Creative Synthesis of Kinematic Mechanisms

Jiong Lin * **Jialong Ning** **Judah Goldfeder** **Hod Lipson** *
Creative Machines Lab, Columbia University, New York, NY 10027
https://jl6017.github.io/GenMech/

Abstract

In this paper, we formulate the problem of kinematic synthesis for planar linkages as a cross-domain image generation task. We develop a planar linkages dataset using RGB image representations, covering a range of mechanisms: from simple types such as crank-rocker and crank-slider to more complex eight-bar linkages like Jansen's mechanism. A shared-latent variational autoencoder (VAE) is employed to explore the potential of image generative models for synthesizing unseen motion curves and simulating novel kinematics. By encoding the drawing speed of trajectory points as color gradients, the same architecture also supports kinematic synthesis conditioned on both trajectory shape and velocity profiles. We validate our method on three datasets of increasing complexity: a standard four-bar linkage set, a mixed set of four-bar and crank-slider mechanisms, and a complex set including multi-loop mechanisms. Preliminary results demonstrate the effectiveness of image-based representations for generative mechanical design, showing that mechanisms with revolute and prismatic joints, and potentially cams and gears, can be represented and synthesized within a unified image generation framework. Code and dataset are available at: GitHub, Hugging Face.

1 Introduction

Kinematic synthesis is a long-standing problem in mechanical engineering. The objective is to design mechanisms that can generate a desired motion trajectory, using mechanical components such as links, sliders, gears, and cams. Within this broad area, the synthesis of planar linkages represents an important but still challenging subproblem. Classic examples such as the Watt's and Stephenson's linkages [24, 2], as well as more complex designs like Jansen's mechanism [26], illustrate how carefully designed linkages can produce sophisticated and purposeful motion. However, the synthesis problem remains challenging due to the complex and irregular nature of the design space. This complexity arises from the need to ensure a specific degree of freedom, manage redundant components, avoid motion singularities [25], and account for multiple mechanisms that may produce the same trajectories [41]. These challenges span both the *analysis* perspective, simulating trajectories from given mechanisms, and *synthesis* perspective, generating mechanisms from motion curves. We demonstrate these symmetric processes in Figure 1.

Analytical approaches to kinematic synthesis date back to the 19th century, most notably with Burmester theory [3], which provides closed-form geometric solutions for linkage design under discrete pose constraints. With the advent of computational methods, optimization-based techniques such as evolutionary algorithms [20, 21] and interactive design tools like LinkEdit [1] have enabled the synthesis of more complex and non-intuitive mechanisms. More recently, the availability of large-scale datasets [12, 29] has enabled learning-based methods that train neural networks, including fully connected architectures, VAEs [13, 6, 30], and multi-modal encoders like CLIP [34, 28], to predict mechanisms from motion examples. However, existing methods often rely on task-specific

*Corresponding authors: jl6017@columbia.edu, hod.lipson@columbia.edu.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Creative AI Track.

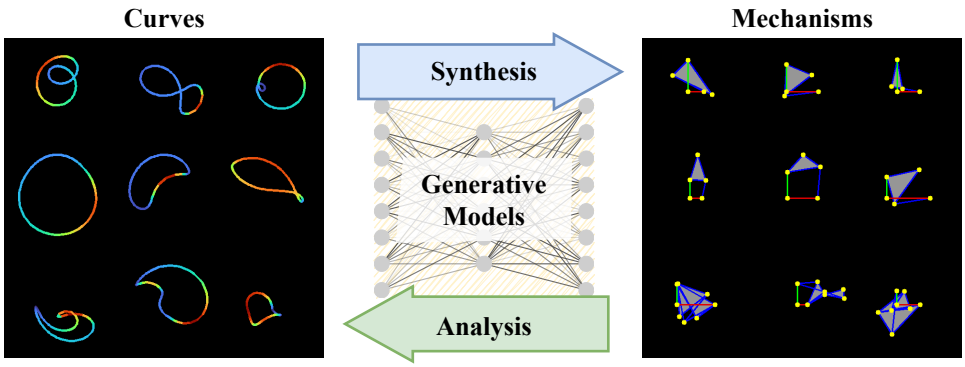

Figure 1: We propose an image based generative framework for kinematic synthesis and analysis. The model supports both synthesis and analysis using the same architecture and representation. Trajectory speed is encoded using color gradients (left), while types of links and joints are encoded using predefined colors (right).

data structures such as graphs or joint coordinate lists, which restrict generalization across different types of mechanisms. In contrast, image-based representations provide a unified and scalable format that naturally encodes both motion and structure, enabling the same model architecture and training pipeline to be applied across a wide variety of kinematic structures.

