MechaFormer: Sequence Learning for Kinematic Mechanism Design Automation

Diana Bolanos[1,2], **Mohammadmehdi Ataei**[1], **Pradeep Kumar Jayaraman**[1]

[1]Autodesk Research, Toronto, Ontario, Canada
[2]Department of Mechanical Engineering, UC Berkeley, Berkeley, CA, USA
dbolanos@berkeley.edu, {mehdi.ataei, pradeep.kumar.jayaraman}@autodesk.com

Abstract

Designing mechanical mechanisms to trace specific paths is a classic yet notoriously difficult engineering problem, characterized by a vast and complex search space of discrete topologies and continuous parameters. We introduce MechaFormer, a Transformer-based model that tackles this challenge by treating mechanism design as a conditional sequence generation task. Our model learns to translate a target curve into a domain-specific language (DSL) string, simultaneously determining the mechanism's topology and geometric parameters in a single, unified process. MechaFormer significantly outperforms existing baselines, achieving state-of-the-art path-matching accuracy and generating a wide diversity of novel and valid designs. We demonstrate a suite of sampling strategies that can dramatically improve solution quality and offer designers valuable flexibility. Furthermore, we show that the high-quality outputs from MechaFormer serve as excellent starting points for traditional optimizers, creating a hybrid approach that finds superior solutions with remarkable efficiency.

Code/Appendix —
https://github.com/dianabolanos/mechaformer

Introduction

Mechanism synthesis is a foundational problem in mechanical design that involves the systematic generation and selection of feasible kinematic topologies and configurations to achieve specific motion or functional requirements (Hartenberg and Danavit 1964; Angeles 2003). It plays a crucial role in various applications, from industrial robotics and manufacturing machinery to biomedical devices and consumer electronics. The overarching goal of mechanism synthesis is to determine the best arrangement of joints and links such that a mechanical system performs a desired task reliably and efficiently, often under constraints like space limitations, joint limits, or load-bearing capacities (Norton and Han 2007). A designer must select a topology (the number and arrangement of links and joints) from a discrete set of graphs while simultaneously optimizing continuous geometric parameters like joint locations. This mixed discrete-continuous space, combined with a highly non-linear and non-convex relationship between parameters and motion, is commonly encountered in mechanical design problems (Bolanos et al. 2023; Martins and Ning 2021), and makes it notoriously hard for analytical and optimization-based methods.

Traditional mechanism synthesis follows two paths, both with unique challenges. Analytical methods rooted in kinematics, such as Burmester theory (Hartenberg and Danavit 1964), produce closed-form solutions for simple four-bar linkages and a handful of target positions but break down for continuous curves, richer topologies, or cases with no solution (Primrose, Freudenstein, and Sandor 1964; Ma and Angeles 1988). Numerical optimization casts synthesis as error minimization inside a chosen topology (Ebrahimi and Payvandy 2015), yet gradient methods cling to initial guesses and get trapped in local minima (Mariappan and Krishnamurty 1996; Sancibrian et al. 2004), while population-based search needs thousands of evaluations (Lin 2010; Acharyya and Mandal 2009; Cabrera, Simon, and Prado 2002). Both approaches usually explore only one topology, so better mechanisms remain hidden unless each alternative is optimized separately, multiplying cost.

Although finding a mechanism that produces a prescribed path is exceptionally difficult, simulating the path traced by a given mechanism is straightforward. This computational asymmetry between inverse and forward kinematics makes the problem well-suited to learning-based approaches. Because forward kinematics are inexpensive, thousands of candidate mechanisms can be evaluated in seconds on modern hardware, enabling large-scale dataset generation and screening (Heyrani Nobari et al. 2022; Nurizada et al. 2025).

Inspired by recent successes in domains like program synthesis (Wang et al. 2021), molecular design (Mazuz et al. 2023), and gear design (Etesam et al. 2025; Cheong et al. 2025; Ataei et al. 2025), we reframe kinematic synthesis as a conditional sequence-to-sequence learning problem. We introduce a domain-specific language (DSL) that serializes any mechanism into a structured string of tokens. This approach transforms the intractable design problem into a more manageable task: translating a target curve into its corresponding mechanism *sentence*.

We present MechaFormer, a Transformer-based architecture specifically trained to perform this translation. Given a target B-spline curve, MechaFormer autoregressively generates a complete mechanism definition. Its core advantage

