ASME Journal of Mechanical Design
Online journal at:
https://asmedigitalcollection.asme.org/mechanicaldesign

Deep Learning Conceptual Design of Sit-to-Stand Parallel Motion Six-Bar Mechanisms

Zhijie Lyu
Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: zhijie.lyu@stonybrook.edu

Anurag Purwar[1]
Computer-Aided Design and Innovation Lab,
Department of Mechanical Engineering,
Stony Brook University,
Stony Brook, NY 11794-2300
e-mail: anurag.purwar@stonybrook.edu

The sit-to-stand (STS) motion is a crucial activity in the daily lives of individuals, and its impairment can significantly impact independence and mobility, particularly among disabled individuals. Addressing this challenge necessitates the design of mobility assist devices that can simultaneously satisfy multiple conflicting constraints. The effective design of such devices often involves the generation of numerous conceptual mechanism designs. This paper introduces an innovative single-degree-of-freedom (DOF) mechanism synthesis process for developing a highly customizable sit-to-stand (STS) mechanical device by integrating rigid body kinematics with machine learning. Unlike traditional mechanism synthesis approaches that primarily focus on limited functional requirements, such as path or motion generation, our proposed design pipeline efficiently generates a large number of 1DOF mechanism geometries and their corresponding motion paths, known as coupler curves. Leveraging a generative deep neural network, we establish a probabilistic distribution of coupler curves and their mapping to mechanism parameters. Additionally, we introduce novel metrics for quantitatively evaluating and prioritizing design concepts. The methodology yields a diverse set of viable conceptual design solutions that adhere to the specified constraints. We showcase various single-degree-of-freedom six-bar linkage mechanisms designed for STS motion, presenting them in a ranked order based on established criteria. While the primary focus is on the integration of STS motion into a versatile mobility assist device, the proposed approach holds broad applicability for addressing design challenges in various applications. [DOI: 10.1115/1.4066036]

Keywords: mechanism synthesis, machine learning, deep neural network, variational autoencoders, sit-to-stand device, conceptual design, kinematics, linkages

1 Introduction

The sit-to-stand (STS) and stand-to-sit motions, collectively denoted as STS henceforth, represent fundamental activities of daily life (ADL) critical for the overall well-being and autonomous functionality of individuals. These motions engage a majority of lower body muscles and are typically executed around 60 times per day, with some variability (±22) as reported in the literature [1]. Unfortunately, the elderly population and individuals affected by neuro-muscular disabilities, such as post-polio syndrome, multiple sclerosis, cerebral palsy, or injuries, encounter difficulties in performing these motions without external assistance. Research indicates that more than 6.8 million Americans rely on various forms of mobility assistive devices [2], and over 2 million individuals aged 64 and above experience challenges in rising from a seated position [3]. The incapacity to execute such fundamental ADL significantly diminishes the quality of life for disabled individuals, potentially leading to premature admission to assisted living facilities. This paper addresses the critical need for developing innovative solutions to enhance mobility and independence in individuals facing challenges in STS motions.

In STS maneuvers, the human body's joints follow distinct motion trajectories. Replicating these trajectories with an external mechanism is crucial for achieving natural kinematic function, as deviations may result in unnecessary stresses on joints and muscles and potential injuries. Unlike robotic exoskeletons, precise alignment of robot and human joint axes may not be essential [4]. However, practical considerations, such as device size, actuation method choice, and joint placement, introduce additional design constraints, necessitating the exploration of multiple mechanism design concepts. Current methods for designing single-degree-of-freedom (DOF) mechanisms that follow a path often yield a limited set of solutions through computationally expensive processes. Given the preference for single-DOF mechanisms due to their simplicity and cost-effectiveness, the problem of mechanism design for STS motion is formulated as "designing a single-DOF linkage mechanism where a floating link generates human body joint motion paths while adhering to geometric constraints suitable for a compact mobility device."

Existing work in path synthesis typically models the given path as a sequence of precision points, framing the problem as an optimization task to minimize errors between the precision points and the

[1]Corresponding author.

Contributed by Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received January 7, 2024; final manuscript received July 15, 2024; published online August 21, 2024. Assoc. Editor: Bin Zi.

generated points. However, for STS motion, capturing the overall shape of the path is critical, even at the expense of precision, while satisfying other critical constraints, such as vertical lift in the sagittal plane. Traditional optimization approaches are acknowledged for their slow convergence, susceptibility to local minima, sensitivity to parameter changes, and dependence on initial conditions [5–9].

Closed-loop linkage mechanisms exhibit nonlinear relationships between input and output variables, making the synthesis problem overdetermined and lacking an analytical solution, especially for higher-order mechanisms. For instance, a six-bar linkage can trace an 18-degree algebraic curve, presenting greater complexity compared to four-bar linkages [10]. In summary, poor representation of motion paths, nonlinear relationships, and interdependencies between input and output that resist simplification into simple analytical mathematical models, and known problems with optimization approaches have impeded progress in the mechanism synthesis area.

In addressing the aforementioned challenges, our proposed solution involves the utilization of a variational autoencoder (VAE) [11], a deep generative neural network model falling under the families of probabilistic graphical models and variational Bayesian methods. While generative adversarial networks (GANs) [12] have demonstrated proficiency in generating realistic images and various synthesis tasks, we advocate for VAE due to considerations related to maintaining training stability, avoiding issues like mode collapse and vanishing gradients intrinsic to GANs, and leveraging a latent space in VAE for the sampling of analogous coupler curves. VAEs have a notable advantage over vanilla neural networks (NN) due to their capacity to produce a myriad of solutions using their latent space. Additionally, by embedding the coupler curve within an image space, we can model the overall shape of the given path rather than relying on a sparse set of precision points. A trained VAE model obviates the need for an initial guess, generating mechanisms approximating the input, and the Gaussian structure of the latent space ensures that minor input modifications do not disproportionately alter the output. In summary, the merits of employing the VAE-based approach lie in its ability to overcome the limitations of traditional methods and its potential scalability for higher-order mechanisms. We term this approach a data-driven paradigm, where raw data suffices, and post-training, queries yield immediate results—a concept aligning with Erdman's vision of future tools benefiting from a *memory* or *previous solutions library* [13].

This paper addresses the kinematic and geometric constraints associated with STS motion. We introduce two key components: (1) a mechanism design pipeline integrated with deep neural networks for the exploration of diverse conceptual designs tailored to STS motion, and (2) a collection of six-bar linkage mechanisms generating the motion path of the shoulder joint while ensuring a constant orientation of a supporting bar along the path. A constant orientation motion along an arbitrary path is also defined as a *parallel motion*. This supporting bar is pivotal in providing upper body support and serving as a fixture for attaching ancillary items. Upon identification of a suitable mechanism, two instances of it are integrated into parallel planes within a walker-like frame. Figure 1 illustrates a prototype of the STS mobility assist, incorporating two planar six-bar linkage mechanisms confined to move in the sagittal plane.

Our methodology, expounded upon in subsequent sections, begins with the creation of a dataset featuring planar four-bar mechanisms. These mechanisms have normalized coupler curves, which are transformed into corresponding image representations. Subsequently, we employ a VAE to learn the distribution of these curve images, associating a unique descriptor with each coupler curve and storing them alongside their respective mechanism parameters. During synthesis, an input curve undergoes descriptor computation by the VAE, and the four-bar mechanisms in the dataset with the closest descriptors are identified as potential solutions. We then derive the four-bar *cognates* of the generated solutions, i.e., mechanisms that generate the same coupler curve [14].

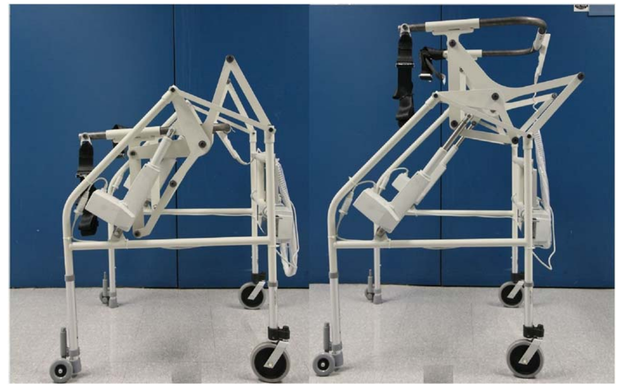

Fig. 1 A mobility assist prototype shown in sitting (left) and standing positions (right) are the two sets of six-bar mechanisms in parallel planes driven by electromechanical actuators. A New York-based manufacturer of physical rehabilitation products partnered with Stony Brook University to bring this device to the market and positioned it as an ambulation therapy aid.

By combining two specific cognates of the four-bar mechanisms using an algorithm presented in the paper, we obtain a set of six-bar parallel motion mechanisms. However, not all six-bar mechanisms may meet the design constraints, necessitating a hard constraint test for filtration. Following this test, we introduce a metric to rank the remaining six-bar candidates, facilitating the selection of the most suitable and optimal mechanisms. This structured approach ensures efficient mechanism synthesis, yielding a meticulously filtered and ranked set of solutions. An illustrative overview of this design pipeline is provided in Fig. 2, with detailed explanations in subsequent sections.

This paper introduces a novel mechanism concept design pipeline with contributions in three key areas:

(1) Development of a complete normalization technique that for the first time effectively resolves the invariance issues concerning translation, rotation, scaling, and reflection of coupler curves.
(2) A computationally efficient algorithm for generating parallel motion six-bar mechanisms based on the cognates of four-bar mechanisms.
(3) Introduction of new quantitative metrics for systematic ranking and evaluation of generated mechanism design concepts.

The paper also highlights limitations associated with utilizing an L^2 norm within the latent space of a VAE to assess the similarity of coupler curves. To address this limitation, an alternative method using a sequence-matching algorithm based on dynamic programming is proposed.

The subsequent sections of this paper are organized as follows. Section 2 provides background information on kinematic synthesis and offers a brief review of relevant literature pertaining to the design of single-DOF mechanisms for physical therapy and rehabilitation devices. Section 3 delves into the normalization technique, image-based representation of coupler curves, the NN architecture, the sequence-matching algorithm, and database preparation. Section 4 details the design specifications, criteria, and application of the VAE in designing various STS mechanisms. Finally, Sec. 5 summarizes the findings.

