2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

LEGO: Latent-space Exploration for Geometry-aware Optimization of Humanoid Kinematic Design

Jihwan Yoon[1], Taemoon Jeong[1], Jeongeun Park[1], Chanwoo Kim[1],
Jaewoon Kwon[2], Yonghyeon Lee[3], Kyungjae Lee[4], and Sungjoon Choi[1*]

***Abstract*— Designing robot morphologies and kinematics has traditionally relied on human intuition, with little systematic foundation. Motion–design co-optimization offers a promising path toward automation, but two major challenges remain: (i) the vast, unstructured design space and (ii) the difficulty of constructing task-specific loss functions. We propose a new paradigm that minimizes human involvement by (i) learning the design search space from existing mechanical designs, rather than hand-crafting it, and (ii) defining the loss directly from human motion data via motion retargeting and Procrustes analysis. Using screw-theory-based joint axis representation and isometric manifold learning, we construct a compact, geometry-preserving latent space of humanoid upper body designs in which optimization is tractable. We then solve design optimization in this latent space using gradient-free optimization. Our approach establishes a principled framework for data-driven robot design and demonstrates that leveraging existing designs and human motion can effectively guide the automated discovery of novel robot design. Project page: `https://jihwan-yoon-page.github.io/legopt/`**

I. Introduction

In robot mechanical design, determining the morphology as well as the lengths of links and axes of joints is typically the very first step. In practice, however, this process often depends heavily on human intuition and lacks a principled, systematic foundation. To move toward design automation, a promising approach is motion–design co-optimization [1], which, given a set of target motions, seeks designs that are inherently suited to executing those motions – a direction to which this paper contributes.

The major difficulty lies in the vast, combinatorial, and unstructured nature of the design space. To make optimization tractable, designers must first restrict the search space – for example, by exploiting task-specific heuristics, symmetry assumptions, or low-dimensional parametric families – before meaningful optimization can be performed [1]. Moreover, each new task often introduces unique constraints and parameters, necessitating repeated and costly redefinition of the design space.

Another challenge lies in designing the loss (or reward) function for a given task. For example, given target robot behaviors (e.g., a specific style of dancing), defining an appropriate loss or reward is nontrivial. Even when possible, it typically requires many iterations, significant human involvement, and substantial time and computational resources.

In this paper, we propose a new paradigm that minimizes human involvement. The main idea is twofold. First, instead of being hand-crafted, the search space is learned from existing mechanical designs used as training data. Although the available designs do not constitute a massive dataset on the scale of vision or language data, they are nevertheless diverse enough to capture meaningful structural patterns and priors for guiding design optimization – a point we demonstrate in this paper.

Second, the loss function is defined directly from human motion data, requiring data but not heuristic or costly manual design. In this regard, we address the key challenge of reconciling the discrepancy between the human skeletal kinematic structure and the robot kinematic structure being optimized by leveraging motion retargeting techniques and Procrustes analysis.

Regarding the search space, the first question that naturally arises is how to numerically represent existing designs. Adopting screw theory [2], [3], we model each joint as a 6D vector $(\boldsymbol{\omega}, \mathbf{q})$, with $\boldsymbol{\omega}$ a unit axis and $\mathbf{q}$ the joint position in a common base frame. We focus on designs with broadly similar morphologies – such as humanoids with conventional kinematic trees – so that a design with N joints corresponds to a point in $\mathbb{R}^{6N}$, with padding applied for those with fewer joints.

The next question is how to identify a lower-dimensional subspace of existing designs that enables efficient optimization. Although each design resides in a high-dimensional space of size $6N$ (e.g., humanoids with over 20 joints correspond to more than 100 dimensions), the set of realizable designs is expected to lie on a structured, lower-dimensional manifold. To capture this, we use an encoder–decoder framework for manifold learning [4]–[7], mapping designs into a compact, lower-dimensional yet expressive latent space.

To further improve structure in the latent space, we em-

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT): (No. RS-2024-00457882, AI Research Hub Project, 50%), (No. RS-2022-II220871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 25%), Ministry of Trade, Industry, and Energy (MOTIE), Korea, under the "Global Industrial Technology Cooperation Center program" supervised by the Korea Institute for Advancement of Technology (KIAT). (Grant No. P0028435, 25%).

[1]Jihwan Yoon, Taemoon Jeong, Jeongeun Park, Chanwoo Kim, and Sungjoon Choi are with the Department of Artificial Intelligence, Korea University, Seoul 02841, Republic of Korea (e-mail: {yoonmungchi, taemoonjeong, baro0906, chanwoo-kim, sungjoon-choi}@korea.ac.kr).

[2]Jaewoon Kwon is with Contoro Robotics, Austin, TX 78758, USA (e-mail: jaewoon@contoro.com).

[3]Yonghyeon Lee is with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (e-mail: yhl@mit.edu).

[4]Kyungjae Lee is with the Department of Statistics, Korea University, Seoul 02841, Republic of Korea (e-mail: kyungjae_lee@korea.ac.kr).

*Corresponding author.

ploy an isometric regularization approach, which encourages smoothness and geometry preservation [7]–[10]. Optimization is then carried out directly in this latent space, with the loss function defined from human motion data – via Procrustes analysis – and solved using Voronoi Optimistic Optimization (VOO) [11].

The contributions of this paper are summarized as follows:

- A screw-theory-based representation that enables intuitive and scalable encoding of kinematic structures.
- A learning-based framework for search space design that leverages existing mechanical designs.
- An integrated pipeline that combines an isometric autoencoder, a human-motion–retargeting-based loss function, and VOO.

II. Related Work: Learning-based Design Optimization

In this section, we review learning-based design methods, focusing on two main directions: latent-model–based optimization [12]–[14], which is closely related to our framework, and reinforcement learning approaches [15], [16].

To address the curse of dimensionality in robot design, some recent approaches propose learning low-dimensional latent spaces that retain design validity while enabling efficient optimization [12]–[14]. Grammar-guided Latent Space Optimization (GLSO) [12] embeds modular robot structures into a continuous latent space using a graph VAE, allowing Bayesian optimization to be performed with high sample efficiency. COIL [13] constrains the latent space to represent only valid designs and uses evolutionary strategies to search it. MorphVAE [14] applies VAE-based learning to soft robot morphology using continuous evolutionary sampling between tasks and generations.

