# Sampling Strategies for Transformer-Based Mechanism Synthesis

**Mohammadmehdi Ataei**
Autodesk Research
mehdi.ataei@autodesk.com

**Diana Bolanos**
Autodesk Research
dbolanos@berkeley.edu

**Pradeep Kumar Jayaraman**
Autodesk Research
pradeep.kumar.jayaraman@autodesk.com

## Abstract

Physical design problems offer unique opportunities for sampling strategies that go beyond standard probabilistic generation. We explore how the inherent structure and symmetries of physical systems enable specialized sampling techniques that operate outside the learned model itself. Using mechanism synthesis as an exemplar, where the goal is to design mechanical linkages that trace desired paths, we demonstrate sampling strategies that exploit physical invariances, leverage simulator-based evaluation, and provide interpretable control over the generation process. These approaches show how understanding the physics of a domain can lead to more effective sampling, yielding both accurate and diverse solutions that serve as strong starting points for traditional optimization.

## 1 Introduction

Mechanism synthesis involves designing mechanical linkages–assemblies of rigid bars connected by joints–that trace specific paths when actuated. This classical engineering problem exemplifies a broader class of physical design tasks where generative models can propose solutions, but a physics simulator ultimately determines what works. Unlike text generation where model confidence often correlates with quality, here the simulator acts as the ground truth oracle.

Physical systems possess inherent symmetries and structures that enable sampling strategies beyond what the model learns directly. The same mechanism can be translated, rotated, or scaled while preserving its kinematic behavior, suggesting that sampling can exploit these invariances without retraining. Similarly, the hierarchical nature of physical design (choosing components before fine-tuning parameters) naturally informs how we structure the sampling process. This paper explores several such strategies that leverage the physics of the domain to improve sampling effectiveness.

We demonstrate how simple yet principled sampling techniques, when combined with simulator evaluation, can dramatically improve solution quality and diversity. By treating the learned model as a proposal mechanism and the simulator as the selection criterion, we shift from pure likelihood-based generation to a physics-informed sampling framework. These strategies are not specific to our domain but illustrate a general principle: understanding the physical nature of a problem enables more effective sampling beyond what machine learning alone provides.

Prior work spans generative modeling Hoskins and Kramer [1993], Ataei et al. [2025], Etesam et al. [2025], Cheong et al. [2025], Fogelson et al. [2023], retrieval Nurizada et al. [2025b], Nobari et al. [2024], and optimization-heavy pipelines Ebrahimi and Payvandy [2015], Pan et al. [2023] for design

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference: Sampling Meets Learning.

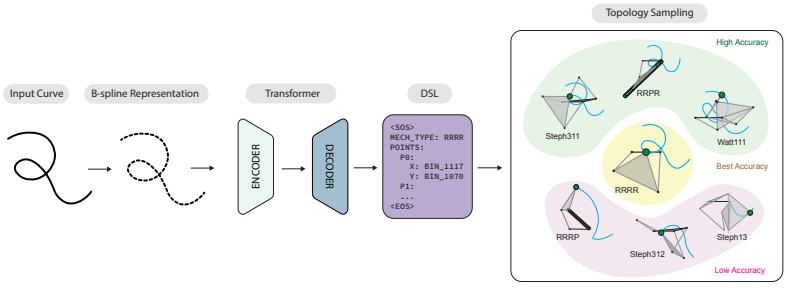


Figure 1: Mechanism-sampling pipeline: a B-spline curve is encoded by a Transformer and decoded to a DSL (topology and joint parameters); sampled candidates are evaluated and selected.

tasks; see recent surveys for a broader view Sonntag et al. [2024], Han et al. [2025]. We complement these threads by treating inference as sampling with evaluator-driven selection, focusing on simple strategies that transfer across domains.

## 2 Methodology

Kinematic path synthesis is an inverse problem: given a target trajectory $\mathcal{C}^*$, find a planar linkage that generates it. We define a mechanism as $\mathcal{M} = (\tau, \mathbf{J}, c)$ where $\tau \in \mathcal{T}$ (e.g., four-bar, Watt six-bar) fixes links, joint types, and connectivity; $\mathbf{J} = \{\mathbf{j}_1, \ldots, \mathbf{j}_n\} \subset \mathbb{R}^2$ are initial joint coordinates; and $c$ selects the coupler joint. Forward kinematics uses a deterministic simulator $\Phi$ producing the curve

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

where $\theta$ is the actuation angle; $\Phi$ is inexpensive to evaluate. The inverse problem is

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

with curve distance $d(\cdot,\cdot)$ (e.g., DTW Tavenard et al. [2020]). This mixed-integer, non-convex program couples discrete $\tau$ with continuous $\mathbf{J}$ and has many local minima, making standard methods unreliable and initialization-sensitive. We instead learn the conditional distribution $p(\mathcal{M} \mid \mathcal{C}^*)$ from simulator-generated $(\mathcal{M}, \mathcal{C})$ pairs, enabling direct sampling of high-quality mechanisms.