2 Background

The primary objective of this paper is to introduce a comprehensive design methodology that integrates rigid body kinematics and machine learning for the synthesis of motion-generating devices. Mechanism synthesis entails determining both the topology, characterized by the interconnection pattern of links and joints, and the

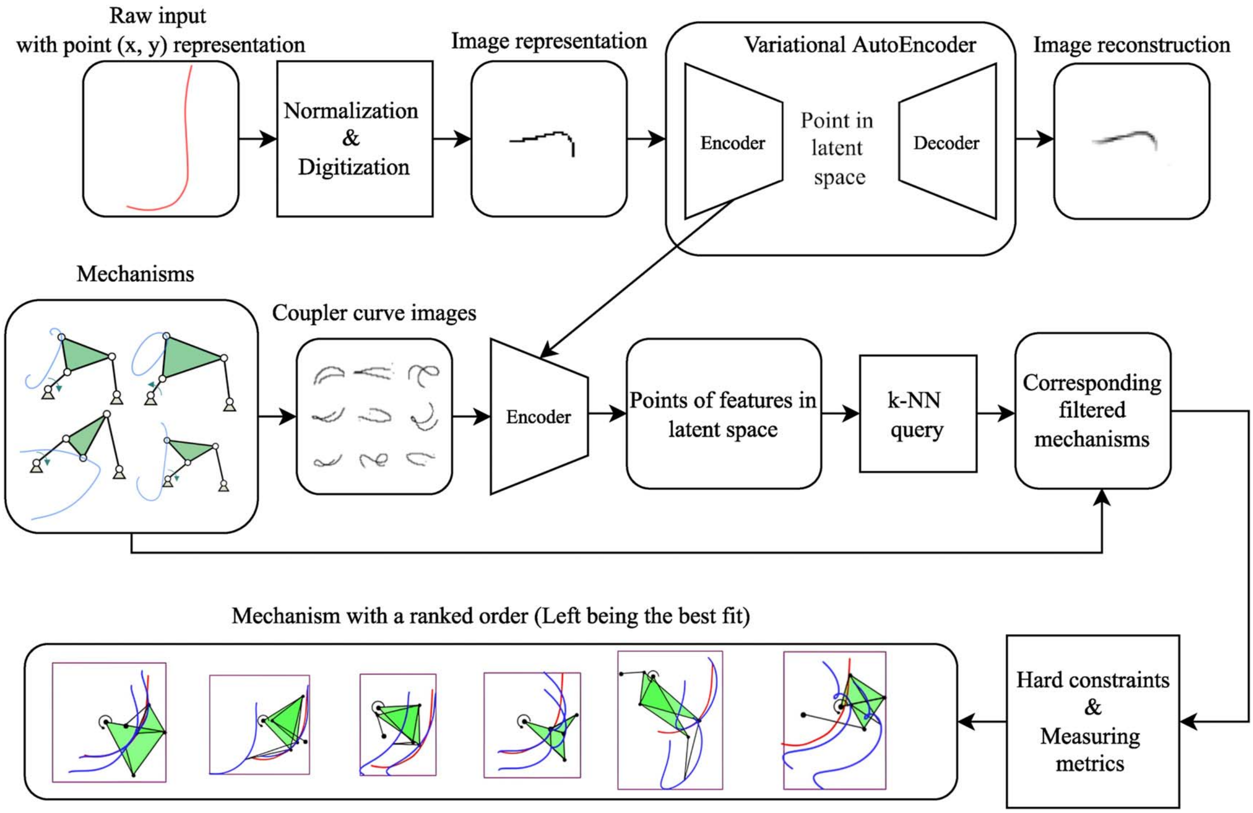

Fig. 2 Design pipeline for coupler curve representation, query, evaluation, and mechanism synthesis

dimensions of links and locations of joints, based on specified motion or path requirements. Input problems are traditionally categorized into path, motion, or function synthesis, as extensively discussed in seminal texts on kinematic synthesis by Norton [15] and Erdman and Sandor [16].

Recent endeavors have seen a growing interest in framing functional tasks in physical therapy and rehabilitation as path or motion generation problems. Various approaches include the design of adjustable single-degree-of-freedom linkage systems for ankle trajectory tracing [17], knee-ankle-foot orthosis using six-bar mechanisms [18], and a 3R open chain for ankle joint trajectory tracking [19]. Machine learning has also been employed for predicting rehabilitation gait [20]. Alternatively, these tasks can be viewed as motion generation problems, leading to the design of cam linkages for lower-limb rehabilitation [21], clustering-based machine-learning techniques for upper-limb rehabilitation [22], and linkage mechanisms with elastic components for creating desired gait patterns [23]. Conditional generative adversarial networks have been explored for designing six-bar mechanisms [24].

Ge et al. [25–27] and Deshpande and Purwar [28,29] proposed a kinematic mapping approach for simultaneous type and dimension synthesis of mechanisms for motion generation. Although implemented in a mechanism design tool [30], this approach often yields impractical mechanisms with circuit, branch, and order defects and provides a limited number of solutions [31]. For example, for a five-position motion generation problem, they obtain only up to four constituent dyads of a planar four-bar mechanism [26]. Bottema and Roth [32], Ravani and Roth [33], and McCarthy [34] provide the most comprehensive treatment of the kinematic mapping approach to the motion synthesis problem.

Traditional optimization and Atlas-based methods have been employed for path synthesis [35,36], while Fourier coefficients and artificial neural networks have been used for representing and synthesizing coupler curves [37,38]. Reinforcement learning has been applied to treat straight-line path synthesis as an iterative optimization problem [39].

Recent work by Purwar et al. [40–43] has incorporated machine learning techniques to synthesize defect-free mechanisms and handle uncertain user inputs. The input curve has been modeled as an image to address variable algebraic complexity [44]. Datasets for data-driven methods have been created [45–47], and invariant representations of coupler curves have been proposed [48]. Challenges in deep learning design of robot mechanisms have been outlined [42]. Building upon this foundation, the present paper establishes a design pipeline with quantitative evaluation criteria for generating linkage mechanisms for STS motion under multiple constraints.

3 Overview of Model

Single-DOF mechanisms, sharing identical link ratios, exhibit affinely invariant coupler curves with variations in position, scale, rotation, and reflection. To compactly represent this information in a database, we employ a normalization process that ensures the affinely invariant nature of coupler curves before digitizing and embedding them into an image space. In this section, we provide a comprehensive overview of our normalization and digitization methods. We also discuss the architecture of our VAE and introduce a curve-matching algorithm that addresses the limitations of using only a neural network for comparing two coupler curves. Additionally, we present a summary of the complete design pipeline and the preparation of the dataset.

3.1 Normalization and Digitization. Our normalization and digitization method draws inspiration from the work of Sener and Unel [49], which presents a general-purpose approach for applying various affine transformations to data points, including translation, shearing, rotation, scaling, and reflection. We adapt this method to obtain an affinely invariant representation of coupler curves, referred to as a canonical representation. This involves centering the coupler curves about the origin of a chosen reference frame, isotropically scaling them to a specified image size, rotating them to maximize the non-Gaussianity of the underlying curve data, and normalizing reflections based on the third-order central moment of the data points. We elaborate on each of these processes in the following.

1. Centering: Given a set of N points on a curve denoted by $\mathbf{X_0}_i \in \mathbb{R}^2; i = 1 \ldots N$, their center point is defined as the average of the Cartesian coordinates, represented by $\bar{\mathbf{x}} = [\bar{x}, \bar{y}] = \sum_{i=1}^{N} \mathbf{X_0}_i/N$. The centering process involves aligning the center point to the origin of a chosen reference frame, expressed by Eq. (1), where the first subscript of $\mathbf{X_0}_i$ denotes the processing phase of the points. The centering matrix is denoted as $\mathbf{M_1}$.

$$\mathbf{X_1}_i = \mathbf{M_1}\mathbf{X_0}_i = \begin{bmatrix} 1 & 0 & -\bar{x} \\ 0 & 1 & -\bar{y} \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X_0}_i; \, i = 1 \ldots N \tag{1}$$

2. Uniform scaling: The scaling factor is determined by the point on the coupler curve farthest from the origin. In our method, a rectangular window encapsulates the entire coupler curve, and the scaling ensures that all points fit within the window. If the farthest point $\mathbf{x_f}$ is at a distance d from the center, and the half-window size is one unit, the point distribution is scaled by $1/d$ as represented in Eq. (2).

$$\mathbf{X}_{2i} = \mathbf{M}_2\mathbf{X}_{1i} = \begin{bmatrix} 1/d & 0 & 0 \\ 0 & 1/d & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{X}_{1i}; \, i = 1 \ldots N \tag{2}$$

3. Rotation invariance: To achieve rotation invariance, an orthogonal matrix $\mathbf{M_3}$ is computed as shown in Eq. (3) using independent component analysis (ICA) [50].

$$\mathbf{X}_{3i} = \mathbf{M_3}\mathbf{X}_{2i} = \begin{bmatrix} M_3 & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}_{2i} \tag{3}$$

ICA is a statistical technique employed for identifying independent directions in feature space. In our case, ICA serves to estimate the unmixing matrix, which is the desired orthogonal matrix. We utilize the FastICA [51] in an iterative approach to compute the transformation matrix $\mathbf{M_3}$. Algorithm 1 outlines the detailed steps of this process. The algorithm includes functions such as nl_function() for component-wise transformation using a nonlinear function (we chose x^3), mean() for computing the average of each column, broadcast() for duplicating rows, Grahm_Schmidt() for performing Grahm–Schmidt orthogonalization, and expand() for homogenizing the matrix. The loop continues until the components of the resulting matrix $\mathbf{M_3}$ converge within a tolerance value ε (set to 10^{-4}) over a maximum of K iterations (set to 200). The final unmixing matrix $\mathbf{M}_3$ is then applied to the data points as shown in Eq. (3).

Algorithm 1 Non-Gaussianity maximization

Input: Centered, uniformly scaled, and $(N, 2)$-sized data X_2, max iteration K, and numerical tolerance ε
Output: The unmixing matrix M_3
Initialize a new matrix M_3 of size (2, 2) with random values
for i in 1 to K:
  $M_{3prev} \leftarrow M_3$ ⊳ Make copy
  $nl\_response \leftarrow$ nl_function$(X_2 \cdot M_3^T)$ ⊳ dot(·) indicates matrix multiplication
  $nl\_average \leftarrow (nl\_response^T \cdot X_2)/N$
  $nl\_prime\_average \leftarrow$ mean(nl_prime($nl\_response$))
  $nl\_prime\_expand \leftarrow$ broadcast($nl\_prime\_average$)
    ⊳ Broadcast $nl\_prime\_average$ from (1,2) to (2,2)
  $M_3 \leftarrow nl\_average - (nl\_prime\_expand \cdot M_3)$
  $M_3 \leftarrow$ Grahm_Schmidt(M_3)
  if all(abs$(M_3 - M_{3prev})) < \varepsilon$: ⊳ Check convergence
    break
return expand(M_3) ⊳ Homogenize

4. Reflection invariance: While mechanisms and their coupler curves may undergo reflection, the absence of normalization can introduce redundancy in the database. The fourth transformation matrix $\mathbf{M}_4$ in Eq. (4), addressing reflection, relies on the absolute values of the third-order central moment of $\mathbf{X}_3$ [49].

$$\mathbf{X}_{4i} = \mathbf{M}_4\mathbf{X}_{3i} \tag{4}$$

Here the matrix $\mathbf{M}_4$ is defined based on the signum function *sgn* and the calculated moments as detailed in the following:

$$\mathbf{M}_4 = \begin{cases} \begin{bmatrix} 0 & \text{sgn}(m_{2,1}) & 0 \\ \text{sgn}(m_{1,2}) & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \text{if} |m_{1,2}| > |m_{2,1}| \\ \begin{bmatrix} \text{sgn}(m_{1,2}) & 0 & 0 \\ 0 & \text{sgn}(m_{2,1}) & 0 \\ 0 & 0 & 1 \end{bmatrix}, & \text{otherwise} \end{cases}$$

For a 2D curve, the p, q-order moment is defined as

$$m_{p,q} = \frac{1}{N}\sum_{i=1}^{N} x_i^p y_i^q \tag{5}$$

Compared to the normalization method proposed by Deshpande and Purwar [44] for four-bar coupler curves, our approach ensures consistent orientation and incorporates reflection invariance. Their method aligns the principal axis of the largest eigenvalue with the x-axis, which can result in different alignments depending on computing architecture or the initial orientation of the coupler curve. Additionally, their method does not account for mirroring, whereas our approach accommodates mirrored coupler curves achievable through mirrored mechanisms. Figure 3 illustrates the disparity between the two methods. The overall transformation matrix $\mathbf{M}$ is given by $\mathbf{M_4M_3M_2M_1}$.