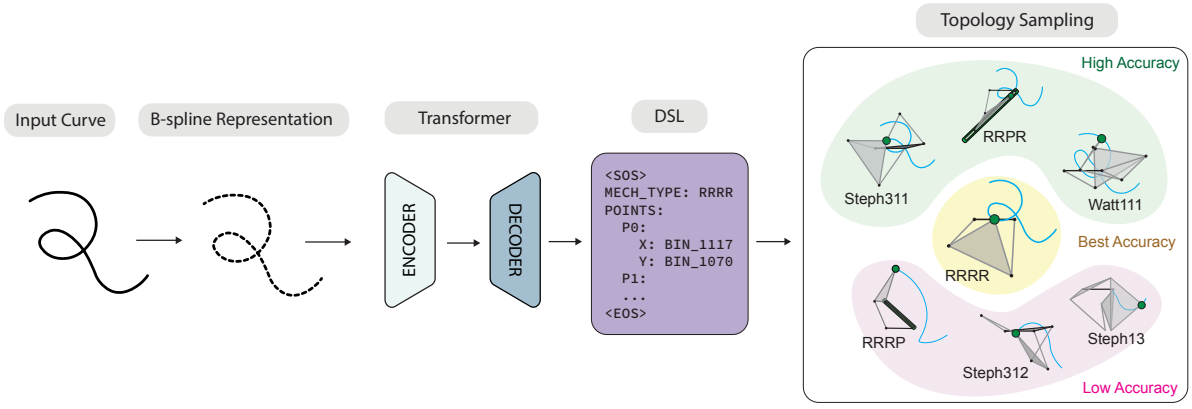

Figure 1: Mechanism-sampling pipeline. The input curve is converted to a B-spline, encoded by a Transformer, and decoded into a DSL that specifies mechanism topology and joint parameters. Topology sampling then generates and evaluates candidates, with clusters indicating accuracy.

lies in its unified approach: the model learns to output topology and geometry simultaneously, allowing it to discover the most appropriate type of mechanism for a given curve and fine-tune its parameters in a single, coherent process.

A major strength of our generative approach is its capacity for interactive and exploratory design, overcoming the limitations of single-point predictions. We introduce and validate a suite of novel sampling strategies that enable a more holistic exploration of the vast and complex design space. These techniques empower designers to rapidly generate a diverse portfolio of candidate solutions, systematically probing variations in mechanism topology, geometry, and even contextual placement within a given environment. By providing a broader view of the possibilities, our method allows for the discovery of superior and non-obvious solutions.

This paper makes the following contributions:

1. **A unified generative framework** that treats mechanism synthesis as a sequence learning problem. This approach seamlessly integrates topology selection and parameter optimization into a single, end-to-end model, allowing it to discover optimal mechanisms from a vast and complex design space without manual partitioning.
2. **State-of-the-art performance** in path synthesis with sampling supporting different topologies. Through comprehensive experiments, we show that MechaFormer surpasses existing learning-based baselines in path-matching accuracy, while also generating a diverse set of valid and novel mechanism designs.
3. **A design framework with targeted sampling strategies.** We introduce and validate sampling strategies, including Best@k, rotational sampling, and topology sampling that substantially improve solution quality and diversity, enabling rapid exploration of candidate mechanisms.
4. **A hybrid optimization workflow.** We show that model outputs provide superior initializations for local optimizers, yielding elite solutions far more efficiently and robustly than direct optimization from random starts.

Related Work

Recent contributions in mechanical synthesis of linkage mechanisms have looked to machine learning techniques for guidance on designing complex physical systems (Sonntag et al. 2024; Han et al. 2025).

Nurizada et al. (2025) introduced a conditional β-VAE model for generating diverse planar four-bar mechanisms from coupler curves. Their work is supported by a large curated dataset and a comprehensive evaluation framework based on reconstruction accuracy, novelty, and diversity. While the model is capable of producing multiple candidate mechanisms per input, its reconstruction performance shows that only about 45% of generated mechanisms achieve a DTW score below 2, which indicates satisfactory curve approximation. This limits its effectiveness for high-precision synthesis tasks and suggests room for improvement in learning the mapping between input curves and mechanism parameters.

Nobari et al. (2022) introduced the LINKS dataset, a large-scale collection of 100 million planar mechanisms ranging from 6 to 15 links, each paired with a motion trajectory. In follow-up work, Nobari et al. (2024) developed a contrastive learning approach for retrieving mechanisms from the dataset, followed by local refinement using quasi-Newton methods. While effective in minimizing trajectory error, many of the retrieved mechanisms are topologically complex and structurally unrealistic, often involving oversized configurations or internal collisions that would be impractical or impossible to manufacture.

Optimization and reinforcement learning-based methods have also been explored for mechanism synthesis. Pan et al. (2023) investigated a range of optimization approaches, including mixed-integer conic programming (MICP), mixed-integer nonlinear programming (MINLP), and simulated annealing (SA), to design mechanisms with up to 14 rigid links. Their hybrid MICP-MINLP approach achieved higher-quality results than SA but incurred high computational costs, often requiring over an hour to solve a sin-

gle instance. Similarly, Fogelson et al. (2023) proposed a reinforcement learning framework that refines mechanism parameters through reward-based feedback while enforcing geometric constraints. Although their method demonstrates better performance than evolutionary baselines, it remains expensive and requires extensive reward engineering to balance feasibility and task performance. These approaches illustrate the promise and limitations of optimization-heavy pipelines, which struggle to scale efficiently.

In contrast to prior work, our approach treats mechanism synthesis as a sequence modeling task, using a transformer-based architecture to map continuous B-spline representations of input curves to discrete DSL tokens representing mechanism topology and joint parameters. This formulation enables generalization across a diverse set of mechanism families and allows for probabilistic sampling of multiple topologies and configurations per input. Furthermore, we demonstrate how this framework can be used to discover better-performing topologies through input rotation and topology sampling, a form of post-hoc analysis that has not been explored in existing generative models.

Problem Formulation

The core task of kinematic path synthesis is to solve an inverse problem. Given a desired target trajectory, the goal is to discover a corresponding planar linkage mechanism that can generate it. We formalize the key components below. We represent a mechanism as $\mathcal{M} = (\tau, \mathbf{J}, c)$, where $\tau \in \mathcal{T}$ is a topology chosen from a finite set of mechanism types (e.g., four-bar, Watt six-bar) that fixes the number of links, the joint types, and their connectivity (revolute or prismatic). The continuous parameters are $\mathbf{J} = \{\mathbf{j}_1, \ldots, \mathbf{j}_n\} \subset \mathbb{R}^2$, the initial joint coordinates for τ. The coupler index c selects the joint whose path must match the target.

