Enhanced Kinematic Synthesis for Concept Generation with Generative and Sequential Models

Shrinath Deshpande

Mechanism Synthesis Pipeline

- Motion /Path /Function Generation Problem (Subjective)

Problem Setup (Requires problem understanding)

- Mathematically Precise Representation (Objective)

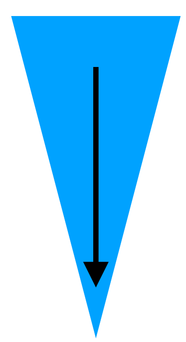
Problem Solving (Algorithmic, Nonlinear Mapping)

- Employ Numerical Solvers/ Optimizers

Problem Setup

Subjective Task

Problem Setup



Precise Problem

Why?

- Required by Solvers

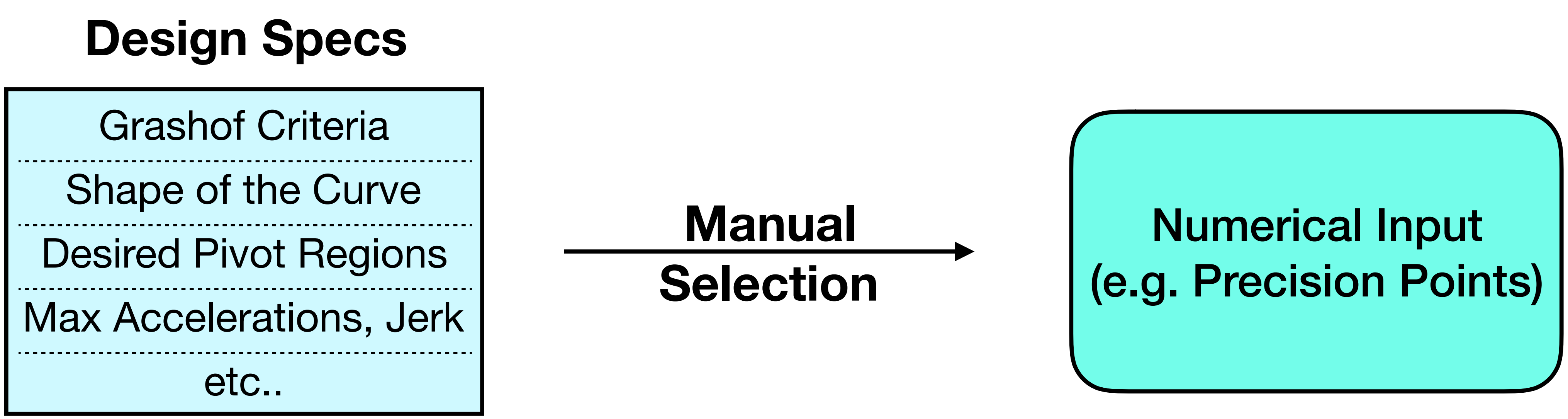
Cons

- Loss of Designer’s Intent (Information Loss)

- Defective Solutions

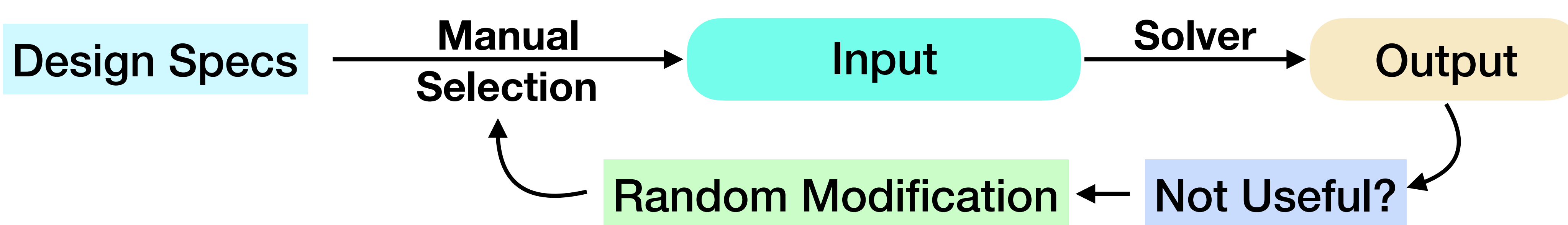
Problem Setup Goals

- Keep the designer's intent intact
  - Understand what aspects are important and Relax the less important constraints
- Keep it simple for solvers to operate
  - Numerical format



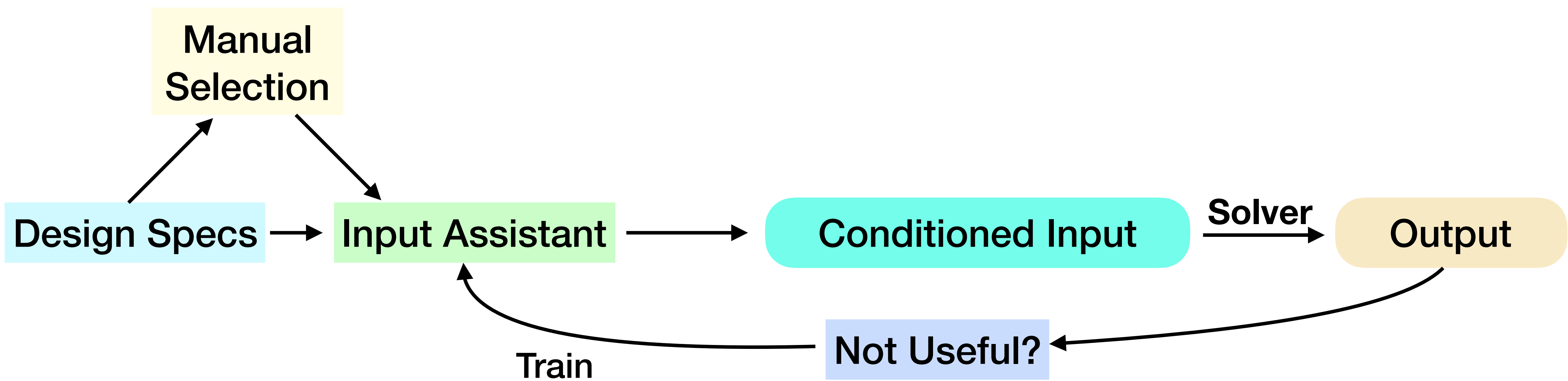
Difficulties in Setting Up the Problem

- Highly Nonlinear Mapping Between Numerical Input and Solver Results
  - Small Changes in precision poses can lead to very different coupler curves



Conditioned Input

- Exploit the nonlinear relationship between input and output
- Intelligently **modify** the Input to maximize the probability of usefulness of output
  - **Condition** the input to have some ‘learned’ properties
  - **Augment** the input with missing information/ free Choices



Conditional Precision Position Input

- X be the set of precision poses which results in useful output (e.g. set of 100 poses with dimension [100, 3])

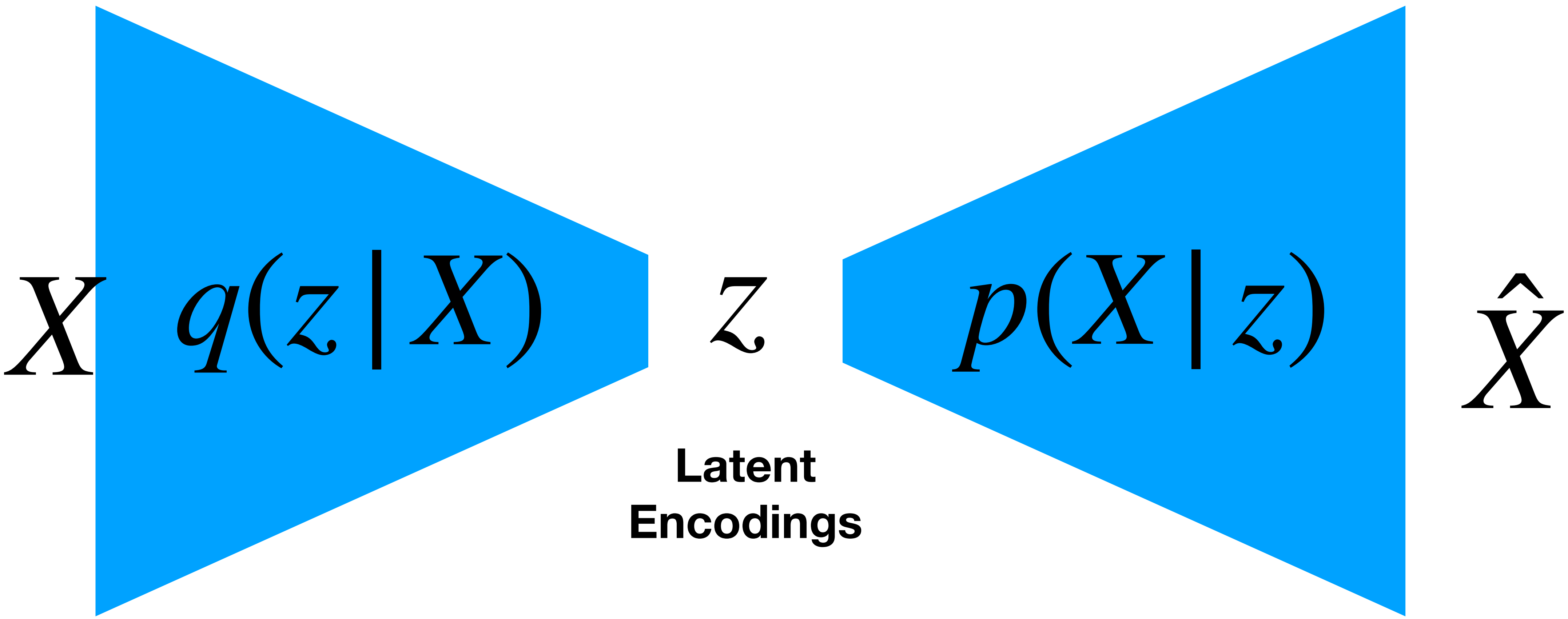
- Goal is to :
  1. approximate the model which generates the observed distribution of X.