5. Digitization: Finally, the planar points are assigned to 64-by-64 equally divided bins, resulting in a 64×64 pixel image of the coupler curve. Figure 3(*d*) displays the digitization outcome for the curves depicted in Fig. 3(*c*). The curve image serves as input to the neural network architecture, generating a corresponding descriptor for database search.

3.2 Neural Network Architecture.

In this paper, we employ a convolutional neural network (CNN)-VAE [11,52,53], which encodes a probabilistic distribution of coupler curves and decodes to generate new curves. The encoding phase involves feeding images of coupler curves to the CNN-VAE, obtaining their latent representations as features for the input path. The hyperparameters for our CNN-VAE network are detailed in Table 1 for the encoder/recognition part and Table 2 for the decoder/generative part.

The loss function used for training the machine learning model is presented in Eq. (6), where L_{VAE} is the sum of the reconstruction loss L_{Rec} and the Kullback–Leibler (KL)-divergence regularizer L_{Reg}. The reconstruction loss is implemented using the binary cross entropy (BCE) function, and the regularization term endows the latent space with a Gaussian distribution. The BCE losses in the training batch of $n = 400$ images are summed up, where X represents the input and ground truth, and $\hat{X}$ is the predicted image from the VAE model. Further notation includes $X_i^{(jk)}$ denoting the grayscale value at pixel location (j, k) of the ith sample image and d representing the number of latent space dimensions (set to 50 in this implementation). Moreover, $\sigma_i^{(j)^2}$ and $\mu^{(j)^2}$ represent the mean and standard deviation, respectively of the squared jth latent dimension value for the ith sample image.

$$L_{\text{VAE}} = L_{\text{Rec}} + L_{\text{Reg}} \tag{6}$$

where

$$L_{\text{Rec}} = -\sum_{i=1}^{n}\sum_{j=1}^{64}\sum_{k=1}^{64} (X_i^{(jk)} \log(\hat{X}_i^{(jk)}) + (1 - X_i^{(jk)}) \log(1 - \hat{X}_i^{(jk)}))$$

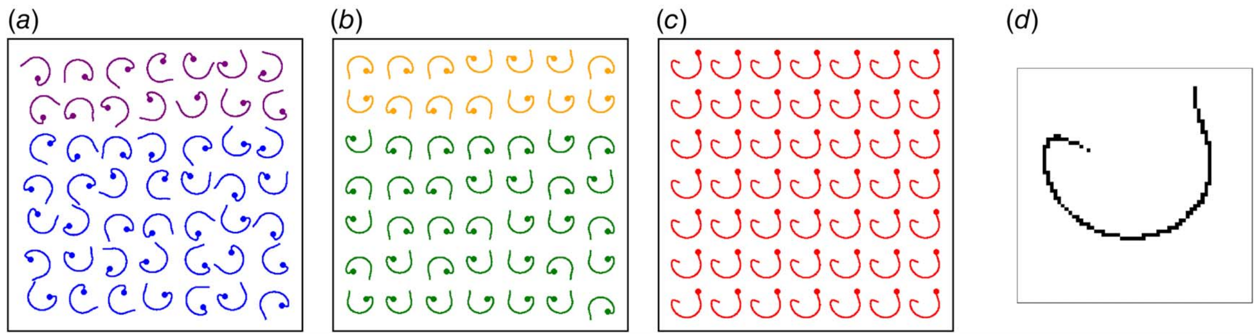

Fig. 3 Part figure (*a*) shows a set of randomly rotated and mirrored coupler curves. The first row shows the curves that are rotated incrementally counterclockwise from left to right, while the curves in the second row are the corresponding reflected versions of the ones in the first row. Then, in the remaining five rows, each curve undergoes a random rotation, translation, scaling, and reflection. For the display purpose, we have plotted them in almost the same size after scale invariance. The corresponding normalization results from Deshpande and Purwar [44] are shown in (*b*). Here, there are four different results because reflection brings two variants, and these two variants are rotated such that their first principal axis is aligned with the *x*-axis. Furthermore, the alignment gives two different results because the alignment can be along the axis's positive direction or the negative direction. Therefore, there are four (2 × 2) different results for these initial curves. Comparatively, the new method presented in this paper has only one normalization result, regardless of the initial orientation or reflection, as shown in (*a*). Finally, the curve is digitized, and (*d*) shows the digitized result from (c).

Table 1 Encoder/Recognition network architecture

Layer	Filter count	Filter size	Stride	Output shape	Activation
Input	—	—	—	(64, 64, 1)	—
Convolution 1	32	(11, 11)	(1,1)	(64, 64, 32)	ReLU
Max pool 1	—	(2, 2)	(2, 2)	(32, 32, 32)	—
Convolution 2	64	(5, 5)	(1, 1)	(32, 32, 64)	ReLU
Max pool 2	–	(2,2)	(2, 2)	(16, 16, 64)	—
Convolution 3	128	(3, 3)	(1, 1)	(16, 16, 128)	ReLU
Max pool 3	—	(2, 2)	(2, 2)	(8, 8, 128)	—
Flatten 1	—	—	—	8192	—
Fully connected 1 (μ)	—	—	—	50	—
Fully connected 2 (σ)	—	—	—	50	(exponential)

Table 2 Decoder/Generative network architecture

Layer	Filter count	Filter size	Stride	Output shape	Activation
Input	—	—	—	50	—
Fully connected	—	—	—	1024	ReLU
Reshape	—	—	—	(16, 8, 8)	—
Transpose convolution 1	128	(3, 3)	(2, 2)	(16, 16, 128)	ReLU
Transpose convolution 2	64	(5, 5)	(2, 2)	(32, 32, 64)	ReLU
Transpose convolution 3	32	(11, 11)	(1, 1)	(64, 64, 32)	None
Transpose convolution 4	1	(3, 3)	(1,1)	(64, 64, 1)	None

and

$$L_{\text{Reg}} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{d}\left(\sigma_i^{(j)^2} + \mu_i^{(j)^2} - \log\left(\sigma_i^{(j)^2}\right) - 1\right)$$

The trained neural network aims to faithfully reproduce the input coupler curve image and ensures that the latent space has a multi-Gaussian probabilistic distribution, allowing for the generation of new and useful coupler curves. Figure 4 visually depicts the training process outcomes after every 30 epochs for an example coupler curve.

3.3 Curve Matching. The CNN-VAE architecture establishes a Gaussian structure in the latent space, implying that (1) similar curve images correspond to similar latent representations, and (2) the similarity between coupler curves is reflected in the numerical values of their respective latent vectors. Although in most instances, an L^2 norm of the difference between two latent vectors is a viable metric for assessing the dissimilarity between two coupler curves, challenges arise when a discontinuity separates the two latent vectors. While the literature in machine learning suggests that the latent space could possess a manifold structure, characterized by smoothness and local convexity [54], systematic exploration of this concept remains an open question. Figure 5 illustrates this issue. The figure demonstrates that the reconstructed coupler curve image undergoes significant changes in certain dimensions. For instance, an identical magnitude of change in the 44th and 45th dimensions of the latent representation results in vastly different images, while minimal changes are observed in the 5th, 25th, and 35th dimensions. Moreover, reducing the dimension of the latent space may yield a blurrier image, indicating that the features of the input curve are not accurately captured. This example underscores the necessity of introducing an alternative metric, beyond a simplistic L^2 norm of the

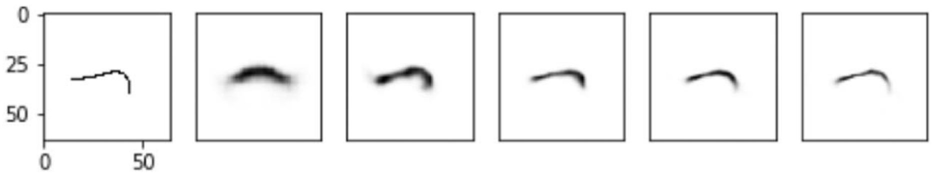

Fig. 4 These are the training results every 30 epochs. The change of output from the VAE model shows the neural network learns to reconstruct a similar image gradually.

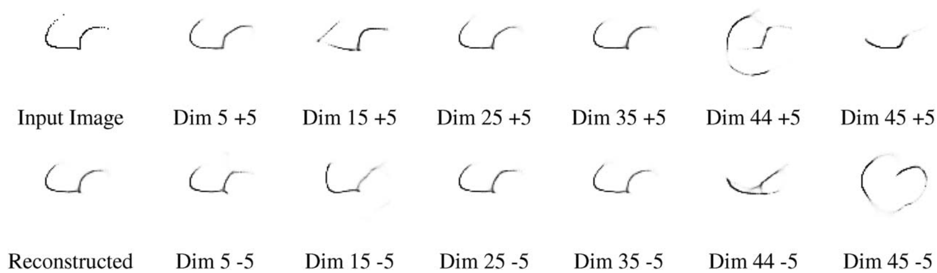

Fig. 5 An input image is sent to the VAE and its 50-dimensional latent representation is computed. The image can be reproduced using the decoder part of the VAE. However, changing the latent vector in different dimensions results in changing the image in different ways. This set of images shows the same change amount in some dimensions (dimensions 5, 15, 25, 35, 44, and 45). While the changes in dimension 25 and 35 result in almost the same image as the reconstructed image, the changes in dimension 44 and 45 result in drastically different images.

difference in latent representations, to effectively compare the similarity of coupler curves.

Our design pipeline generates mechanisms and their associated coupler curves for comparison with the input path. However, due to potential differences in length and time sequence between the coupler curves and the input path, a straightforward direct comparison becomes impractical. To address this challenge, we introduce a sequence-matching algorithm. The concept of curve sequence matching involves associating each point of the input curve with a point in the coupler curve in a non-decreasing order.