Given a target curve $\mathcal{C}^*$, inference is a decision problem under a simulator-defined utility rather than a pure likelihood objective. Let a decoded sequence be $\mathbf{y} = (\tau, \mathbf{q}_{1:m}, c)$ where $\tau$ is the discrete topology token, $\mathbf{q}_{1:m}$ are quantized coordinates, and $c$ is the coupler index. A deterministic map $\Gamma$ converts tokens to a mechanism $\mathcal{M} = \Gamma(\mathbf{y}) = (\tau, \mathbf{J}, c)$ by dequantizing $\mathbf{q}_{1:m}$ to continuous joints $\mathbf{J} = Q_B^{-1}(\mathbf{q}_{1:m})$. The evaluator defines

$$U(\mathbf{y} \mid \mathcal{C}^*) = -\,d\big(\Phi(\Gamma(\mathbf{y})),\, \mathcal{C}^*\big). \tag{3}$$

Because $U$ acts on simulated geometry, $\log p(\mathbf{y} \mid \mathcal{C}^*)$ is an imperfect proxy. We therefore approximate the Bayes decision $\arg\max_{\mathbf{y}} U(\mathbf{y} \mid \mathcal{C}^*)$ by evaluator-in-the-loop sampling: draw a finite proposal set from a tempered model $p_t$ with temperature $t$ and select by $U$,

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y} \in S_{t,k}(\mathcal{C}^*)} U(\mathbf{y} \mid \mathcal{C}^*), \qquad S_{t,k}(\mathcal{C}^*) = \{\mathbf{y}^{(i)} \sim p_t(\mathbf{y} \mid \mathcal{C}^*)\}_{i=1}^{k}. \tag{4}$$

This departs from typical language decoding where token likelihood more closely aligns with task utility and justifies structured sampling tailored to kinematics.

Inputs are $K = 64$ B-spline control points; outputs are a DSL: a topology token (24 types) and canonical-frame joint coordinates quantized to $B = 200$ bins (see appendix for ablation). A Transformer encoder–decoder maps points to tokens, trained with token-level cross-entropy.

**Discrete-continuous tokens and sampling order** The chain rule factorization induced by token order is central for design and usability. We order the sequence so that topology is decided first and geometry conditioned on it,

$$p(\mathbf{y} \mid \mathcal{C}^*) = p(\tau \mid \mathcal{C}^*)\, p(\mathbf{q}_{1:m}, c \mid \tau, \mathcal{C}^*). \tag{5}$$

At inference, prefix-conditioning on $\tau$ implements coarse-to-fine exploration over a hybrid space: for a user–or policy–selected subset $\mathcal{T}' \subseteq \mathcal{T}$ we draw conditionals $\{\mathbf{y}_{>\tau}^{(i)} \sim p_t(\mathbf{y}_{>\tau} \mid \tau, \mathcal{C}^*)\}$ and select by $U\big((\tau, \mathbf{y}_{>\tau}^{(i)}) \mid \mathcal{C}^*\big)$. Placing $\tau$ first isolates the discrete combinatorics, stabilizes sequence length, and provides an interpretable control knob: designers can fix or prioritize mechanism families before sampling continuous dimensions, which improves usability in interactive settings.

**Canonical Frame Normalization** A fundamental challenge in learning from geometric data is handling nuisance variables; the synthesized path is invariant to similarity transforms. We remove translation, rotation, and scale by anchoring on ground joints $\mathbf{j}_{g_1}$ and $\mathbf{j}_{g_2}$ and applying the affine map

$$\mathcal{N}(\mathbf{x}) = s\,\mathbf{R}(-\theta)\,(\mathbf{x} - \mathbf{j}_{g_1}), \quad s = \frac{1}{\|\mathbf{j}_{g_2} - \mathbf{j}_{g_1}\|_2}, \quad \theta = \operatorname{atan2}\big((\mathbf{j}_{g_2} - \mathbf{j}_{g_1})_y, (\mathbf{j}_{g_2} - \mathbf{j}_{g_1})_x\big), \tag{6}$$

with $\mathbf{R}(\theta)$ the 2D rotation matrix. This yields $\mathcal{N}(\mathbf{j}_{g_1}) = (0, 0)$ and $\mathcal{N}(\mathbf{j}_{g_2}) = (1, 0)$ and is applied to curves and joints during dataset construction and decoding. The inverse map $\mathcal{N}^{-1}(\mathbf{x}') = \mathbf{R}(\theta)\,(\mathbf{x}'/s) + \mathbf{j}_{g_1}$ restores world coordinates.

By fixing the frame, we can explore base orientation efficiently via rotations of the input curve. Let $\mathbf{R}(\alpha)$ denote a rotation, $\mathcal{C}_\alpha^* = \{\mathbf{R}(\alpha)\mathbf{p} : \mathbf{p} \in \mathcal{C}^*\}$, and let $\mathbf{y}_\alpha^{(i)} \sim p_t(\mathbf{y} \mid \mathcal{C}_\alpha^*)$ with decoded mechanism $\mathcal{M}_\alpha^{(i)} = \Gamma(\mathbf{y}_\alpha^{(i)})$ expressed in the canonical frame. We evaluate candidates against the original target by inverse-rotating the mechanism,