Forward kinematics is given by a deterministic simulator Φ that maps $\mathcal{M}$ and an actuation variable (e.g., input crank angle θ) to the coupler position and thus to the curve

$$\mathcal{C} = \Phi(\mathcal{M}) = \{\Phi(\mathcal{M}, \theta) \mid \theta \in [0, 2\pi)\} \subset \mathbb{R}^2. \quad (1)$$

This forward process, Φ, is computationally inexpensive and straightforward to evaluate.

The inverse problem (finding an optimal mechanism $\mathcal{M}^*$ that generates a trajectory closely matching a desired target curve $\mathcal{C}^*$) can be framed as an optimization problem:

$$\mathcal{M}^* = \underset{\mathcal{M}=(\tau,\mathbf{J},c)}{\arg\min}\ d(\Phi(\mathcal{M}), \mathcal{C}^*) \quad (2)$$

where $d(\cdot, \cdot)$ is a distance metric between two curves, such as Dynamic Time Warping (DTW) (Tavenard et al. 2020). This formulation is notoriously difficult to optimize because it is both mixed-integer and non-convex: the search couples a discrete topology $\tau \in \mathcal{T}$ with continuous joint coordinates $\mathbf{J} \in \mathbb{R}^{2n}$, and the objective $d(\Phi(\mathcal{M}), \mathcal{C}^*)$ has many local minima in $\mathbf{J}$, making gradient-based and many gradient-free methods unreliable and highly sensitive to initialization.

To address these challenges, we reframe kinematic synthesis as a probabilistic modeling task rather than optimization. We learn the conditional distribution $p(\mathcal{M}|\mathcal{C}^*)$ from a dataset of $(\mathcal{M}, \mathcal{C})$ pairs generated by the forward simulator Φ, enabling direct sampling of high-quality mechanisms for any target curve.

Methodology

Canonical Frame Normalization for Invariance and Exploratory Sampling

A fundamental challenge in learning from geometric data is handling nuisance variables. In our case, a mechanism's function is invariant to similarity transformations. Raw joint coordinates force models to waste capacity learning this invariance. We eliminate this redundancy by applying canonical normalization to every mechanism, anchoring on its two ground joints $\mathbf{j}_{g_1}$ and $\mathbf{j}_{g_2}$. An affine transformation maps all joints such that $\mathbf{j}'_{g_1} = (0, 0)$ and $\mathbf{j}'_{g_2} = (1, 0)$.

This normalization provides two key benefits. First, it collapses infinite placements to one canonical representation, letting the model focus on mapping relative geometry to curve shape. Second, by fixing the mechanism's frame, we enable design exploration through input curve transformation, the foundation of our rotational sampling strategy discussed later (Figure 2). To find optimal mechanism orientation for a given curve, we rotate the input by angles $\{\alpha_1, \ldots, \alpha_m\}$, generate mechanisms for each rotation, then apply inverse rotations to the generated mechanisms. This efficiently searches for optimal base orientation in real-world coordinates, made tractable by our representation.

Abstracting Topology as a Single Conditional Choice

We represent mechanism topology as a single categorical variable rather than having the model assemble a graph of links and joints. Each of the 24 valid, kinematically distinct topologies in our library maps to a unique token (e.g., `RRRR`, `Watt2T1A1`, `Steph1T1`), and the model's first task is to predict one token.

This abstraction guarantees kinematic feasibility and focuses learning where it matters. Constructing valid chains directly is combinatorially hard and constrained by rules such as Grubler's criterion (Gogu 2005) and connectivity, so encoding topology as a token ensures every design starts from a valid, well-defined type. With topology fixed, the model maps curve features to the most suitable known mechanism, then infers dimensions. The detailed connectivity graph for the chosen type is stored externally and guides generation of joint coordinates.

This choice does preclude novel topologies, which is an acceptable trade-off. Most practical mechanisms use a small set of foundational four-bar and six-bar linkages that our 24 types cover well. The real engineering challenge is creative dimensioning within these proven families, and that is exactly what the model optimizes.

Input Representation: B-Spline Control Points

To provide the model with a compact, fixed-size representation of the target curve $\mathcal{C}^*$, we parameterize it using a cubic B-spline. Any given target curve, initially represented as a set of ordered points, is fitted to a B-spline with a fixed