It is crucial to note that the lengths of the two sequences need not be identical. To achieve this, we employ the dynamic time warping (DTW) algorithm [55]. DTW identifies the sequential mapping that minimizes the sum of differences between matched points in two given sequences. Comparatively, chamfer distance, a conventional method that computes the smallest sum of two sets of points, cannot detect the order difference. By dynamically programming the alignment process, DTW determines an optimal path that minimizes dissimilarity, offering a robust measure of similarity between sequences. The fundamental approach involves computing a cumulative cost matrix, initialized with the first element set to zero and the remaining elements in the first row and first column set to infinity. This matrix is filled using dynamic programming, considering the minimum cumulative cost from three possible alignments at each step and incorporating the distance between corresponding points in the two sequences, computed using the L^2 norm.

The cumulative cost in the bottom-right corner of the matrix signifies the dissimilarity between the two sequences, while the aligned pairs illustrate the optimal alignment. Algorithm 2 provides a comprehensive depiction of the entire DTW process.

Algorithm 2 Dynamic time warping

Input: Curve sequences P and Q
Output: r_{DTW} ▷ Dissimilarity measure
$DTWtable \leftarrow \text{ones}(\text{len}(P)+1, \text{len}(Q)+1) * N$ ▷ Set N to an arbitrarily large number
$DTWtable[0, 0] = 0$
for i in 1tolen(P):
  for j in 1tolen(Q):
    $prev = \min([DTWtable[i-1, j], DTWtable[i, j-1], DTWtable[i-1, j-1]])$
    $DTWtable[i, j] = prev + \text{diff}(P[i-1], Q[j-1])$ ▷ diff: L^2 norm
  return $DTWtable[\text{len}(P), \text{len}(Q)]$

3.4 Mechanism Database Preparation. In this section, we first discuss how we prepare a dataset of planar four-bar mechanisms, which could trace both open and closed curves with different number of points on them. We use a grid-based generation of planar four-bar mechanisms where pivots and coupler points of four-bar mechanisms are located randomly at discrete locations of a grid. Following Deshpande and Purwar [44], the size of the grid was chosen to be 7×7. A four-bar linkage mechanism (Fig. 6) can be described as a sequence of five points collected in a vector of joints **c** as

$$\mathbf{c} = (j_0, j_1, j_2, j_3, j_4)$$

where a joint is represented as $j_i = (x_i, y_i)$.

Without any loss of generality, we fix j_0 at (0, 0) and randomly locate j_1, j_2, and j_3 on the grid points; see Fig. 7 as an example. We place the joint j_4 on a coupler body frame attached to the link defined by j_1 and j_3. This is done to ensure that we are not wasting computational resources locating coupler points on invalid mechanisms. Figure 8 shows a few coupler curves from the same mechanism defined by (j_0, j_1, j_2, j_3) while varying j_4. We follow a three-step process to create the mechanisms: first, we

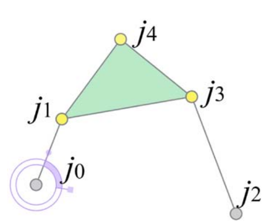

Fig. 6 Four-bar mechanism topology for four-bar mechanisms saved in the database

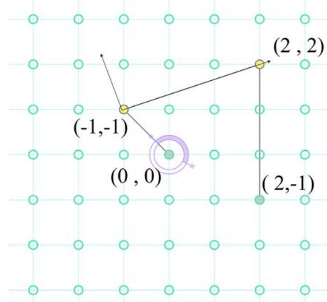

Fig. 7 Illustration on the combinations. The joints, except the one at (0, 0), can be on any point denoted with a circle as long as the combination of these joints leads to a valid mechanism.

generate mechanisms based on the combinations of (j_0, j_1, j_2, j_3), and obtain unique mechanisms according to link-length ratio combinations. Second, we simulate all of these mechanisms and compute the transformation matrix from the coupler body frame to the fixed frame. Third, we compute the position of j_4 by multiplying the position of j_4 in the moving frame with the transformation matrix.

For the first step, the values of x_i and y_i are set to integral values in $[-3, 3]$. Therefore, the number of combinations is $7^6 = 117,649$, where a lot of duplication exists. Thus, we use the ratio of link lengths to filter out repeated or undesired combinations according to the rules as follows:

(1) Remove all combinations where j_i and j_k are co-located $(i \neq k)$. Such combinations would form a structure and cannot move at all when one of them is a moving joint and the other is a fixed joint, or it becomes a rotating rigid body when two coincident points are fixed joints.
(2) We denote the ground link to be the reference link l_0, l_1 to be the link connecting j_0 and j_1, l_2 to be the link connecting j_1 and j_3, and l_3 to be the link connecting j_2 and j_3. Then, we check all combinations of $(l_1/l_0, l_2/l_0, l_3/l_0)$. If the link ratios are the same, the mechanism is not added to the database.

These rules result in 4957 unique link-length ratio combinations, 1407 of which can do full rotations according to the Grashof condition.

Then, for the second step, we use our recently proposed planar N-bar simulation algorithm [56] to simulate mechanisms and create

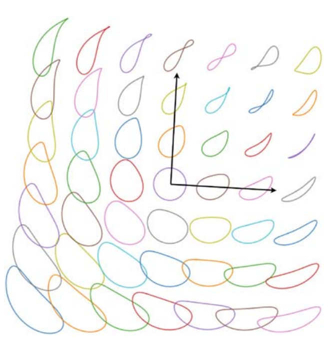

Fig. 8 This figure shows a distribution of coupler curves generated when the coupler point is varied by a unit distance in both the *x*- and *y*- directions. The four-bar mechanism in this figure is defined by $(j_0, j_1, j_2, j_3) = ((0, 0), (-1, -1), (-3, -3), (-2, 3))$ labeled in Fig. 6. The circle in the center of this figure corresponds to is produced when $^B p_4 = [0, 0, 1]^T$, i.e., when the joint j_4 coincides with j_1. As $^B p_4$ changes, different coupler curves are produced. For example, the coupler curve on the immediate right of the circle corresponds to $^B p_4 = [1, 0, 1]^T$. The coupler curves overlap each other in the fixed frame; however, for visual clarity, we have drawn them here spaced out.

coupler curves. We remove mechanisms, where a link adjacent to the ground link can not rotate more than 20 deg. Such mechanisms produce coupler curves of limited length to be practically useful. After removing such mechanisms, the number of mechanisms shrinks to 4909. The transformation matrix can be expressed as

$$ {}^F_B T = \begin{bmatrix} \cos\alpha & -\sin\alpha & t_x \\ \sin\alpha & \cos\alpha & t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{7} $$

where F stands for the ground/fixed frame, B stands for the moving body frame, and α is the angle between the positive x-axis and the link l_2. Here, t_x and t_y are x_1 and y_1, respectively since the origin of the moving frame coordinate system is located at j_1. Then, finally, the coupler curves can be computed with the expression as follows:

$$ {}^F p_4 = {}^F_B T\, {}^B p_4 \tag{8} $$

where $^B p_4 = [x_4, y_4, 1]^T$. If j_4 co-locates with j_1 or j_3, coupler curves are circles. We do not want circles to overwhelm our dataset, so we disallow locating j_4 at one of the moving pivots. With that restriction, the total number of combinations/mechanisms in the database is $4909 \times 47 = 230{,}723$.

We use 200,000 randomly picked coupler curves from the database as the training set, which enables our CNN-VAE neural network to represent different types of coupler curves.

3.5 Design Pipeline. Our design pipeline begins with the training of a CNN-VAE, where we minimize the reconstruction and KL-divergence loss. The trained encoder is then employed to index the four-bar mechanisms in the database. Coupler curves are inputted into the encoder network, and the mean vectors (μ) representing the distribution of the curve are saved for subsequent queries. When a new coupler curve is introduced to the encoder, its mean vector is compared to those in the database using k-nearest-neighbor search. The k-nearest mechanisms, based on the shortest distances using L^2 norm, are selected as candidate solutions.

These candidates are subsequently transformed to align with the size and orientation of the raw input curve, determining their positions in the plane. An algorithm then combines two cognates of the four-bar mechanisms to generate a six-bar mechanism for achieving parallel motion. We further filter out six-bar mechanisms that fail to satisfy the design specifications. Finally, the DTW algorithm is applied to rank the mechanisms based on spatial positioning and point sequence matching fitness. An overview of this pipeline is presented in Fig. 2.

The essential data saved include one positional state of each four-bar mechanism, the topological representation of the mechanism, and their descriptors, i.e., the mean vector (μ) for each mechanism and the neural network parameters.

4 Sit-to-Stand Mechanism Synthesis

In this section, we leverage our design pipeline to generate a plethora of conceptual design solutions for planar six-bar mechanisms, aligning with the natural kinematics of the sit-to-stand movement. We present the design specifications for the STS mechanism, outline the generation of six-bar mechanisms with parallel motions, introduce performance metrics, and finally, present and discuss the results.

4.1 Design Specifications. Research has shown that, despite variations in the specific motion paths adopted by individual users during STS maneuvers, the overarching movement pattern remains consistent [57]. Additionally, Kamnik et al. [58] have asserted that, in the context of STS motion, the upper body is restricted to movement within the sagittal plane while undergoing orientation changes in the anteroposterior direction. Consequently, it is reasonable to infer that a substantial proportion of individuals

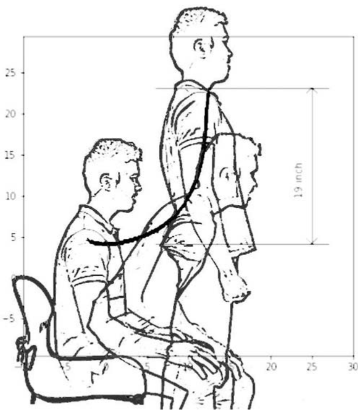

Fig. 9 This paper's goal is to find several mechanism concepts that can generate a constant orientation motion along the STS J-curve for the shoulder position

unable to execute a standing-up motion possess the ability to control their upper body orientation. Carr [59] has advocated for the limitation of arm movement during the STS motion, particularly for the elderly, emphasizing that the positioning of the arms influences the body's center of mass. Research has also indicated that when the arms are unrestrained, there is a notable increase in undesirable ankle angular displacement. Considering this, we impose the following requirements on the chosen mechanisms:

(1) Perform the sit-to-stand motion by following the natural trajectory of the shoulder joint, known as the J-curve (see Fig. 9).
(2) Provide a floating link along the chosen path, maintaining its orientation.
(3) Achieve a vertical displacement of the shoulder joint by at least 19 in., accommodating individuals of average height between 5 ft and 6 ft, approximately 95% of the population.