In this work, we propose a cross domain generative framework that jointly models mechanical structures and their motion trajectories. Trained on a curated dataset of paired images of mechanisms and curves, the model learns a shared latent representation that bridges the two domains. Once trained, it enables both kinematic synthesis and analysis. Our contributions are as follows: 1). **A new dataset** of paired RGB images of planar mechanisms and their corresponding motion trajectories, covering diverse linkage types. 2). **An end-to-end image generative model** that learns a shared latent representation across the mechanism and trajectory domains. 3). **Demonstration of cross domain synthesis and analysis**, enabling high fidelity translation from trajectory to mechanism and from mechanism to trajectory.

2 Related Work

Paper	Dataset	Curve Rep.	Mechanism Rep.	Method
Lipson [21] (2008)	Not fixed	Coordinates	Operator tree	Genetic Programming
Vermeer et al.[40] (2018)	Not fixed	Coordinates	Operator tree	Reinforcement Learning
Deshpande et al.[7] (2021)	4-links and 6-links	Image	Coordinates	VAE+KNN
Pan et al.[33] (2023)	Not fixed	Coordinates	Graph	Optimization
Fogelson et al.[9] (2023)	Max 16-links	Coordinates	Graph	GCN+Reinforcement Learning
Nobari et al.[12, 28] (2024)	Max 20-links	Coordinates	Graph	CLIP+BFGS
Lee et al.[18] (2024)	4-links	Image	Graph	cGAN
Nurizada et al.[30] (2025)	4-links & sliders	Image	Coordinates	β-VAE
Ours	Max 12-links & sliders	Image & Velocities	Image or Video	Shared-latent VAEs

Table 1: Summary of selected relevant kinematic synthesis papers.

Kinematic synthesis. Early work attempted to solve kinematic synthesis by interpolation. Simulation was used to generate a large database of mechanisms, and a neural network guided synthesis by interpolating a new mechanism from similar ones in the database [39]. Other approaches used a tree-based mechanism representation, relying on genetic programming [20] or reinforcement learning [9] to search the space of trees and find mechanisms with the desired properties.

Recent work has mostly focused on deep learning. A variety of generative models have been explored, including cGANs [18] and VAEs [30]. Contrastive Learning has also been applied to enable rapid retrieval from massive mechanism databases [28]. The rise of data driven approaches has further spurred the release of several high quality datasets [29, 12] and even attempts at 3D synthesis [4]. Related to synthesis, mechanism design tools have also been explored. One example is mechanism editing, whereby a user can interact with an existing mechanism, and the solver allows for the user to add new properties while specifying properties that need to be preserved [1]. Another

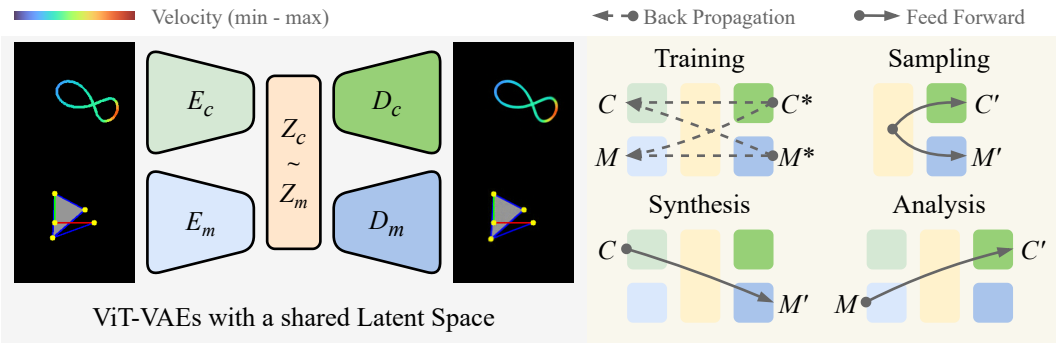

Figure 2: Overview of our shared-latent VAE framework for cross-domain kinematic synthesis and analysis. Curve and mechanism images (C, M) are encoded into latent embeddings (Z_c, Z_m), which are aligned in a shared latent space. During training, both reconstruction ($\hat{C}$, $\hat{M}$) and cross-domain prediction ($\hat{C}_m$, $\hat{M}_c$) are supervised via back-propagation. At inference time, the model enables synthesis (from C to M') and analysis (from M to C') through feed-forward decoding. Following MAE [10], we adopt an asymmetric ViT encoder–decoder design.

tool automated conversion from hand drawn sketches to digital representations [31]. Yet another presented an interactive design system for the creation of mechanical characters [5] and linkage based characters [38], and multi-stable structures [42].

