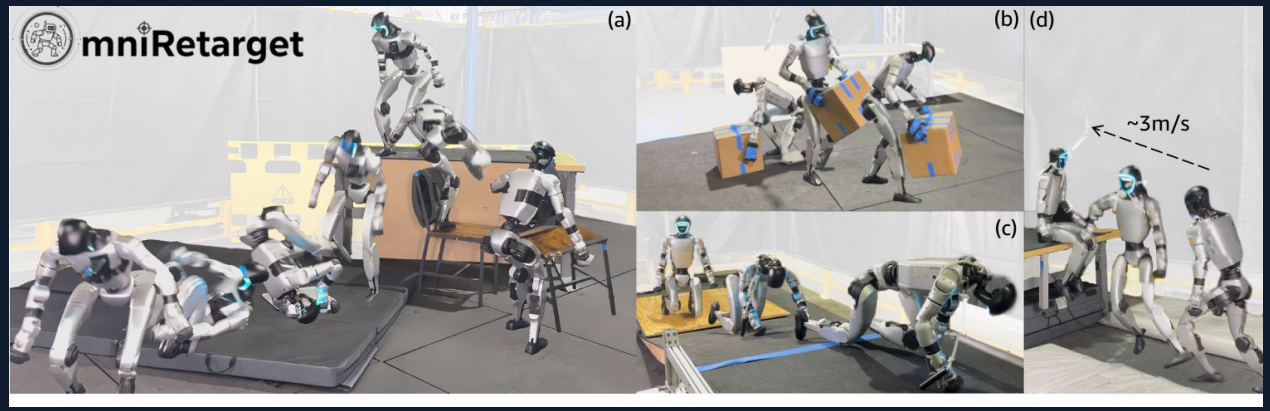
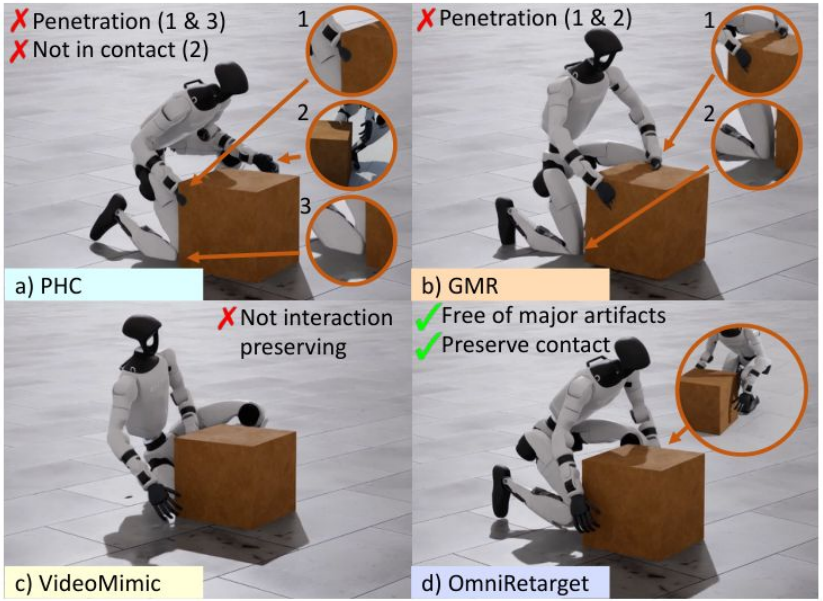
OmniRetarget

Interaction-Preserving Data Generation for Humanoid Whole-Body Loco-Manipulation and Scene Interaction

최원준



손 위치를 맞춰도 상자를 잡을 수 있을까?



신체 구조가 다르다

팔 길이와 몸의 비율이 달라

같은 좌표를 그대로 따라가기 어렵다

상호작용이 깨진다

손이 상자를 관통하거나,

잡아야 할 표면에서 떨어질 수 있다

주요 관절 위치의 유사성과 올바른 접촉은 서로 다른 평가 기준

Source: Fig. 7, Sec. V-B

Task: 사람의 상자 들기를 로봇 동작으로

사람의 팔 길이와 로봇의 팔 길이가 달라도, 손–상자 접촉을 유지해야 한다

사람의 시범
상자를 잡고 들어 올림

→

OmniRetarget
관계 보존 + 로봇 제약

→

Reference Motion
로봇의 기준 동작

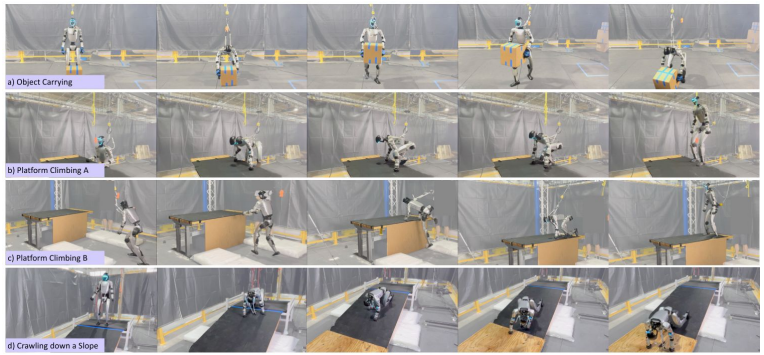
함께 필요한 정보

상자의 형상·위치 + 바닥·플랫폼 + 로봇 구조·관절 한계

Source: Fig. 2, Sec. III-A

OmniRetarget

사람의 시범을 기반으로 물체·지면과의 관계를 포함한 로봇 학습용 동작 데이터 만들기



상자 운반 · 플랫폼 오르기 · 경사면 기어가기

Source: Fig. 5, Sec. V-A

기존 방법과 무엇이 다른가?