The parallel motion requirement is based on the assumption that users will rest their hands on a horizontal support bar, necessitating constant orientation during the STS motion. As planar four-bar mechanisms cannot generate this specific motion, we explore potential six-bar mechanisms to fulfill this requirement. Additionally, spatial constraints such as the mechanism's envelope, link ratio, and overall dimensions are considered. These include:

(1) The entire mechanism should fit within a 40×40 in. rectangle, ensuring integration in a compact device.
(2) The length ratio of the largest link to the smallest link should not exceed 4.5 for optimal transmission angle.

Figure 9 illustrates the size of the 40×40 in. window, while Fig. 10 depicts the J-curve and its preprocessed result. This curve

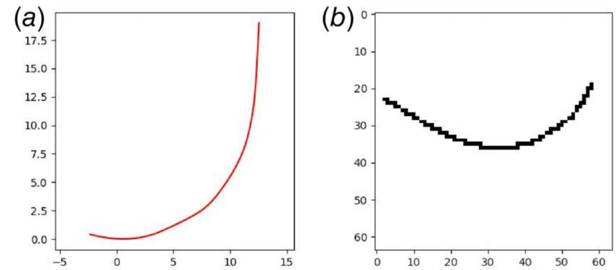

Fig. 10 Part figure (*a*) is the J-shaped curve to be synthesized. The curve is captured from the image in Fig. 9 and (*b*) is the preprocessed result using the techniques in Sec. 3.1.

is encoded in the latent space, enabling a nearest-neighbor search to produce four-bar mechanisms that trace the curve. The subsequent section outlines an algorithmic approach for computing the cognates of four-bar mechanisms and combining them to create parallel motion six-bar mechanisms.

4.2 Generating Parallel Motion Six-Bar Mechanisms. According to Roberts–Chebyschev theorem [60], three four-bar linkages called curve cognates [61] can trace the same coupler curve. A six-bar Watt I linkage can be generated by combining two of the three cognates of a four-bar linkage. This six-bar mechanism can perform a parallel motion along the given curve. Most texts on the kinematics of machines, such as by Uicker et al. [62] present a graphical approach to finding cognates and forming six-bar mechanisms. Sherman et al. [14] have presented a method based on kinematic equations to construct cognates of four-bar and six-bar linkages by permuting link rotations. Here, we present a simple algorithm meant for fast implementation. Figure 11 shows three example J-curve cognates.

The first step in building the six-bar linkage mechanism is to find the cognates for the four-bar linkage mechanism inside it. Algorithm 3 shows the process of finding two cognates from a four-bar mechanism, where *findTriangleAngles*(a, b, c) is a function that computes the inner angle formed by three vertices (a, b, c). In the algorithm, j_1, j_3, and j_4 (see Fig. 6) correspond to points a, b, and c, respectively. The function *findPosofC*$(a, b, \alpha, \beta, \gamma)$ computes the third vertex c according to the two points and the three angles α, β, and γ. The function *computeSlope*(p_1, p_2) takes two points as input to compute the slope of the line joining p_1 to p_2. In the function *findPosofC*$(a, b, \alpha, \beta, \gamma)$, two values satisfy the constraints, while only one is suitable for the cognate.

As shown in Fig. 11(*d*), the basic idea behind this algorithm is that (1) the topology constructed by the three four-bar cognates has four similar triangles, which are $\triangle j_1j_3j_4$, $\triangle j_0j_2p_5$, $\triangle j_4p_3p_4$, and $\triangle p_1j_4p_2$, and (2) the topology has three parallelograms,

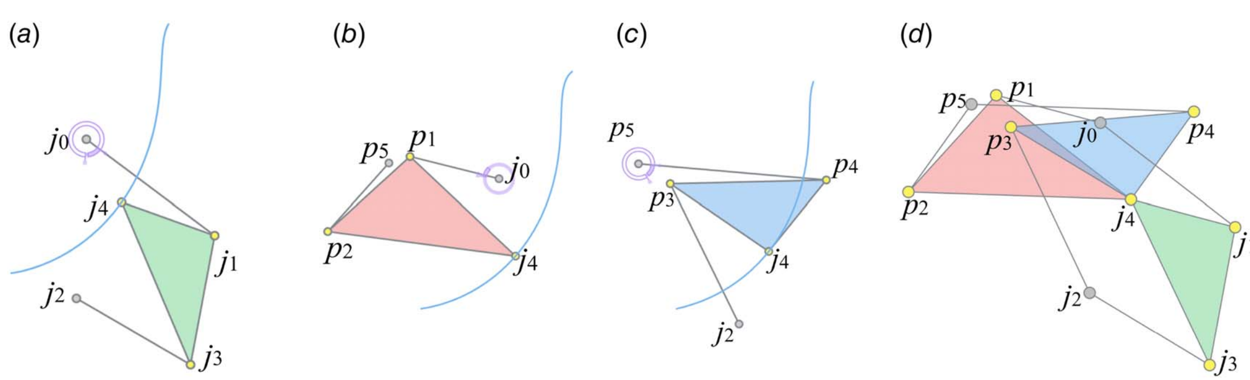

Fig. 11 Part figures (*a*), (*b*), and (*c*) show three four-bar cognates, whose coupler point j_4 share the same trajectory. When one of the three cognates is known, the other two cognates can be computed according to Algorithm 3, which is based on the geometry shown in (*d*).

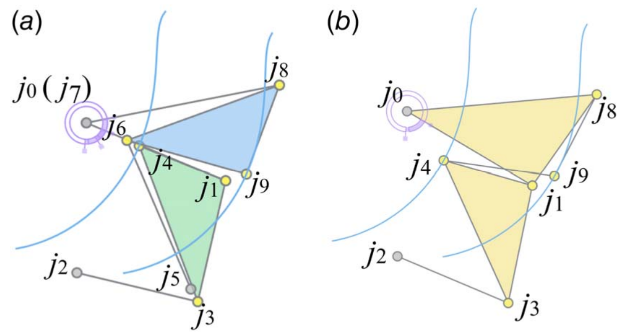

Fig. 12 These two figures show the generation of a six-bar with parallel motion. First, we take two cognates (Figs. 11(*a*) and 11(*c*)) output from Algorithm 3, and make two fixed joints coincide. Then, it takes all five joints in the first four-bar and two joints of the second four-bar to create a six-bar linkage. The resultant six-bar linkage can generate a parallel motion because its two joints trace two parallel curves: (*a*) two joints in the two four-bars are coincided and (*b*) the two four-bars are merged into a six-bar.

namely $j_0j_1j_4p_1$, $j_3j_2p_3j_4$, and $p_2j_4p_4p_5$. Specifically, p_1 and p_3 are computed using the parallel vectors because the other three nodes in their corresponding parallelograms $j_0j_1j_4p_1$ and $j_3j_2p_3j_4$ are known, respectively. Next, p_2, p_4, and p_5 are computed using the four similar triangles. While each *findPosofC*$(a, b, \alpha, \beta, \gamma)$ gives two results, only one combination of j_4, p_2, p_4, and p_5 can create the third parallelogram which is $p_2j_4p_4p_5$. Hence, there are two pairs of parallel vectors, i.e., $j_4\vec{p_2} \parallel p_5\vec{p_4}$ and $p_2\vec{p_5} \parallel p_4\vec{j_4}$, and these two parallel relationships are used to find the correct combination of p_2, p_4, and p_5.

Algorithm 3 Find cognate combinations

Input: The combination of joints j_0, j_1, j_2, j_3, j_4 for a four-bar linkage
Output: Two cognates with the same topology as input
$j_0, j_1, j_2, j_3, j_4 \leftarrow ((x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4))$
$\alpha, \beta, \gamma \leftarrow \mathit{findTriangleAngles}(j_1, j_3, j_4)$
$p_1 \leftarrow j_4 - j_1 + j_0$
$p_3 \leftarrow j_4 - j_3 + j_2$
for p_5 in $\mathit{findPosofC}(j_0, j_2, \alpha, \beta, \gamma)$:
  for p_4 in $\mathit{findPosofC}(j_4, p_3, \alpha, \beta, \gamma)$:
    for p_2 in $\mathit{findPosofC}(p_1, j_4, \alpha, \beta, \gamma)$:
      $k1 = \mathit{computeSlope}(p_5, p_4)$
      $k2 = \mathit{computeSlope}(p_4, j_4)$
      $k3 = \mathit{computeSlope}(j_4, p_2)$
      $k4 = \mathit{computeSlope}(p_2, p_5)$
      if $(abs(k1 - k3)) + (abs(k2 - k4)) < 1e - 3$:
        return $[(j_0, p_1, p_5, p_2, j_4), (j_2, p_3, p_5, p_4, j_4)]$

Furthermore, only two of the three cognates can achieve the desired parallel motion along the J-curve, which are given by $(j_0, j_1, j_2, j_3, j_4)$ and $(j_2, p_3, p_5, p_4, j_4)$ output by the above algorithm. These two cognates are shown in Figs. 11(*a*) and 11(*c*), respectively. A six-bar mechanism is created based on Algorithm 4, where the returned ordered combination of joints corresponds to the right image in Fig. 12.

Algorithm 4 Generate parallel six-bar mechanism

Input: The combination of joints j_0, j_1, j_2, j_3, j_4 for four-bar linkage
Output: Six-bar linkage mechanism parameter
Get 2^{nd} cognate parameters from Algorithm 3 as $cog2$
$j_0, j_1, j_2, j_3, j_4 \leftarrow ((x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4))$
$j_5, j_6, j_7, j_8, j_9 \leftarrow cog2$
$j_8 \leftarrow j_8 + j_0 - j_7$
$j_9 \leftarrow j_9 + j_0 - j_7$
return $(j_0, j_1, j_2, j_3, j_4, j_8, j_9)$

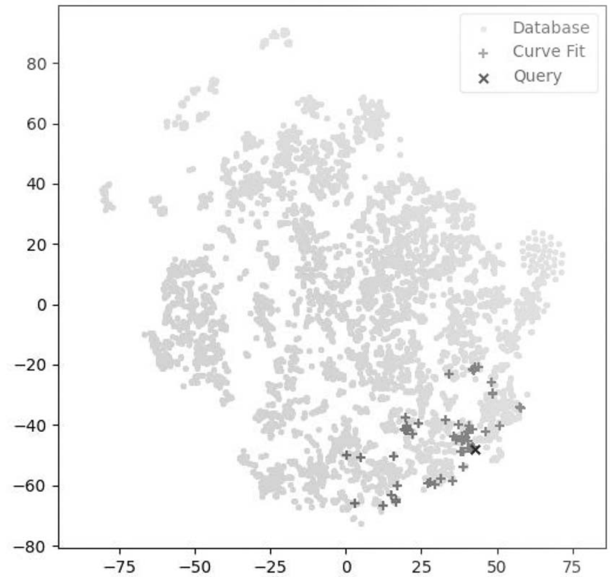

Fig. 13 T-SNE plot for the latent representation distribution of curves. Here, 5000 coupler curves are randomly picked from the database to illustrate their latent representation distribution. The "+" correspond to the mechanisms that satisfy all the constraints specified in Sec. 4.1, which is a subset of the mechanism set shown with gray dots. The "x" sign point corresponds to the image representation of the input coupler curve.