Generative models have progressed from Variational Autoencoders (VAEs) [17], which learn probabilistic latent representations, to Diffusion Models such as DDPMs [14], which iteratively transform noise into data through denoising steps. Latent Diffusion Models (LDMs) [36] perform diffusion in a compressed latent space, while Flow Matching [19] offers a continuous-time formulation that learns transformation between distributions. Cross-domain generative models aim to translate data from one domain to another while preserving semantic consistency. Applications includes image-to-image translation [37, 16], few-shot domain adaptation [32], and cross-modal 3D synthesis [22, 23]. Multi-modal generative models extend this idea to learning joint representations across different modalities, such as video and audio [27]. By jointly modeling multiple domains or modalities, these methods enable richer synthesis capabilities and bidirectional translation between diverse input types. Previous works have primarily focused on datasets such as SVHN to MNIST [15] or style translation [35] between domains. In this work, we introduce kinematic synthesis as a new cross-domain generative modeling task, well-defined and application-relevant, and present a data generation pipeline with preliminary results on three datasets using the shared-latent VAEs.

3 Approach

3.1 Shared-latent VAEs

Overview of our method shown in Figure 2. Let C and M denote the input curve and mechanism images. E_c, E_m are the encoders, and D_c, D_m are the decoders for the curve and mechanism branches. The latent codes are $Z_c = E_c(C)$ and $Z_m = E_m(M)$, sampled from a shared latent distribution $Z \sim \mathcal{N}(0, I)$. The reconstructions are $\hat{C} = D_c(Z_c)$ and $\hat{M} = D_m(Z_m)$, while the cross-domain predictions are $\hat{M}_c = D_m(Z_c)$ and $\hat{C}_m = D_c(Z_m)$. The training loss is defined as:

$$\mathcal{L}_{\text{Shared-latent-VAEs}} = \|C - \hat{C}\|^2 + \|M - \hat{M}\|^2 + \beta \cdot [\text{KL}(q(Z_c|C)\|p(Z)) + \text{KL}(q(Z_m|M)\|p(Z))] + \lambda \cdot \|Z_c - Z_m\|^2 + \gamma \cdot (\|\hat{M}_c - M\|^2 + \|\hat{C}_m - C\|^2)$$

The loss function consists of five components. The **reconstruction loss** measures the image reconstruction error for both the curve and mechanism domains, ensuring that each encoder-decoder pair can accurately reproduce its input. The **KL divergence** term regularizes the latent codes Z_c and Z_m to follow a standard normal distribution, promoting smoothness and structure in the latent space. The **latent similarity loss** encourages the curve and mechanism embeddings to be close, facilitating a shared latent representation across domains. Lastly, the **cross-domain prediction loss** ensures that the latent codes contain sufficient information to decode meaningful counterparts in the other domain, supporting both synthesis and analysis tasks.

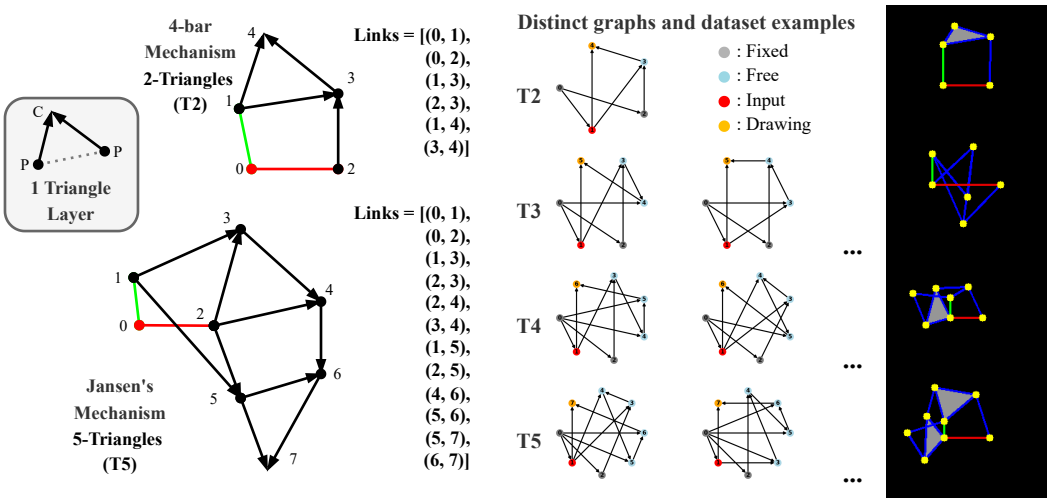

Figure 3: Examples of dataset construction. **Left:** Example mechanisms (4-bar mechanism and Jansen's linkage) built with 2 and 5 triangle layers, with their corresponding sequences of link connections. **Right:** For each complexity level (T2–T5), two representative graphs from different isomorphism classes are shown, along with rendered mechanism examples from the dataset.