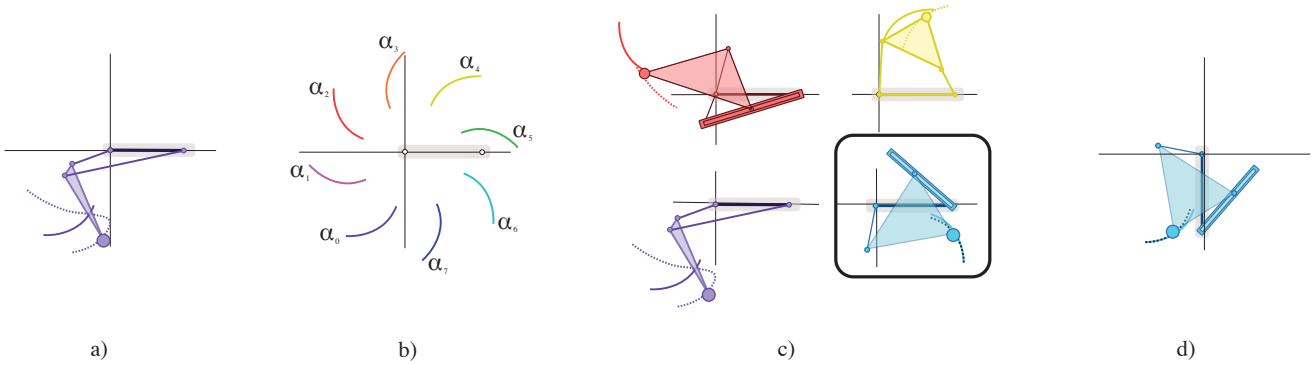

Figure 2: a) User-defined input curve shown by the solid line, and model-generated mechanism and coupler trajectory shown by the dashed line. b) Original curve is rotated eight times by $\Delta\alpha = 45°$. c) Four example mechanisms at different rotation angles. The mechanism at α_6 is selected. d) The mechanism base is rotated to retain the position of the user-defined input curve relative to the origin. A simple target curve is used to clearly illustrate the process.

number of $K = 64$ control points. The resulting set of control points, $\mathbf{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_K\}$ with $\mathbf{p}_k \in \mathbb{R}^2$, captures the essential shape of the curve. This sequence of control points $\mathbf{X} \in \mathbb{R}^{K\times 2}$ serves as the input to our model.

Output Representation: A Domain-Specific Language for Mechanisms

To make the mechanism $\mathcal{M}$ amenable to sequence generation, we developed a DSL that serializes a mechanism's topology and continuous joint coordinates into a discrete sequence of tokens, $\mathbf{y} = (y_1, y_2, \ldots, y_L)$.

Vocabulary The vocabulary $\mathcal{V}$ of our DSL consists of three types of tokens:

- **Structural Tokens:** Special tokens like 'SOS' (Start of Sequence), 'EOS' (End of Sequence), and delimiters like 'MECH_TYPE', 'POINTS', 'P_i', 'X:', 'Y:'.
- **Topology Tokens:** A finite set of tokens representing each of the 24 mechanism topologies (including 4-bar and 6-bar linkages, with a combination of revolute or prismatic joints) in our design space, e.g., 'RRRR', 'RRRP'.
- **Coordinate Tokens:** A set of integer tokens representing quantized coordinate values. We discretize the continuous coordinate space $[-\kappa, \kappa]$ into $B = 200$ uniform bins (see appendix for the ablation that justified this choice). A continuous coordinate value v is mapped to a bin index $b(v)$ via:

$$b(v) = \left\lfloor \frac{(v + \kappa) \cdot (B - 1)}{2\kappa} \right\rfloor \tag{3}$$

This transforms the regression of continuous joint positions into a multi-class classification problem.

Sequence Structure A full mechanism sequence $\mathbf{y}$ follows a strict, human-readable structure that first declares the topology and then lists the normalized coordinates of its free joints. A ground link is assumed to be normalized with its joints at $(0, 0)$ and $(1, 0)$, so these are not included in the sequence. Table 1 provides a detailed example.

Token	Description
`<SOS>`	Start of sequence token.
`MECH_TYPE RRRR`	Declares the mechanism's topology. Here, a four-bar linkage with four revolute joints.
`POINTS`	Structural token indicating the start of the joint coordinate block.
`P_1 X: BIN_125 Y: BIN_180`	Specifies the coordinates for the first free joint. The continuous coordinates (x_1, y_1) have been quantized to integer bin indices (125, 180).
`P_2 X: BIN_45 Y: BIN_78`	Specifies the coordinates for the second free joint, quantized to bins (45, 78).
...	(Additional joints)
`<EOS>`	End of sequence token.

Table 1: Breakdown of the Domain-Specific Language (DSL) used to represent a mechanism as a token sequence.

Sequence-to-Sequence Model Architecture

We map an input sequence of control points $\mathbf{X}$ to a sequence of mechanism tokens $\mathbf{y}$ using a Transformer encoder-decoder (Vaswani et al. 2017). Training maximizes

$$\log p(\mathbf{y} \mid \mathbf{X}) = \sum_{t=1}^{L} \log p(y_t \mid \mathbf{y}_{<t}, \mathbf{X}). \tag{4}$$

The encoder self-attends over the B-spline control points to obtain contextual features. The decoder autoregressively predicts tokens using masked self-attention over $\mathbf{y}_{<t}$ and cross-attention to the encoder output.

Hybrid Method: Generative Seeding for Local Optimization

MechaFormer generates high-quality mechanisms but may not reach perfect local optima due to its probabilistic nature. We bridge this gap by combining our generative

model's global search with local optimization precision, using MechaFormer's output as an intelligent initial seed.

For target curve $\mathcal{C}^*$, we first sample an initial mechanism $\mathcal{M}_{gen} = (\tau_{gen}, \mathbf{J}_{gen}, c_{gen})$ from $p(\mathcal{M}|\mathcal{C}^*)$. This handles the most difficult aspects: selecting a promising topology τ_{gen} and providing near-optimal initial geometry $\mathbf{J}_{gen}$.