Other methods, such as MORPH [15] and N-LIMB [16], integrate reinforcement learning with differentiable or universal control policies to co-optimize morphology and behavior. However, these approaches are computationally inefficient, lack search-space reduction, and often explore spaces so large that they lead to designs that are not practically manufacturable.

Our approach differs in three major ways: (i) prior works rely on graph or voxel representations, whereas we adopt a screw-theory-based representation; (ii) they depend on synthetic datasets generated under human-defined constraints, whereas we leverage real, existing robot designs; and (iii) their loss functions are manually designed, while ours is directly defined from human motion data. Taken together, these distinctions advance our framework toward more automated and less human-involved robot design. Additionally, while all of these methods remain limited to simulation studies, we present a real robot design, demonstrating the real-world applicability of our approach.

III. Preliminaries

Our framework consists of three main components: (i) representation of the kinematic structure, (ii) data-driven dimensionality reduction of the search space, and (iii) optimization via retargeting in the reduced space. To support these components, this section introduces screw theory, manifold learning, and motion retargeting with Procrustes analysis, which together serve as the fundamental building blocks of our design framework.

A. Screw Theory

According to the Chasles–Mozzi theorem, any rigid-body displacement can be expressed as a screw motion, i.e., a rotation combined with a translation along a fixed line in space called the screw axis [2], [17], [18]. The screw axis is defined as the line passing through a point $\mathbf{q}$ and oriented by a unit vector $\boldsymbol{\omega}$. The displacement consists of a rotation about $\boldsymbol{\omega}$ by an angle θ, together with a translation parallel to $\boldsymbol{\omega}$ of magnitude θh, where h denotes the screw pitch. Then, the corresponding rigid-body motion is characterized by this screw motion:

$$\exp\left(\begin{bmatrix}[\boldsymbol{\omega}] & \mathbf{v}\\ 0 & 0\end{bmatrix}\right) \in \mathrm{SE}(3) \subset \mathbb{R}^{4\times 4}, \tag{1}$$

where

$$[\boldsymbol{\omega}] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2}$$

and $\mathbf{v} = -\boldsymbol{\omega} \times \mathbf{q} + h\boldsymbol{\omega}$. Thus, $(\boldsymbol{\omega}, h, \mathbf{q})$ provides a complete parametrization of a rigid-body motion via its screw axis: $\boldsymbol{\omega}$ specifies the axis direction, h the screw pitch, and q a point on the axis. In particular, the case $h = 0$ corresponds to pure rotation, which we use as a compact geometric representation of each joint axis throughout the paper to effectively represent robot models.

B. Manifold Learning and Isometric Regularization

We view an autoencoder, consisting of an encoder f that maps a data point $\mathbf{x}$ to a lower-dimensional latent variable $\mathbf{z}$ and a decoder $g : \mathbf{z} \mapsto \mathbf{x}$, as learning a chart of the data manifold [4], [5], i.e., the decoder g parametrizes the data manifold. The pullback metric on the latent space is given by $G(\mathbf{z}) = J_g(\mathbf{z})^\top J_g(\mathbf{z})$, where J_g denotes the Jacobian of g. The deviation of $G(\mathbf{z})$ from the identity matrix $\mathbf{I}$ quantifies local geometric distortion (to see why, consider $d\mathbf{x} = J_g(\mathbf{z})d\mathbf{z}$ and $d\mathbf{x}^\top d\mathbf{x} = d\mathbf{z}^\top J_g^\top J_g d\mathbf{z}$).

Isometry regularization encourages the latent space to preserve the geometry of the data manifold by reducing distortion [7], [8], [10], i.e., encouraging $J_g^\top J_g \approx \alpha\mathbf{I}$ (isometry up to scale), and has also been shown to yield a (intrinsically) flatter manifold [6]. Specifically, the *isometric* regularization term is derived as

$$\mathcal{L}_{\text{iso}} = \frac{\mathbb{E}_z[\mathrm{Tr}((J_g^\top J_g)^2)]}{\mathbb{E}_z[\mathrm{Tr}(J_g^\top J_g)]^2}, \tag{3}$$

where $\mathbb{E}_z$ is over latent space data distribution, which in practice can be efficiently computed by using Jacobian-vector and vector-Jacobian products [7]. Isometric regularization stabilizes training with limited data and yields a well-conditioned, geometry-preserving latent design space, along with a smooth manifold well suited for our kinematic optimization.

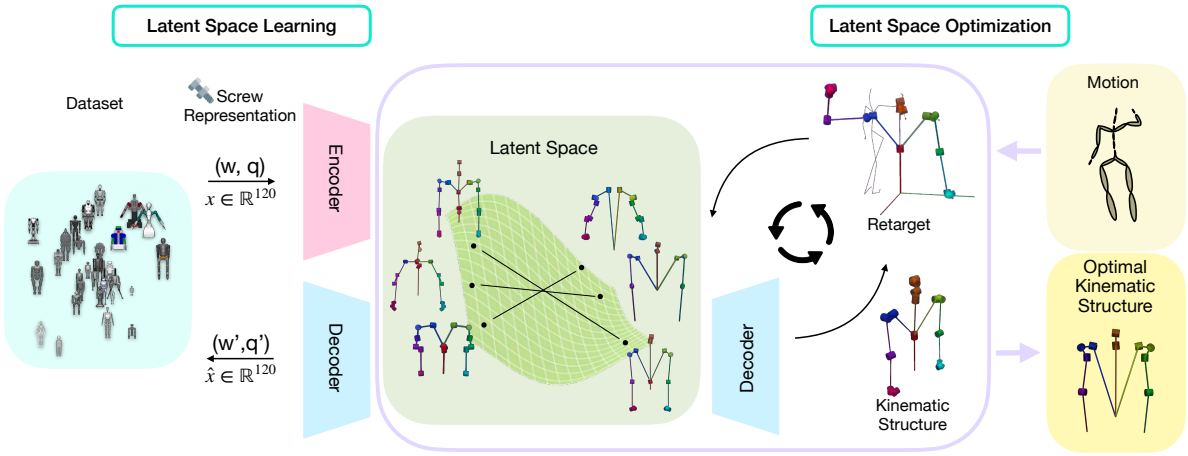

Fig. 1: Total pipeline for humanoid kinematic structure optimization. First, a dataset of robots is converted to a unified **Screw Representation** ((ω, q)). An autoencoder with an **isometric regularization loss** learns a compact 2D latent space from these representations. Next, **Voronoi Optimistic Optimization (VOO)** iteratively samples and decodes candidates from the latent space. Each candidate is evaluated by its motion retargeting performance using **Procrustes analysis**. The process converges to an optimal kinematic structure that best performs the given task.