Methods (방법)	Hard Kinematic Constraints (운동학적 필수 제약)	Interaction w/ Object (물체 상호작용)	Interaction w/ Terrain (지형 상호작용)	Data Augmentation (데이터 증강)
IMMA	Yes	No	No	No
PHC	No	No	No	No
GMR	No	No	No	No
VideoMimic	Soft Penalty (오차 점수에 반영)	No	Yes	No
OmniRetarget	**Yes**	**Yes**	**Yes**	**Yes**

상호작용과 Hard Constraints를 함께 다루고, 장면 변형까지 지원

Source: Table I, Appendix B

핵심 아이디어

관계 보존

Interaction Mesh
신체·물체·환경의 점들을 연결해
접촉의 상대 관계를 표현

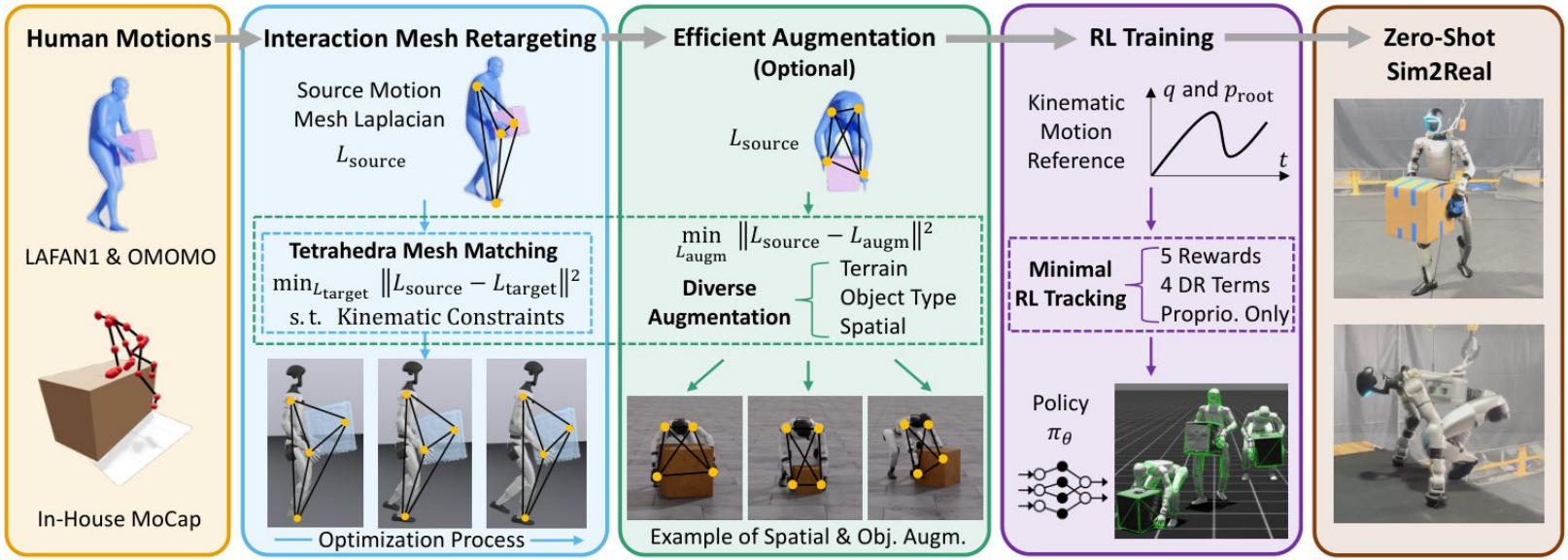
가능한 자세를 찾기

Hard Constraints
충돌·관절·속도 제한과
지지하는 발의 위치를 유지

보존할 관계를 정하고, 허용되는 로봇 자세 안에서 오차 줄이기

Source: Sec. III-A, Eqs. (1)–(3)