$$X = G(z)$$

  2. Analytical approximation to the posterior probability of the unobserved variables z

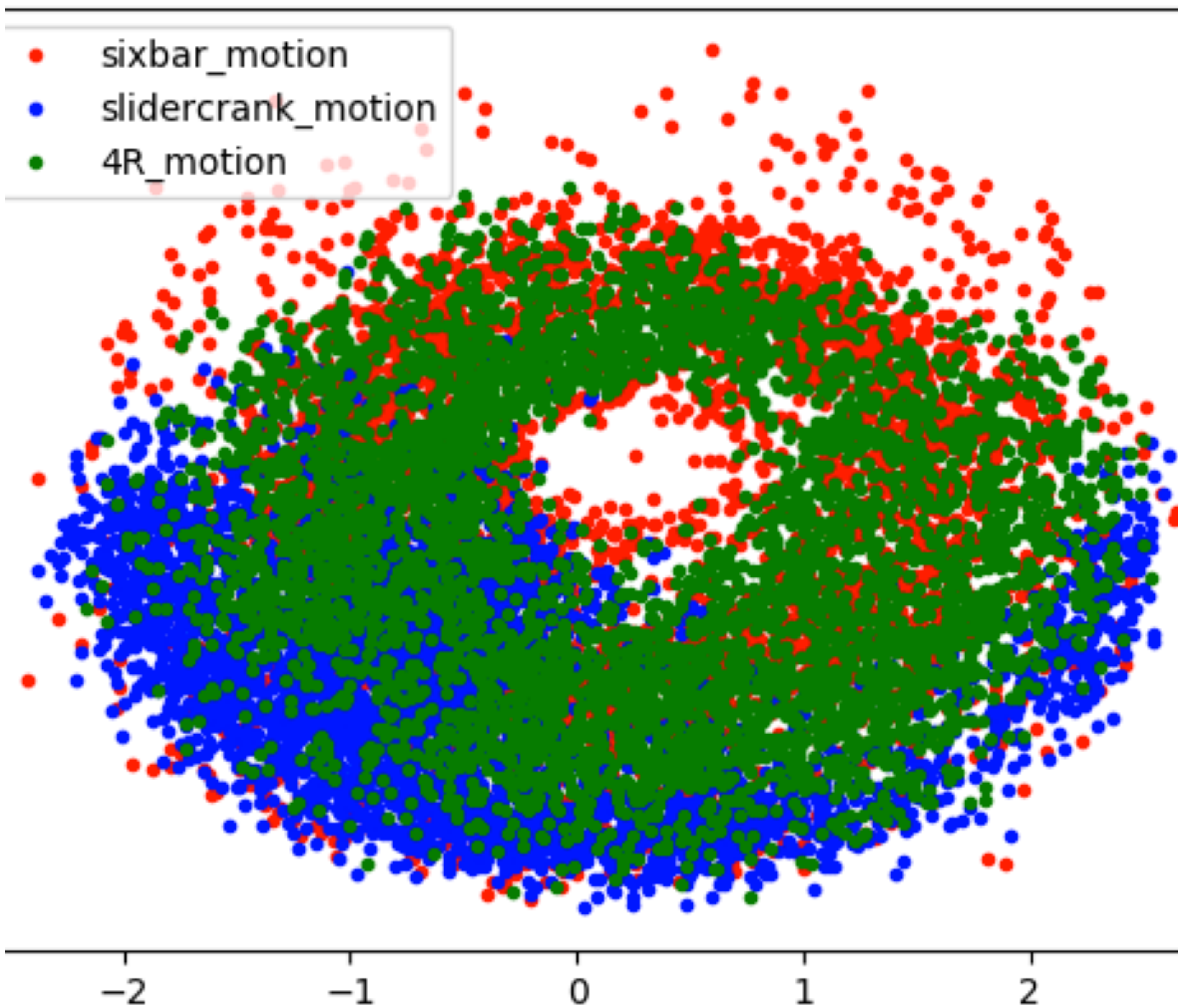
$$p(z \mid X) = \frac{p(X \mid z)p(z)}{p(X)} = F(X)$$

Variational Auto-Encoder



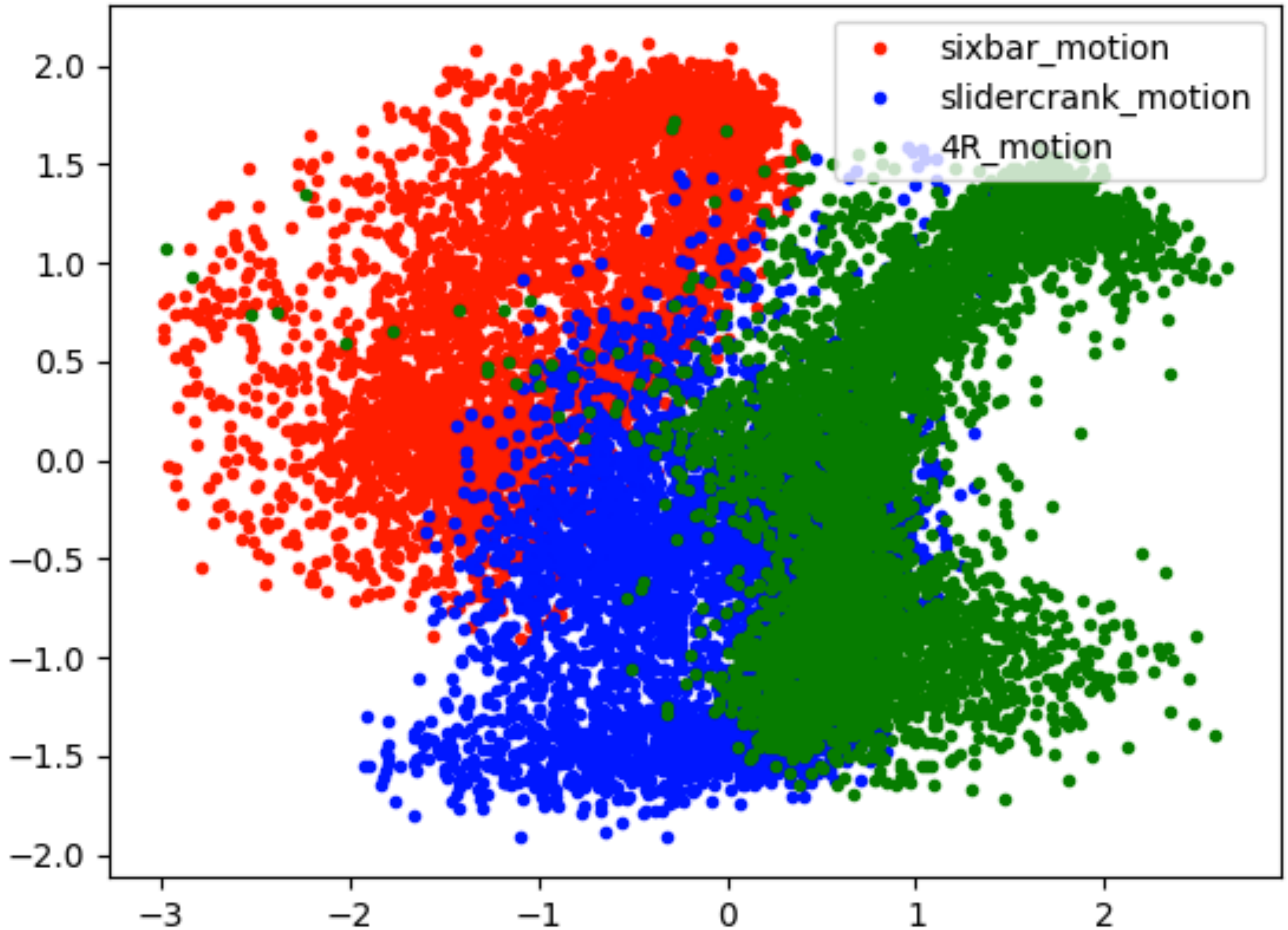
$$L = \underbrace{(X - \hat{X})^2}_{\textbf{Reconstruction Loss}} + \sum_j \underbrace{KL(q_j(z \mid X) \,||\, p(z))}_{\textbf{KL Divergence Loss}}$$

Latent Space For Coupler Path



- Latent Embeddings of coupler motions sampled four random Fourbar, Sixbar and Slider Crank linkages