C. Motion Retargeting and Procrustes Analysis

We follow a motion retargeting strategy inspired by prior works on humanoid motion transfer [19]–[21]. Because direct joint mapping is generally infeasible due to morphological differences, we define the target positions of the robot's joints of interest (JOI) using scaled directional vectors from the human skeleton. The corresponding joint angles are then computed by solving a numerical inverse kinematics problem under joint limit constraints. To quantify the similarity between the source and retargeted motions, we employ Procrustes analysis [22], an alignment method that removes differences in translation, rotation, and scale between trajectories. This yields the Procrustes Aligned Mean Per Joint Position Error (PA-MPJPE) [23]–[27], which serves as a normalized measure of motion similarity and provides the basis for evaluating candidate kinematic structures in our optimization framework.

IV. Method

In this section, we present our pipeline for motion-specific humanoid upper body design optimization. We begin by formulating the problem as a structure–motion co-optimization task, where the goal is to identify a kinematic structure that minimizes the discrepancy between a target human motion and its retargeted execution on the robot. We then introduce a screw-theoretic representation (w, q) to encode upper body humanoid structures in global coordinates, followed by a data curation process that unifies 30 diverse robot models into a consistent 120-dimensional feature vector. Next, we learn a compact latent design space using an isometric-regularized autoencoder, which preserves geometric continuity while reducing dimensionality. Finally, we perform optimization in this latent space with VOO, combining motion retargeting and Procrustes analysis to identify feasible, symmetric upper-body designs tailored to specific motions.

A. Problem Formulation

We are given a source motion M_{src}, represented as a set of joint trajectories captured from a source kinematic model (e.g., a human skeleton). Our objective is to identify a new kinematic structure $\mathbf{x} \in \mathcal{X}$, together with its retargeted motion, such that the retargeted motion closely matches M_{src}. Here, $\mathcal{X}$ denotes an abstract design space of admissible upper-body kinematic structures, whose precise instantiation will be specified later.

To this end, we define a retargeting operator $M_{\text{tar}} = \mathcal{R}(\mathbf{x};\, M_{\text{src}})$ which maps the source motion M_{src} onto a candidate kinematic structure $\mathbf{x}$, yielding the corresponding retargeted motion M_{tar} of $\mathbf{x}$. The operator $\mathcal{R}$ is treated as a black-box procedure: it does not require differentiability and, in practice, is not differentiable due to the discrete nature of joint mappings and inverse kinematics. The optimization problem is thus formulated as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} \mathcal{L}\Big(M_{\text{src}},\ \mathcal{R}(\mathbf{x};\, M_{\text{src}})\Big), \tag{4}$$

where the loss function $\mathcal{L}$ measures the discrepancy between the source motion and the retargeted motion; in our experiments, we instantiate $\mathcal{L}$ using Procrustes analysis [28]. Given the optimal kinematic structure $\mathbf{x}^*$, its corresponding retargeted motion M^* is $M^* = \mathcal{R}(\mathbf{x}^*;\, M_{\text{src}})$.

This general formulation captures the essence of our goal: to search over possible kinematic structures in $\mathcal{X}$ and evaluate them by how well the retargeted motion aligns with the original source motion. In the following sections, we describe how this problem is addressed efficiently by introducing a learned latent design space and performing optimization therein.

B. Screw Theory-based Representation

Screw theory has been widely used to unify rotation and translation within a compact, globally consistent for-

mulation [2]. For our goal of comparing and retargeting motions in various upper body morphologies, global joint level positional fidelity is essential: the traditional form $(\boldsymbol{\omega}, \mathbf{v})$, where $\mathbf{v}$ is the moment vector, does not directly encode absolute joint positions and depends on the choice of reference point, which can hinder spatial normalization and cross-morphology comparisons.

We therefore represent each joint by $S = (\boldsymbol{\omega}, \mathbf{q}), \quad \boldsymbol{\omega}, \mathbf{q} \in \mathbb{R}^3$, where $\boldsymbol{\omega}$ denotes the direction (unit vector) of the global joint axis and $\mathbf{q}$ a point on that axis given in the world frame. This $(\boldsymbol{\omega}, \mathbf{q})$ parameterization preserves explicit positional information, which is advantageous for motion retargeting and structural normalization. When needed (e.g., for revolute joints), the moment vector can be recovered as $\mathbf{v} = -\boldsymbol{\omega} \times \mathbf{q}$.

A practical challenge is that robots differ in the number of joints per anatomical group, whereas the learning stage requires fixed-length inputs. To avoid injecting artificial directional bias during padding, we assign an orientationless placeholder to missing axes by setting $\boldsymbol{\omega} = \mathbf{0}$. Because this study considers only revolute joints, a zero axis should be interpreted solely as a placeholder rather than as an indicator of prismatic motion. For missing positions, we preserve spatial context by imputing $\mathbf{q}$ with the mean of the existing joints in the same group; when a group is entirely absent, $\mathbf{q}$ is inherited from the nearest parent group along the kinematic hierarchy. This scheme retains positional structure through $\mathbf{q}$ while keeping the axis channel uninformative wherever data are missing.