We then fix the topology τ_{gen} and refine the joint coordinates through local optimization. Within trust region $\mathcal{B}$ around $\mathbf{J}_{gen}$, we minimize path-following error:

$$\mathbf{J}^* = \underset{\mathbf{J} \in \mathcal{B}(\mathbf{J}_{gen})}{\arg\min}\ d(\Phi((\tau_{gen}, \mathbf{J}, c_{gen})), \mathcal{C}^*) \tag{5}$$

We solve this using L-BFGS-B, a quasi-Newton method with box constraints. By starting from MechaFormer's high-likelihood initialization rather than random points, this hybrid approach avoids poor local minima and consistently achieves superior solutions.

Dataset and Training

We train MechaFormer on a subset of the dataset introduced by Nurizada et al. (2025), which contains 3 million single-DOF planar linkage mechanisms with their corresponding coupler curves. We filter the dataset to include only mechanism types with at least 20,000 instances, resulting in 846,480 training samples (with 83,499 held out for validation) across 24 distinct topologies. Our model architecture contains approximately 19 million parameters with a 256-dimensional hidden size, 8 attention heads, and 6 layers each for encoder and decoder. The model is trained using the AdamW optimizer with a cosine learning rate schedule and warm-up, employing cross-entropy loss with label smoothing for 30 epochs on distributed GPUs (DGX A100). Implementation details, including the training code, processed data, and model weights, are provided in the appendix.

Experiments

We evaluate model performance on 1,000 validation curves drawn from the dataset introduced by Nurizada et al. (2025). All evaluations were conducted with a model temperature of 0.1 (see appendix for sampling study). We use DTW to assess reconstruction fidelity, with values below 2 considered satisfactory, scores between 2 and 3 deemed moderate, and values above 3 indicative of poor reconstruction. DTW computations are performed using the *tslearn* library (Tavenard et al. 2020).

The normalization used for DTW analysis across curves of different scales is defined as:

$$\mathcal{N}(\mathcal{X}) = \frac{\mathcal{X} - \mu_i}{\sigma_{RMS_i}} \tag{6}$$

where both input and output curves are normalized by the input curve's mean (μ_i) and root mean square (RMS) deviation (σ_{RMS_i}), enabling fair comparison between curves of different scales. We report both medians (η_{DTW}) and means (μ_{DTW}) since means are skewed by outliers (see appendix), in addition to a percentage of samples that fell within satisfactory scores. Success Rate is also presented to quantify convergence on feasible mechanism configurations. Lastly, we present the mean bi-directional Chamfer Distance (μ_{CD}) as a supplementary accuracy metric. We calculate this metric by computing the average of the minimum Euclidean distances from each point on one curve to the nearest point on the other curve, in both directions.

To provide a more comprehensive visual understanding of performance, we include representative examples of input curves, generated mechanisms, and their corresponding evaluation metrics in the appendix.

Best @ k

We evaluate the benefit of sampling multiple candidates by generating $k \in \{1, 2, 4, 8, 16, 32\}$ mechanisms per validation curve and selecting the best based on DTW. Table 2 shows monotonic improvement as k increases: median DTW decreases from 3.09 (k=1) to 1.61 (k=32), while maintaining over 99% Success Rate throughout. This demonstrates that sampling multiple candidates significantly improves solution quality without sacrificing robustness.

Rotational Sampling

Our canonical normalization enables efficient exploration of mechanism orientations by rotating the input curve rather than the mechanism itself. We generate mechanisms for the input curve rotated at $45°$ increments ($\alpha_i = i \cdot 45°$ for $i = 0, \ldots, 7$), then apply inverse rotations to recover real-world placements. Table 2 shows this strategy improves median DTW from 3.09 to 1.76, demonstrating that optimal base orientation significantly impacts accuracy. Figure 2 illustrates this eight-rotation process, where selecting the best orientation (e.g., α_6 in the example) yields superior curve-following performance. While translation sampling could be similarly implemented by shifting the input curve, we leave this exploration for future work.

Topology Sampling

Since topology tokens appear first in our DSL sequence, we can prefix-constrain generation to explore all 24 mechanism types for each input curve. Table 2 shows an 86.7% Success Rate. Despite lower Success Rates, this sampling identifies the best-performing topology when design flexibility permits, as illustrated in Figure 1.

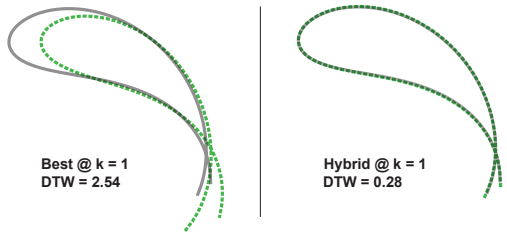

Figure 3: An example of curve alignment before and after implementing the optimization routine.