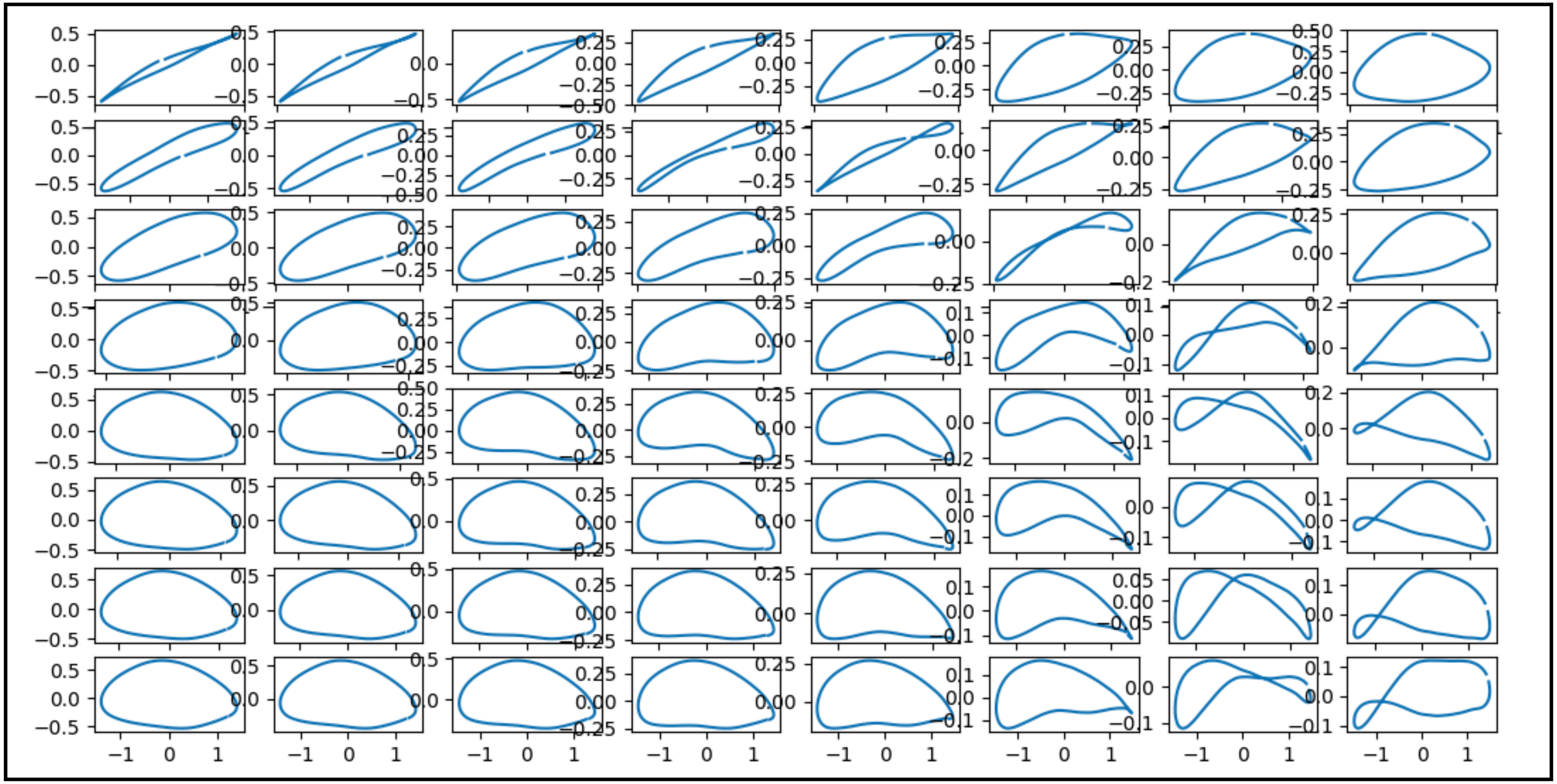
Latent Space For Coupler Motion



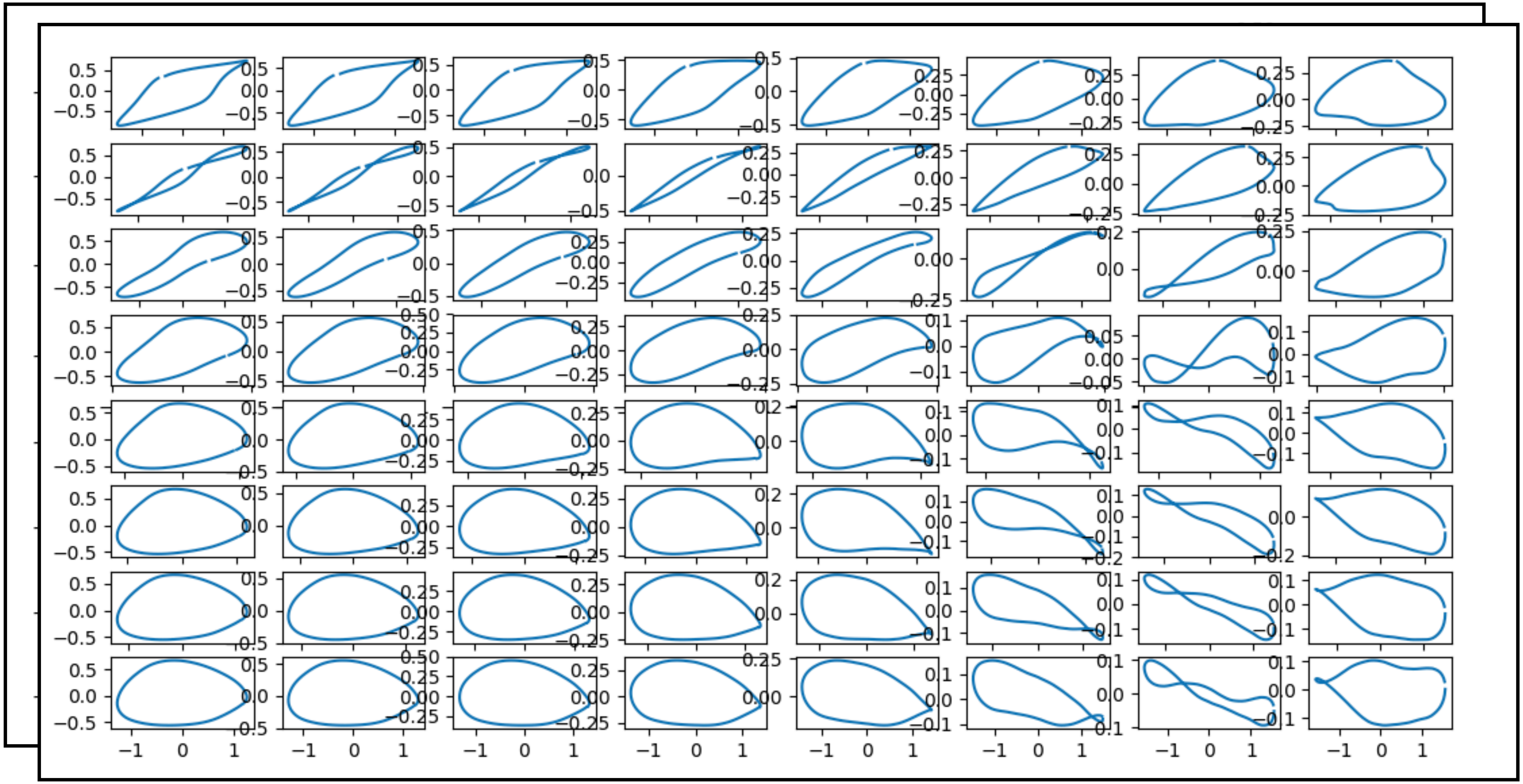
- Latent Embeddings of coupler motions sampled four random Fourbar, Sixbar and Slider Crank linkages