While the $(\boldsymbol{\omega}, \mathbf{q})$ representation is interpretable and globally consistent, the resulting design vector remains high-dimensional, making direct search inefficient and brittle. In the next subsections, we therefore learn a low-dimensional *latent design space* from these screw-theoretic features and impose isometric regularization to obtain an optimization-friendly embedding.

C. Humanoid Model Data Curation

We constructed our dataset using 30 diverse humanoid and bi-arm robot models [29]–[52], including representative examples such as Talos [30], Baxter [32], Valkyrie [37], and PR2 [38], selected to cover a wide range of sizes, joint configurations, and application domains. The complete list of 30 robots used in the dataset is visualized in Figure 3. Each robot was simulated in MuJoCo [53] and posed in an A-pose (or the closest feasible configuration) to standardize joint orientations, particularly at the elbows. For robots without existing MJCF files, URDF models were converted to MJCF. Global joint positions and axes were recorded in the world coordinate frame.

Joints were categorized into 14 anatomical groups: torso and neck (no side), plus shoulder girdle, shoulder, upper arm, elbow, forearm, and wrist (left/right counterparts). Virtual joints were inserted only for missing wrists, as they serve as end-effectors and are necessary for maintaining structural consistency.

For each group, we determined the maximum joint count across all robots and padded missing entries accordingly, imputing positions with the group mean when the group was partially present and inheriting from the nearest parent when it was entirely absent. The axis channel of padded entries followed the orientationless placeholder rule defined in our screw-theoretic representation, keeping the axis information intentionally uninformative wherever data were missing.

Positions were normalized by the shoulder-to-base distance to ensure size invariance. To reduce redundancy and enforce symmetry, only right-side joints (excluding torso and neck) were retained in the data representation. This compact representation maintains morphological completeness through mirror reconstruction during optimization. The final representation for each robot was a $(20, 6)$ array—20 joints with $(\boldsymbol{\omega}, \mathbf{q})$ screw parameters—flattened into a 120-dimensional vector for downstream learning.

D. Design manifold Learning

We trained an autoencoder, an encoder f_θ and decoder g_θ modeled with neural networks, with isometric regularization [7] on the 120-dimensional screw-based vectors from our curated dataset. The network comprises two hidden layers (64 and 32 units) and a latent dimension of 2 (with additional tests at 3), using Tanh activations and L2 reconstruction loss. Isometric regularization encourages the local Jacobian to approximate the identity matrix, preserving geometric continuity and reducing curvature in the latent space – especially beneficial for small datasets to improve stability and generalization.

E. Optimization within the Manifold

Given the learned latent design space, we search latent vectors $\mathbf{z}$ whose decoded kinematic structures $g_\theta(\mathbf{z})$ can best reproduce the source motion M_{src}. Specifically, we minimize

$$\mathcal{L}_{\text{total}}(\mathbf{z}; M_{\text{src}}) = \mathcal{L}_{\text{PA-MPJPE}}\Big(M_{\text{src}},\, \mathcal{R}(g_\theta(\mathbf{z}); M_{\text{src}})\Big) + \lambda_{\text{joint}}\, N_{\text{tot}}\big(g_\theta(\mathbf{z})\big). \tag{5}$$

where $\mathcal{R}$ retargets M_{src} onto the decoded structure, and the PA-MPJPE [54] loss aligns motions via a Procrustes transform and then computes mean per-joint position error. $N_{\text{tot}}(\cdot)$ counts the *total* number of active joints in the mirrored full upper body; a joint j is considered active if $\|\boldsymbol{\omega}_j\|_2 \geq \varepsilon$, which ignores padded placeholders. This penalty discourages gratuitous articulation and biases the search toward parsimonious, task-specific designs.

We optimize $\mathcal{L}_{\text{total}}$ over $\mathbf{z}$ using VOO, a gradient-free method effective in low-dimensional continuous spaces. Because our representation stores only right-side joints (plus torso and neck), we mirror the decoded right side across the sagittal plane to obtain the full upper body before evaluation and count N_{tot}.

V. Experiments

In this section, we evaluate our proposed method on three upper-body CMU Mocap sequences through two primary experiments. First, we demonstrate the effectiveness of our screw-theory representation within a AE latent space by comparing it against the traditional DH parameterization and

direct optimization in the naive 120-dimensional space, as detailed in Section V-B. Second, we validate that applying isometric regularization improves designs and that a compact, two-dimensional latent space is sufficient for effective optimization, as shown in Section V-C.

A. Experimental Setup

Motions. We use three CMU Mocap sequences as M_{src}: *141_16 Wave Hello* (arm-dominant; negligible torso), *18_15 Chicken Dance* (moderate torso yaw), and *79_02 Swimming* (torso yaw+pitch with dynamic arms) [55].

Search space & optimizer. Candidate structures are decoded as $g_\theta(\mathbf{z})$ from $(\boldsymbol{\omega}, \mathbf{q})$ features and mirrored to obtain the full upper body; for the retargeter $\mathcal{R}$, we used optimization-based retargeting method [21]. We search over $\mathbf{z}$ with VOO under a fixed evaluation budget shared across methods.

Objective & hyperparameters. We minimize PA-MPJPE [54] penalized with number of joint, $J = \text{PA-MPJPE} + \lambda_{\text{joint}} N_{\text{tot}}$, counting a joint active if $\|\boldsymbol{\omega}_j\|_2 \geq \varepsilon$; we set $\lambda_{\text{joint}} = 3.5$ and $\varepsilon = 0.5$. For DH baselines, near-zero links are clamped to zero when $|d| \leq 0.01$ or $|a| \leq 0.01$.

Protocol. All methods use the same evaluation budget and ten shared seeds; the decoder g_θ is frozen during optimization. VOO searches the latent box $\mathbf{z} \in [-15, 15]^d$ (d=2 for main results; d=3 for the ablation) with n_{init}=16 uniformly drawn starts and T=30 subsequent iterations (one evaluation each), i.e., n_{eval}=$n_{\text{init}} + T = 46$ evaluations per run. Unless otherwise noted, we also report the total objective averaged over the three motions together with its two components (PA-MPJPE and N_{tot}).