$$S = \bigcup_{\alpha \in \mathcal{A}} \{\tilde{\mathcal{M}}_\alpha^{(i)} = \mathbf{R}(-\alpha) \cdot \mathcal{M}_\alpha^{(i)}\}_{i=1}^{k}, \qquad \hat{\mathcal{M}} = \arg\min_{\mathcal{M} \in S} d\big(\Phi(\mathcal{M}), \mathcal{C}^*\big), \tag{7}$$

where the group action $\mathbf{R}(-\alpha) \cdot \mathcal{M}$ rotates all joints of $\mathcal{M}$. Figure 2 illustrates this rotation-only exploration. Translations $\mathbf{t}$ can be integrated by the $\mathrm{SE}(2)$ action $\mathbf{x} \mapsto \mathbf{R}(\alpha)\mathbf{x} + \mathbf{t}$ with an analogous selection rule, but translation is largely neutralized by $\mathcal{N}$; we therefore present rotation sampling here as the most impactful and compact strategy.

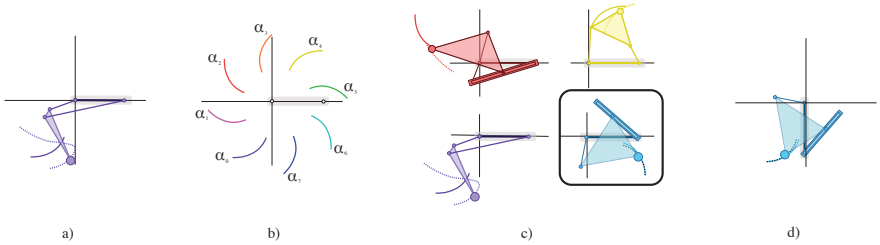


Figure 2: Rotation sampling: a) target curve (solid) and decoded trajectory (dashed); b) curve rotated 8 times by $\Delta\alpha = 45°$; c) mechanisms across angles with the best at $\alpha_6$; d) inverse-rotate the selected mechanism back to the original frame.

**Hybrid Method** The transformer generates strong mechanisms but may miss perfect local optima. We close this gap by using its output as an intelligent seed for precise local optimization. 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 $\tau_{gen}$ and providing near-optimal initial geometry $\mathbf{J}_{gen}$. We then fix the topology $\tau_{gen}$ and refine the joint coordinates through local optimization. Within trust region $\mathcal{B}$ around $\mathbf{J}_{gen}$, we minimize path-following error:

$$\mathbf{J}^* = \arg\min_{\mathbf{J} \in \mathcal{B}(\mathbf{J}_{gen})} d(\Phi((\tau_{gen}, \mathbf{J}, c_{gen})), \mathcal{C}^*) \tag{8}$$

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

| | | $\eta_{\text{DTW}} \downarrow$ | $\mu_{\text{DTW}} \downarrow$ | DTW $< 3.0$ | DTW $< 2.0$ | $\mu_{\text{CD}} \downarrow$ | Success Rate $\uparrow$ |
|---|---|---|---|---|---|---|---|
| **Transformer** | **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 @** $\alpha$ | 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% |
| **TF + 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 $\mu_{\text{CD}}$ of 0.135 from a study conducted on this dataset Nurizada et al. [2025a].

Table 1: Comparison of sampling strategies (varying $k$, rotation $\alpha$, topology search) vs other baselines; metrics include DTW (median and mean; $< 2$ satisfactory, 2–3 moderate, $> 3$ poor), mean Chamfer Distance, and Success Rate (% feasible); all methods use identical hyperparameters (see appendix).

### 2.1 Dataset and Training

We train on a filtered subset of Nurizada et al. [2025a] spanning 24 planar linkage topologies and hold out a validation split. Full dataset curation, architecture hyperparameters, training schedules, and implementation details are provided in Appendix. We evaluate on 1,000 validation curves using DTW (primary) and CD (supplementary).

## 3 Experiments and Discussion

Table 1 highlights the impact of physics-informed sampling. Best-of-$k$ sampling uses the simulator as an oracle, reducing median DTW from 3.090 ($k = 1$) to 1.605 ($k = 32$) with near-perfect feasibility. Rotation sampling exploits SE(2) symmetries to match this performance through orientational invariance. Topology-first sampling reflects engineering principles, guiding exploration of discrete-continuous structures. The hybrid method, combining generative initialization with local optimization, achieves 0.887 median DTW, showing that physics-aware sampling yields superior starting points.

Table 2 shows that our sampling approach enhances optimization. Direct L-BFGS-B from random initialization yields limited improvement (18.99 $\rightarrow$11.86 median DTW), while the hybrid method starting from best-of-$k$ samples converges rapidly, refining 0.99$\rightarrow$0.51 median DTW for best-of-32. Physics-informed sampling thus provides high-quality initialization in favorable regions. Increasing temperature trades accuracy for diversity, boosting variety with minimal feasibility loss (Appendix Fig.7). Overall, sampling-driven seeding outperforms random initialization under equal budgets, with further ablations in the appendix. These results show framing inference as learning-plus-selection yields practical gains: simple sampling strategies paired with an evaluator consistently improve quality, diversity, and usability. Using mechanism design as an example, structured decoding provides effective proposals, and evaluator-guided re-ranking identifies strong solutions and seeds for refinement.

