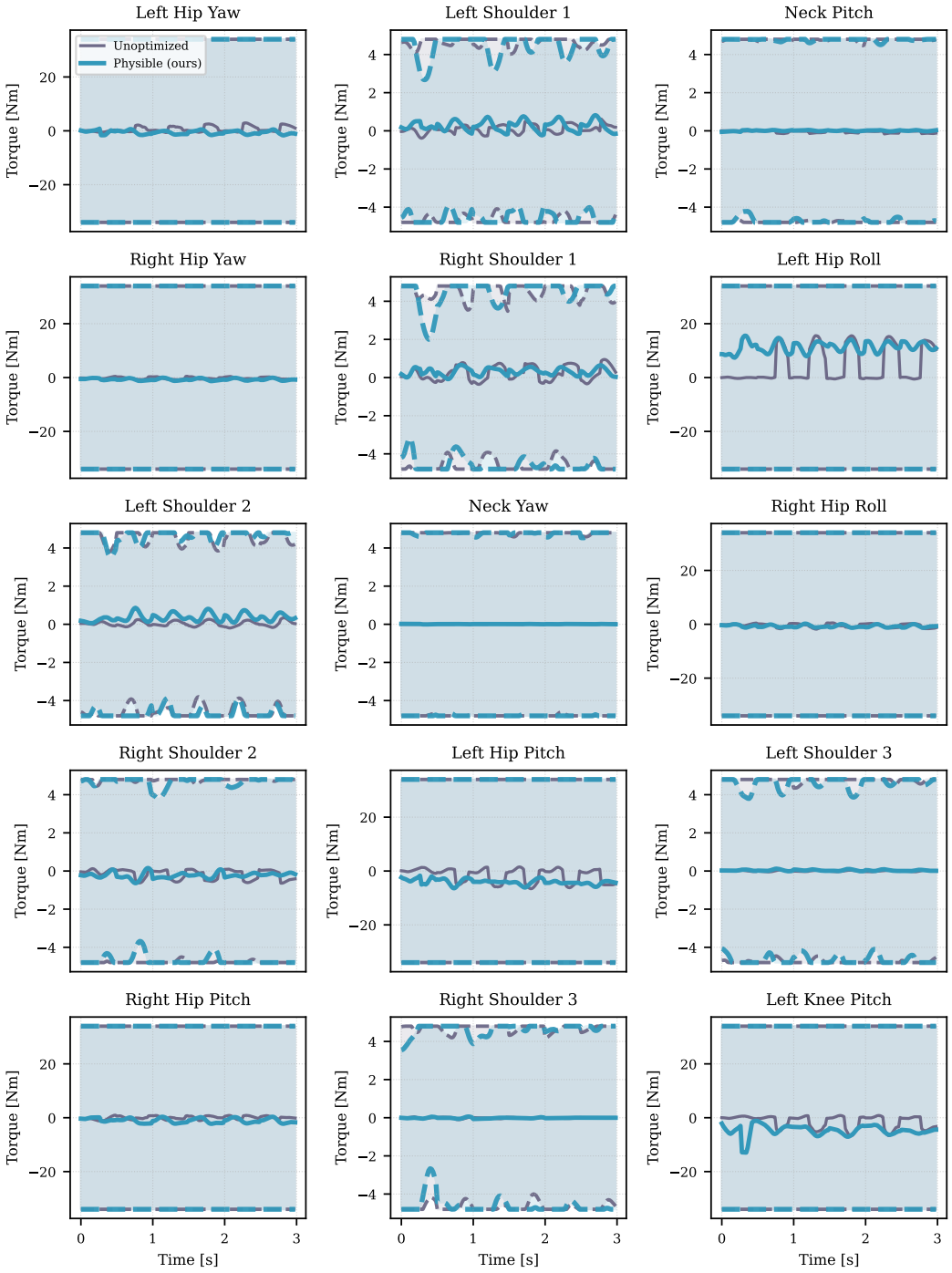


Figure 47: Behavior comparison of the *joint torques (including limits)* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot - Part 1.