B. Representation and Manifold of Robot Structure

We present a quantitative comparison of our method in Table I to show the effectiveness of screw-based representation and manifold learning. We report the best-within-budget total objective (mean±std over 10 seeds) *per* motion and also for the joint objective over *set* of all three motions (sum of the three retargeting terms plus one joint-count penalty). To evaluate our approach, we compare our method against two baselines: Denavit–Hartenberg parameters (DH) [56], [57], which uses a learned manifold of traditional DH parameters, and a direct screw-theory based representation (baseline), which performs direct optimization on the high-dimensional parameter space. We implemented standard DH parameterization, augmenting it with an axis offset to account for the humanoid upper-body topology—from the torso to the neck and shoulders—and for robots whose wrists terminate as links without an explicit joint, we introduced separate tool center points (TCP) for each wrist in addition to the 14 anatomical joints.

Compared with DH [56], [57] parameter-based representation, the reductions are 39.9% (Wave), 32.4% (Dance), 32.9% (Swimming) with a single-motion average of 35.0%, and 43.5% on the all-motions objective. The superior result of the screw-theoric representation for optimization indicates that the screw-theoretic representation with isometric regularization provides both superior accuracy and stability, especially when a single structure must accommodate heterogeneous motions.

TABLE I: Best-within-budget total objective on three motions (mean±std over 10 seeds; lower is better). Objective is PA-MPJPE + $\lambda_{\text{joint}} N_{\text{tot}}$ with $\lambda_{\text{joint}} = 3.5$ and $\varepsilon = 0.5$.

Rep.	Motion(↓) Wave	Dance	Swimming	Motion Set(↓)
Baseline	152.2 ± 13.9	153.0 ± 10.2	154.8 ± 15.4	274.4 ± 33.9
DH [56]	162.1 ± 17.4	148.8 ± 17.3	150.9 ± 20.9	315.8 ± 27.3
Ours	$\mathbf{97.5 \pm 3.6}$	$\mathbf{100.6 \pm 5.0}$	$\mathbf{101.3 \pm 3.6}$	$\mathbf{178.4 \pm 10.8}$

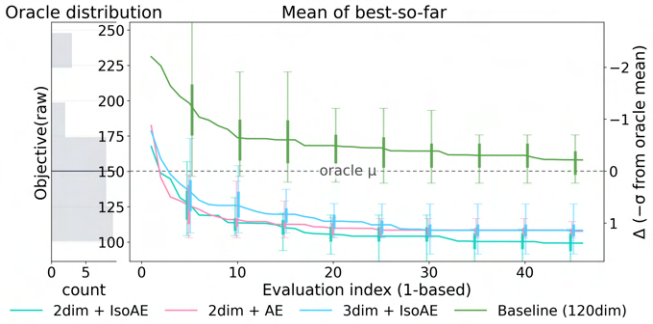

Fig. 2: Mean best-so-far objective across ten shared seeds(lower is better). The error bars represent the interquartile range (thick) and min–max (thin). Left inset: oracle distribution with dashed mean μ on motion *chicken dance*.

By comparing our full method to baseline, which uses the same screw-theory parameters but optimizes in the full 120-dimensional space without manifold learning, we observe massive and consistent performance gains. This approach yields objective score reductions of 36.0%, 34.2%, and 34.6% on the individual motions, and 35.0% on the joint all-motions task. This large margin highlights the difficulty of navigating high-dimensional parameter spaces, as learning a structured manifold of valid robot designs makes the optimization problem more tractable.

C. Role of Isometric Regularization and Latent Dimension

Figure 2 shows the mean best-so-far objective across ten shared seeds under a fixed evaluation budget on *18_15 chicken dance* motion compared with the use of the isometric regularizer (2D AE [58], [59]) and latent dimension size (2D IsoAE and 3D IsoAE). 2D IsoAE (Ours) achieves the lowest mean best-so-far objective over the entire 46-evaluation budget. It drops sharply in the first 5–10 evaluations, falls below the oracle mean μ early, and then continues to improve monotonically before stabilizing around steps 30–35; its dispersion is tighter than 3D IsoAE, indicating that a two-dimensional latent design space suffices. Compared with 2D AE (no isometry), 2D IsoAE remains uniformly better across all steps, evidencing the benefit of the isometric regularizer. In contrast, baseline (screw representation with 120 dimensions) starts higher, improves slowly, and stays near or above μ with wide interquartile range (IQR)/min–max bands, highlighting the difficulty of high-dimensional search without representation learning.