## References


Mohammadmehdi Ataei, Hyunmin Cheong, Jiwon Jun, Justin Matejka, Alexander Tessier, and George Fitzmaurice. Transformer-based interfaces for mechanical assembly design: A gear train case study. *arXiv preprint arXiv:2504.08633*, 2025.

Hyunmin Cheong, Mohammadmehdi Ataei, Amir Hosein Khasahmadi, and Pradeep Kumar Jayaraman. e-simft: Alignment of generative models with simulation feedback for pareto-front design exploration. *arXiv preprint arXiv:2502.02628*, 2025.

| | DTW (median ± std) | | | | | Success | | Time | |
|---|---|---|---|---|---|---|---|---|---|
| Approach | 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 2: Optimization performance on 10 samples (`maxfun`=2000); `nfev` = objective evaluations, `maxfun` = runs reaching the evaluation cap; transformer generation time negligible.


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

Yasaman Etesam, Hyunmin Cheong, Mohammadmehdi Ataei, and Pradeep Kumar Jayaraman. Deep generative model for mechanical system configuration design. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):16496–16504, Apr. 2025. doi: 10.1609/aaai.v39i16.33812. URL `https://ojs.aaai.org/index.php/AAAI/article/view/33812`.

Mitchell B Fogelson, Conrad Tucker, and Jonathan Cagan. Gcp-holo: Generating high-order linkage graphs for path synthesis. *Journal of Mechanical Design*, 145(7):073303, 2023.

Xu Han, Ping Zhao, Xiran Zhao, and Bin Zi. Review on machine learning-based approaches for the kinematic analysis and synthesis of mechanisms. *Frontiers of Mechanical Engineering*, 20(2):11, 2025.

Josiah C Hoskins and Glenn A Kramer. Synthesis of mechanical linkages using artificial neural networks and optimization. In *IEEE International Conference on Neural Networks*, pages 822–J. IEEE, 1993.

Amin Heyrani Nobari, Akash Srivastava, Dan Gutfreund, Kai Xu, and Faez Ahmed. Link: Learning joint representations of design and performance spaces through contrastive learning for mechanism synthesis. *arXiv preprint arXiv:2405.20592*, 2024.

Anar Nurizada, Rohit Dhaipule, Zhijie Lyu, and Anurag Purwar. 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, 2025a.

Anar Nurizada, Zhijie Lyu, and Anurag Purwar. Path generative model based on conditional $\beta$-variational auto encoder for four-bar mechanism design. *Journal of Mechanisms and Robotics*, 17 (6):061004, 2025b.

Zherong Pan, Min Liu, Xifeng Gao, and Dinesh Manocha. Joint search of optimal topology and trajectory for planar linkages. *The International Journal of Robotics Research*, 42(4-5):176–195, 2023.

Sebastian Sonntag, Vincent Brünjes, Janosch Luttmer, Burkhard Corves, and Arun Nagarajah. 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, page V007T07A003. American Society of Mechanical Engineers, 2024.

Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. Tslearn, a machine learning toolkit for time series data. *Journal of Machine Learning Research*, 21(118):1–6, 2020. URL `http://jmlr.org/papers/v21/20-091.html`.

# A Training and Data Details

We use the dataset of Nurizada et al. [2025a] consisting of single-DOF planar linkages with paired coupler curves. Our subset filters mechanism families with at least 20,000 instances, yielding 846,480 training samples and 83,499 validation examples across 24 topologies. The transformer totals $19$M parameters with hidden size 256, 6 encoder and 6 decoder layers, and 8 attention heads. Training uses AdamW, cosine decay with warm-up, cross-entropy loss with label smoothing, and runs for 30 epochs on DGX A100 GPUs. Code, processed data, and checkpoints accompany the submission.

# B Implementation Details

**Data Preprocessing.** The raw mechanism dataset[1] undergoes several preprocessing steps to create a suitable representation for sequence learning. Each mechanism in the dataset is first normalized to a canonical coordinate frame through a sequence of transformations that place the two ground (fixed) joints at standardized positions: the first at the origin $(0, 0)$ and the second at $(1, 0)$. This normalization eliminates the infinite variations due to translation, rotation, and scaling while preserving the essential kinematic relationships. For the target curves, we employ cubic B-spline fitting with a fixed number of 64 control points, providing a compact yet expressive representation that captures curve shapes while maintaining a consistent input dimension for the neural network. The continuous joint coordinates are discretized into 200 uniform bins spanning the range $[-10, 10]$, transforming the regression problem into a multi-class classification task that the Transformer can handle more effectively.

**Domain-Specific Language.** The DSL serializes each mechanism into a structured token sequence following a strict grammar. Each sequence begins with a start-of-sequence token, followed by the mechanism type declaration (e.g., `MECH_TYPE: RRRR`), then a `POINTS:` marker, and finally the quantized coordinates of each free joint. Each joint is represented as `P_i X: BIN_j Y: BIN_k`, where $i$ indexes the joint and $j, k$ are the bin indices for the x and y coordinates respectively. This structured format ensures that the model learns both the syntax and semantics of valid mechanism descriptions while maintaining interpretability.