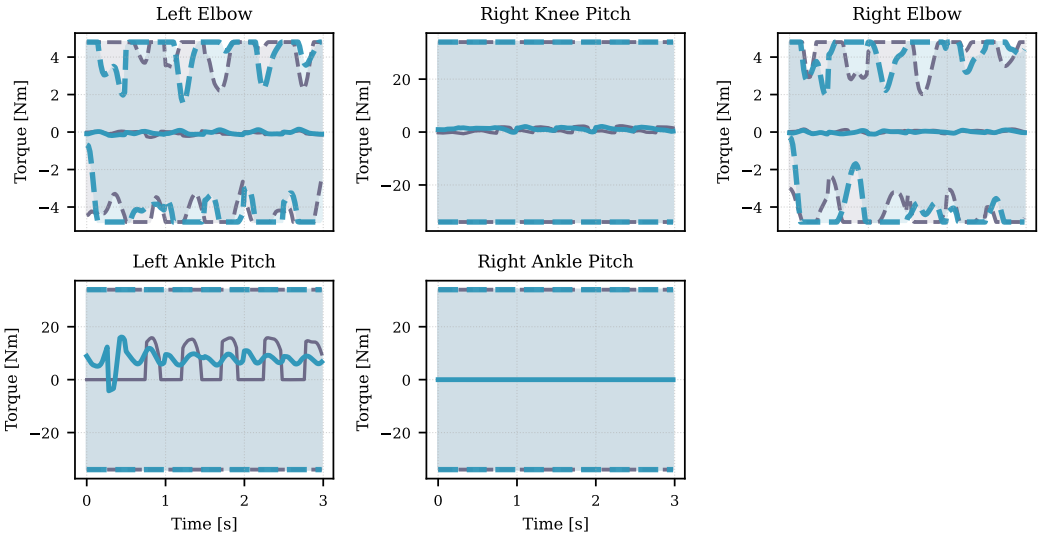


Figure 48: Behavior comparison of the *joint torques (including limits)* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot - Part 2.