Distribution of Coupler Paths

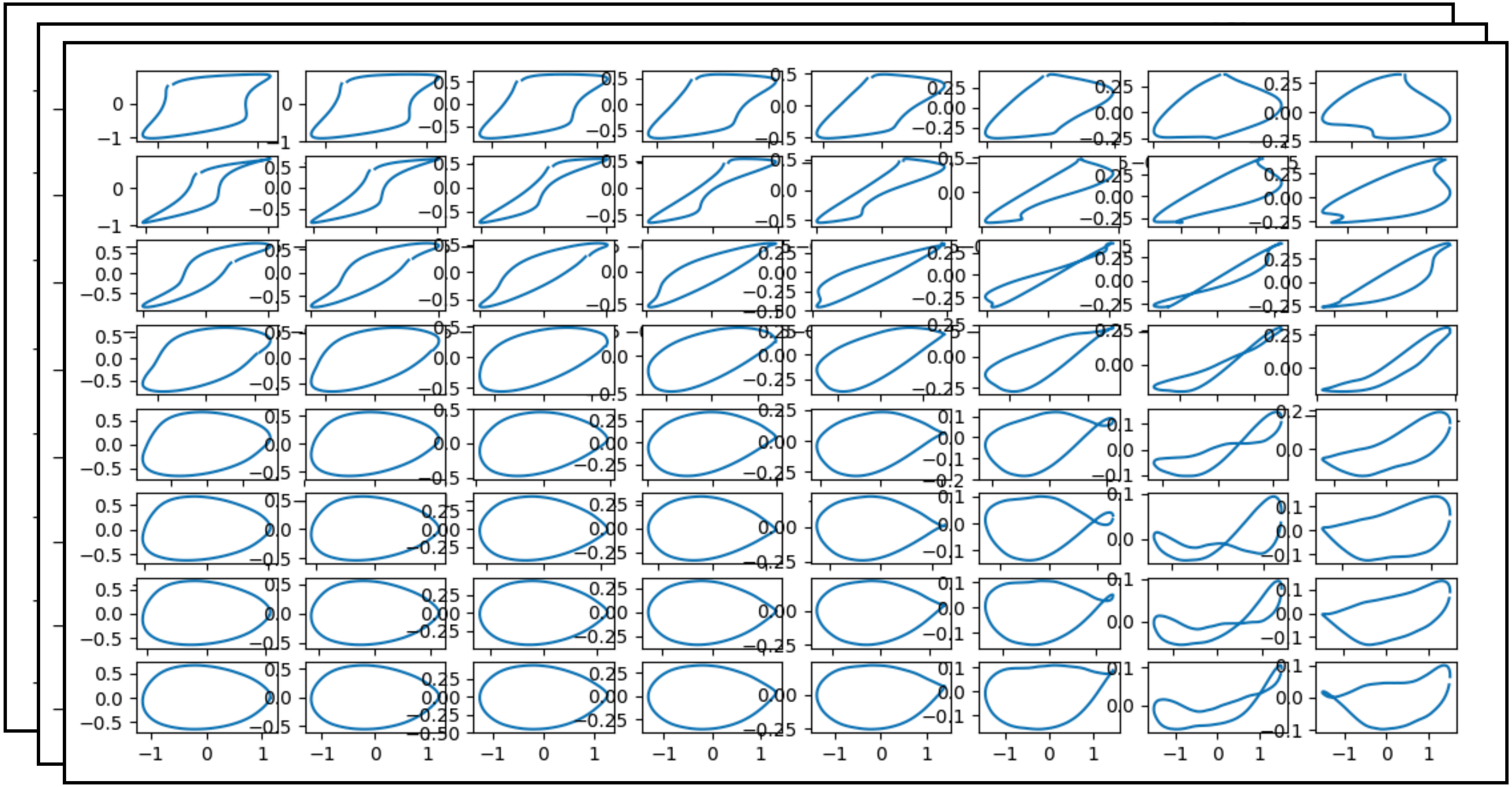
Distribution of Coupler Paths



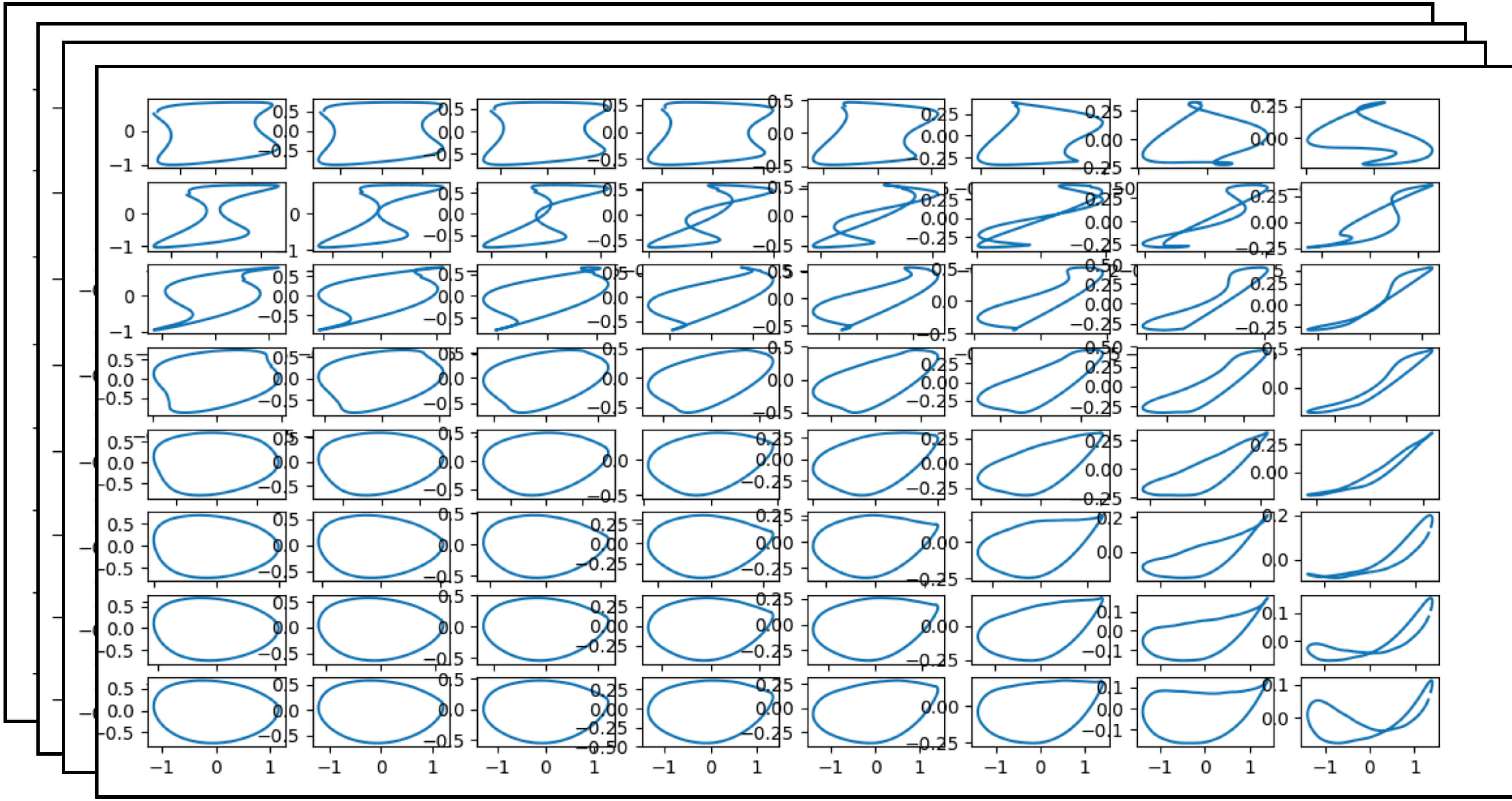
Distribution of Coupler Paths



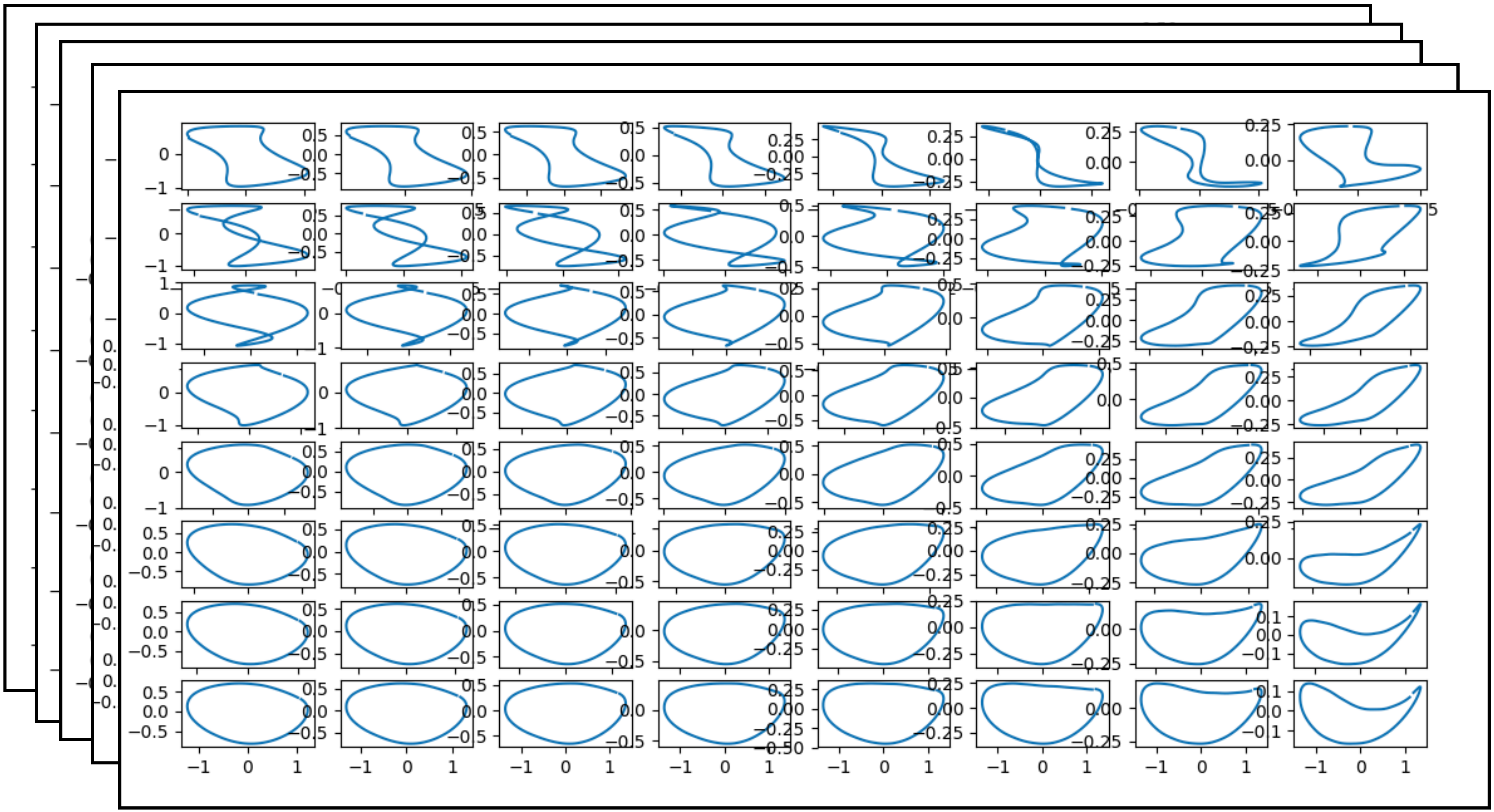
Distribution of Coupler Paths



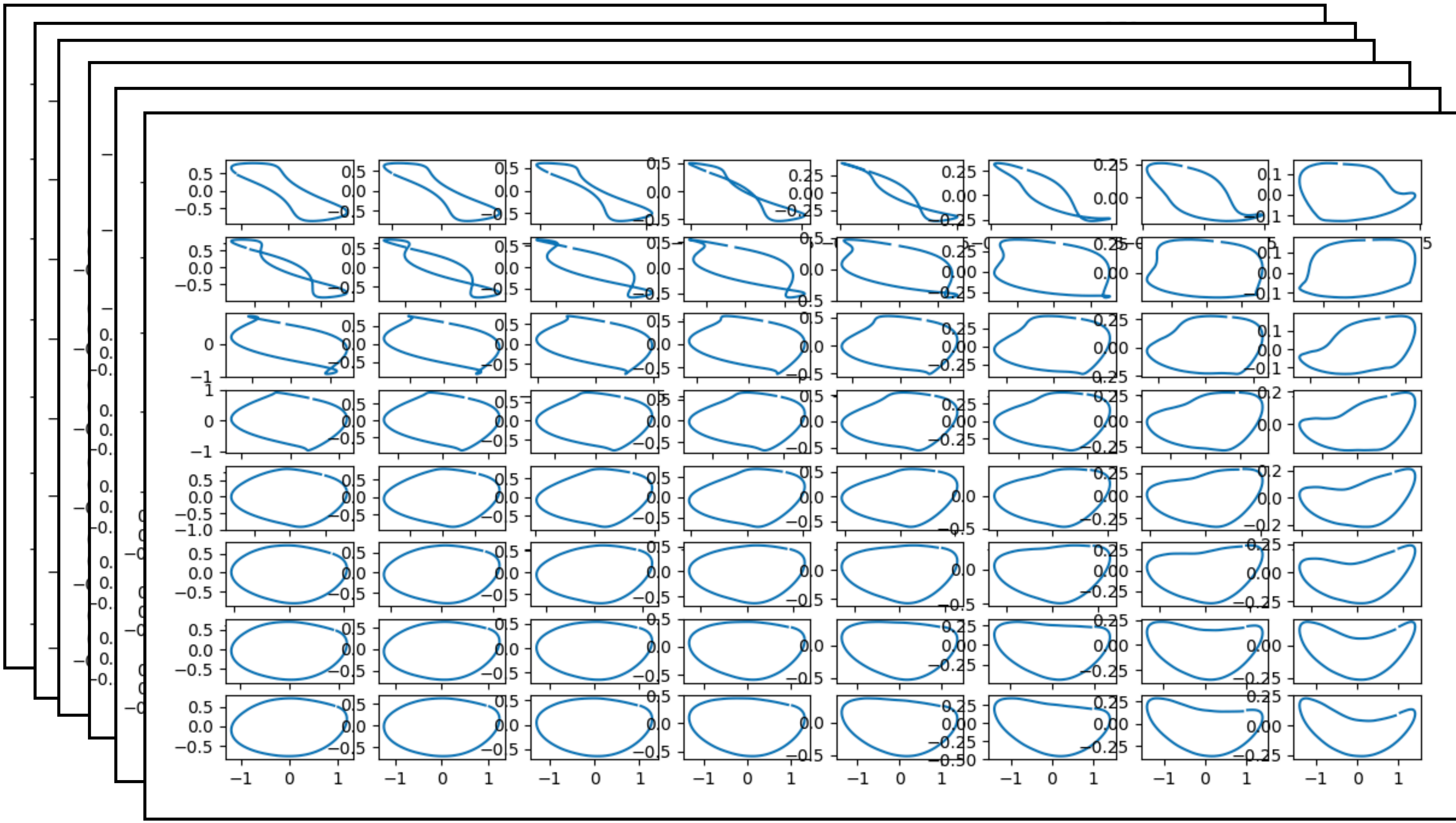
Distribution of Coupler Paths



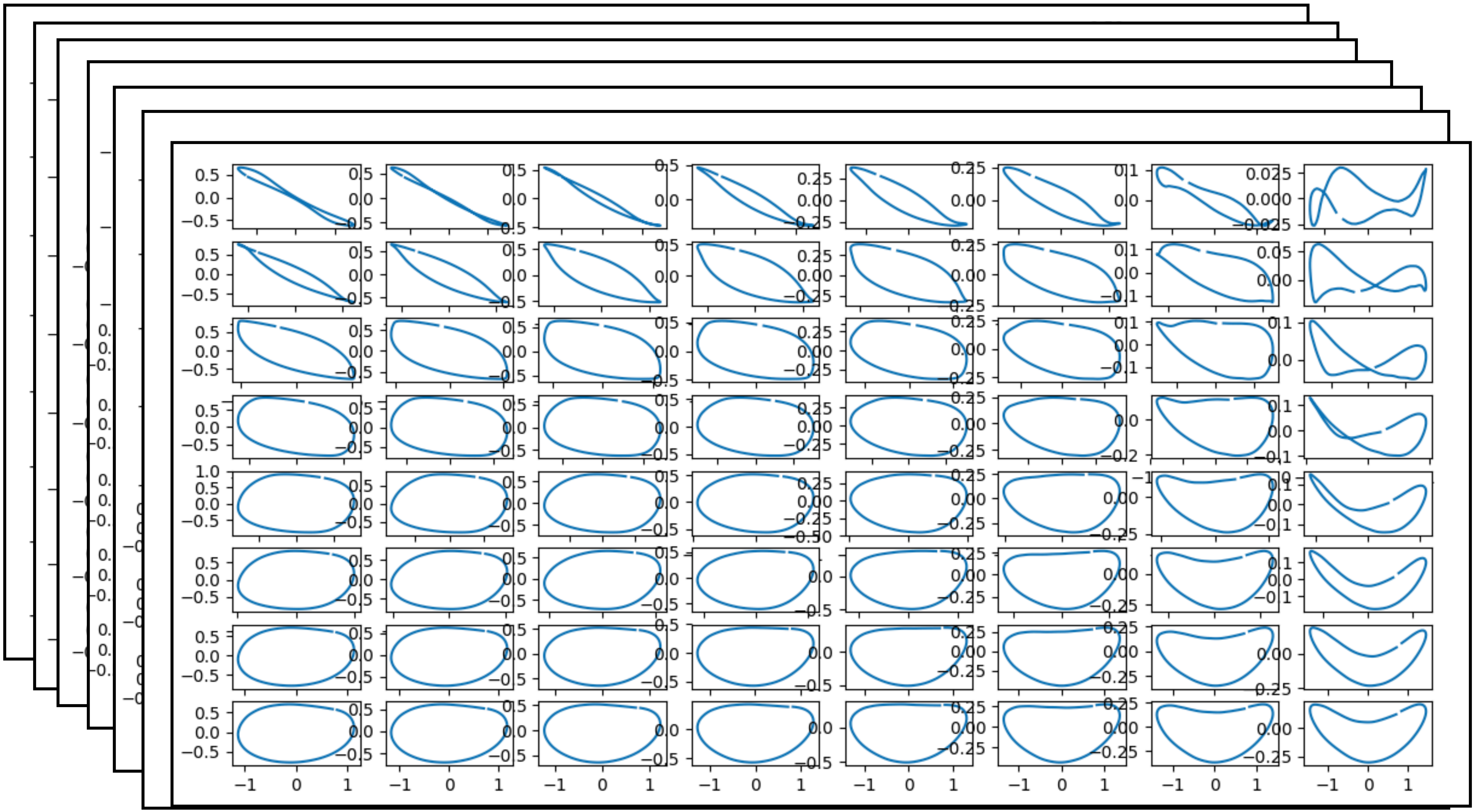
Distribution of Coupler Paths



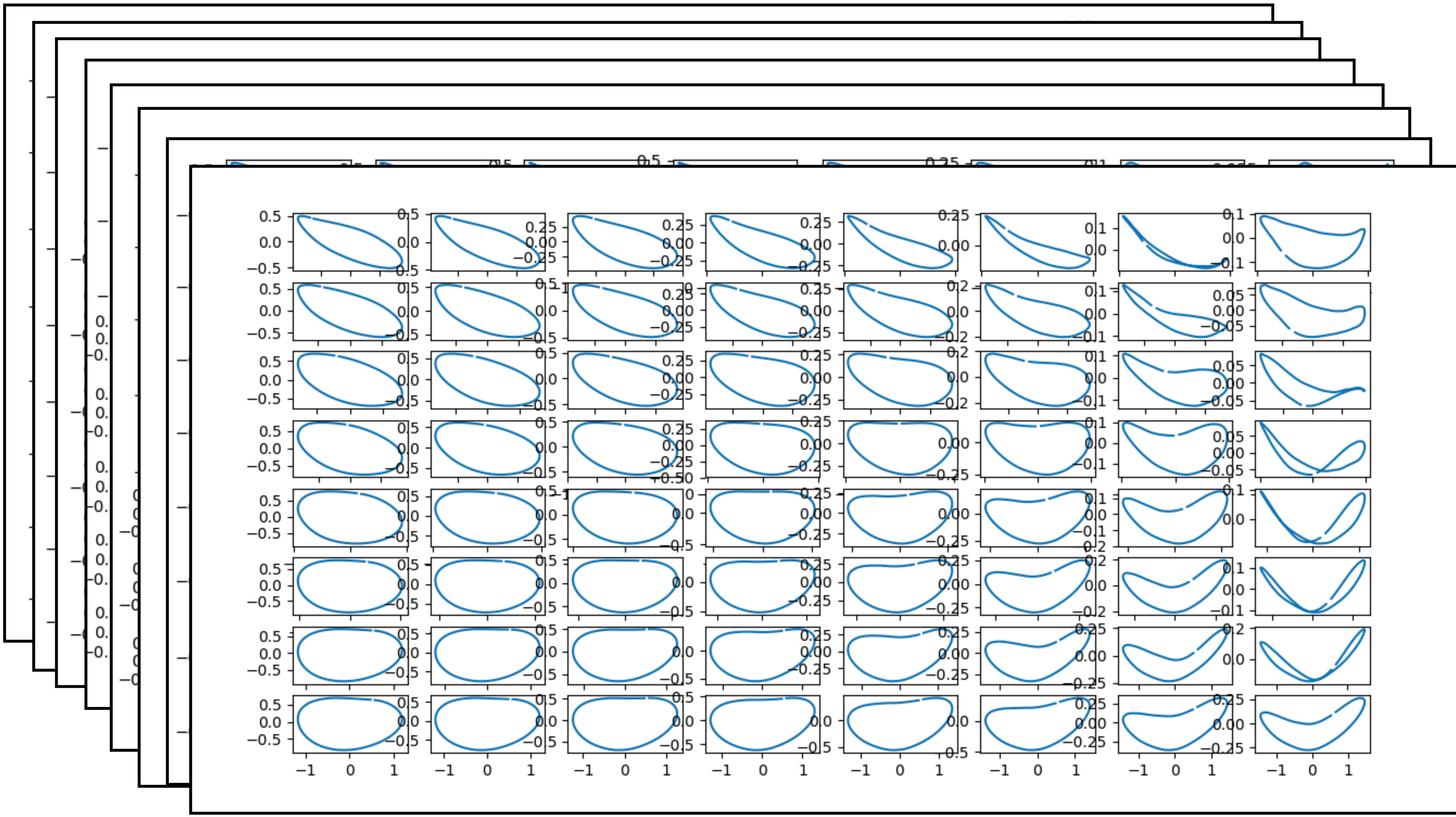
Distribution of Coupler Paths



Distribution of Coupler Paths



Distribution of Coupler Paths



Variational Inference

$$p(z\,|\,x) = \frac{p(x\,|\,z)p(z)}{p(x)}$$