**Model Architecture and Training.** We employ a standard Transformer encoder-decoder architecture. The encoder takes as input a sequence of 64 B-spline control points (each point represented as 2D coordinates) and processes them through six self-attention layers to build a contextualized representation of the target curve. The decoder autoregressively generates the DSL token sequence, starting from a start-of-sequence token and producing one token at a time through six masked self-attention layers with cross-attention to the encoder output. Each token prediction is made from a vocabulary of 232 tokens (including special tokens, DSL structure tokens, mechanism type tokens, and 200 coordinate bins). The architecture incorporates several modern improvements including flash attention for computational efficiency, RMSNorm for training stability, gated linear units (GLU) with Swish activation in the feedforward layers, rotary positional embeddings for better length generalization, and QK normalization in attention layers. Table B summarizes all hyperparameters used in our experiments. Training is performed using distributed data parallel (DDP) across $8\times$ NVIDIA A100 GPUs. The training takes about one hour to complete. The complete training and inference code are made publicly available to facilitate reproducibility and future research.

[1] https://www.kaggle.com/datasets/purwarlab/four-six-and-eight-bar-mechanisms-with-curves

| Hyperparameter | Value |
|---|---|
| *Model Architecture* | |
| Hidden dimension ($d_{model}$) | 256 |
| Attention heads | 8 |
| Encoder layers | 6 |
| Decoder layers | 6 |
| Total parameters | $\sim$19M |
| *Training* | |
| Optimizer | Adam |
| Learning rate | $1 \times 10^{-4}$ |
| Weight decay | $1 \times 10^{-5}$ |
| Batch size (per GPU) | 256 |
| Total batch size | 2048 |
| Epochs | 30 |
| LR schedule | ReduceLROnPlateau |
| LR reduction factor | 0.5 |
| LR patience | 3 epochs |
| Gradient clipping | 1.0 |
| *Loss Function* | |
| Loss type | Cross-entropy |
| Padding token ignored | ✓ |
| *Data Processing* | |
| B-spline control points | 64 |
| B-spline degree | 3 |
| Coordinate range | $[-10, 10]$ |
| Coordinate bins | 200 |
| Min. instances per type | 20,000 |
| Train/validation split | 90/10 |
| *Architecture Features* | |
| Flash attention | ✓ |
| RMSNorm | ✓ |
| GLU feedforward | ✓ |
| Rotary position embeddings | ✓ |
| QK normalization | ✓ |
| Swish activation | ✓ |
| No bias in feedforward | ✓ |

Table 3: Hyperparameters used for training.

## C Coordinate Discretization Ablation

To determine the optimal discretization granularity for joint coordinates, we conducted an ablation study varying the number of bins $B$ used to quantize the continuous coordinate space $[-10, 10]$. We evaluated three bin sizes: $B \in \{50, 200, 2000\}$, specifically chosen to represent coarse, medium, and fine discretization levels respectively. These values span two orders of magnitude to comprehensively assess how quantization resolution influences model accuracy.

The bin size directly influences model accuracy through two competing effects. With $B = 50$ (bin width = 0.4 units), the coarse quantization introduces substantial discretization error—each predicted coordinate can only take one of 50 possible values, limiting the precision with which joint positions can be specified. This manifests as poor reconstruction accuracy (DTW = 8.0628) since the model cannot place joints with sufficient precision to accurately trace the target curves.

At the opposite extreme, $B = 2000$ (bin width = 0.01 units) provides high spatial resolution but paradoxically yields the worst performance (DTW = 12.2427). This degradation occurs because fine-grained discretization creates a sparse, high-dimensional output space where each of the 2000 bins appears infrequently in the training data. The model struggles to learn robust patterns across this

sparse categorical distribution, leading to poor generalization despite the theoretical capability for precise coordinate specification.

The optimal configuration at $B = 200$ (bin width = 0.1 units) balances these competing factors. It provides sufficient spatial resolution for accurate joint placement while maintaining a learnable output distribution where each bin appears frequently enough in the training data for the model to learn meaningful patterns. This finding guided our choice of $B = 200$ for all experiments reported in the main paper, achieving a median DTW of 3.0900 that represents nearly $3\times$ improvement over either extreme.

| Bin Size | DTW Median |
|---|---|
| 50 | 8.0628 |
| 200 | **3.0900** |
| 2000 | 12.2427 |

Table 4: DTW median values for different bin sizes. We run these experiments with Best @ $k$ = 1 and 10 different samples.

## D Temperature Sampling

The purpose of this study was to identify an appropriate sampling temperature $T$ for generating mechanisms across experiments. Since temperature influences the stochasticity of the model, selecting a suitable value ensures both quality and diversity in generated outputs while avoiding performance degradation due to overly deterministic or overly random behavior. We evaluated the average best normalized DTW between the predicted and target trajectories across different temperature values $T \in \{0.001, 0.1, 0.5, 1.0\}$ and sampling counts $k \in \{1, 2, 4, 8\}$. As shown in Table 5, $T = 0.1$ achieved the best performance when $k = 1$. Furthermore, as $k$ increased, $T = 0.1$ remained competitive, with DTW scores comparable to other temperatures. Based on this balance between quality and consistency, we selected $T = 0.1$ as the default temperature for all experiments.

| Temperature | k = 1 | k = 2 | k = 4 | k = 8 |
|---|---|---|---|---|
| T = 0.001 | 6.8652 | 6.8652 | 6.8652 | 6.8652 |
| T = 0.1 | **6.8316** | 5.3570 | 4.2911 | 3.6301 |
| T = 0.5 | 7.5084 | **5.2166** | **3.8533** | **3.1371** |
| T = 1.0 | 9.6690 | 6.3901 | 4.5011 | 3.4234 |

Table 5: Average best normalized DTW for different values of sampling temperature $T$ and number of samples $k$. Lower is better.