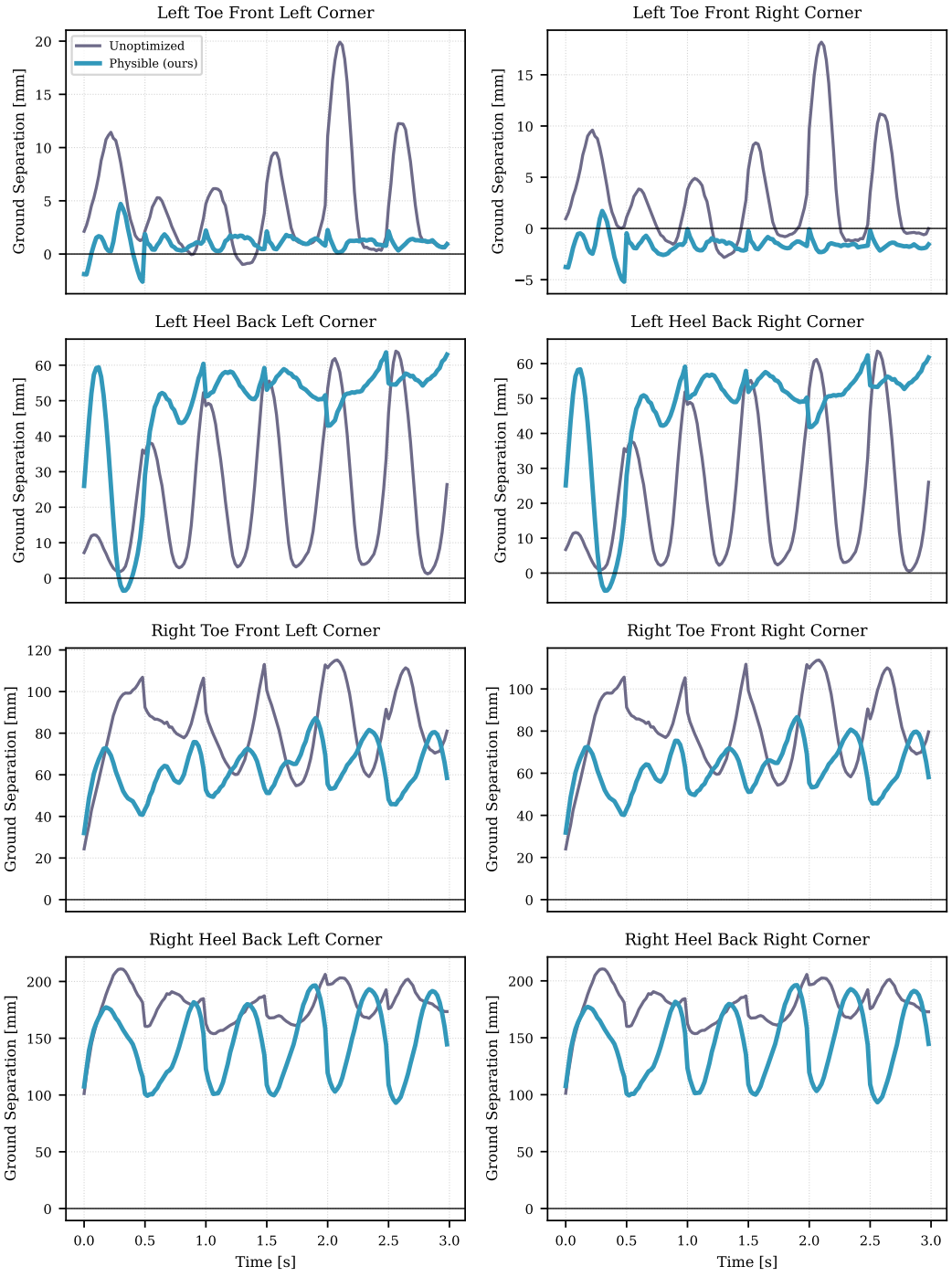


Figure 49: Behavior comparison of the *ground separation* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