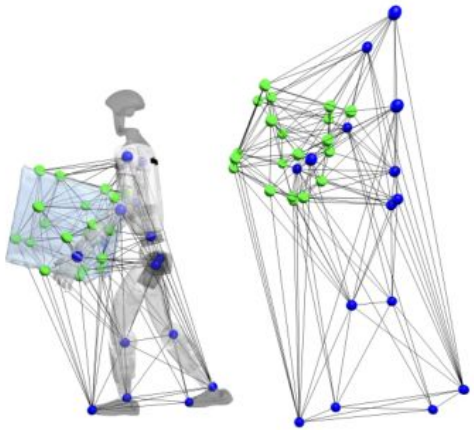
전체 Pipeline



Reference 생성과 augmentation을 먼저 수행하고, RL로 물리적 수행을 학습

Source: Fig. 2

Interaction Mesh



Vertices

신체 주요 점 + 물체·환경 표면 점

Edges

사면체화로 정한 이웃 관계

Interaction Mesh: 몸·물체·환경 사이의 관계 지도

Source: Fig. 9, Sec. III-A

Laplacian Coordinate

$$L(p_{t,i}) = p_{t,i} - \sum_{j \in \mathcal{N}(i)} w_{ij} \cdot p_{t,j},$$

01 점의 좌표

P = (3, 4)

A = (1, 0)

B = (5, 0)

02 평균 계산 → 03 차이 계산

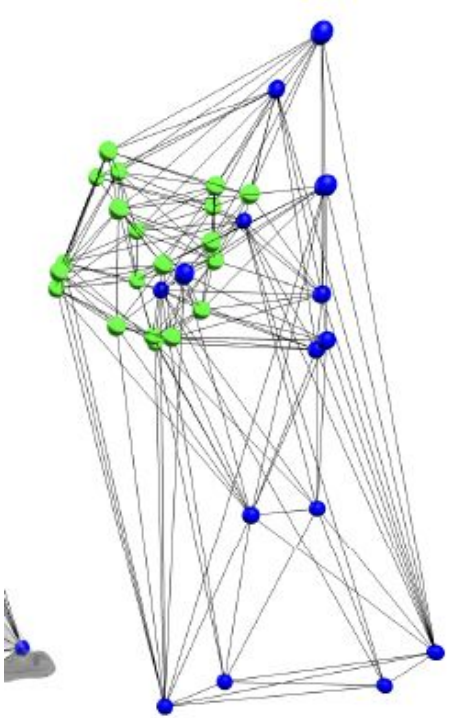
이웃 평균 = (3, 0)

L(P) = (3, 4) – (3, 0) = (0, 4)

이웃 평균에서 어느 방향으로 얼마나 떨어져 있는지를 나타내는 벡터

Source: Eq. (1) · 숫자는 설명용 예시

Delaunay Tetrahedralization



P의 이웃

N(P) = { A, B, C, D, E }

평균은 연결된 이웃 전체에 대해 계산

여러 사면체가 꼭짓점과
모서리를 공유하며 연결된다

Source: Sec. III-A · 아래 예시는 개념 설명

Delaunay와 Uniform Weights

Delaunay Tetrahedralization

3차원 점들을 여러 사면체로 연결해 각 점의 이웃을 정의

빈 외접구 조건

각 사면체의 외접구 내부에 다른 입력 점이 없는 연결

Uniform Weights

$$w_{ij} = 1/|\mathcal{N}(i)|$$

각 점의 이웃 평균을 만들기 위한 가중치값

Source: Sec. III-A · 가중치 선택의 우수성 비교는 제시하지 않음

Laplacian Deformation Energy

SOURCE

사람 + 물체 + 환경

이미 알려진 시범의 관계

TARGET

로봇 + 물체 + 환경

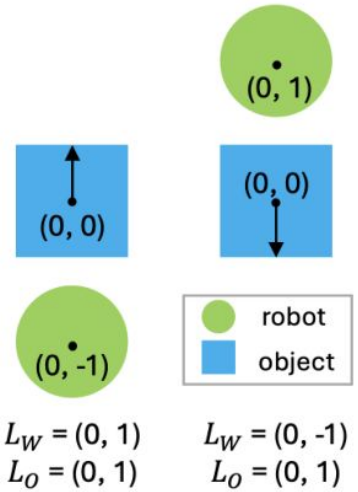
후보 자세 q에 따른 관계

$$E_L = \sum_{p_{t,i}^{\text{source}} \in \mathcal{P}_t^{\text{source}}, p_{t,i}^{\text{target}} \in \mathcal{P}_t^{\text{target}}} \| L(p_{t,i}^{\text{source}}) - L(p_{t,i}^{\text{target}}) \|^2 .$$

각 대응점의 관계 차이를 제곱 합

Source: Eq. (2) · Energy는 변형 오차 점수

Object Local Frame



World Frame (세계 좌표계)

상자가 회전하면

Laplacian 벡터도 회전한다

Object Frame (물체 좌표계)

상자에 붙어 함께 움직이는 축

같은 상대 관계는 그대로 표현

동일한 상호작용을 전역 회전 때문에 다른 관계로 평가하지 않도록 Local Object Frame 사용

Source: Fig. 8, Appendix C.2

Constrained Optimization

$$q_t^\star = \arg\min_{q_t} \sum_i \left\| L(p_{t,i}^{\text{source}}) - L(p_{t,i}^{\text{target}}(q_t)) \right\|^2 + \left\| q_t - q_{t-1} \right\|_Q^2$$

Objective (줄일 오차)