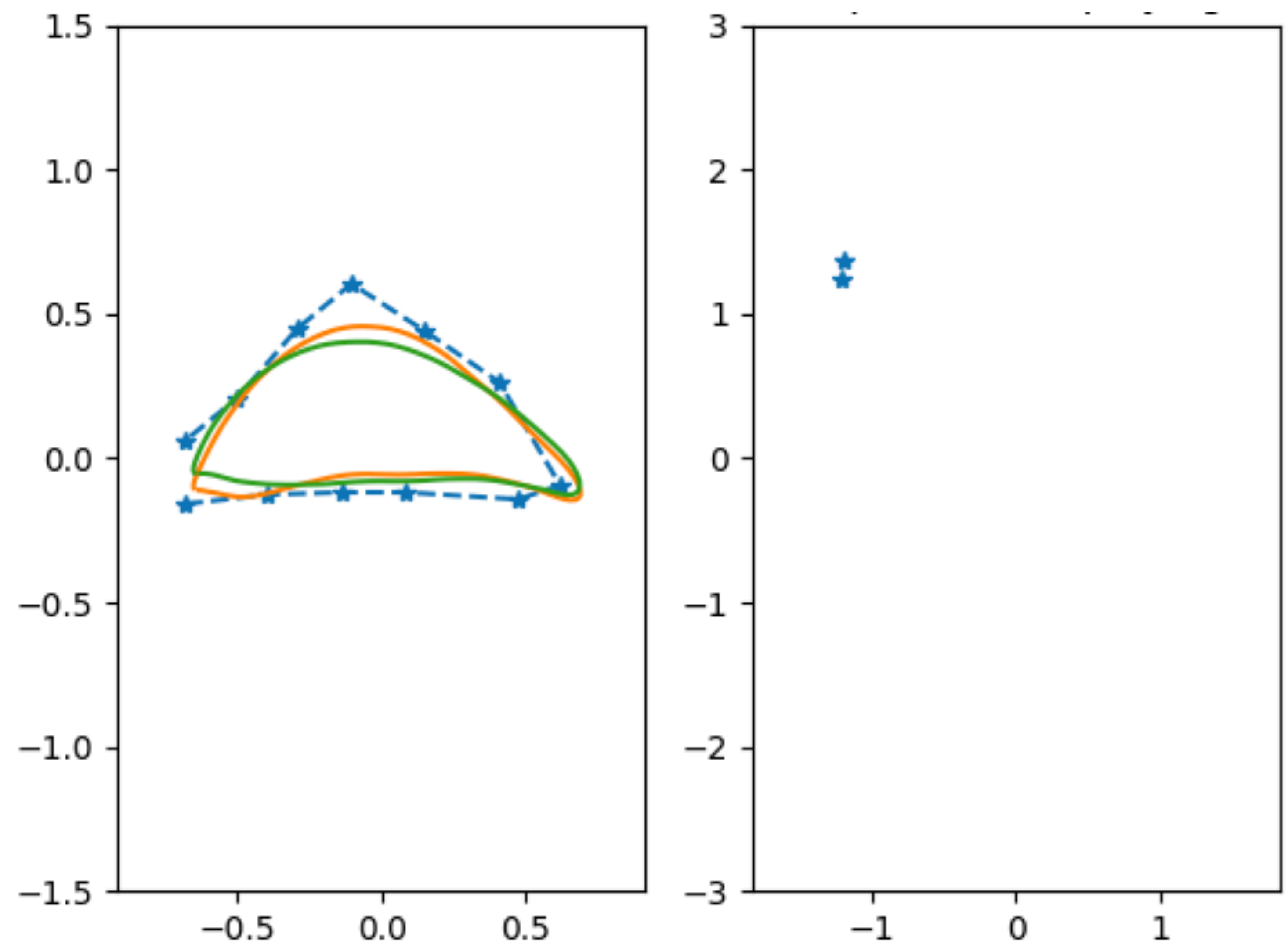
Encoder's Approximation

$$F(x) \approx p(z\,|\,x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$

Coupler Paths **Latent Space**



Variational Inference

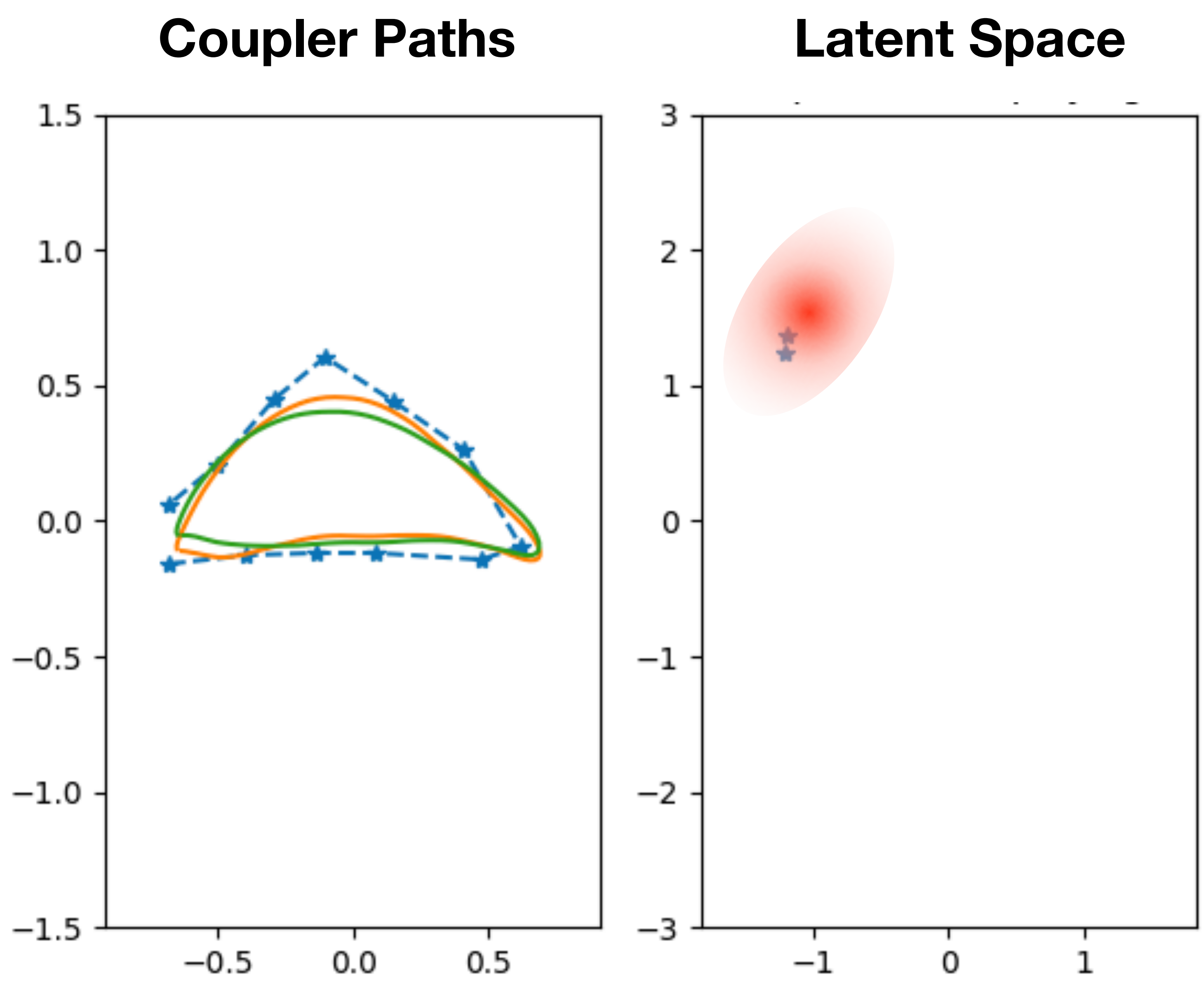
$$p(z\,|\,x) = \frac{p(x\,|\,z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z\,|\,x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

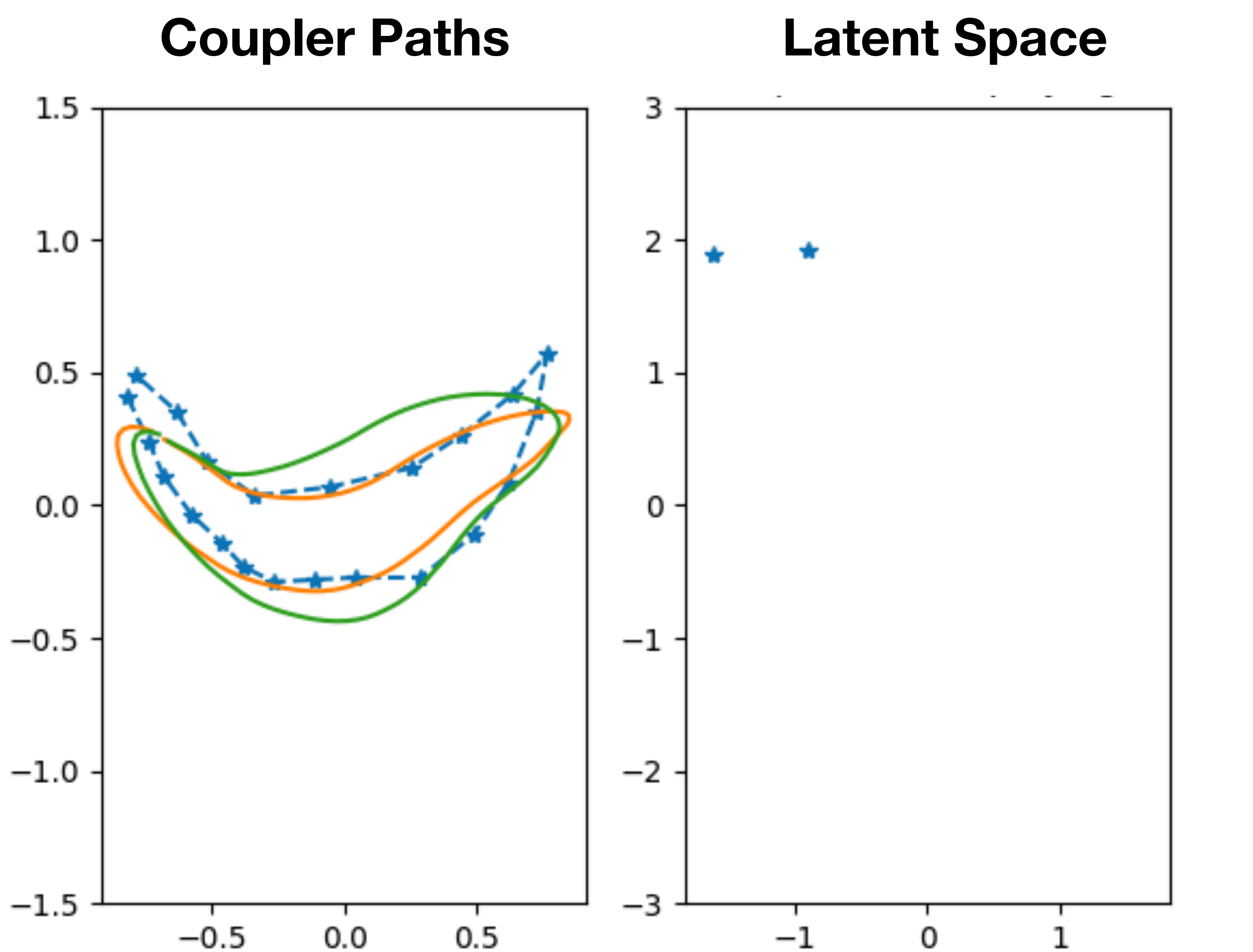
$$p(z\,|\,x) = \frac{p(x\,|\,z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z\,|\,x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