Table 3 These are the statistics for the 114 found mechanisms

	s_1	s_2	s_3	*score*
Mean	1.00	1.01	1.27	3.28
Std. Dev.	0.15	0.17	0.40	0.41
Max	1.54	1.42	1.61	4.08
Min	0.71	0.78	0.01	1.78

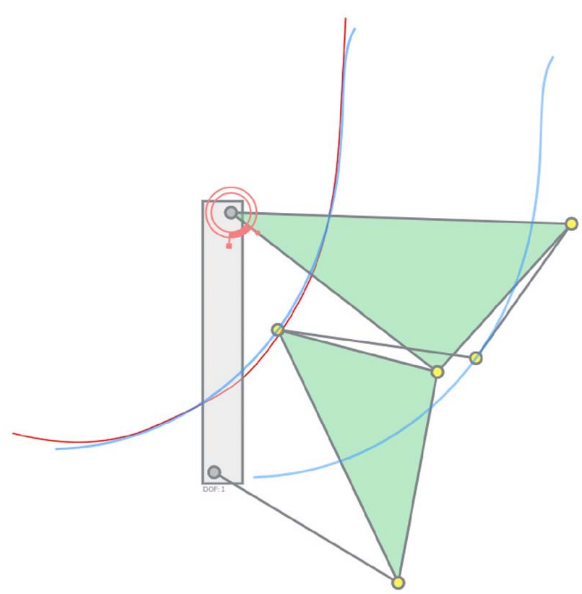

Fig. 14 This is the best six-bar parallel mechanism found through the pipeline. Specifically, the two parallel coupler curves are drawn, and the STS curve is fitted by the left coupler curve.

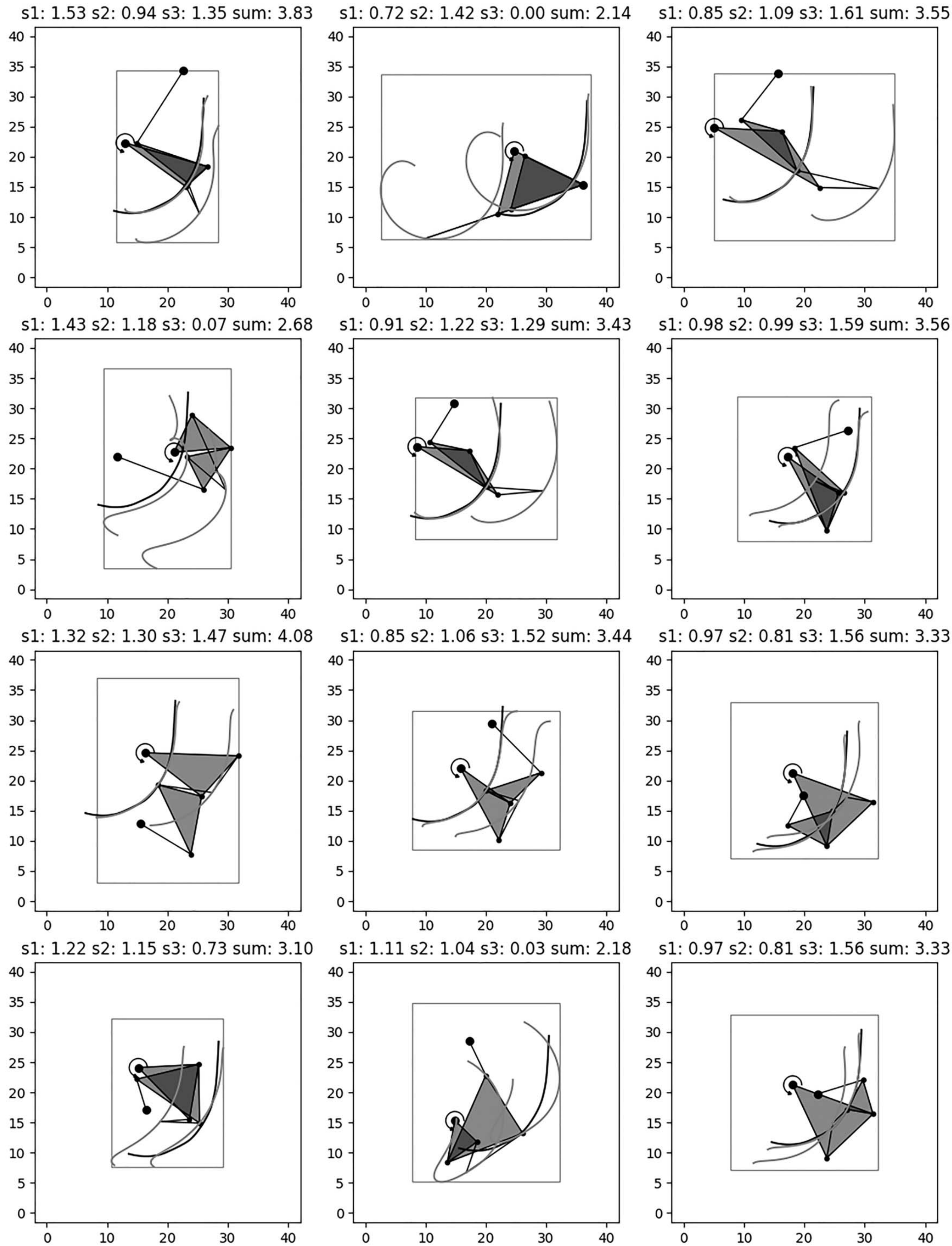

Fig. 15 The labeled length unit is an inch for all the 12 automatically selected six-bar Watt I mechanisms. Each column shows mechanisms sorted according to different metrics: the first column for the first metric r_{rect}, the second for r_{length}, and the third for r_{DTW}. The scores computed according to Sec. 4.3 are displayed on top of each mechanism image. The inner gray rectangle envelopes the mechanism throughout its entire motion. The black curve represents the input J-shaped curve. The two gray curves for each mechanism represent the parallel motion of two coupler curves. For visual clarity, the two coupler links are shaded transparent gray. The black dots with a slightly bigger size represent the fixed joints, and the rotating arrow indicates the actuator joint for the mechanism.

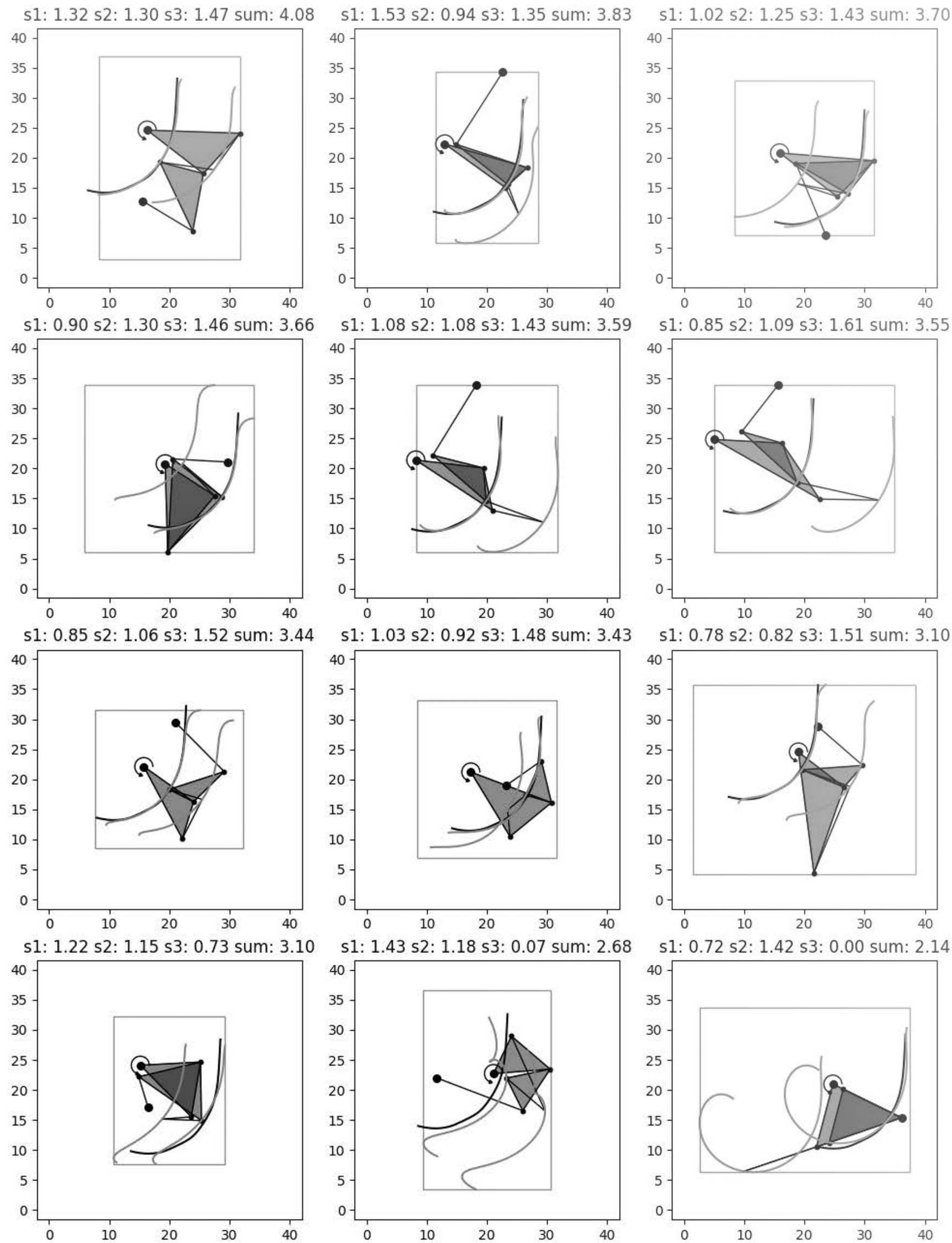

Fig. 16 The labeled length unit is an inch for all the 12 best six-bar Watt I mechanisms, which are ordered with a descending *score* from left-to-right and top-to-bottom. The top-left mechanism receives the highest overall *score* described in Sec. 4.3 out of the 114 mechanisms, and the bottom-right mechanism receives the lowest *score* out of the 12. Each component of the score is also displayed with (*s*1, *s*2, *s*3) on top of each image. The gray rectangle envelopes the mechanism throughout its entire motion. The black curve represents the input J-shaped curve. The two gray curves for each mechanism represent the parallel motion of two coupler curves. For visual clarity, the two coupler links are shaded transparent gray. The black dots with a slightly bigger size represent the fixed joints, and the rotating arrow indicates the actuator joint for the mechanism.

The six-bar mechanism is then translated, scaled, rotated, and reflected according to the curve fitting between the curve traced by j_4 and the input J-curve. The transformation matrix is obtained by multiplying its own normalization matrix and the inverse matrix for the input curve.