Source와 Target의 관계 차이

직전 프레임 대비 급격한 자세 변화

Hard Constraints

Collision Avoidance $\phi_j(q_t) \geq 0, \forall j$

Joint / Velocity Limits $q_{\min} \leq q_t \leq q_{\max}$

$v_{\min} \cdot dt \leq q_t - q_{t-1} \leq v_{\max} \cdot dt$

Foot Sticking $p_t^F = p_{t-1}^F, \forall \text{stance foot},$

운동학적 조건을 만족해도 동역학적 균형과 접촉력까지 보장하지 않음

Source: Eqs. (3a)–(3e)

Optimization: Input / Output

q(t)의 구성	Floating base의 위치·방향과 로봇의 모든 관절 각도
Forward Kinematics (순기구학)	후보 q(t)를 넣으면 각 Body Keypoint 위치 $f_i(q(t))$를 계산
Warm Start (이전 해에서 시작)	직전 프레임의 해에서 시작 · 반복 수정 후 현재 q(t)* 출력

Source: Sec. III-A, Appendix D · 첫 프레임 초기 추정은 논문에 구체적으로 명시되지 않음

Frame-by-Frame Optimization

현재 시범·장면과 직전 프레임의 로봇 자세를 입력

1 · 시작
이전 프레임의 자세

→

2 · FK
손·발 위치 계산

→

3 · 비교
Mesh 오차·제약

→

4 · 수정
관절·몸체 조정

2–4를 반복하며 개선 (최대 10회 반복)