## E Distributions

Figure 3 shows the distribution of DTW scores for different sampling counts $k$, plotted as a boxplot on a linear scale. Each box illustrates the interquartile range (IQR), with the central line indicating the median DTW value for each $k$. We chose to highlight median values in our main text over means due to the presence of large outliers, which can significantly skew the average and misrepresent the typical performance. The boxplot makes this effect clear, particularly at lower $k$ values where a few poorly performing samples inflate the upper range. The median provides a more robust summary statistic under these conditions, capturing the central tendency of the distribution more accurately.

## F L-BFGS-B

In this study, we use the best outcomes from the Best @ $k$ = 32 study, and search through the joint spaces for each of the free joints. This approach does not search through different topologies, as these are non-differentiable values. We initiate this problem using the following optimization statement:

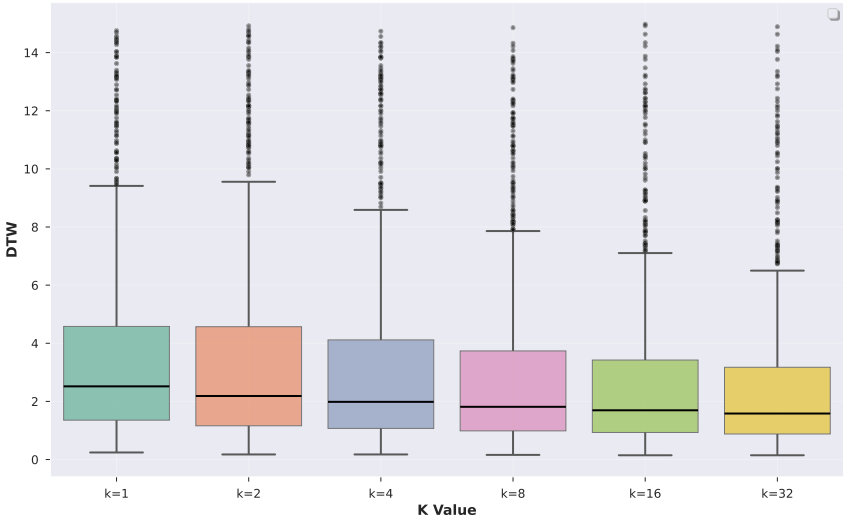


Figure 3: DTW distribution across different values of $k$. Each box shows the interquartile range with the median marked. Extreme outliers are present and motivate comparisons across medians.

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \text{DTW}(\mathcal{N}(\mathcal{X}_I), \mathcal{N}(\mathcal{X}_C(\mathbf{x}))) \\
\text{subject to} \quad & x_i \in [x_i^0 - \delta_i, x_i^0 + \delta_i], \quad i \in \{1, \ldots, n\} \\
& \mathbf{j}'_{g_1} = (0, 0) \\
& \mathbf{j}'_{g_2} = (1, 0) \\
\text{where} \quad & \delta_i = \max(0.5, 0.5|x_i^0|), \\
& \mathcal{N}(\mathcal{X}) = \frac{\mathcal{X}_I - \mu_I}{\sigma_{RMS_I}}
\end{aligned}
$$

as defined by: $\mathbf{x}$ is the vector of mechanism coordinates to be optimized, $\mathcal{X}_I$ is the input trajectory curve, $\mathcal{X}_C(\mathbf{x})$ is the coupler trajectory for coordinates $\mathbf{x}$, $\mathcal{N}(\cdot)$ is the normalization function, $\mu$ is the mean of the input trajectory points, $\sigma_{RMS_I}$ is the RMS variance of the input trajectory curve, $x_i^0$ are the initial coordinate values, $\delta_i$ are the bounds for each coordinate, and $\mathbf{j}'_{g_1}$ and $\mathbf{j}'_{g_2}$ are fixed ground points at $(0, 0)$ and $(1, 0)$ respectively.

This optimization routine is solved using the L-BFGS-B algorithm with the following parameters:

$$
\begin{aligned}
\texttt{maxiter} &= 50 \\
\texttt{maxfun} &= 100 \\
\texttt{ftol} &= 10^{-6} \\
\texttt{gtol} &= 10^{-6} \\
\texttt{eps} &= 10^{-3}
\end{aligned}
$$

## G Examples

Figure 4 presents nine representative examples of generated mechanisms along with their corresponding target (input) and generated (output) trajectories. Each subplot displays a unique mechanism sample, visualizing the coupler path traced by the mechanism in relation to the desired curve. The DTW score is annotated in each plot, providing a quantitative measure of trajectory alignment. We deliberately selected a range of samples with varying performance levels to highlight the diversity in accuracy: from high-performing mechanisms with low DTW values (green) to low-performing

ones with large trajectory mismatches (red). Figure 5 shows 100 examples of different curves in the dataset used to train our model.