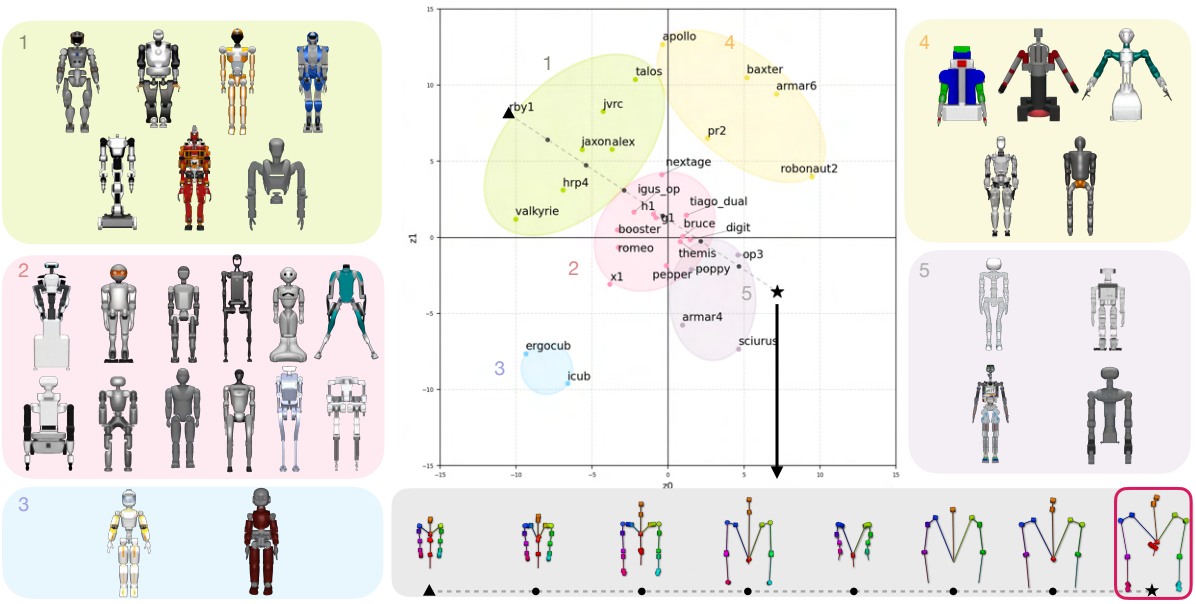

Fig. 3: Latent map of 30 robots (2D IsoAE). Colors show k-means clusters (k=5) with representative thumbnails, which demonstrate that the robots are semantically grouped based on their morphological similarity (e.g., full-body humanoids, dual-arm manipulators). The bottom strip interpolates eight points from `rby1` to a starred design (sufficient for *Swimming*); joint birth/death is visible as markers appear/disappear under the activation rule.

Beyond the ranking, three properties stand out. *Sample efficiency*: within the first ten evaluations, 2D IsoAE already sits well below μ and separates on the normalized right axis, while 3D IsoAE and 2D AE lag and the 120D baseline hovers around μ. *Robustness*: the IQR of 2D IsoAE tightens quickly and its min–max whiskers shrink after roughly 20 evaluations; by the final budget, the IQR bands of 2D IsoAE have little overlap with those of 3D IsoAE or 2D AE, indicating lower seed sensitivity. *Dimension effect*: under a fixed budget, increasing latent dimensionality dilutes VOO's sampling density (Voronoi cells effectively shrink in higher d), which hurts exploitation—consistent with the slower progress of 3D IsoAE and the 120D direct baseline.

D. Discussion: structure of the learned design space

Beyond its quantitative performance, our model learns a meaningful latent space for robot design. This is visualized in Figure 3, which shows how the manifold groups similar structures into distinct k-means clusters (k=5) with representative thumbnails. The clusters align with intuitive morphology families: a "large full-body humanoids" group (e.g., Talos [30], JAXON [39], rby1 [52], etc.) occupies the upper-left region; dual-arm mobile-manipulators and upper-torso platforms (Baxter [32], PR2 [38], etc.) reside in the upper-right; compact social/service humanoids (Pepper, OP3, H1, etc.) concentrate near the origin; the iCub family (iCub [34], ergoCub [33]) forms a distinct basin in the lower-left; and a lighter research-platform cluster (e.g., ARMAR-4 [46], Sciurus [44]) appears in the lower-right. These groupings suggest that the isometric latent space preserves semantically meaningful kinematic traits (torso involvement, shoulder-girdle complexity, anthropomorphic proportion), providing a navigable atlas for design search.

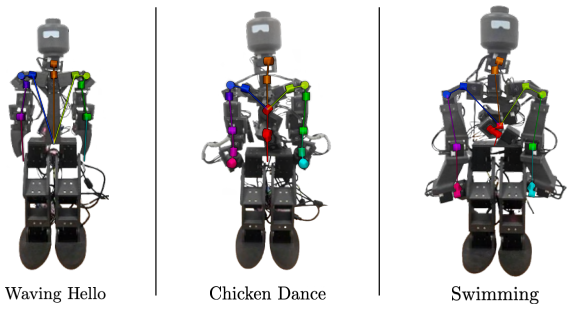

Fig. 4: Hardware prototypes generated by our design framework under manufacturing constraints. Each prototype is optimized for a specific task ('Waving Hello', 'Chicken Dance', 'Swimming'), resulting in different kinematic structures.

Interpolating through the latent space reveals both smooth morphing of link layouts and discrete changes like joint birth/death, showing that our decoder accommodates variable joint counts. Notably, optimal solutions for certain motions (starred) can lie on cluster boundaries, suggesting they blend features from different design families. Overall, the space provides: (i) coherent clustering, (ii) smooth and discrete kinematic changes, and (iii) actionable starting points for future optimization.

E. Hardware

To empirically validate the pipeline, we fabricated a sufficiently good hardware prototype under manufacturing and schedule constraints. All kinematic structures were denormalized by applying a single scale factor such that, relative to a 170mm lower-body reference, the base-to-neck distance equals 124mm. Assembly was performed in the same A-pose world frame used throughout the dataset. Joint axes and refer-