현재 로봇 자세 q(t)* → 다음 프레임의 시작점

Source: Sec. III-A, Algorithm 5, Appendix D

Frame-by-Frame Optimization

현재 자세 주변에서 근사

Mesh 오차·충돌·발 위치를 근사
SOCP를 풀어 수정량 dq 계산

작게 수정하고 다시 계산

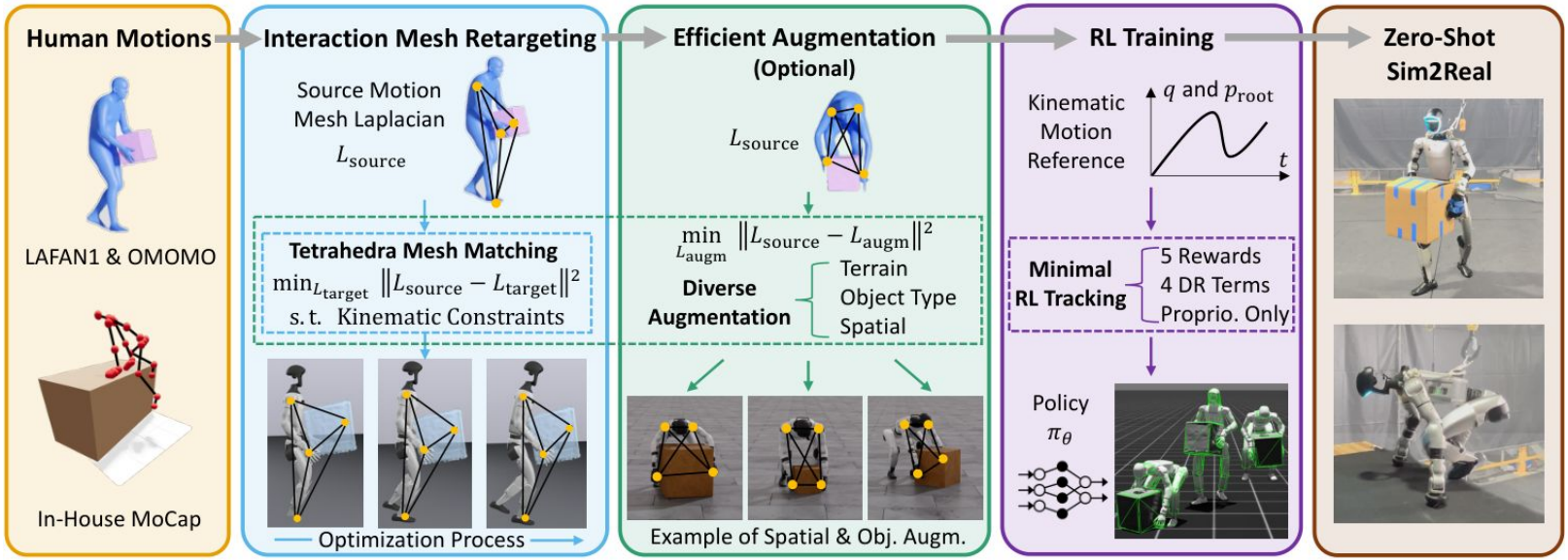
새 자세 = 현재 자세 + dq
수정한 자세에서 다시 근사

$$\bar{q}_{n+1} = \bar{q}_n + dq_n^{\star}.$$

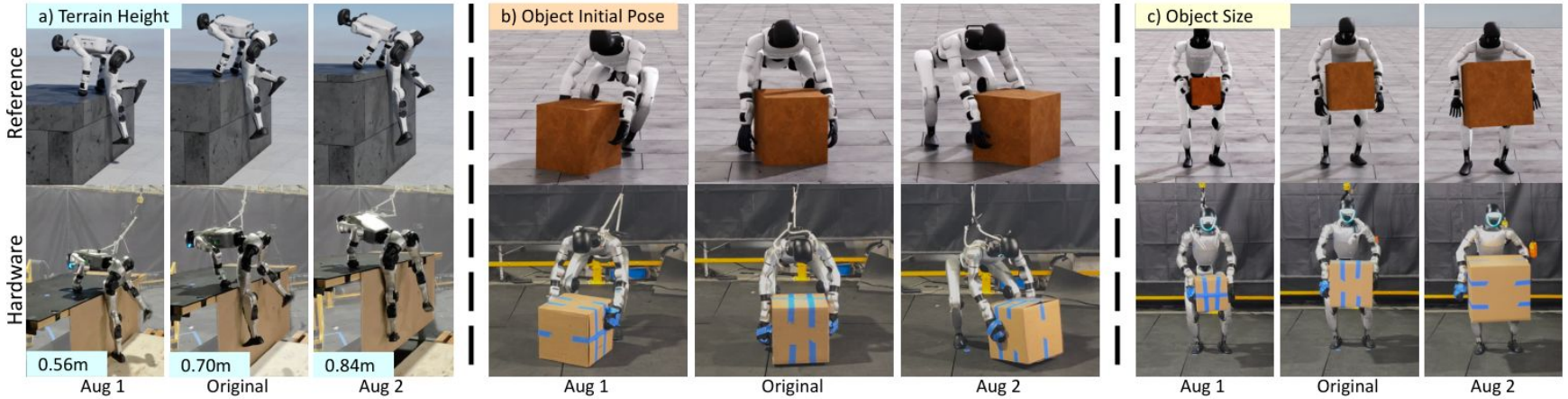
수정량 제한과 반복 종료

신뢰 영역: $\| dq \|_2 \leq 0.2$
수렴하면 종료 · 프레임당 최대 10회

Source: Appendix D, Eqs. (15a)–(15g)


Human Motions
LAFAN1 & OMOMO
In-House MoCap
Interaction Mesh Retargeting
Source Motion
Mesh Laplacian
L_{source}
Tetrahedra Mesh Matching
$\min_{L_{\text{target}}} \|L_{\text{source}} - L_{\text{target}}\|^2$
s. t. Kinematic Constraints
Optimization Process
Efficient Augmentation
(Optional)
L_{source}
$\min_{L_{\text{augm}}} \|L_{\text{source}} - L_{\text{augm}}\|^2$
Diverse Augmentation
Terrain
Object Type
Spatial
Example of Spatial & Obj. Augm.
RL Training
Kinematic Motion Reference
q and p_{root}
t
Minimal RL Tracking
5 Rewards
4 DR Terms
Proprio. Only
Policy
π_θ
Zero-Shot Sim2Real

Data Augmentation



물체 위치·크기와 플랫폼 높이를 바꾸고, 해당 조건의 Reference를 다시 최적화

Source: Fig. 4, Sec. III-B

바뀐 상자 위치에서 동작을 시작하려면?

동작 시작 전

상자 시작 위치·방향에 차이를 추가

움직이기 전에는 그 차이를 유지

동작 시작 후

추가한 위치·회전 차이를 점차 감소

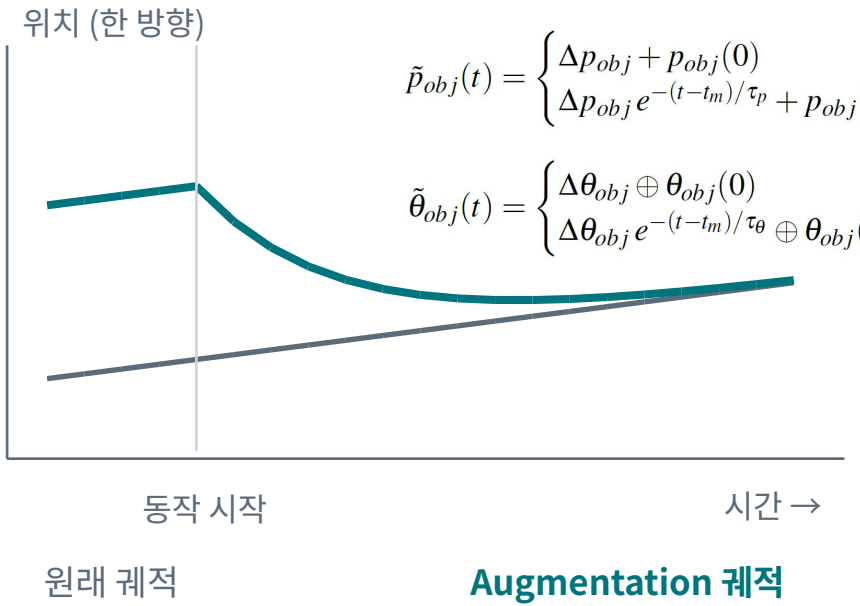
원래 궤적에 부드럽게 가까워짐

Reference 다시 생성

새 상자 궤적을 고정하고

로봇 자세를 다시 최적화

원래 궤적과 Augmentation 궤적



Source: Appendix C.1, Eqs. (14a)–(14b) · 아래 그림은 위치 차이의 개념 예시

Reference를 실제 움직임으로

Reference Motion

무엇을 따라 할 것인가?