3.2 Dataset

We construct a dataset of 1-DOF planar mechanisms by recursively applying triangle-based operators, starting from two fixed joints and one input joint, as shown in Figure 3. There are multiple operators to preserve the 1-DOF property while expanding the mechanism, such as T-operator and D-operator [21]. Following recent work, we use the J-operator from the LINKS dataset [12] and the recurrent triangle solver introduced in LinkEdit [1] to build mechanisms stacking triangle layers. Our goal is to include historically significant mechanisms such as Watt's, Stephenson's, and Jansen's, all of which share key properties: they begin with two fixed nodes, have one drawing node, and can be constructed with triangle layers. [†]

To generate a compact yet expressive dataset spanning from simple four-bar linkages to complex mechanisms such as Jansen's, we start with all possible combinations and then apply a set of structural filters. We begin with three initial nodes and recursively add one node at each layer by selecting two existing nodes as parents. This results in a total of $\prod_{i=3}^{k+2} \binom{i}{2}$ combinations for k triangle layers, which corresponds to the first row in Table 2. For example, when $k = 2$, the number of combinations is $\binom{3}{2} \cdot \binom{4}{2} = 18$; when $k = 5$, $\binom{3}{2} \cdot \binom{4}{2} \cdot \binom{5}{2} \cdot \binom{6}{2} \cdot \binom{7}{2} = 56{,}700$. We only retain mechanisms that satisfy the following conditions: 1) a single drawing node, since mechanisms with multiple output points can be described by simpler substructures; 2) no redundant triangles, as adding a triangle onto an already rigid structure does not increase structural variety; 3) exactly two fixed nodes, to match the properties of the target mechanisms and avoid introducing additional grounded points; and 4) one representative graph per isomorphic class, removing any structure that differs only by the sequence of construction. As a result, we include 396 distinct linkage structures, along with an equal number of crank-slider-based structures obtained by changing the starting point of graph construction.

Filters | Triangle layers	**T0**	**T1**	**T2**	**T3**	**T4**	**T5**	**Total**
Initial: All combinations	1	3	18	180	2700	56700	59602
Filter 1: One drawing node	0	1	5	31	257	2803	3097
Filter 2: Two fixed nodes	0	0	1	11	107	1227	1346
Filter 3: No redundant links	0	0	1	8	68	632	709
Filter 4: Isomorphic graphs	0	0	1	7	47	341	396

Table 2: Number of graphs retained after applying each structural filter, for the number of triangle layers from 0 (T0) to 5 (T5).

[†]Sub-variants such as Watt-II and Stephenson-III, which involve three fixed nodes, are not included in the current version dataset. The dataset can be extended to include these cases by adjusting the filtering parameters.