		$\eta_{\text{DTW}} \downarrow$	$\mu_{\text{DTW}} \downarrow$	DTW < 3.0	DTW < 2.0	$\mu_{\text{CD}} \downarrow$	Success Rate $\uparrow$
MechaFormer	**Best @** k						
	$k = 1$	3.090	$6.831^{\pm 9.468}$	48.7%	35.3%	0.250	99.2%
	$k = 2$	2.652	$5.357^{\pm 6.699}$	54.6%	41.8%	0.205	99.4%
	$k = 4$	2.154	$4.291^{\pm 5.499}$	61.9%	47.4%	0.171	**99.5%**
	$k = 8$	1.877	$3.630^{\pm 4.848}$	66.2%	52.3%	0.152	**99.5%**
	$k = 16$	1.735	$3.073^{\pm 3.609}$	69.5%	56.1%	0.134	99.3%
	$k = 32$	**1.605**	$2.739^{\pm 3.212}$	**72.4%**	**60.1%**	0.123	99.4%
	Best @ α	1.757	$\mathbf{2.675^{\pm 2.534}}$	72.0%	55.6%	0.121	99.0%
	Best @ Topology	2.090	$2.857^{\pm 2.468}$	68.4%	47.6%	**0.119**	86.7%
MF + BFGS	**Hybrid @** k						
	$k = 1$	1.591	$4.264^{\pm 6.133}$	65.0%	56.0%	0.139	**97.6%**
	$k = 16$	0.912	$1.946^{\pm 2.535}$	79.6%	71.7%	0.084	94.4%
	$k = 32$	**0.887**	$\mathbf{1.796^{\pm 2.349}}$	**82.0%**	**73.9%**	**0.077**	93.4%
Baseline	**KNN**	3.799	$5.356^{\pm 6.311}$	36.7%	20.7%	0.212	100.0%
	L-BFGS-B	14.258	$18.430^{\pm 18.504}$	6.8%	3.9%	0.6603	95.0%

Prior work reports a best μ_{CD} of 0.135 from a study conducted on this dataset (Nurizada et al. 2025).

Table 2: Performance comparison of sampling strategies (varying k, rotation angle α, topology search) versus optimization baselines. DTW measures reconstruction accuracy: < 2 (satisfactory), $2 - 3$ (moderate), > 3 (poor). Both median and mean DTW are reported due to high-variance outliers. Mean Chamfer Distance is reported as supplementary accuracy metric. All methods used identical hyperparameters (see appendix). Success Rate indicates percentage of kinematically feasible mechanisms.

Hybrid Method

We combine MechaFormer's Best @ k sampling with local optimization: first generating k mechanisms and selecting the best, then refining its joint coordinates using L-BFGS-B within trust regions. Table 2 shows this hybrid approach with $k = 32$ achieves the lowest median DTW of 0.887, demonstrating that initializing from the best of multiple model samples dramatically outperforms random initialization. Figure 3 illustrates the improvement in DTW achieved through the use of hybrid optimization techniques.

Baselines

Comparison to Prior Work We compare our approach against two recent works in mechanism synthesis. First, Nurizada et al. (2025) achieved a best μ_{CD} of 0.135. Our model surpasses this with μ_{CD} of 0.119 using sampling and 0.077 using our hybrid method (Table 2). While both studies use subsets of the same dataset, direct comparison has limitations as the specific sample selections may differ. Second, Nurizada, Lyu, and Purwar (2025) reported η_{DTW} of 2.441 using a conditional β-VAE on 12 million 4-bar mechanisms. Though our dataset is smaller (846,480 samples) and more diverse (24 topologies including 6-bar), the consistent DTW computation enables meaningful comparison. MechaFormer achieves η_{DTW} of 1.605 (Best@32) and 0.887 (Hybrid@32), indicating that its architecture offsets reduced data scale while managing increased mechanism complexity.

KNN To verify MechaFormer learns meaningful representations beyond memorization, we use the trained model's encoder to embed validation curves, retrieve their nearest neighbors from the training set based on cosine similarity in the embedding space, and evaluate the retrieved mechanisms. Table 2 shows MechaFormer significantly outperforms this baseline (median DTW: 3.090 vs 3.799), confirming the model generates novel solutions rather than retrieving stored examples from its learned representation space.

L-BFGS-B We evaluate direct optimization by using *just* the topology from MechaFormer's Best@k=32 output, applying the same ground normalization constraints ($\mathbf{j}'_{g_1} = (0, 0)$, $\mathbf{j}'_{g_2} = (1, 0)$), but randomly initializing all other joint coordinates. Using identical optimization parameters as the Hybrid study, Table 2 shows poor performance: median DTW of 14.258 and a 95% convergence Success Rate, demonstrating the difficulty of optimization from random initialization in this non-convex design space.

Diversity

We quantify generative diversity to assess the model's ability to produce varied yet valid designs. For each input curve, we generate k mechanisms and compute two diversity metrics. Topological diversity measures the fraction of unique mechanism types among k samples, calculated as the number of unique topologies divided by the total 24 possible topologies. Joint diversity quantifies variation among mechanisms of the same topology by computing the average pairwise distance between their joint parameters, normalized by the square root of the number of free joints to account for varying mechanism complexities.