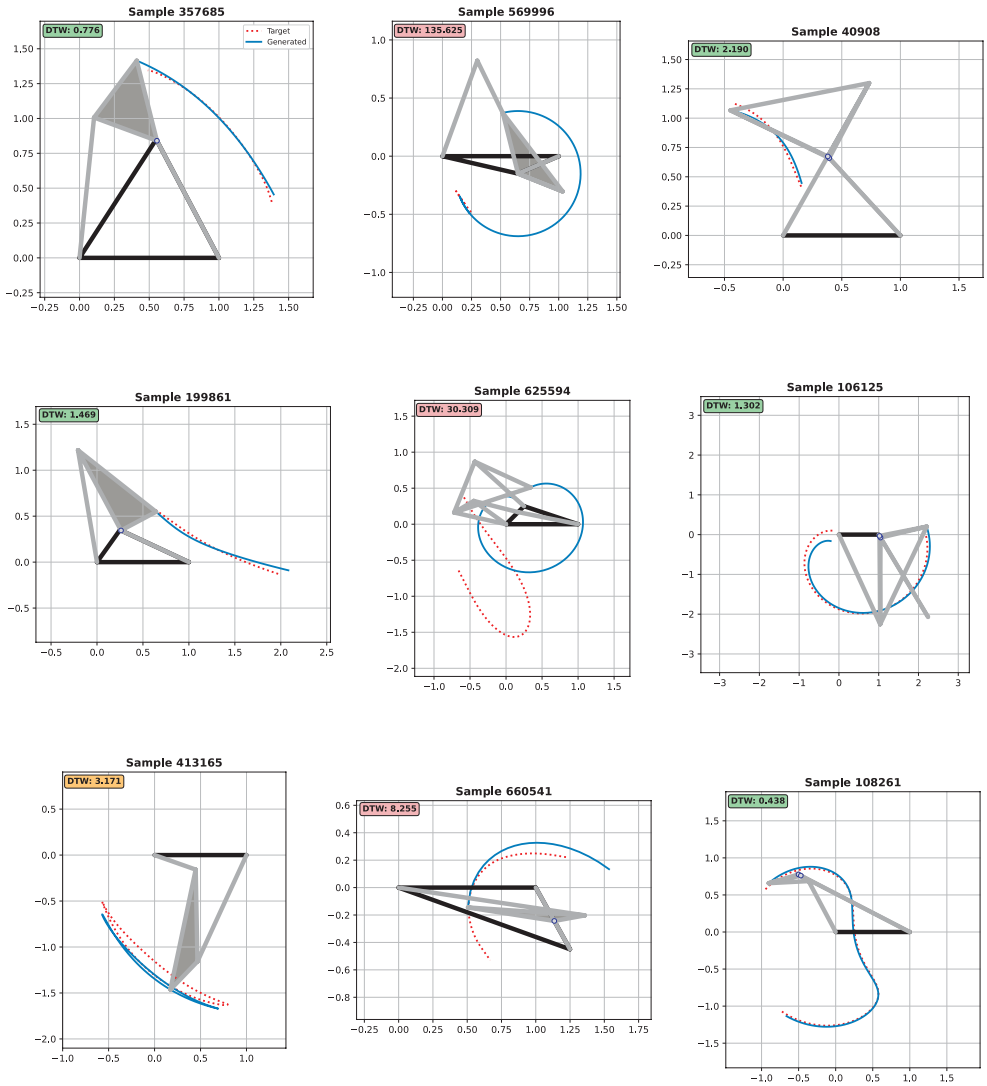


Figure 4: Nine sample mechanisms and their output trajectories in addition to the input curve. High, medium, and low accuracy performing outcomes are represented.