4.3 Ranking the Mechanisms That Satisfy All Constraints. The goal of this paper is to find a large number of conceptual design solutions that meet design specifications as presented earlier. These solutions are further evaluated and ranked based on the following three criteria:

(1) Maximize the aspect ratio r_{rect} of the height to width for the mechanism envelope to reduce the bulk in the depth direction.
(2) Minimize the ratio r_{length} of the largest-to-smallest link length. The minimum of this value is 1, i.e., all links have the same length. Notably, the moving triad is considered to be composed of three binary links, and their link lengths are considered in the link-length ratio computation. Smaller link-length ratios lead to a better transmission angle.
(3) Minimize the sequence match error between the generated coupler curve and the input curve.

For the last criterion, maximizing the match of two sequences means minimizing the difference between two curves. Thus, the value of r_{DTW} should be as small as possible. To satisfy this constraint and fit with our goal, the DTW algorithm is run twice for two given point sequences because the proposed DTW Algorithm 2 only detects in one direction, but we match the point sequence regardless of their direction. Then, the minimum of the two mappings is chosen to be r_{DTW} for this curve.

Based on the three criteria, we create a cumulative rank score

$$f_{score} = w_1 \cdot s_1 + w_2 \cdot s_2 + w_3 \cdot s_3 \tag{9}$$

Furthermore, we want s_1, s_2, and s_3 have the same magnitude of impact for the final score when the weights w_1, w_2, and w_3 are all equal to 1. Therefore, the ratings need to be scaled. We choose a simple linear function for r_{rect} and s_1 as shown in Eq. (10). Here, $\bar{r}_{rect}$ is the average rating for aspect ratio in the filtered six-bar mechanisms.

$$s_1 = \frac{r_{rect}}{\bar{r}_{rect}} \tag{10}$$

Next, s_2 should be low when r_{length} value is large. We introduce the power function as shown in Eq. (11). This function is monotonically decreasing, and it is equal to 1 when r_{length} is equal to the average link-length ratio rating $\bar{r}_{length}$.

$$s_2 = 2 \times 0.5^{(r_{length}/\bar{r}_{length})} \tag{11}$$

Similarly, for the DTW score metric, the power function is applied as shown in Eq. (12).

$$s_3 = 2 \times 0.5^{(r_{\mathrm{DTW}}/\bar{r}_{\mathrm{DTW}})} \tag{12}$$

where $\bar{r}_{\mathrm{DTW}}$ is the average DTW matching error. Here, each component is a ratio and thus dimensionless. The weights w_1, w_2, w_3 can be chosen to relatively weigh each of the criteria.

4.4 Results and Discussions. Following a kNN search with k set to 5000 and a subsequent filtering phase that excludes six-bar parallel mechanisms failing to meet the constraints outlined in Sec. 4.1, we obtain a set of 114 six-bar mechanisms. The T-SNE plot [63] in Fig. 13 illustrates the filtered and selected coupler curves. Although most curve representations closely resemble the input representation, a few outliers exhibit significant deviations from the input. This suggests the possibility of obtaining a curve that deviates from the input curve when relying solely on kNN. To address this, a sequence-matching algorithm is implemented as the ranking metric, mitigating this issue. Utilizing three metrics, the best-fitted mechanism out of the 114 is determined. Figure 14 showcases this optimal design. Table 3 provides statistics for each metric, with s_3 exhibiting the highest deviation, indicating that some synthesized mechanisms yield poor curve-fitting results. This underscores the importance of the curve-matching metric in the pipeline.

For comparative analysis, Fig. 15 presents 12 selected mechanism design concepts arranged in descending order based on the three metrics (s_1, s_2, s_3) in each column. In the first column, the top left mechanism design concept is most suitable if the aspect ratio of the mechanism envelope is a priority. For a smaller link ratio or a higher s_2 value, the mechanism in the top middle is preferable. If matching the target trajectory is the primary goal, then the mechanism in the top right column is a better choice. Additionally, we consider the overall *score* value. Figure 16 displays 12 selected mechanism design concepts sorted by the overall *score*, emphasizing the solutions that satisfy all design constraints.

4.5 Computational Complexity and Time. The machine learning-driven computations presented in our work can be generally divided into three parts: data generation, training, and inference. Using an Intel Core i9 CPU with 32 GB memory, it took 2 h to create a database of 230,723 four-bar mechanisms for the STS path. The training time was approximately 5 h. Subsequently, the inference time can be divided into sampling time and filtering to match the design constraints and criteria. This took about 20 min. During inference, the DTW algorithm took 90% of the computation time because the time to compute the difference between two path sequences is comparable to $O(nm)$, where n and m are the two sequence lengths (number of points) respectively, and this algorithm is performed twice (both directions) for all 114 solution candidates. The search in the dataset is done within a minute, using a k-d tree. The time complexity is $O(dk \log n)$, where d is the data dimension (50), k is the k value for k-nearest-neighbor (5000), and n is the number of samples in the dataset (230,723). Other tasks, like the normalization, parallel six-bar mechanism generation, and mechanism rating took less than 1 min altogether.

5 Conclusions and Future Outlook

In this paper, we have presented a novel design pipeline for the conceptual generation of linkage mechanisms for a sit-to-stand motion. We presented a comprehensive data normalization method adapted from the pattern recognition field. Then, we showed that this approach effectively produces many design concepts while simultaneously satisfying design constraints. We also presented several metrics to assess different criteria and a composite score for quantitatively evaluating selected mechanisms. At the end of this process, a mobility assist device can be created by selecting two copies of identical six-bar mechanisms and integrating them in parallel planes of a custom walker frame, which helps users get up from a seated position.

The method presented in this paper is broadly applicable to a dataset of mixed types of linkage mechanisms, such as four-, six-, eight-, and ten-bar mechanisms. However, one of the challenges is the efficient simulation of all the different types of mechanisms of which there can be a large variety. For example, an eight-bar mechanism has 153 unique linkages with a ground-connected actuator that can be constructed from the 71 distinct eight-bar mechanisms [64]. While the number of mechanisms that may need to be simulated depends on the application, we estimate that for the general path synthesis problem, all 153 configurations of eight-bar mechanisms may require more than 1M mechanisms. Once the database generation is complete, there is an additional cost of training NNs, path matching, and filtering. Furthermore, a 1M dataset generally means a longer search time and a larger storage space. Unless there is a direct mapping that can combine the type and dimensions of mechanisms, one can expect to create different NNs for different types. This is where a graph-based VAE for mechanism design can help reduce this complexity. However, this is an unexplored topic. Extension to spatial mechanisms is nontrivial due to a combinatorial explosion of mechanism parameters, which would require even larger data sets. Moreover, there are no general-purpose simulation algorithms for spatial mechanisms yet available that can readily create datasets. Besides, there are yet-to-be-researched fundamental issues related to the representation of the input, such as 3D coupler path and motion, mapping from the input to the mechanism parameters, and effective and efficient NN architectures, which are discussed in a position paper [42].