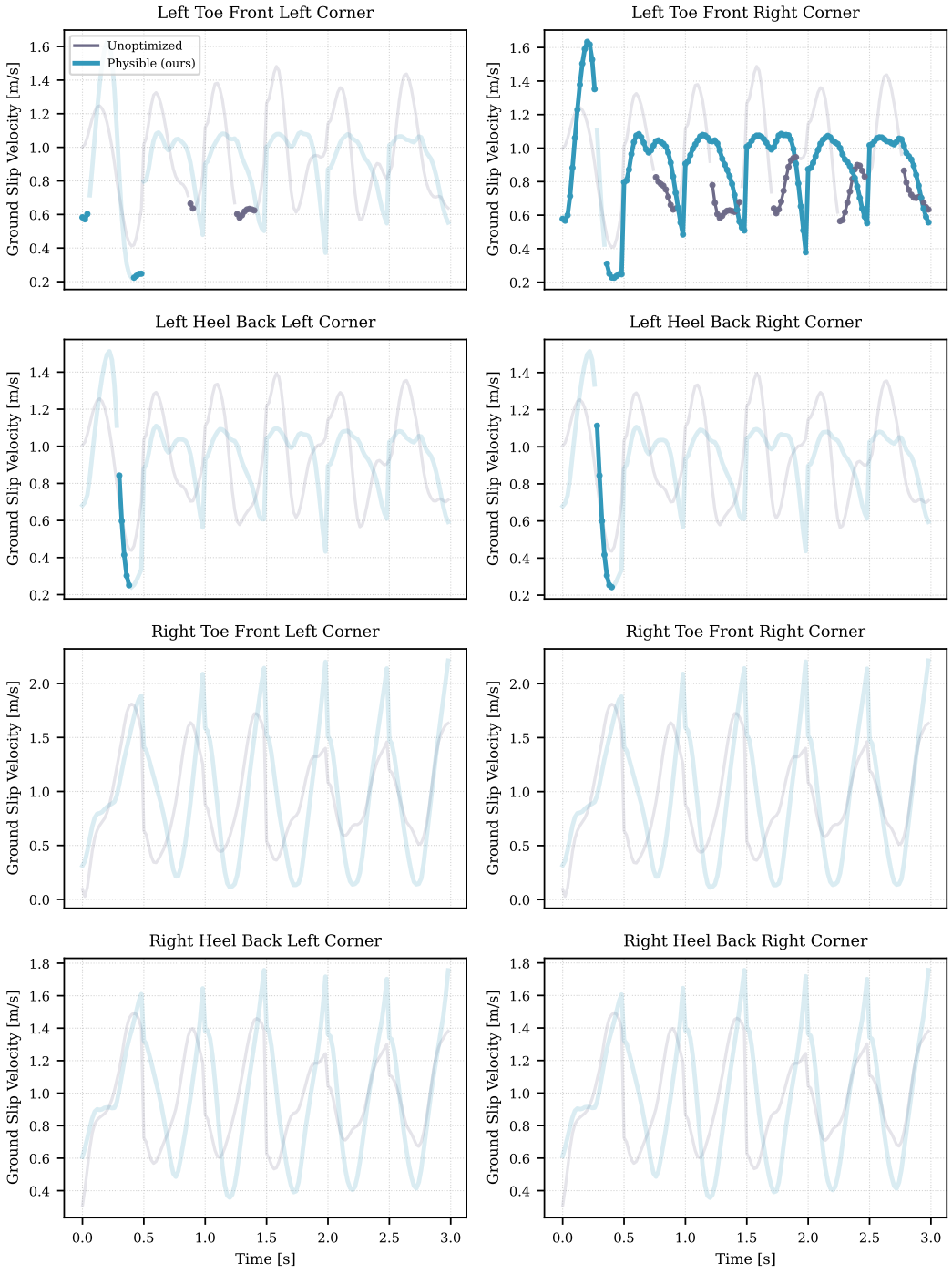


Figure 50: Behavior comparison of the *ground slip velocity* for the dynamic feasibility optimization for 150 time steps of the *left-hop* walking style on the bipedal robot.

H.2.3 Uphill and Downhill Walking: Unoptimized vs. Physible

We compare unoptimized and Physible-optimized bipedal robot motions in the *neutral* style for uphill ($+25^\circ$) and downhill (-25°) scenarios in Fig. 51. Each sequence spans $3\,\mathrm{s}$ at $50\,\mathrm{Hz}$, with friction coefficient $\mu_{\mathrm{c},p} = 1.0$. The fixed command and experiment settings are given in Appendix F.2. The renderings and root pitch use a slope-aligned coordinate frame: x points along the direction of travel and z is normal to the surface, so the horizontal ground plane in the renderings represents the slope.

We evaluate the mismatch between planned and feasible reconstructed root accelerations using the Euclidean norms of the three-dimensional linear and angular residuals, $\|\Delta\mathbf{a}\|_2$ and $\|\Delta\boldsymbol{\alpha}\|_2$. Their root-mean-square values over each sequence decrease from 5.95 to $3.13\,\mathrm{m\,s^{-2}}$ for uphill motion (47%) and from 5.00 to $0.82\,\mathrm{m\,s^{-2}}$ for downhill motion (84%). The corresponding angular mismatch decreases from 3.01 to $0.32\,\mathrm{rad\,s^{-2}}$ (89%) and from 3.00 to $0.12\,\mathrm{rad\,s^{-2}}$ (96%), respectively.

Optimization produces opposite pitch adaptations in the two scenarios: root pitch shifts toward positive values uphill and negative values downhill. The planned and reconstructed accelerations agree more closely after optimization, especially downhill. The uphill sequence retains appreciable linear mismatch, visible in both the along-slope and slope-normal components. These examples therefore show improved dynamic consistency under the slope-specific gravity vectors.