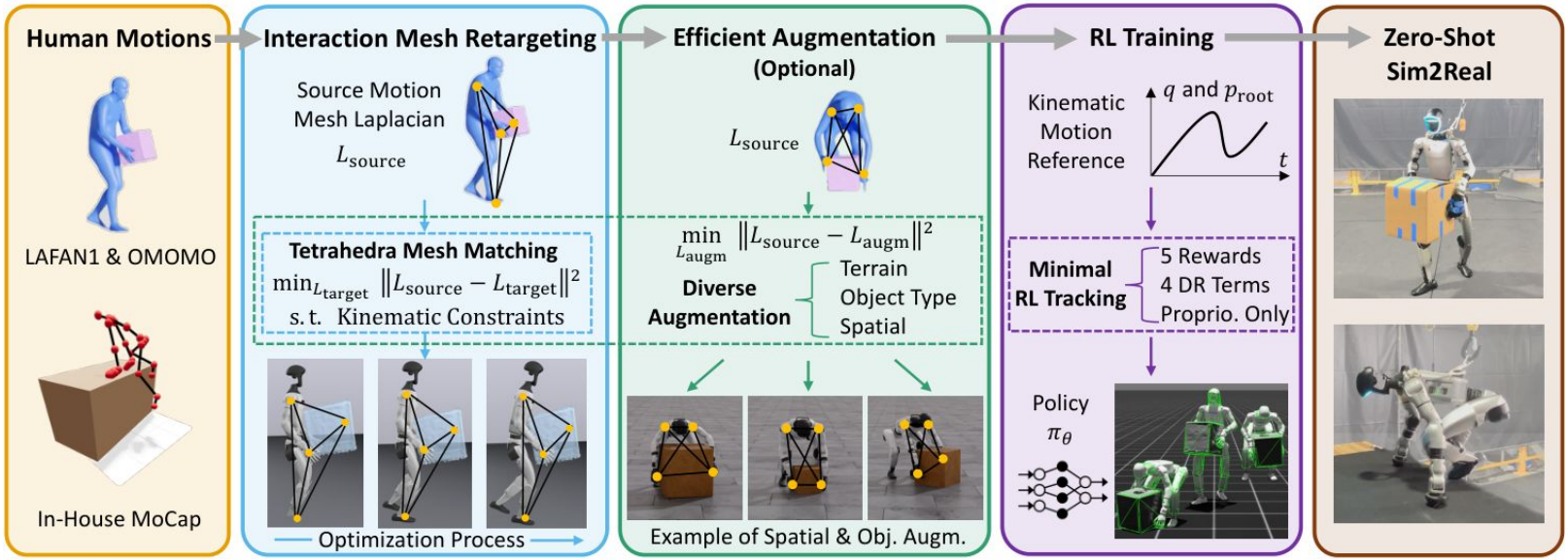
로봇 기준 자세의 연속
관절·접촉 조건을 고려

RL Policy

지금 어떻게 움직일 것인가?

현재 상태에서 action 결정
중력·마찰·관성 속에서 수행

Source: Sec. IV


Human Motions
LAFAN1 & OMOMO
In-House MoCap
Interaction Mesh Retargeting
Source Motion
Mesh Laplacian
L_{source}
Tetrahedra Mesh Matching
$\min_{L_{\text{target}}} \|L_{\text{source}} - L_{\text{target}}\|^2$
s. t. Kinematic Constraints
Optimization Process
Efficient Augmentation
(Optional)
L_{source}
$\min_{L_{\text{augm}}} \|L_{\text{source}} - L_{\text{augm}}\|^2$
Diverse Augmentation
Terrain
Object Type
Spatial
Example of Spatial & Obj. Augm.
RL Training
Kinematic Motion Reference
q and p_{root}
t
Minimal RL Tracking
5 Rewards
4 DR Terms
Proprio. Only
Policy
π_θ
Zero-Shot Sim2Real

Policy와 Simulator

학습 중: 명령을 내리고, 물리적 결과를 경험하며 Policy를 수정한다

입력: Reference + 현재 로봇 상태 + 이전 action

Policy
행동을 결정

→

Simulator
물리적 결과 계산

→

Reward
기준 수행을 평가

RL 알고리즘이 reward를 이용해 Policy를 업데이트

Source: Sec. IV · 역할을 설명하는 개념 흐름

Policy는 어떤 정보를 받는가?