Figure 4 shows how temperature controls the exploration-exploitation tradeoff: $t = 0.1$ yields consistent but limited

Approach	DTW (median ± std)					Success		Time	
	Initial	25%	50%	75%	Final	DTW < 3.0	`maxfun`	`nfev`	(s)
L-BFGS-B	$18.99^{\pm 11.26}$	$11.87^{\pm 9.26}$	$11.87^{\pm 9.26}$	$11.86^{\pm 9.53}$	$11.86^{\pm 9.53}$	3/10	0/10	$511^{\pm 316}$	$2.22^{\pm 6.26}$
Hybrid @ $k = 1$	$4.36^{\pm 5.16}$	$1.45^{\pm 2.26}$	$1.43^{\pm 1.68}$	$1.41^{\pm 1.61}$	$1.15^{\pm 1.55}$	8/10	3/10	$641^{\pm 791}$	$3.88^{\pm 17.73}$
Hybrid @ $k = 16$	$1.04^{\pm 5.19}$	$0.77^{\pm 1.07}$	$0.73^{\pm 0.94}$	$0.71^{\pm 0.93}$	$0.71^{\pm 0.94}$	9/10	2/10	$602^{\pm 672}$	$3.54^{\pm 9.08}$
Hybrid @ $k = 32$	$0.99^{\pm 5.19}$	$0.57^{\pm 4.69}$	$0.53^{\pm 4.69}$	$0.51^{\pm 4.31}$	$0.51^{\pm 4.33}$	9/10	1/10	$567^{\pm 443}$	$2.85^{\pm 8.80}$

Table 3: Optimization performance for 10 samples with `maxfun` set at 2000. The `nfev` is the number of objective evaluations used, and the `maxfun` indicates the number of runs that reached the evaluation cap. MechaFormer generation time is negligible.

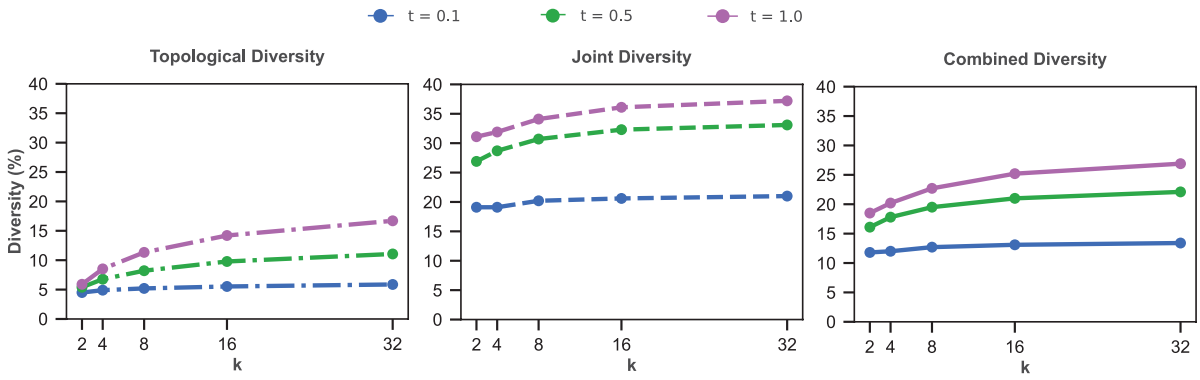

Figure 4: Topological, joint, and combined diversity trends across varying model temperatures and k-sampled evaluations. The Combined Diversity metric is the mean of the Topological and Joint Diversity values.

variations (13.4% combined diversity), while $t = 1.0$ produces richer variations (26.9%) at the potential cost of some invalid designs. This allows designers to tune between conservative refinement and creative exploration.

Optimization Analysis

To understand the value of model-guided initialization, we conduct a deeper analysis comparing L-BFGS-B optimization from random starts versus our hybrid approach. We use SciPy's L-BFGS-B implementation with tolerance parameters `ftol`=10^{-9} and `gtol`=10^{-9}, trust regions bounded by $\delta_i = \max(0.5, 0.5|x_i^0|)$, and extended optimization budget (`maxfun`=2000 vs. 100 initially, `maxiter`=50).

Table 3 shows results from 10 randomly selected validation samples. All methods achieve 90% of their final improvement within 25% of iterations, with median function evaluations (∼600) well below the 2000 limit, indicating convergence to local optima rather than computational constraints. The critical finding is the 20× quality gap: L-BFGS-B from random initialization achieves median DTW of 11.86 with only 30% Success Rate ($DTW < 3.0$), while hybrid approaches reach 0.71 (k=16) to 0.51 (k=32) with 80-90% success. This demonstrates that MechaFormer provides initializations in high-quality basins of attraction, and even with identical optimization parameters and extended budgets, random starts consistently converge to poor local minima due to the non-convex nature of the design space.

Discussion and Conclusions

MechaFormer demonstrates that reframing mechanism synthesis as conditional sequence generation yields substantial advantages over traditional approaches. Our DSL representation unifies topology selection and geometry optimization, eliminating exhaustive enumeration while enabling direct discovery of appropriate mechanisms from curve features.

Our sampling strategies unlock complementary benefits: Best@k sampling reduces median DTW from 3.09 to 1.61, rotational sampling leverages canonical normalization to find optimal base orientations (DTW: 1.76), and temperature control enables designers to balance accuracy and diversity (13.4% to 26.9%). Most significantly, our hybrid approach achieves DTW of 0.887, over 20× better than direct optimization, by providing high-quality initializations that navigate the non-convex design space.

While limited to planar mechanisms with revolute/prismatic joints and a fixed topology library, MechaFormer establishes sequence learning as a powerful paradigm for mechanism design. Future work includes extending to spatial linkages, incorporating manufacturing constraints, and developing real-time CAD integration.

References

Acharyya, S.; and Mandal, M. 2009. Performance of EAs for four-bar linkage synthesis. *Mechanism and Machine Theory*, 44(9): 1784–1794.

Angeles, J. 2003. *Fundamentals of robotic mechanical systems: theory, methods, and algorithms*. Springer.