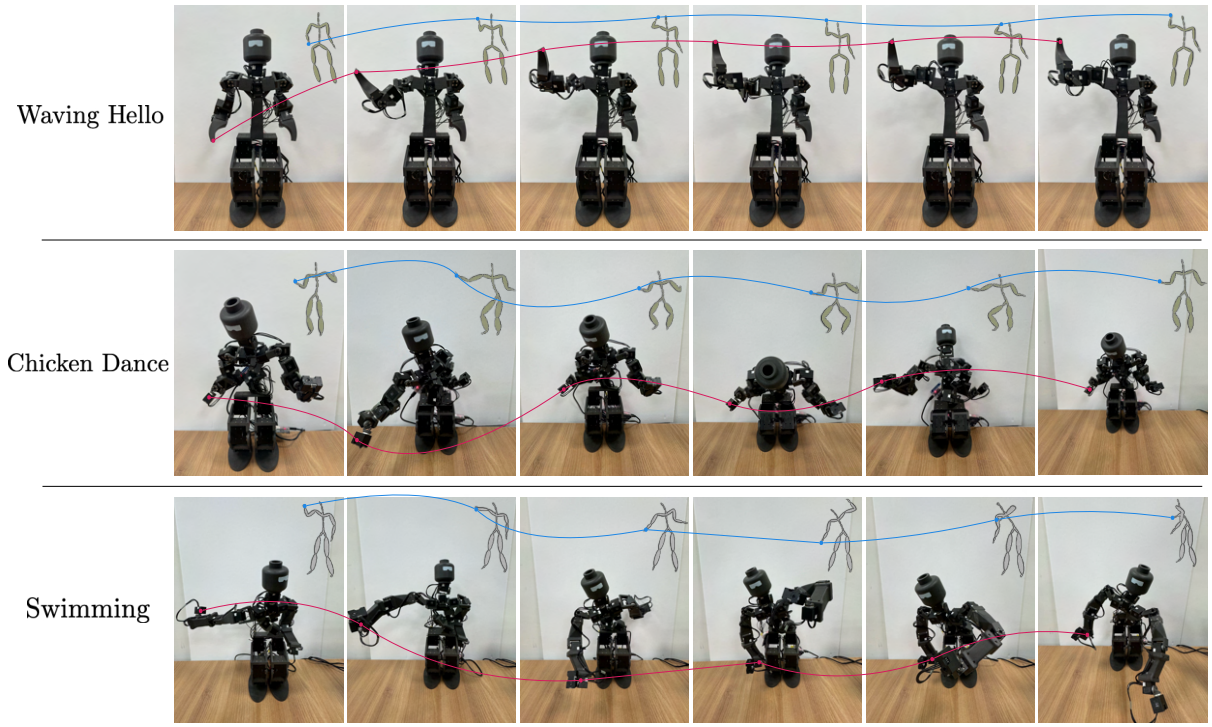

Fig. 5: Demonstration of the three motion-optimized robots from Fig. 4 successfully executing Waving Hello, Chicken Dance, and Swimming via motion retargeting. For each robot snapshot, the human figure (upper right) shows the key pose from the source human motion (BVH file) used as input, while the photo shows the retargeted execution on the robot's unique kinematics. The red and blue lines visualize the 3D end-effector trajectories of robot and source human motion.

ence positions were nominally preserved; where unavoidable to accommodate packaging and assembly, we permitted local position adjustments up to 10mm (direction-agnostic). Left–right geometry was obtained by geometric mirroring about the sagittal plane, and all subsequent retargeting and evaluation used the as-built CAD/MJCF model. On the as-built platform, we executed the retargeted joint trajectories for all three source motions—*Wave Hello*, *Chicken Dance*, and *Swimming*—using the same world frame and activation rule as in simulation. Without any architectural changes to the pipeline, the as-built upper-body humanoids smoothly tracked the commanded poses (see Fig. 5).

F. Limitations

While the retargeting-based objective promotes task fidelity and parsimony, our decoder and search do not yet enforce hard manufacturability constraints. In particular, we do not impose (i) kinematic feasibility such as discrete joint types and range limits, (ii) geometric validity including self-collision avoidance, mesh/packaging/clearance constraints, and (iii) dynamic plausibility with actuation limits, torque/energy budgets, and contact stability. Future work will integrate these constraints via a structure-aware decoder (e.g., feasibility-projected decoding), lightweight feasibility filters (reject/repair with collision and range checks), and simulator-backed penalties (inverse dynamics, torque limits), moving toward multi-objective design and real-hardware validation.

VI. Conclusion

We presented a motion-to-structure pipeline for upper-body humanoid design optimization. The approach hinges on three components: (i) a screw-theoretic joint representation $(\boldsymbol{\omega}, \mathbf{q})$ that encodes global joint axes and positions and enables consistent cross-morphology reasoning; (ii) a compact latent design space learned with an isometric-regularized autoencoder to provide a smooth, optimization-friendly manifold; and (iii) gradient-free search in this manifold using VOO, driven by a *total objective* that combines PA-MPJPE after retargeting with a mild joint-count penalty. Together, these elements turn human motion priors into actionable guidance for selecting kinematic structures, achieving $\sim$34% lower average objective than direct search and $\sim$39% than DH+IsoAE across three motions.

References

[1] J. Kwon, S. Kim, and F. C. Park, "Physically Consistent Lie Group Mesh Models for Robot Design and Motion Co-Optimization," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9501–9508, 2022.

[2] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.

[3] F. C. Park, "Computational aspects of the product-of-exponentials formula for robot kinematics," *IEEE Trans. Autom. Control*, vol. 39, no. 3, pp. 643–647, 1994.

[4] Y. Lee, "A Geometric Perspective on Autoencoders," *arXiv preprint arXiv:2309.08247*, 2023.

[5] Y. Lee, H. Kwon, and F. Park, "Neighborhood Reconstructing Autoencoders," *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, pp. 536–546, 2021.

[6] Y. Lee and F. C. Park, "On Explicit Curvature Regularization in Deep Generative Models," in *TAG-ML Workshops*. Proc. Int. Conf. Mach. Learn. (ICML), 2023, pp. 505–518.

[7] Y. Lee *et al.*, "Regularized Autoencoders for Isometric Representation Learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[8] J. Lim *et al.*, "Graph Geometry-Preserving Autoencoders," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2024.

[9] Y. Lee, "Motion Manifold Primitives With Parametric Curve Models," *IEEE Trans. Robot.*, 2024.

[10] H. Heo *et al.*, "Isometric Regularization for Manifolds of Functional Data," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

[11] B. Kim *et al.*, "Monte Carlo Tree Search in Continuous Spaces Using Voronoi Optimistic Optimization with Regret Bounds," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 9916–9924.

[12] J. Hu, J. Whitman, and H. Choset, "GLSO: Grammar-guided Latent Space Optimization for Sample-efficient Robot Design Automation," in *Proc. Conf. Robot Learn. (CoRL)*, 2022.

[13] P. J. Bentley *et al.*, "COIL: Constrained Optimization in Learned Latent Space – Learning Representations for Valid Solutions," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, 2022.

[14] J. Song *et al.*, "MorphVAE: Advancing Morphological Design of Voxel-Based Soft Robots with Variational Autoencoders," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 9, 2024, pp. 10 368–10 376.

[15] Z. He and M. Ciocarlie, "MORPH: Design Co-optimization with Reinforcement Learning via a Differentiable Hardware Model Proxy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2024.

[16] C. Schaff and M. R. Walter, "N-LIMB: Neural Limb Optimization for Efficient Morphological Design," *arXiv preprint arXiv:2207.11773*, 2022.