Reference Motion

기준 관절 위치·속도
기준 대비 골반 위치·방향 오차

Proprioception (자기 상태)

골반 선속도·각속도
현재 관절 위치·속도

Previous Action

직전 행동을 함께 입력
급격한 명령 변화를 줄이는 학습에 활용

Source: Sec. IV, OmniRetarget · arXiv:2509.26633v3

물체를 직접 보지 않는 Policy

REFERENCE

생성 단계에서 물체·지형을 고려했으므로 필요한 동작에 그 정보가 반영

REWARD

Simulator는 물체 상태를 알아 body / object tracking을 평가 가능

LIMIT

같은 입력으로 구분할 수 없는 임의의 물체 위치 변화에는 대응이 어려움

실제 실행에서는 학습한 RM을 사용

Source: Sec. IV · 관측과 reward 정보는 서로 다름

5가지의 Reward

1–2 · Tracking	Body: 몸의 위치·방향·선속도·각속도 Object: 물체 위치·방향 (해당 과제)
3–4 · Regularization	Action Rate: 급격한 행동 변화에 벌점 Soft Joint Limit: 관절 한계 위반에 벌점
5 · Self-Collision	로봇의 몸 부위끼리 충돌하지 않는가 BeyondMimic의 가중치·설정을 그대로 사용

Source: Sec. IV, OmniRetarget · arXiv:2509.26633v3

학습 종료 조건과 Policy 구성

Tracking 실패 시 종료

몸이 기준 동작에서 크게 벗어나면 종료
물체 오차: 거리 1.0 m 초과 / 방향 45° 초과

동작별 Policy 구성

상자 이동 동작은 하나의 Multi-task Policy
플랫폼 오르기는 Reference별 Policy

학습에서 실제 로봇으로

시뮬레이션에서 Policy를 학습
추가 실제 학습 없이 Unitree G1에 적용

Source: Sec. IV, OmniRetarget · arXiv:2509.26633v3

Domain Randomization

Aspect	Data Augmentation	Domain Randomization
Parameters	상자 초기 위치·방향·크기 플랫폼 높이	질량·무게중심·관성 형상·센서 노이즈 등
Procedure	조건에 맞는 Reference 다시 최적화	물리 조건을 바꾸며 Policy 학습
Box Example	**오른쪽 상자를 잡는 새 기준 동작 생성**	**같은 기준 동작을 더 무거운 상자로 연습**

물체 DR: 질량 0.1–2 kg · COM ±0.08 m · 관성 50–150% · 형상 ±10%

Source: Sec. IV · 물체 초기 위치·방향의 DR은 명시되지 않음

Evaluation

01 Kinematic Quality (동작 품질)

Penetration ↓
관통 시간·깊이
Foot Skating ↓
미끄러짐 시간·속도

02 Contact Preservation (접촉 유지)

의도한 접촉을
유지한 시간 비율 ↑

03 RL Success (학습 성공률)

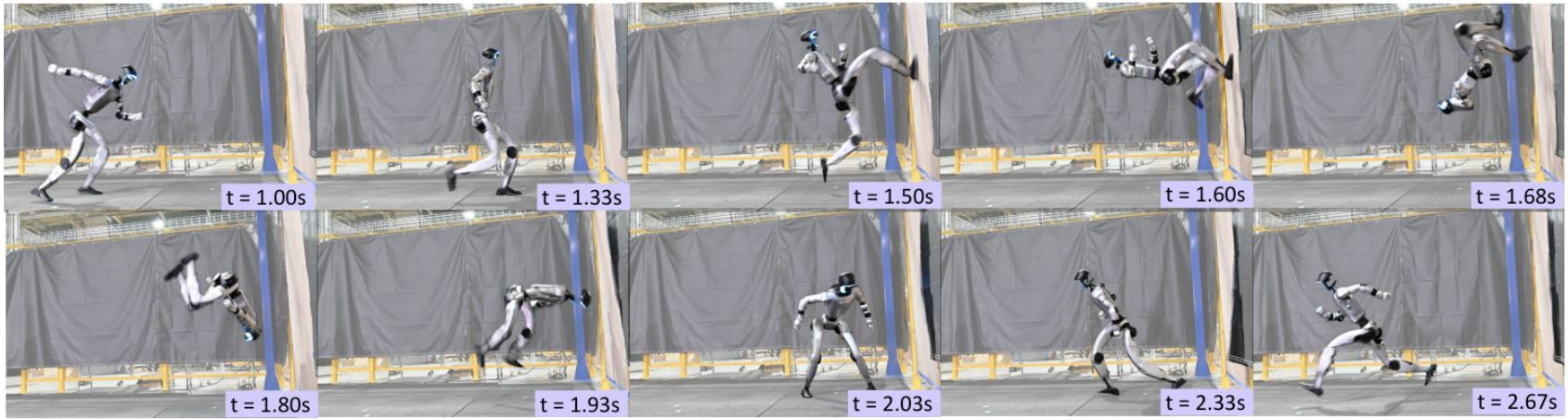
39개 어려운 동작
같은 학습 설정

시뮬레이션 성공률 ↑

접촉 유지 점수만 높아도 관통이 크면 좋은 Reference라고 보기 어렵다

Source: Sec. V-B, Table II

Hardware / Data Augmentation



Wall-flip: 실제 실험 5/5 성공

Augmentation 데이터 시뮬레이션 성공률 79.1% · nominal 평가 82.2%

Source: Fig. 6, Sec. V-A

Robot–Object: Kinematic Quality

Method	Penetration (관통) Duration ↓ (시간 비율)	Penetration (관통) Max Depth ↓ (최대 깊이, cm)	Foot Skating (발 미끄러짐) Duration ↓ (시간 비율)	Foot Skating (발 미끄러짐) Max Vel. ↓ (최대 속도, cm/s)	Contact Preservation (접촉 유지) Duration ↑ (시간 비율)
PHC	0.68	5.11	0.05	1.40	0.96
GMR	0.83	8.50	0.02	1.46	0.99
VideoMimic	0.60	7.48	0.12	1.50	0.77
OmniRetarget	**0.00**	**1.34**	**0**	**0**	**0.96**