Ataei, M.; Cheong, H.; Jun, J.; Matejka, J.; Tessier, A.; and Fitzmaurice, G. 2025. Transformer-Based Interfaces for Mechanical Assembly Design: A Gear Train Case Study. *arXiv preprint arXiv:2504.08633*.

Bolanos, D.; Varela, K.; Sargent, B.; Stephen, M. A.; Howell, L. L.; and Magleby, S. P. 2023. Selecting and optimizing origami flasher pattern configurations for finite-thickness deployable space arrays. *Journal of Mechanical Design*, 145(2): 023301.

Cabrera, J.; Simon, A.; and Prado, M. 2002. Optimal synthesis of mechanisms with genetic algorithms. *Mechanism and machine theory*, 37(10): 1165–1177.

Cheong, H.; Ataei, M.; Khasahmadi, A. H.; and Jayaraman, P. K. 2025. e-simft: Alignment of generative models with simulation feedback for pareto-front design exploration. *arXiv preprint arXiv:2502.02628*.

Ebrahimi, S.; and Payvandy, P. 2015. Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms. *Mechanism and Machine Theory*, 85: 189–204.

Etesam, Y.; Cheong, H.; Ataei, M.; and Jayaraman, P. K. 2025. Deep generative model for mechanical system configuration design. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16): 16496–16504.

Fogelson, M. B.; Tucker, C.; and Cagan, J. 2023. GCP-HOLO: Generating high-order linkage graphs for path synthesis. *Journal of Mechanical Design*, 145(7): 073303.

Gogu, G. 2005. Chebychev–Grübler–Kutzbach's criterion for mobility calculation of multi-loop mechanisms revisited via theory of linear transformations. *European Journal of Mechanics-A/Solids*, 24(3): 427–441.

Han, X.; Zhao, P.; Zhao, X.; and Zi, B. 2025. Review on machine learning-based approaches for the kinematic analysis and synthesis of mechanisms. *Frontiers of Mechanical Engineering*, 20(2): 11.

Hartenberg, R.; and Danavit, J. 1964. *Kinematic synthesis of linkages*. New York: McGraw-Hill.

Heyrani Nobari, A.; Srivastava, A.; Gutfreund, D.; and Ahmed, F. 2022. Links: A dataset of a hundred million planar linkage mechanisms for data-driven kinematic design. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 86229, V03AT03A013. American Society of Mechanical Engineers.

Lin, W.-Y. 2010. A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage. *Mechanism and Machine Theory*, 45(8): 1096–1107.

Ma, O.; and Angeles, J. 1988. Performance evaluation of path-generating planar, spherical and spatial four-bar linkages. *Mechanism and machine theory*, 23(4): 257–268.

Mariappan, J.; and Krishnamurty, S. 1996. A generalized exact gradient method for mechanism synthesis. *Mechanism and Machine Theory*, 31(4): 413–421.

Martins, J. R.; and Ning, A. 2021. *Engineering design optimization*. Cambridge University Press.

Mazuz, E.; Shtar, G.; Shapira, B.; and Rokach, L. 2023. Molecule generation using transformers and policy gradient reinforcement learning. *Scientific Reports*, 13(1): 8799.

Nobari, A. H.; Srivastava, A.; Gutfreund, D.; Xu, K.; and Ahmed, F. 2024. Link: Learning joint representations of design and performance spaces through contrastive learning for mechanism synthesis. *arXiv preprint arXiv:2405.20592*.

Norton, R. L.; and Han, J. 2007. *Design of machinery*, volume 4. McGraw-Hill Science/Engineering/Math.

Nurizada, A.; Dhaipule, R.; Lyu, Z.; and Purwar, A. 2025. A dataset of 3M single-DOF planar 4-, 6-, and 8-bar linkage mechanisms with open and closed coupler curves for machine learning-driven path synthesis. *Journal of Mechanical Design*, 147(4): 041702.

Nurizada, A.; Lyu, Z.; and Purwar, A. 2025. Path generative model based on conditional β-variational auto encoder for four-bar mechanism design. *Journal of Mechanisms and Robotics*, 17(6): 061004.

Pan, Z.; Liu, M.; Gao, X.; and Manocha, D. 2023. Joint search of optimal topology and trajectory for planar linkages. *The International Journal of Robotics Research*, 42(4-5): 176–195.

Primrose, E.; Freudenstein, F.; and Sandor, G. 1964. Finite Burmester theory in plane kinematics. *Journal of Applied Mechanics*, 31(4): 683–693.

Sancibrian, R.; Viadero, F.; Garcıa, P.; and Fernández, A. 2004. Gradient-based optimization of path synthesis problems in planar mechanisms. *Mechanism and machine theory*, 39(8): 839–856.

Sonntag, S.; Brünjes, V.; Luttmer, J.; Corves, B.; and Nagarajah, A. 2024. Machine learning applications for the synthesis of planar mechanisms—a comprehensive methodical literature review. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88414, V007T07A003. American Society of Mechanical Engineers.

Tavenard, R.; Faouzi, J.; Vandewiele, G.; Divo, F.; Androz, G.; Holtz, C.; Payne, M.; Yurchak, R.; Rußwurm, M.; Kolar, K.; and Woods, E. 2020. Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research*, 21(118): 1–6.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, Y.; Wang, W.; Joty, S.; and Hoi, S. C. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation. *arXiv preprint arXiv:2109.00859*.