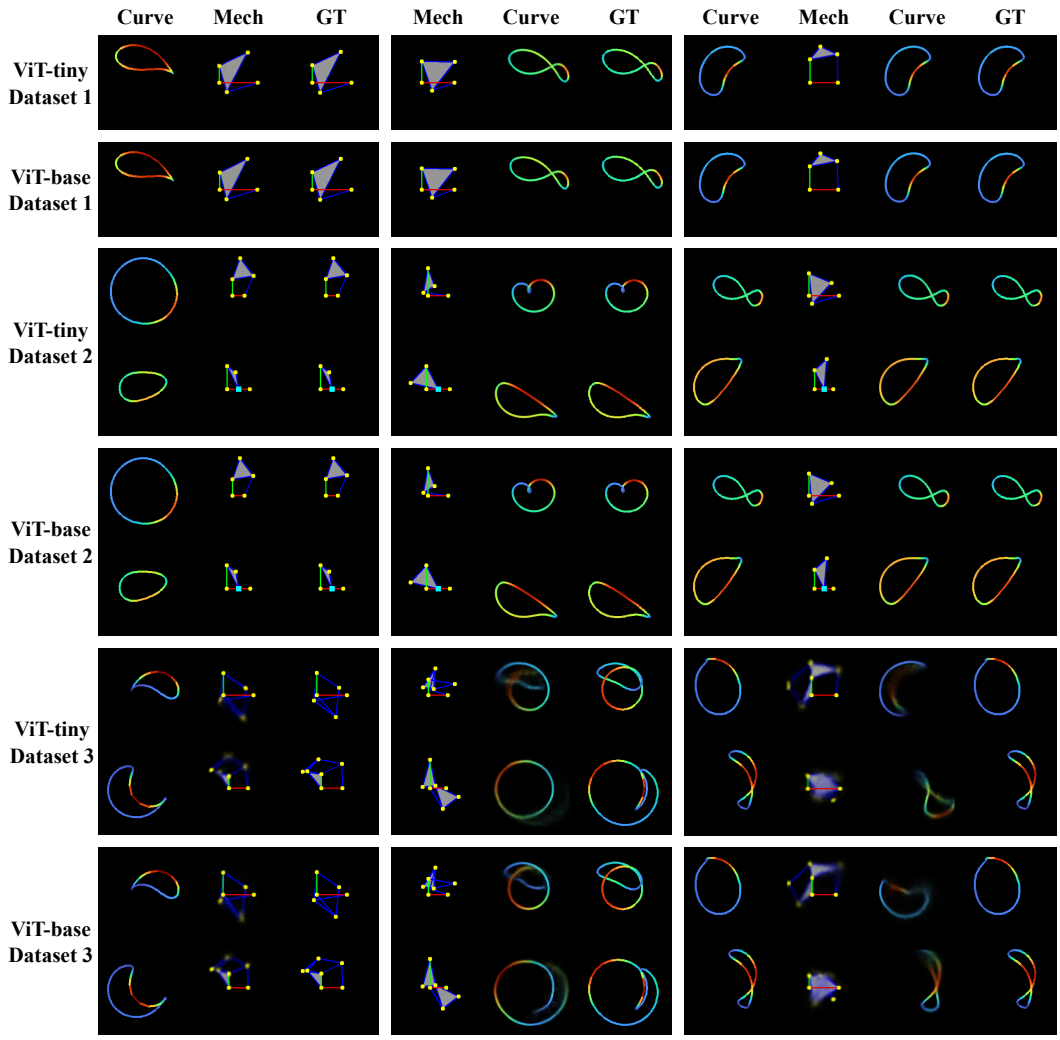

Figure 4: Qualitative results. Columns: (1) Curve to synthesized mechanism, compared with ground-truth mechanism; (2) Mechanism to calculated curve, shown beside the ground-truth curve; (3) Curve-to-mechanism-to-curve, compared with ground-truth curve. Rows list ViT-tiny and ViT-base [8] models on three datasets.

Color encoding is applied to both curve and mechanism images. For curves, the drawing-end speed is mapped to a color scale. For mechanisms, each component is assigned a predefined color: the base link is red, the input link is green, other links are blue, and joints are yellow. This encoding enables explicit validation when deriving parametric models. Inspired by the design of the *Strandbeest's* [26] walking leg, we observe that flattened contact points and steady horizontal motion of the output nodes promote stable forward movement and long locomotion overlap. Incorporating speed information into kinematic synthesis is both beneficial and easily integrated in RGB image representations.

4 Experiments

We open-source the data generation code and provide three example datasets used to validate the proposed shared-latent VAE for image-based synthesis and analysis. **Dataset-1** contains 100K four-bar mechanisms, including both crank–rocker and double–crank cases, referred to as T2. **Dataset-2** contains 200K mixed four-bar mechanisms and crank–slider mechanisms, referred to as T2ST2. **Dataset-3**, denoted as T4, is a subset of our complete collection of distinct graphs, comprising 55 graphs up to T4, with 10K samples collected for each graph. We repeated the data collection process and used the resulting test set for all three experiments.

Models	Dataset 1 (T2) PSNR ↑			Dataset 2 (T2 & ST2) PSNR ↑		
	Synthesis	Analysis	C2M2C	Synthesis	Analysis	C2M2C
ViT-tiny	27.63 ± 2.28	26.59 ± 2.21	26.17 ± 2.36	29.48 ± 3.27	27.97 ± 2.60	28.01 ± 2.72
ResNet-18	**28.58** ± 3.15	26.83 ± 2.61	27.03 ± 2.43	30.83 ± 4.54	28.20 ± 3.00	27.82 ± 3.34
ViT-base	28.54 ± 3.26	**26.90** ± 2.43	**27.18** ± 2.22	**30.87** ± 4.76	**28.43** ± 3.07	**28.40** ± 3.20

Table 3: **Baseline comparison.** PSNR (mean ± std) for image accuracy in synthesis, analysis, and curve-to-mechanism-to-curve (C2M2C) tasks, on two datasets for three models [8, 11].

Ablations	Dataset 1 (T2) PSNR ↑			Dataset 2 (T2 & ST2) PSNR ↑		
	Synthesis	Analysis	C2M2C	Synthesis	Analysis	C2M2C
Black-white input	25.61 ± 3.10	23.42 ± 2.40	22.86 ± 2.35	27.38 ± 3.91	24.70 ± 2.83	24.22 ± 3.24
w/o analysis loss	26.43 ± 1.87	12.50 ± 0.38	12.49 ± 0.38	29.03 ± 3.20	12.52 ± 0.33	12.52 ± 0.33
w/o synthesis loss	12.41 ± 0.29	26.30 ± 1.89	18.55 ± 1.46	12.55 ± 0.30	27.01 ± 2.41	18.71 ± 1.28
w/o cross-domain loss	18.21 ± 0.96	19.22 ± 1.17	18.63 ± 1.73	18.61 ± 1.26	18.91 ± 1.57	19.03 ± 1.56
Complete loss (ViT-tiny)	**27.63** ± 2.28	**26.59** ± 2.21	**26.17** ± 2.36	**29.48** ± 3.27	**27.97** ± 2.60	**28.01** ± 2.72

Table 4: **Ablation study.** Effect of input color and loss function terms on PSNR (mean ± std) for synthesis, analysis, and C2M2C tasks.