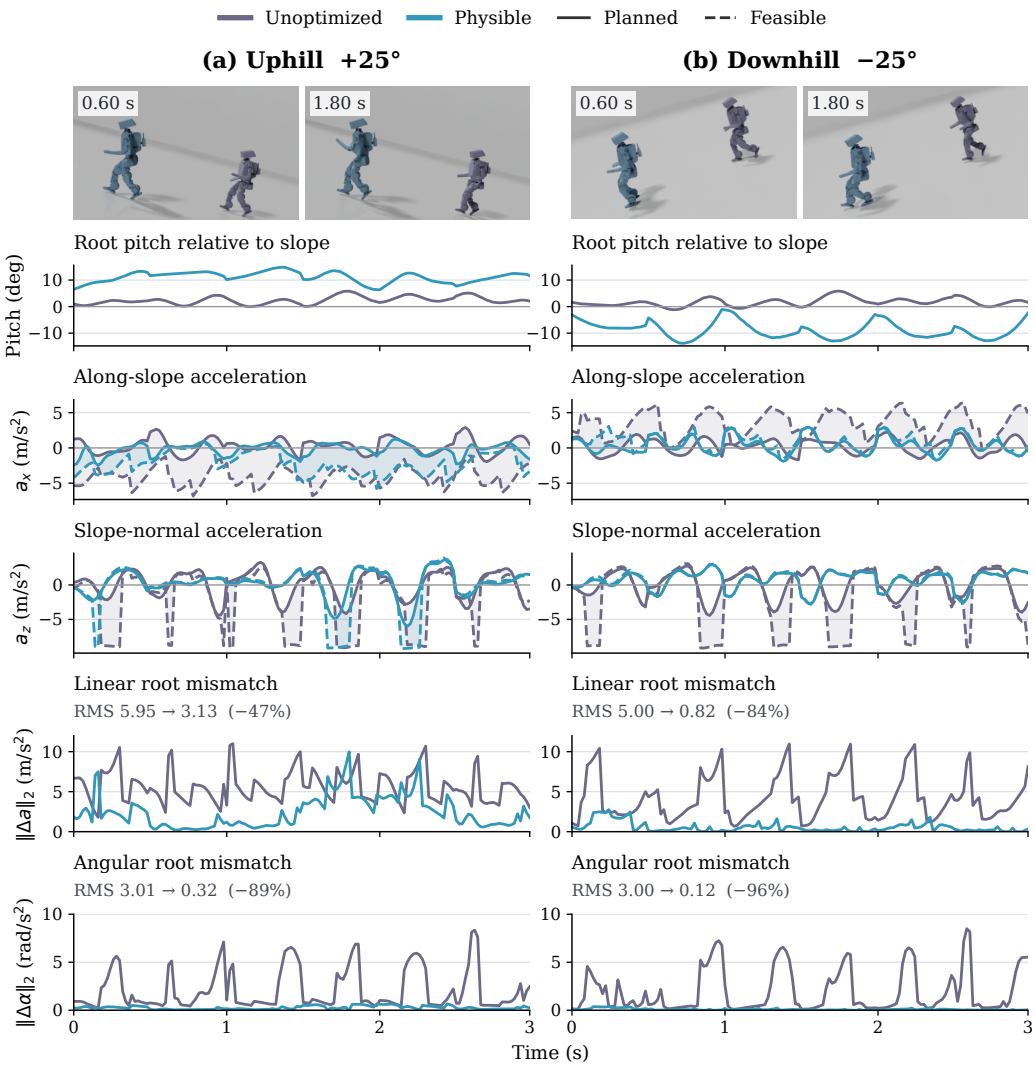


Figure 51: Dynamic feasibility optimization of the bipedal robot in the *neutral* style for (a) uphill ($+25°$) and (b) downhill ($-25°$) motion. Purple denotes unoptimized motion and blue denotes Physible. The snapshots show each pair at video timestamps 0.60 and $1.80\,\mathrm{s}$. All quantities use the slope-aligned frame with rotated gravity. From top to bottom, the plots show root pitch, along-slope and slope-normal accelerations (solid: planned; dashed: feasible reconstruction), and the norms of the linear and angular root acceleration residuals. Shading between acceleration curves highlights their mismatch. RMS annotations summarize each residual norm over the full sequence.