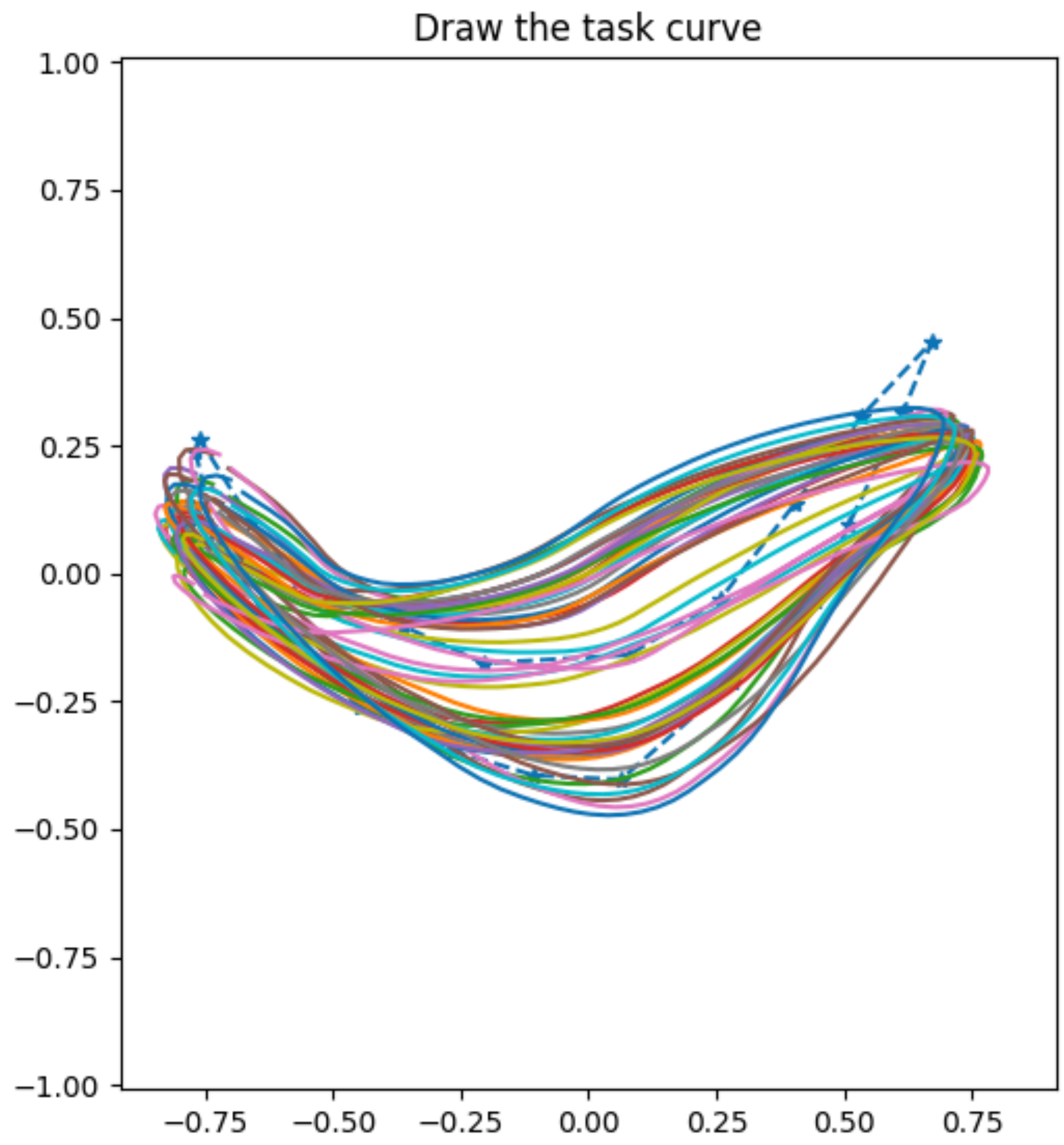
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

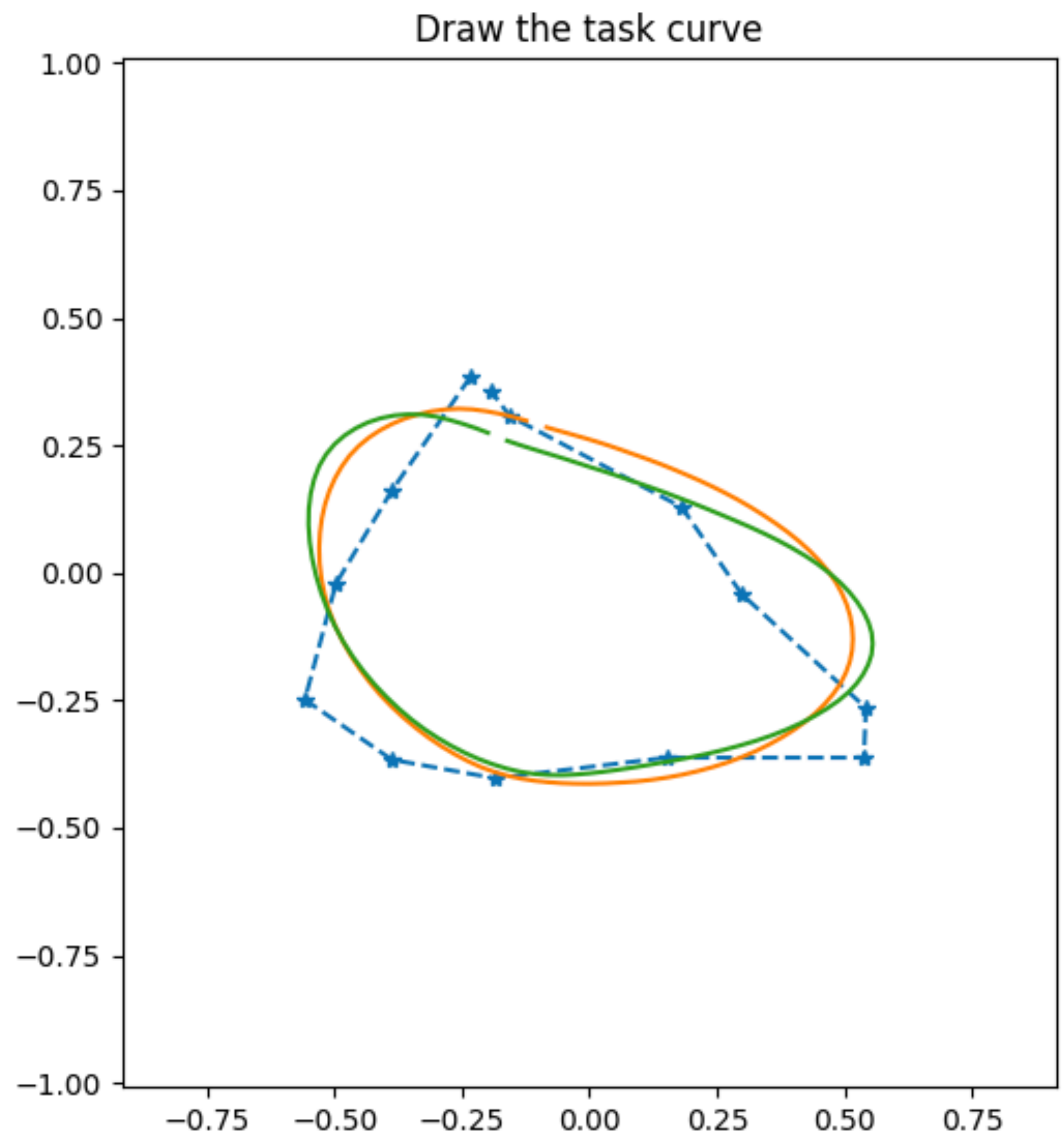
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

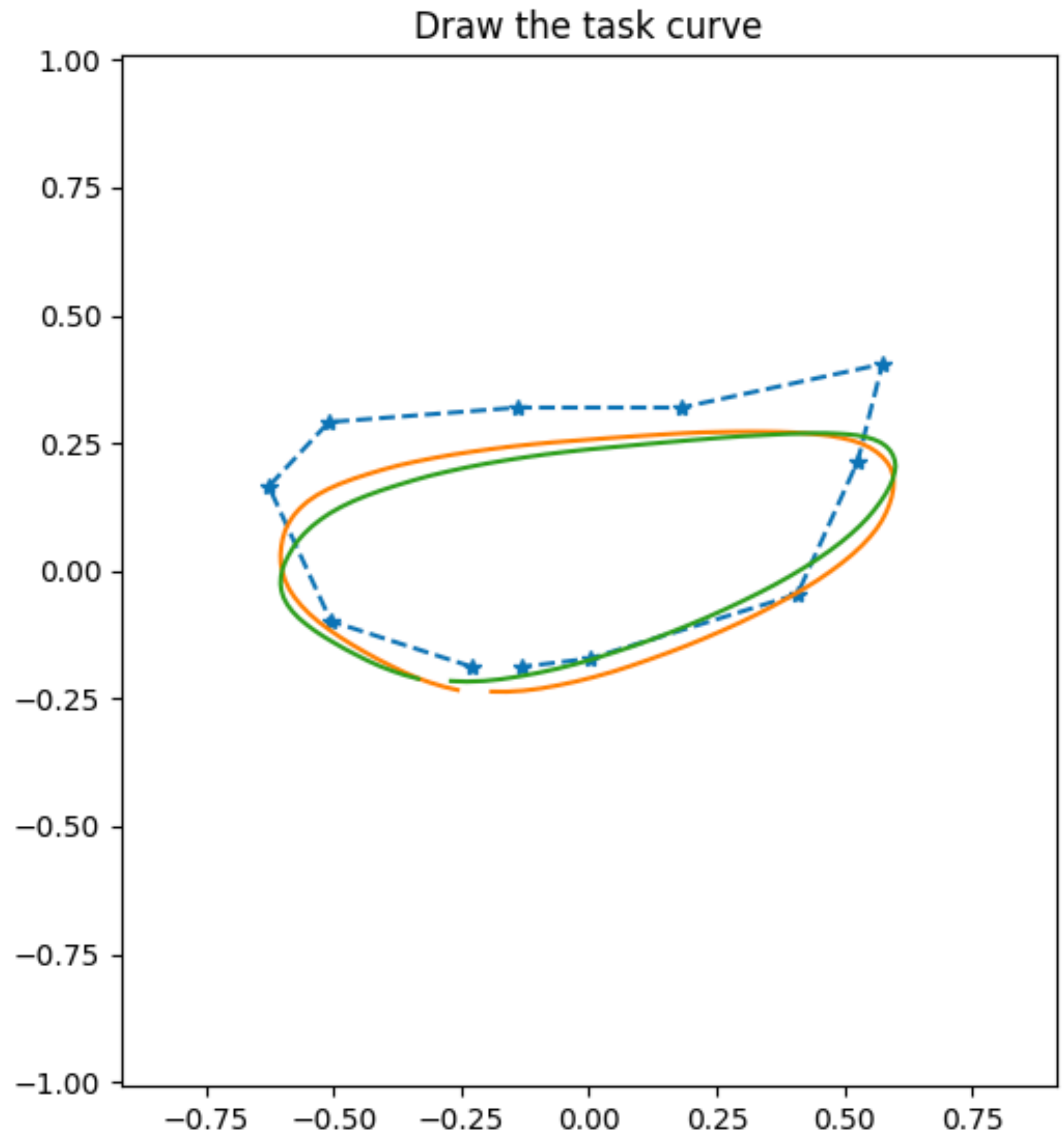
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