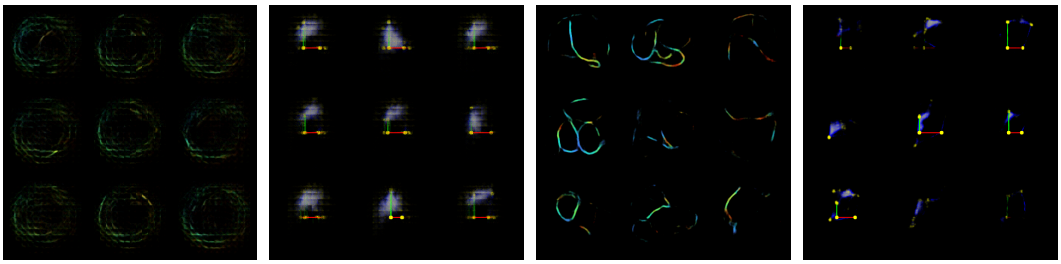

Figure 5: Samples from the latent space, shown in a 3×3 grid: from left to right: ViT-base with $\beta = 10$ (curves, mechanisms) and CNN with $\beta = 1$ (curves, mechanisms).

Qualitative results. We validate our method on three tasks: curve-to-mechanism synthesis, mechanism-to-curve analysis, and a two-stage generation where synthesis is followed by analysis. Figure 4 presents results for all three tasks across the three datasets. ViT-base generally produces more accurate outputs than ViT-tiny, particularly as the complexity of the mechanisms increases.

Quantitative results. Tables 3 and 4 report PSNR for all tasks. ViT-base achieves the highest scores across datasets compared with ResNet and ViT-tiny, especially in two-stage generation. Ablations show that removing synthesis or cross-domain losses significantly reduces accuracy, and color input outperforms black-and-white. The complete loss configuration yields the best overall performance.

5 Discussion

Limitations and future work. Figure 5 shows latent space samples for curves and mechanisms using ViT-base with $\beta = 10$ and ResNet with $\beta = 1$. The current sampling quality is limited, with blurred outputs and low-quality edges. This suggests that the VAE structure with high-capacity decoders may not have learned a well-structured normal distribution in the latent space.

In future work, large-scale datasets and more advanced generative models are encouraged for this task. In our data generation pipeline, we provide the optional recordings of joint coordinates and videos of one period's mechanism motion, enabling future exploration of image-to-graph and image-to-video generation. Image-based representations for complex mechanism structures suffer from occlusion and ambiguity, whereas video representations can provide richer temporal and structural information.

Conclusion. In this paper, we introduced kinematic synthesis as a cross-domain generative modeling task and proposed a shared-latent VAE framework for translating between mechanisms and motion curves. We constructed a scalable dataset covering simple to complex planar mechanisms and demonstrated synthesis, analysis, and two-stage generation on multiple datasets. Preliminary results show that image-based representations enable a unified representation and end-to-end generative synthesis of kinematic mechanisms. This work lays the groundwork for applying generative modeling into mechanism design, with potential applications in robotics.

Acknowledgments: This work was supported in part by the US National Science Foundation AI Institute for Dynamical Systems (DynamicsAI.org) (grant no. 2112085). The author thanks Jinchen Ruan for designing the hardware machine for the real-world validation.

References

[1] M. Bächer, S. Coros, and B. Thomaszewski. Linkedit: Interactive linkage editing using symbolic kinematics. *ACM Transactions on Graphics (TOG)*, 34(4):1–8, 2015.

[2] S. Bai, D. Wang, and H. Dong. A unified formulation for dimensional synthesis of stephenson linkages. *Journal of Mechanisms and Robotics*, 8(4):041009, 2016.

[3] M. Ceccarelli and T. Koetsier. Burmester and allievi: A theory and its application for mechanism design at the end of 19th century. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 42568, pages 291–300, 2006.

[4] Y. Cheng, P. Song, Y. Lu, W. J. J. Chew, and L. Liu. Exact 3d path generation via 3d cam-linkage mechanisms. *ACM Transactions on Graphics (TOG)*, 41(6):1–13, 2022.

[5] S. Coros, B. Thomaszewski, G. Noris, S. Sueda, M. Forberg, R. W. Sumner, W. Matusik, and B. Bickel. Computational design of mechanical characters. *ACM Transactions on Graphics (TOG)*, 32(4):1–12, 2013.