H.3 DOWNSTREAM RL TRACKING

We evaluate downstream tracking of the unoptimized and Physible-optimized bipedal two-foot-jump motions using the protocol in Appendix F.7. Relative to the policy trained on the unoptimized motion, the policy trained on the optimized motion reduces body position, body orientation, and joint position tracking error by 16.4%, 11.9%, and 7.4%, respectively, indicating that the increased dynamical consistency makes the reference easier to track. In contrast, global root position and orientation error increase by 39.6% and 41.3%. We attribute this to the step-wise nature of our dynamic feasibility optimization: accelerations are corrected locally in time rather than through the resulting integrated trajectory, so small residual inconsistencies accumulate as drift in the global root motion, while the more local and repetitive joint and body motion is largely unaffected. Consistent with this interpretation, the RL reward tracks the motion-frame root pose together with root-relative body motion, but it does not explicitly penalize accumulated global root-pose drift. The tracking results therefore show that reducing local dynamic residuals can improve body and joint tracking without improving every aspect of the complete trajectory.

I Detailed Limitations and Future Work

Our evaluation is currently limited to bipedal robotic embodiments. Future work should study additional robot topologies and embodiments (e.g., manipulators, quadrupeds, and mobile manipulators).

During development, we informally observed consistent qualitative behavior across a small number of additional optimization seeds and configurations, without apparent convergence issues associated with local minima. However, these observations do not establish robustness across seeds; we leave a systematic study of the variability of optimization results to future work.

The method further relies on accurate knowledge of robot dynamics, inertial parameters, contact geometry, and contact characteristics (e.g., friction). In practice, these quantities are uncertain and time-varying, and perception errors can directly affect contact reasoning. A natural next step is to incorporate uncertainty into the optimization (e.g., robust/risk-sensitive feasibility) and to evaluate how performance degrades under dynamics/contact mismatch.

Moreover, we currently assume that all ground contacts are sticking. While this simplifies ID and feasibility evaluation, it rules out behaviors that deliberately leverage controlled slipping (e.g., moonwalk-like styles). Future work could therefore co-optimize contact-mode decisions jointly with the motion.

Relatedly, we infer the enforced contact pattern for the *contact pattern preservation* constraint via a majority vote over a batch of unbiased denoised motion sequences. Although effective in practice, this choice is restrictive, as it discourages substantial changes to the initial contact pattern during optimization. Future work should investigate alternatives that simultaneously mitigate ground penetration and "floating" artifacts in *kinematic feasibility* optimization without overly constraining contact-pattern changes.

More broadly, the proposed optimization does not guarantee that the injected/optimized noise remains within the diffusion model's training distribution. Therefore, a common failure mode of the optimization is a *style collapse* where objectives and constraints might be satisfied, but the optimized motion does not adhere to the commanded motion style anymore. Beyond noise normalization, future work should investigate additional distribution-preserving mechanisms to better keep optimized motions in-distribution while still improving feasibility.

Furthermore, the approach requires trading off multiple (sometimes competing) objectives and constraints, which can make it challenging to identify suitable weights in the fitness function. In addition, achieving strict satisfaction of physical constraints would require many iterations and can push the optimization into regions where the diffusion model drifts out-of-distribution, degrading motion style. Future work should therefore explore enforcing physical feasibility constraints earlier in the pipeline—including during motion retargeting and in the construction/selection of the training dataset—so that the unoptimized diffusion output is closer to physical feasibility and requires smaller inference-time corrections.

Building on the previous point, a particularly tricky issue is *ground penetration*. Physically feasible motion (e.g., non-floating behavior, low tangential *slip* velocities near the ground, and dynamically feasible accelerations) relies on active contacts for constraint enforcement. While, in principle, one would like *exactly zero* penetration—ensuring both contact and no interpenetration—this is in practice generally infeasible from an optimization standpoint. We therefore introduce a *contact-pattern preservation* constraint that permits a limited amount of penetration. However, the appropriate penetration threshold is not obvious: setting it too low can substantially increase problem difficulty and slow convergence. Future work should develop more synergistic approaches that better resolve this physical-feasibility "tension".

From a practical standpoint, repeated horizon-wide feasibility evaluations (including inner-loop solves of the ID) can be computationally expensive for long horizons or real-time use. Promising directions include warm-starting (possibly with a learned modification of the diffusion noise (Eyring et al., 2025)), multi-resolution optimization (Antonova et al., 2023), and amortized surrogates for (parts of) the feasibility evaluation to reduce wall-clock time without sacrificing reliability.