GMR의 접촉 유지 0.99도 큰 관통 깊이 8.50 cm와 함께 봐야 한다

Source: Table II, OMOMO · 평균 표시, 표준편차는 발표자 노트

Robot–Terrain: Contact Preservation

Method (방법)	Penetration (관통) Duration ↓ (시간 비율)	Penetration (관통) Max Depth ↓ (최대 깊이, cm)	Foot Skating (발 미끄러짐) Duration ↓ (시간 비율)	Foot Skating (발 미끄러짐) Max Vel. ↓ (최대 속도, cm/s)	Contact Preservation (접촉 유지) Duration ↑ (시간 비율)
PHC	0.66	7.74	0.15	2.03	0.45
GMR	0.91	5.72	0.04	1.75	0.67
VideoMimic	0.83	5.97	0.14	1.85	0.47
OmniRetarget	**0.01**	**1.37**	**0**	**0**	**0.72**

손·발과 지형의 접촉을 더 잘 유지하면서 관통과 발 미끄러짐을 줄인다

Source: Table II, In-House MoCap · 평균 표시, 시간비율은 논문 정의로 정규화

물체 없는 동작에서는 두 방법 모두 성공

Method (방법)	Penetration (관통) Duration ↓ (시간 비율)	Penetration (관통) Max Depth ↓ (최대 깊이, cm)	Foot Skating (발 미끄러짐) Duration ↓ (시간 비율)	Foot Skating (발 미끄러짐) Max Vel. ↓ (최대 속도, cm/s)	RL Success Rate ↑ (강화학습 성공률)
Unitree	0.09	3.22	0.06	1.46	100%
OmniRetarget	**0.00**	**1.07**	**0**	**0**	**100%**

운동학적 품질 개선됨!

Source: Table II, LAFAN1 · 평균 표시

더 좋은 Reference가 RL 수행을 돕는다

82.20%

상자 상호작용 · 시뮬레이션

94.73%

지형 상호작용 · 시뮬레이션

Method (방법)	Robot–Object (로봇–물체)	Robot–Terrain (로봇–지형)
PHC	71.28%	52.63%
GMR	50.83%	78.94%
VideoMimic	3.85%	51.75%
OmniRetarget	**82.20%**	**94.73%**

Source: Table II · 시뮬레이션 성공률 평균, 분산값은 발표자 노트

평균 성공률만 보면 놓치는 것

Robot–Object

물체 특성 변화에 얼마나 강한가?

OmniRetarget은 높은 성공률에 집중

Robot–Terrain

한 번도 성공하지 못한 동작

GMR 4개 · OmniRetarget 1개

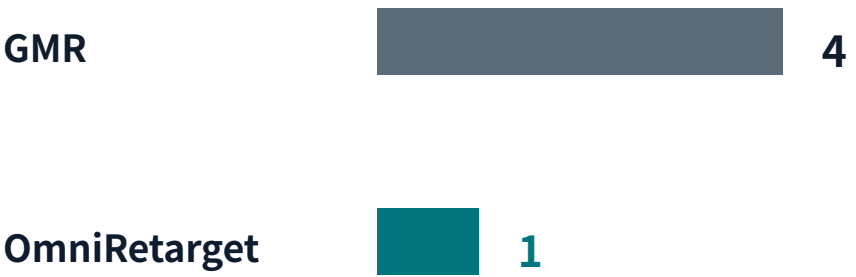
평균과 동작별 결과 함께 보기

PHC·VideoMimic: 거의 절반의

지형 동작에서 한 번도 성공하지 못함

완전 실패한 지형 동작 수

성공률 0%인 동작 · 적을수록 좋음



Appendix E 본문의 수치를 요약

Source: Appendix E, Fig. 10 · nominal motions, 시뮬레이션 평가

Limitations

01	**Computational Cost**	Mesh와 반복 최적화의 비용을 baseline과 정량 비교하지 않음
02	**Kinematics vs. Dynamics**	작은 관통이 남을 수 있고, 안정적인 수행에는 이후 RL이 필요
03	**Scene Observation**	기본 Policy는 물체·지형을 직접 관측하지 않아 임의 장면 적응에 한계

Source: Sec. V–VI · 계산 비용은 발표자의 비판적 검토

Conclusion / Q&A

상호작용을 보존하는
로봇 학습용 동작 데이터 생성

01 Interaction Mesh로 관계 표현

02 Optimization + Data Augmentation으로 Reference 생성

03 RL로 물리 환경에서의 수행 학습

Source: Sec. VI