[6] S. Deshpande and A. Purwar. Computational creativity via assisted variational synthesis of mechanisms using deep generative models. *Journal of Mechanical Design*, 141(12):121402, 2019.

[7] S. Deshpande and A. Purwar. An image-based approach to variational path synthesis of linkages. *Journal of Computing and Information Science in Engineering*, 21(2):021005, 2021.

[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] M. B. Fogelson, C. Tucker, and J. Cagan. Gcp-holo: Generating high-order linkage graphs for path synthesis. *Journal of Mechanical Design*, 145(7):073303, 2023.

[10] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] A. Heyrani Nobari, A. Srivastava, D. Gutfreund, and F. Ahmed. 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, page V03AT03A013. American Society of Mechanical Engineers, 2022.

[13] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[14] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[15] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. Pmlr, 2018.

[16] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim. Learning to discover cross-domain relations with generative adversarial networks. In *International conference on machine learning*, pages 1857–1865. Pmlr, 2017.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] S. Lee, J. Kim, and N. Kang. Deep generative model-based synthesis framework of four-bar linkage mechanisms with target conditions. *Journal of Computational Design and Engineering*, 11(5):318–332, October 2024.

[19] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[20] H. Lipson. A relaxation method for simulating the kinematics of compound nonlinear mechanisms. *Journal of Mechanical Design*, 128(4):719–728, 2006.

[21] H. Lipson. Evolutionary synthesis of kinematic mechanisms. *AI EDAM*, 22(3):195–205, 2008.

[22] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023.

[23] X. Long, Y.-C. Guo, C. Lin, Y. Liu, Z. Dou, L. Liu, Y. Ma, S.-H. Zhang, M. Habermann, C. Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9970–9980, 2024.

[24] Z. Luo, J. Shang, G. Wei, and L. Ren. A reconfigurable hybrid wheel-track mobile robot based on watt ii six-bar linkage. *Mechanism and Machine Theory*, 128:16–32, 2018.

[25] G. Maloisel, E. Knoop, B. Thomaszewski, M. Bächer, and S. Coros. Singularity-aware design optimization for multi-degree-of-freedom spatial linkages. *IEEE Robotics and Automation Letters*, 6(4):6585–6592, 2021.

[26] S. Nansai, N. Rojas, M. R. Elara, R. Sosa, and M. Iwase. On a jansen leg with multiple gait patterns for reconfigurable walking platforms. *Advances in Mechanical Engineering*, 7(3):1687814015573824, 2015.

[27] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, et al. Multimodal deep learning. In *ICML*, volume 11, pages 689–696, 2011.

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

[29] A. Nurizada, R. Dhaipule, Z. Lyu, and A. 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, 2025.

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

[31] A. Nurizada and A. Purwar. Transforming hand-drawn sketches of linkage mechanisms into their digital representation. *Journal of Computing and Information Science in Engineering*, 24(1):011010, 2024.

[32] U. Ojha, Y. Li, J. Lu, A. A. Efros, Y. J. Lee, E. Shechtman, and R. Zhang. Few-shot image generation via cross-domain correspondence. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10743–10752, 2021.

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

[34] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[35] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro, and D. Cohen-Or. Encoding in style: a stylegan encoder for image-to-image translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2287–2296, 2021.

[36] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[37] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *International Conference on Learning Representations*, 2017.

[38] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross. Computational design of linkage-based characters. *ACM Transactions on Graphics (TOG)*, 33(4):1–9, 2014.

[39] A. Vasiliu and B. Yannou. Dimensional synthesis of planar mechanisms using neural networks: application to path generator linkages. *Mechanism and Machine Theory*, 36(2):299–310, 2001.

[40] K. Vermeer, R. Kuppens, and J. Herder. Kinematic synthesis using reinforcement learning. In *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE)*, number V02AT03A009 in Volume 2A: 42nd Mechanisms and Robotics Conference, pages 1–12. American Society of Mechanical Engineers, 2018.

[41] S. Zarkandi. A novel optimization-based method to find multiple solutions for path synthesis of planar four-bar and slider-crank mechanisms. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 235(21):5385–5405, 2021.

[42] R. Zhang, T. Auzinger, and B. Bickel. Computational design of planar multistable compliant structures. *ACM Transactions on Graphics (TOG)*, 40(5):1–16, 2021.