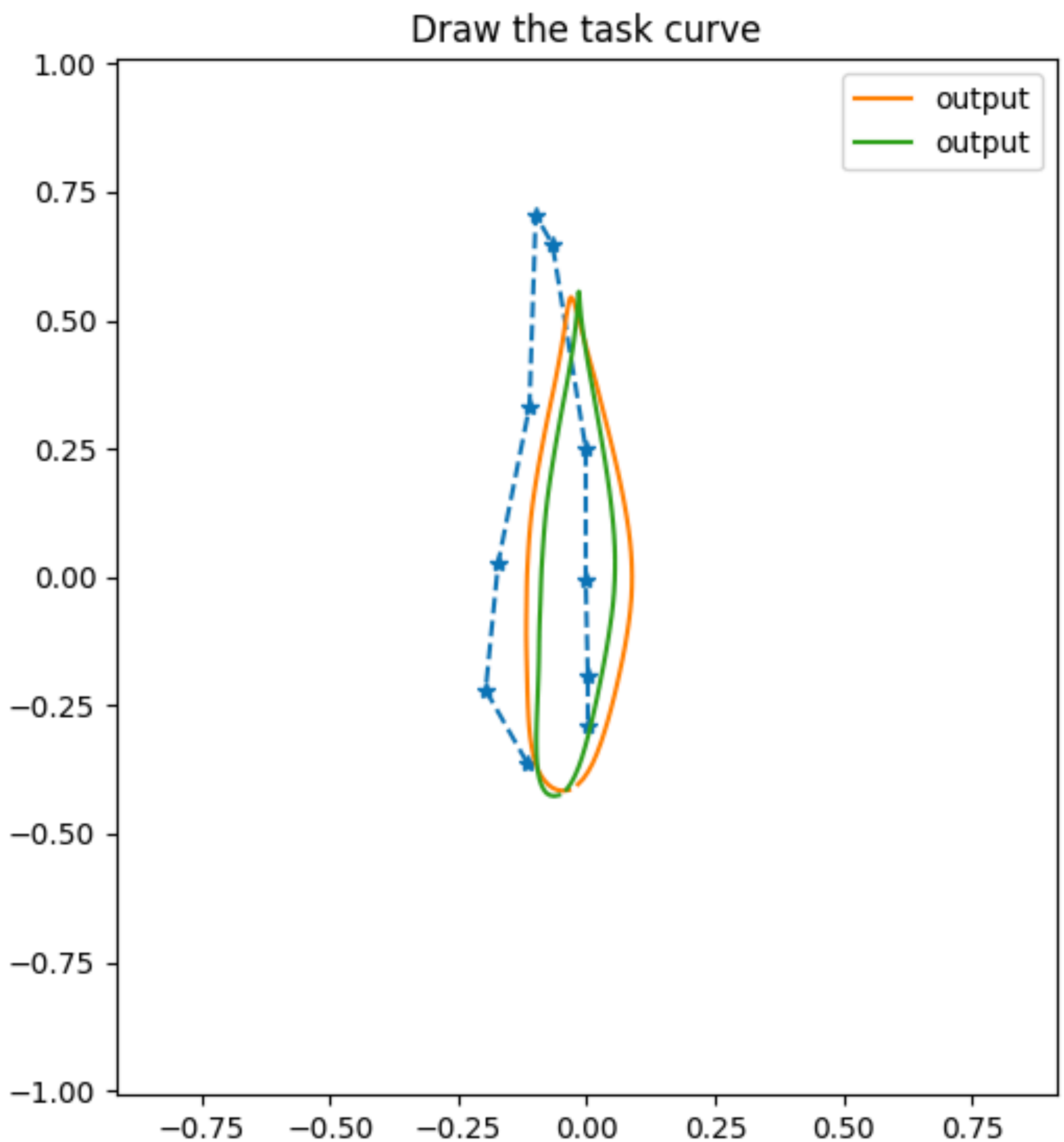
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z|x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

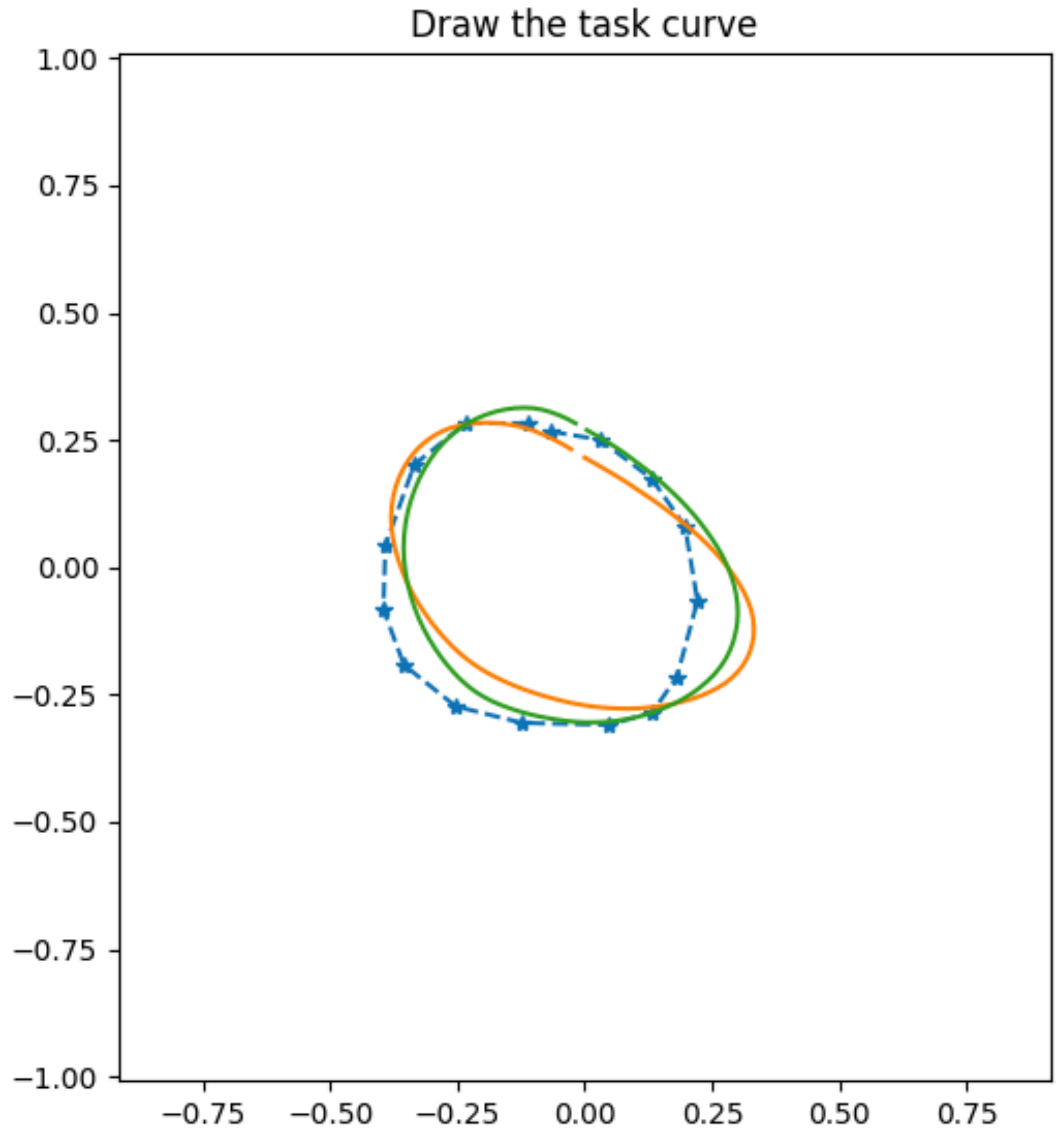
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

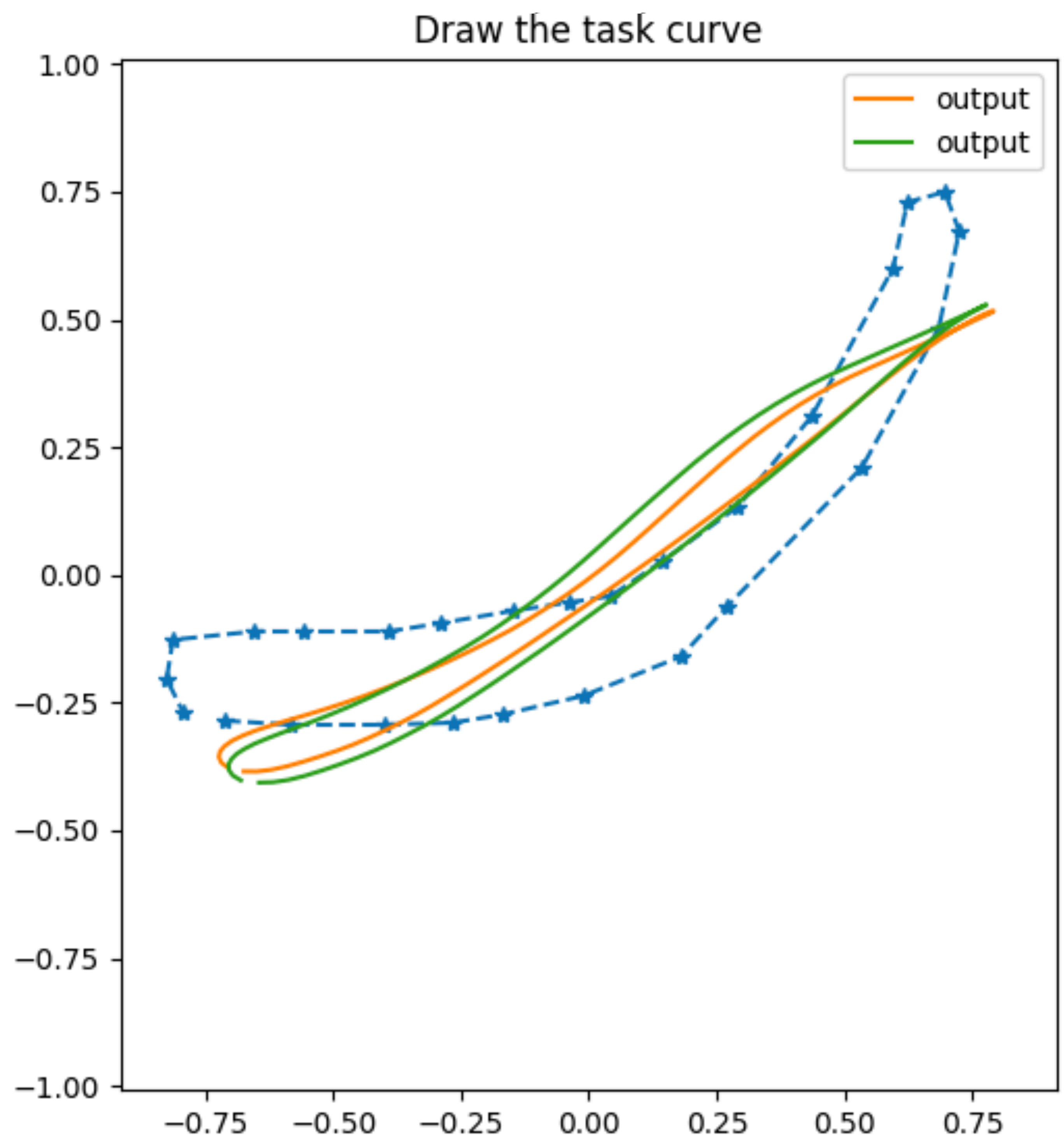
$$p(z|x) = \frac{p(x|z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z|x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