[17] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[18] F. C. Park *et al.*, "Geometric Algorithms for Robot Dynamics: A Tutorial Review," *Appl. Mech. Rev.*, vol. 70, no. 1, p. 010803, 2018.

[19] S. Choi and J. Kim, "Towards a Natural Motion Generator: a Pipeline to Control a Humanoid based on Motion Data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 4373–4380.

[20] T. Jeong *et al.*, "CoRe: A Hybrid Approach of Contact-Aware Optimization and Learning for Humanoid Robot Motions," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2025, pp. 293–300.

[21] T. Jeong *et al.*, "Robust and Expressive Humanoid Motion Retargeting via Optimization-Based Rig Unification," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2025, pp. 21 619–21 626.

[22] C. Goodall, "Procrustes methods in the statistical analysis of shape," *J. R. Stat. Soc. B*, vol. 53, no. 2, pp. 285–321, 1991.

[23] J. Li *et al.*, "HybrIK: A Hybrid Analytical-Neural Inverse Kinematics Solution for 3D Human Pose and Shape Estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 3383–3393.

[24] J. Liu *et al.*, "Normalized Human Pose Features for Human Action Video Alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 11 521–11 531.

[25] E. Gärtner *et al.*, "Differentiable Dynamics for Articulated 3d Human Motion Reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 13 190–13 200.

[26] Z. Yang *et al.*, "SynBody: Synthetic Dataset with Layered Human Models for 3D Human Perception and Modeling," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 20 282–20 292.

[27] E. Gärtner *et al.*, "Trajectory Optimization for Physics-Based Reconstruction of 3d Human Pose from Monocular Video," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 13 106–13 115.

[28] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[29] A. Adu-Bredu, "DigitRobot.jl," https://github.com/adubredu/DigitRobot.jl, approved copyright permission for use in this paper by Agility Robotics.

[30] O. Stasse, T. Flayols *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2017, pp. 689–695.

[31] K. Zakka, Y. Tassa, and MuJoCo Menagerie Contributors, "MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo." [Online]. Available: http://github.com/google-deepmind/mujoco_menagerie

[32] S. Cremer, L. Mastromoro, and D. O. Popa, "On the performance of the baxter research robot," in *Proc. IEEE Int. Symp. Assem. Manuf. (ISAM)*, 2016, pp. 106–111.

[33] iCub Tech, "ergoCub software," https://github.com/icub-tech-iit/ergocub-software, 2022.

[34] G. Metta *et al.*, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Netw.*, vol. 23, no. 8–9, pp. 1125–1134, 2010.

[35] IHMC Robotics, "IHMC Alex SDK," https://github.com/ihmcrobotics/ihmc-alex-sdk.

[36] C. Mastalli, G. Saurel, and example-robot-data Contributors, "Example Robot URDFs." [Online]. Available: https://github.com/Gepetto/example-robot-data

[37] N. A. Radford *et al.*, "Valkyrie: NASA's First Bipedal Humanoid Robot," *J. Field Robot.*, vol. 32, no. 3, pp. 397–419, 2015.

[38] B. Pitzer, S. Osentoski *et al.*, "PR2 Remote Lab: An Environment for Remote Development and Experimentation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2012, pp. 3200–3205.

[39] K. Kojima, T. Karasawa *et al.*, "Development of Life-Sized High-Power Humanoid Robot JAXON for Real-World Use," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2015.

[40] M. Okugawa *et al.*, "Proposal of inspection and rescue tasks for tunnel disasters — Task development of Japan virtual robotics challenge," in *Proc. IEEE Int. Symp. Saf. Secur. Rescue Robot. (SSRR)*, 2015, pp. 1–2.

[41] M. Lapeyre, P. Rouanet *et al.*, "Poppy: Open Source 3D Printed Robot for Experiments in Developmental Robotics," in *Proc. IEEE Int. Conf. Dev. Learn. Epigenet. Robot. (ICDL-Epirob)*, 2014.

[42] M. A. Diftler, J. S. Mehling *et al.*, "Robonaut 2 – The First Humanoid Robot in Space," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011.

[43] Tokyo Opensource Robotics Kyokai Association, "Nextage Open ros packages," https://github.com/tork-a/rtmros_nextage, accessed: 2024.

[44] RT Corporation, "Sciurus17 description package," https://github.com/rt-net/sciurus17_description, disclaimer: Development based on this model falls outside the scope of responsibility of RT Corporation.

[45] P. Allgeuer *et al.*, "igus® humanoid open platform," *KI – Künstliche Intell.*, vol. 30, no. 2, pp. 223–227, 2016.

[46] T. Asfour *et al.*, "ARMAR-4: A 63 DOF torque controlled humanoid robot," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2013, pp. 390–396.

[47] T. Asfour *et al.*, "ARMAR-6: A high-performance humanoid for human–robot collaboration in real-world scenarios," *IEEE Robot. Autom. Mag.*, 2019.

[48] SoftBank Robotics, "Pepper robot meshes," https://github.com/ros-naoqi/pepper_meshes, converted to MJCF for compatibility with MuJoCo environment.

[49] K. Kaneko, F. Kanehiro *et al.*, "Humanoid robot HRP-4: Humanoid robotics platform with lightweight and slim body," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2011.

[50] Westwood Robotics, "BRUCE and THEMIS simulation models," https://github.com/Westwood-Robotics/BRUCE_simulation_models, https://github.com/Westwood-Robotics/THEMIS-Simulation-Model, used with Permission.

[51] Agibot Tech, "Agibot X1 training assets," https://github.com/AgibotTech/agibot_x1_train.

[52] Rainbow Robotics, "RBY1 sdk," https://github.com/RainbowRobotics/rby1-sdk.

[53] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 5026–5033.

[54] R. Dabral, A. Mundhada *et al.*, "Learning 3d human pose from structure and motion," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 668–683.

[55] CMU Graphics Lab, "Carnegie–mellon mocap database," 2007, online: http://mocap.cs.cmu.edu/.

[56] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME J. Appl. Mech.*, vol. 22, no. 2, pp. 215–221, 1955.

[57] R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*. New York: McGraw-Hill, 1964.

[58] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[59] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.