A Real-World Validation

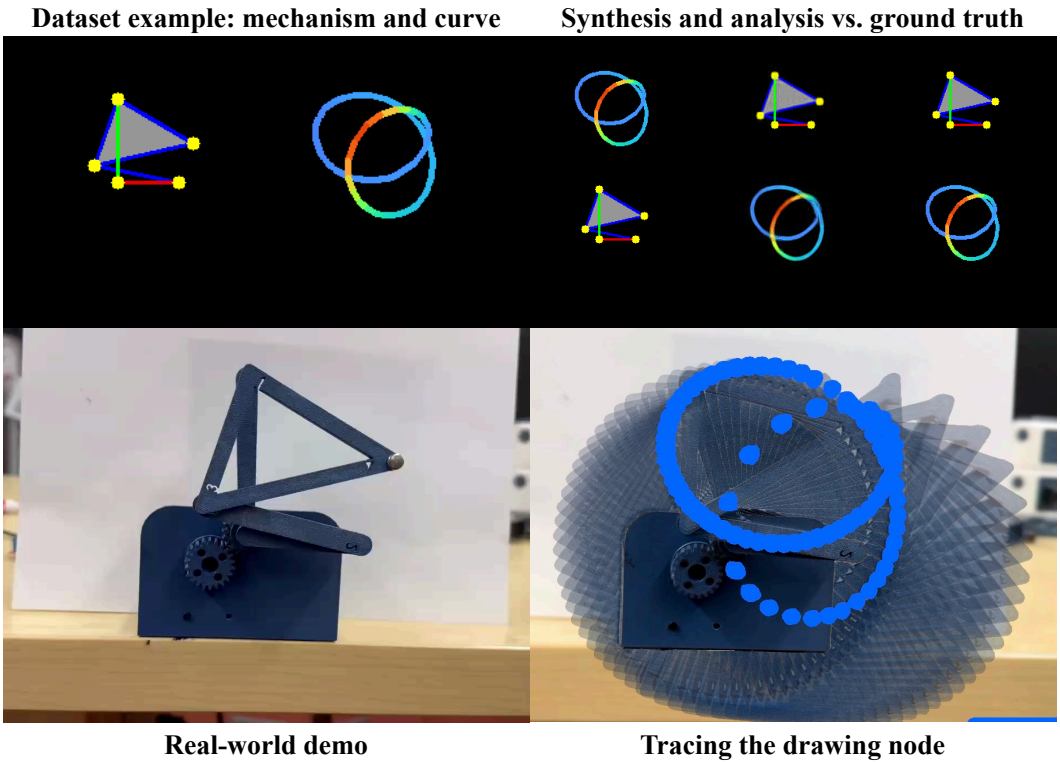

Figure 6: Examples of mechanism–curve pairs and real-world validation. Top left: a dataset example showing a mechanism and its corresponding drawing curve. Top right: synthesis (curve-to-mechanism) and analysis (mechanism-to-curve) results compared with ground truth, using the ViT-Tiny model. Bottom: a 3D-printed mechanism driven by a servo motor and the traced drawing trajectory of its end-effector.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the main contributions: dataset creation, shared-latent VAE framework, and bidirectional synthesis and simulation. Claims match the presented results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] .

   Justification: Discussed in Section 5 Discussion: limitations in sampling quality, edge sharpness, and possible occlusion issues in image-based representation; suggestions for larger datasets and better generative models.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA] .

Justification: The paper does not contain formal theorems or proofs

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes] .

Justification: Datasets, code, and implementation details are provided (Abstract line 16, Section 4 Experiments). Hyperparameters and dataset construction steps are described.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes] .

   Justification: Links to GitHub and Hugging Face are included in the abstract; dataset generation code is open-sourced.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] .

   Justification: Training/test splits, dataset sizes, model architectures (ViT-tiny, ViT-base, ResNet-18), loss terms, and evaluation metrics (PSNR with mean±std) are specified in Sections 3–4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes] .

   Justification: Results include mean and standard deviation for all quantitative metrics (Tables 3 and 4).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] .

   Justification: All experiments were conducted on a single NVIDIA RTX 5090 GPU. Training ViT-tiny on a 100k-sample dataset takes approximately 8 hours; ViT-base on the same dataset takes about 20 hours.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes] .

   Justification: The work uses synthetic data and does not involve human subjects, privacy-sensitive information, or unethical applications; it complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Positive impacts include advancing generative design for robotics and mechanical systems. Potential negative impacts involve unsafe or unreliable mechanisms if deployed without validation; open datasets are intended for research use only.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The released models and datasets have low misuse potential and do not contain harmful or sensitive content.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: External algorithms/tools (e.g., LINKS dataset, LinkEdit) are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: A new dataset of paired RGB mechanism–curve images is documented in Section 3.2 and released along with the dataset generation code under the MIT License. Instructions for use and licensing are included in the repository.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The work does not involve human subjects or crowdsourced data collection.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable; no human subjects are involved.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development does not involve LLMs as an original or non-standard component; LLMs were not used for the experiments or methodology.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.