## H Additional Figures

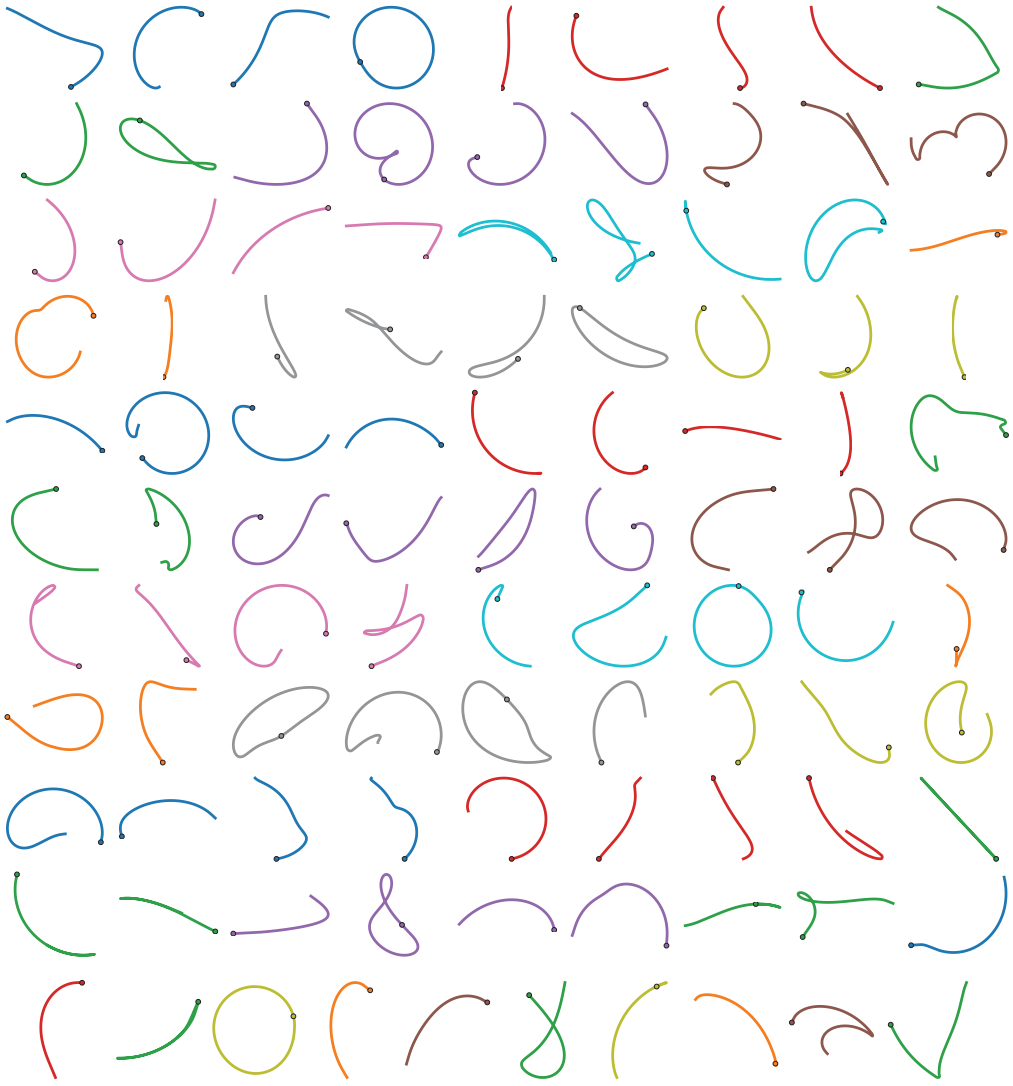

Figure 5: 100 sample curves used in the dataset.

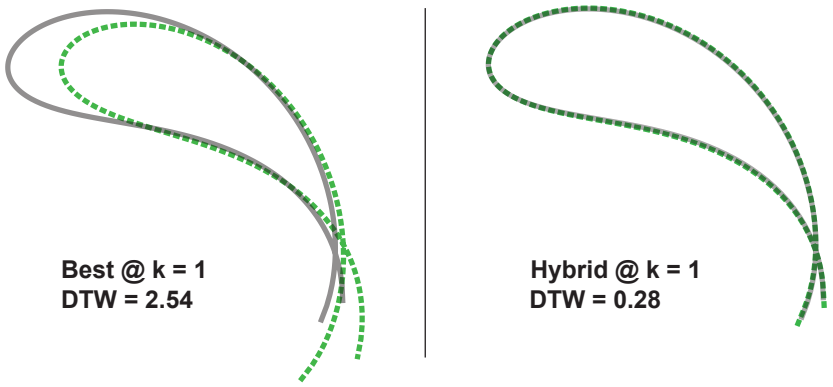


Figure 6: An example of curve alignment before and after implementing the optimization routine.

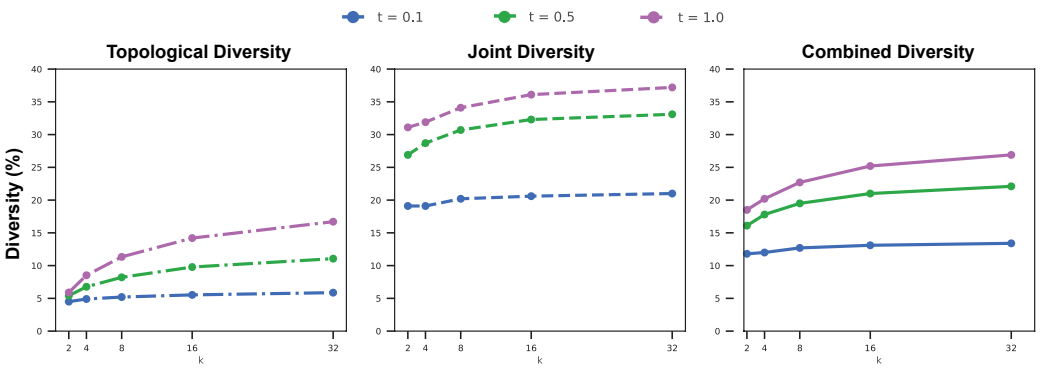


Figure 7: Topological, joint, and combined diversity trends across varying model temperatures and $k$-sampled evaluations. The Combined Diversity metric is the mean of the Topological and Joint Diversity values.