Acknowledgment

This work has been financially supported by the National Science Foundation under a research grant #1563413 to Stony Brook University. All findings and results presented in this paper are those of the authors and do not represent those of the funding agencies.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Kerr, A., Dall, P., and Ryan, C., 2010, "Frequency of the Sit-to-Stand Task: A Pilot Study of Free-Living Adults," Appl. Ergon., **41**(1), pp. 58–61.
[2] Gell, N. M., Wallace, R. B., LaCroix, A. Z., Mroz, T. M., and Patel, K. V., 2015, "Mobility Device Use in Older Adults and Incidence of Falls and Worry About Falling: Findings From the 2011–2012 National Health and Aging Trends Study," J. Am. Geriatrics Soc., **63**(5), pp. 853–859.
[3] Russell, J. N., Hendershot, G. E., LeClere, F., Howie, L. J., and Adler, M., 1994, "Trends and Differential Use of Assistive Technology Devices: United States, 1994," Adv. Data, **1997**(292), pp. 1–9.
[4] Heidari, O., Wolbrecht, E. T., Perez-Gracia, A., and Yihun, Y. S., 2018, "A Task-Based Design Methodology for Robotic Exoskeletons," J. Rehabil. Assistive Technol. Eng., **5**, p. 2055668318800672.
[5] Torres-Moreno, J., Cruz, N., Álvarez, J., Redondo, J., and Giménez-Fernandez, A., 2022, "An Open-Source Tool for Path Synthesis of Four-Bar Mechanisms," Mech. Mach. Theory, **169**, p. 104604.
[6] Rao, R., Savsani, V., and Vakharia, D., 2012, "Teaching-Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large Scale Problems," Inf. Sci., **183**(1), pp. 1–15.
[7] Cabrera, J., Ortiz, A., Nadal, F., and Castillo, J., 2011, "An Evolutionary Algorithm for Path Synthesis of Mechanisms," Mech. Mach. Theory, **46**(2), pp. 127–141.
[8] Storn, R., and Price, K., 1997, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces," J. Global Optim., **11**(4), pp. 341–359.
[9] Vasiliu, A., and Yannou, B., 2001, "Dimensional Synthesis of Planar Mechanisms Using Neural Networks: Application to Path Generator Linkages," Mech. Mach. Theory, **36**(2), pp. 299–310.
[10] Plecnik, M. M., 2015, "The Kinematic Design of Six-bar Linkages Using Polynomial Homotopy Continuation," Ph.D. thesis, University of California, Irvine, CA.
[11] Kingma, D. P., and Welling, M., 2014, "Auto-Encoding Variational Bayes," Computing Research Repository, abs/1312.6114.
[12] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., 2014, "Generative Adversarial Networks," *Advances in Neural Information Processing Systems*, Vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger, eds., Curran Associates, Inc.
[13] Erdman, A. G., 1981, "Computer-Aided Design of Mechanisms: 1984 and Beyond," Mech. Mach. Theory J., **20**(4), pp. 245–250.
[14] Sherman, S. N., Hauenstein, J. D., and Wampler, C. W., 2021, "A General Method for Constructing Planar Cognate Mechanisms," ASME J. Mech. Rob., **13**(3), p. 031009.
[15] Norton, R., 2011, *Design of Machinery: An Introduction to the Synthesis and Analysis of Mechanisms and Machines*, 5th ed., McGraw Hill, New York.
[16] Erdman, A. G., and Sandor, G. N., 1991, *Mechanism Design: Analysis and Synthesis*, 2nd ed., Vol. 1, Prentice-Hall, Englewood Cliffs, NJ.
[17] Tsuge, B. Y., and McCarthy, J. M., 2016, "An Adjustable Single Degree-of-Freedom System to Guide Natural Walking Movement for Rehabilitation," ASME J. Med. Devices, **10**(4), p. 044501.
[18] Ghosh, S., Robson, N. P., and McCarthy, J. M., 2020, "Kinematic Design and Evaluation of a Six-Bar Knee-Ankle-Foot Orthosis," ASME J. Eng. Sci. Med. Diagn. Ther., **3**(2), p. 021111.
[19] Yu, C., Ye, J., Jia, J., Zhao, X., Chen, Z., and Chen, J., 2021, "Design, Synthesis, and Experiment of Foot-Driven Lower Limb Rehabilitation Mechanisms," ASME J. Mech. Rob., **14**(2), p. 021001.
[20] Loya, A., Deshpande, S., and Purwar, A., 2020, "Machine Learning Driven Individualized Gait Rehabilitation: Classification, Prediction, and Mechanism Design," ASME J. Eng. Sci. Med. Diagn. Ther., **3**(2), p. 021105.
[21] Zhao, P., Zhu, L., Zi, B., and Li, X., 2019, "Design of Planar 1-DOF CAM-Linkages for Lower-Limb Rehabilitation Via Kinematic-Mapping Motion Synthesis Framework," ASME J. Mech. Rob., **11**(4), p. 041006.
[22] Zhao, P., Zhang, Y., Guan, H., Deng, X., and Chen, H., 2021, "Design of a Single-Degree-of-Freedom Immersive Rehabilitation Device for Clustered Upper-Limb Motion," ASME J. Mech. Rob., **13**(3), p. 031006.
[23] Kong, K., and Tomizuka, M., 2012, "Design of a Rehabilitation Device Based on a Mechanical Link System," ASME J. Mech. Rob., **4**(3), p. 035001
[24] Kapsalyamov, A., Hussain, S., Brown, N. A., Goecke, R., Hayat, M., and Jamwal, P. K., 2023, "Synthesis of a Six-Bar Mechanism for Generating Knee and Ankle Motion Trajectories Using Deep Generative Neural Network," Eng. Appl. Artif. Intell., **117**(Part A), p. 105500.
[25] Ge, Q., Purwar, A., Zhao, P., and Deshpande, S., 2017, "A Task-Driven Approach to Unified Synthesis of Planar Four-Bar Linkages Using Algebraic Fitting of a Pencil of G-Manifolds," ASME J. Comput. Inf. Sci. Eng., **17**(3), p. 031011.
[26] Ge, Q. J., Zhao, P., Purwar, A., and Li, X., 2012, "A Novel Approach to Algebraic Fitting of a Pencil of Quadrics for Planar 4R Motion Synthesis," ASME J. Comput. Inf. Sci. Eng., **12**(4), p. 041003.
[27] Zhao, P., Li, X., Purwar, A., and Ge, Q. J., 2016, "A Task-Driven Unified Synthesis of Planar Four-Bar and Six-Bar Linkages With R- and P-Joints for Five-Position Realization," ASME J. Mech. Rob., **8**(6), p. 061003.
[28] Deshpande, S., and Purwar, A., 2017, "A Task-Driven Approach to Optimal Synthesis of Planar Four-Bar Linkages for Extended Burmester Problem," ASME J. Mech. Rob., **9**(6), p. 061005.
[29] Deshpande, S., and Purwar, A., 2022, "A Unified Approach to Dyad and Triad Synthesis for Planar Mechanisms for Motion Generation," *Proceedings of the 2022 USCToMM Symposium on Mechanical Systems and Robotics*, P. Larochelle, J. M. McCarthy, eds., Springer International Publishing, New York, pp. 243–261.
[30] Purwar, A., Deshpande, S., and Ge, Q. J., 2017, "Motiongen: Interactive Design and Editing of Planar Four-Bar Motions Via a Unified Framework for Generating Pose- and Geometric-Constraints," ASME J. Mech. Rob., **9**(2), p. 024504.
[31] Chase, T. R., and Mirth, J. A., 1993, "Circuits and Branches of Single-Degree-of-Freedom Planar Linkages," ASME J. Mech. Des., **115**(2), pp. 223–230.
[32] Bottema, O., and Roth, B., 1979, *Theoretical Kinematics*, Dover Publication Inc., New York.
[33] Ravani, B., and Roth, B., 1983, "Motion Synthesis Using Kinematic Mappings," ASME J. Mech. Transm. Autom. Des., **105**(3), pp. 460–467.
[34] McCarthy, J. M., 1990, *Introduction to Theoretical Kinematics*, The MIT Press, Cambridge, MA.
[35] Wu, J., Ge, Q. J., Gao, F., and Guo, W. Z., 2011, "On the Extension of a Fourier Descriptor Based Method for Planar Four-Bar Linkage Synthesis for Generation of Open and Closed Paths," ASME J. Mech. Rob., **3**(3), p. 031002.
[36] Mcgarva, J. R., 1994, "Rapid Search and Selection of Path Generating Mechanisms From a Library," Mech. Mach. Theory, **29**(2), pp. 223–235.
[37] Ullah, I., and Kota, S., 1997, "Optimal Synthesis of Mechanisms for Path Generation Using Fourier Descriptors and Global Search Methods," ASME J. Mech. Des., **119**(4), pp. 504–510.
[38] Khan, N., Ullah, I., and Al-Grafi, M., 2015, "Dimensional Synthesis of Mechanical Linkages Using Artificial Neural Networks and Fourier Descriptors," Mech. Sci., **6**(1), pp. 29–34.
[39] Vermeer, K., Kuppens, R., and Herder, J., 2018, "Kinematic Synthesis Using Reinforcement Learning," Volume 2A: 44th Design Automation Conference, Quebec City, Quebec, Canada, Aug. 26–29, American Society of Mechanical Engineers, p. V02AT03A009.
[40] Deshpande, S., and Purwar, A., 2019, "A Machine Learning Approach to Kinematic Synthesis of Defect-Free Planar Four-Bar Linkages," ASME J. Comput. Inf. Sci. Eng., **19**(2), p. 021004.
[41] Deshpande, S., and Purwar, A., 2019, "Computational Creativity Via Assisted Variational Synthesis of Mechanisms Using Deep Generative Models," ASME J. Mech. Des., **141**(12), p. 121402.
[42] Purwar, A., and Chakraborty, N., 2023, "Deep Learning-Driven Design of Robot Mechanisms," ASME J. Comput. Inf. Sci. Eng., **23**(6), p. 060811.
[43] Sharma, S., and Purwar, A., 2022, "A Machine Learning Approach to Solve the Alt-Burmester Problem for Synthesis of Defect-Free Spatial Mechanisms," ASME J. Comput. Inf. Sci. Eng., **22**(2), p. 021003
[44] Deshpande, S., and Purwar, A., 2020, "An Image-Based Approach to Variational Path Synthesis of Linkages," ASME J. Comput. Inf. Sci. Eng., **21**(2), p. 021005.
[45] Nobari, A. H., Srivastava, A., Gutfreund, D., and Ahmed, F., 2022, "LINKS: A Dataset of a Hundred Million Planar Linkage Mechanisms for Data-Driven Kinematic Design," Volume 3A: 48th Design Automation Conference (DAC), St. Louis, MO, Aug. 14–17, American Society of Mechanical Engineers, p. V03AT03A013.
[46] Chen, Y., Lu, C., Fan, W., Feng, J., and Sareh, P., 2023, "Data-Driven Design and Morphological Analysis of Conical Six-Fold Origami Structures," Thin-Walled Struct., **185**, p. 110626.

[47] Chen, Y., Shi, J., He, R., Lu, C., Shi, P., Feng, J., and Sareh, P., 2023, "A Unified Inverse Design and Optimization Workflow for the Miura-Oring Metastructure," ASME J. Mech. Des., **145**(9), p. 091704.
[48] Nurizada, A., and Purwar, A., 2023, "An Invariant Representation of Coupler Curves Using a Variational AutoEncoder: Application to Path Synthesis of Four-Bar Mechanisms," ASME J. Comput. Inf. Sci. Eng., **24**(1), p. 011008.
[49] Sener, S., and Unel, M., 2006, "Geometric Invariant Curve and Surface Normalization," International Conference on Image Analysis and Recognition. Póvoa de Varzim, Portugal, Sept. 18–20, pp 445–456.
[50] Hyvärinen, A., Karhunen, J., and Oja, E., 2004, *Independent Component Analysis*, Wiley.
[51] Hyvarinen, A., 1999, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis," IEEE Trans. Neural Networks, **10**(3), pp. 626–634.
[52] Bottou, L., 2010, "Large-Scale Machine Learning With Stochastic Gradient Descent," Proceedings of COMPSTAT'2010, Paris, France, Aug. 27–31, Springer, pp. 177–186.
[53] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986, "Learning Representations by Back-Propagating Errors," Nature, **323**(6088), pp. 533–536.
[54] Oring, A., Yakhini, Z., and Hel-Or, Y., 2021, "Autoencoder Image Interpolation by Shaping the Latent Space," Proceedings of the 38th International Conference on Machine Learning, M. Meila, T. Zhang, eds., Vol. 139, PMLR, pp. 8281–8290.
[55] Berndt, D. J., and Clifford, J., 1994, "Using Dynamic Time Warping to Find Patterns in Time Series," Proceedings of AAAI Workshop on Knowledge Discovery in Databases, Seattle, WA, July 31–Aug. 1, pp. 359–370.
[56] Lyu, Z., Purwar, A., and Liao, W., 2023, "A Unified Real-Time Motion Generation Algorithm for Approximate Position Analysis of Planar N-Bar Mechanisms," ASME J. Mech. Des., **146**(6), p. 063302.
[57] Li, J., Xue, Q., Yang, S., Han, X., Zhang, S., Li, M., and Guo, J., 2021, "Kinematic Analysis of the Human Body During Sit-to-Stand in Healthy Young Adults," Medicine, **100**(22), p. e26208.
[58] Kamnik, R., Bajd, T., Williamson, J., and Murray-Smith, R., 2005, "Rehabilitation Robot Cell for Multimodal Standing-Up Motion Augmentation,"
[59] Carr, J. H., 1992, "Balancing the Centre of Body Mass During Standing Up," Physiother. Theor. Pract., **8**(3), pp. 159–164.
[60] Roberts, S., 1875, "On Three-Bar Motion in Plane Space," Proc. Lond. Math. Soc., **1**(1), pp. 14–23.
[61] Roth, B., 1965, "On the Multiple Generation of Coupler-Curves," ASME J. Eng. Ind., **87**(2), pp. 177–183.
[62] Uicker, J. J., Pennock, G. R., and Shigley, J. E., 2011, *Theory of Machines and Mechanisms*, 4th ed., Oxford University Press, New York.
[63] van der Maaten, L., and Hinton, G., 2008, "Visualizing Data Using T-SNE," J. Mach. Learn. Res., **9**(86), pp. 2579–2605.
[64] Parrish, B. E., Michael McCarthy, J., and Eppstein, D., 2015, "Automated Generation of Linkage Loop Equations for Planar One Degree-of-Freedom Linkages, Demonstrated Up to 8-Bar," ASME J. Mech. Rob., **7**(1), p. 011006.