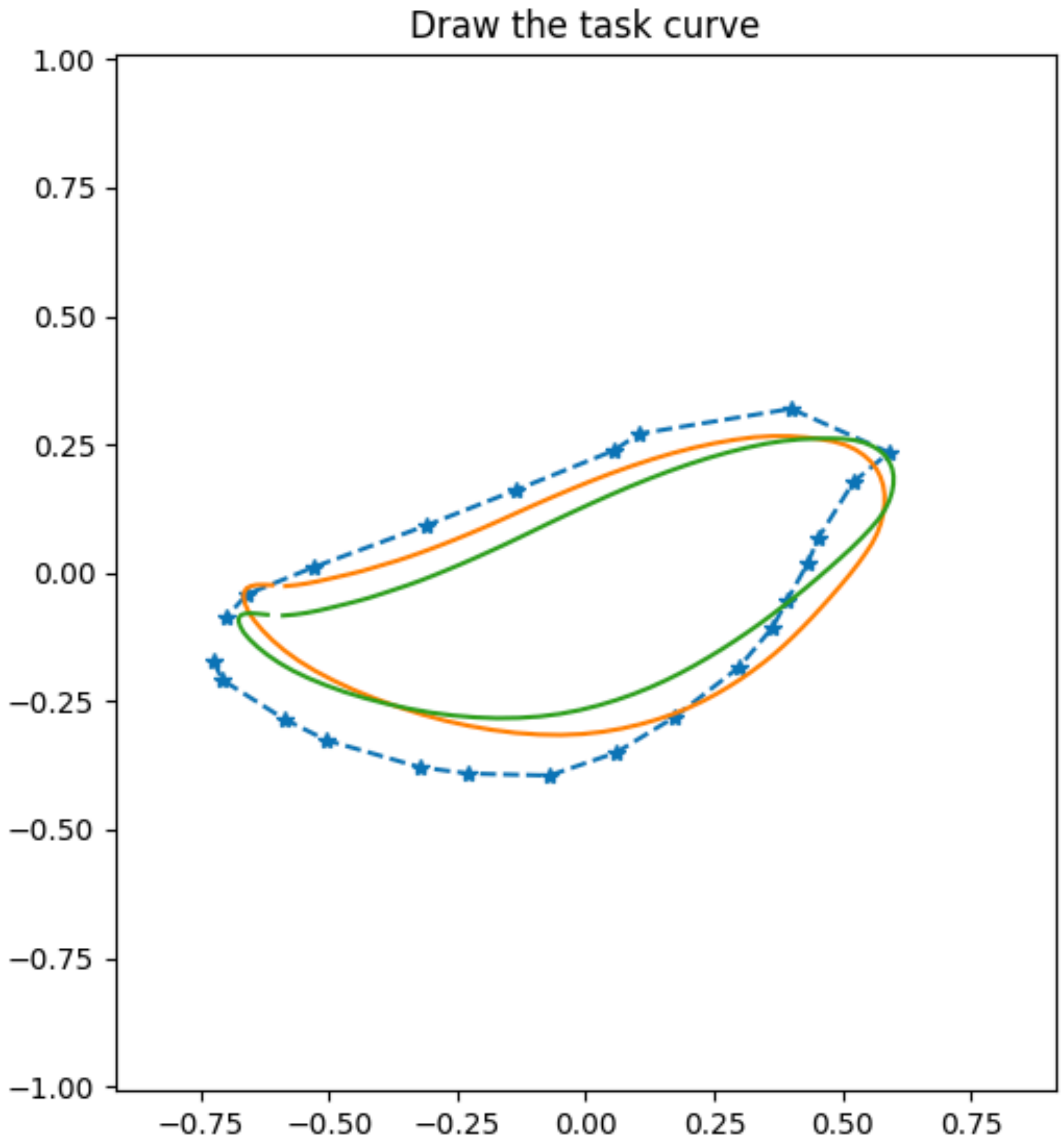
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

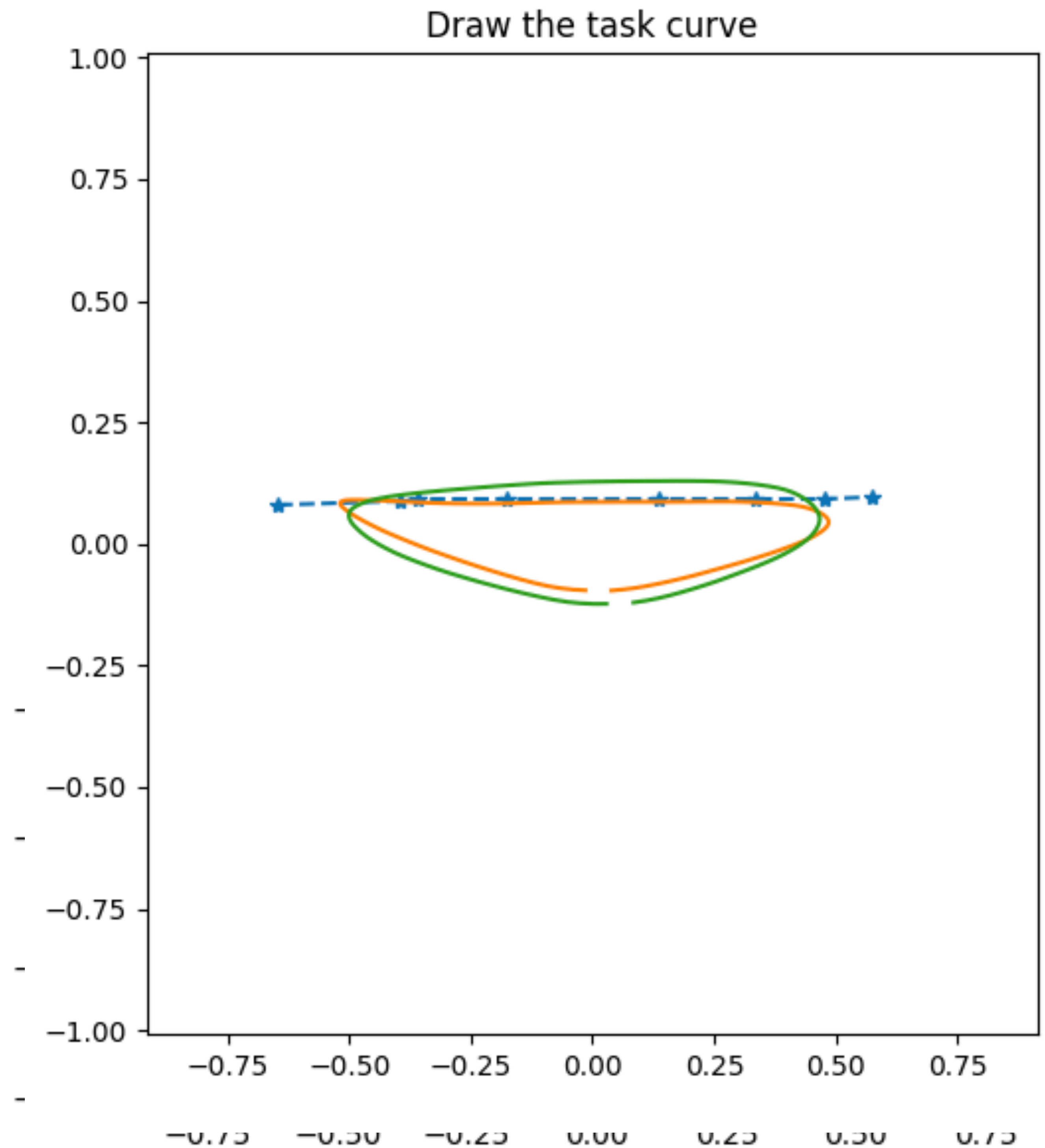
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Variational Inference

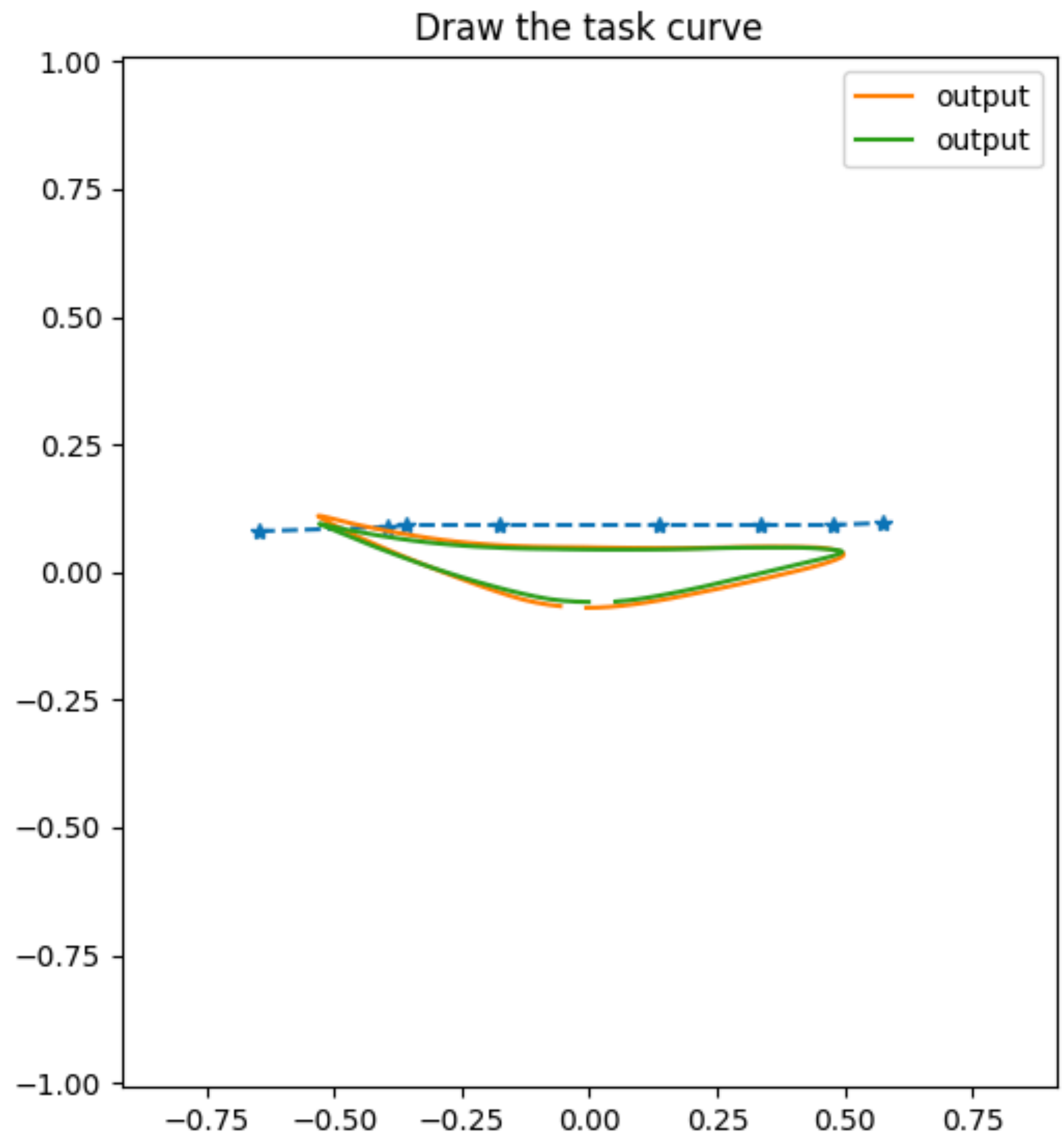
$$p(z \mid x) = \frac{p(x \mid z)p(z)}{p(x)}$$

Encoder's Approximation

$$F(x) \approx p(z \mid x)$$

Decoder's Generation

$$\hat{x}_{samples} = G(z)$$



Input Augmentation: Orientation Prediction

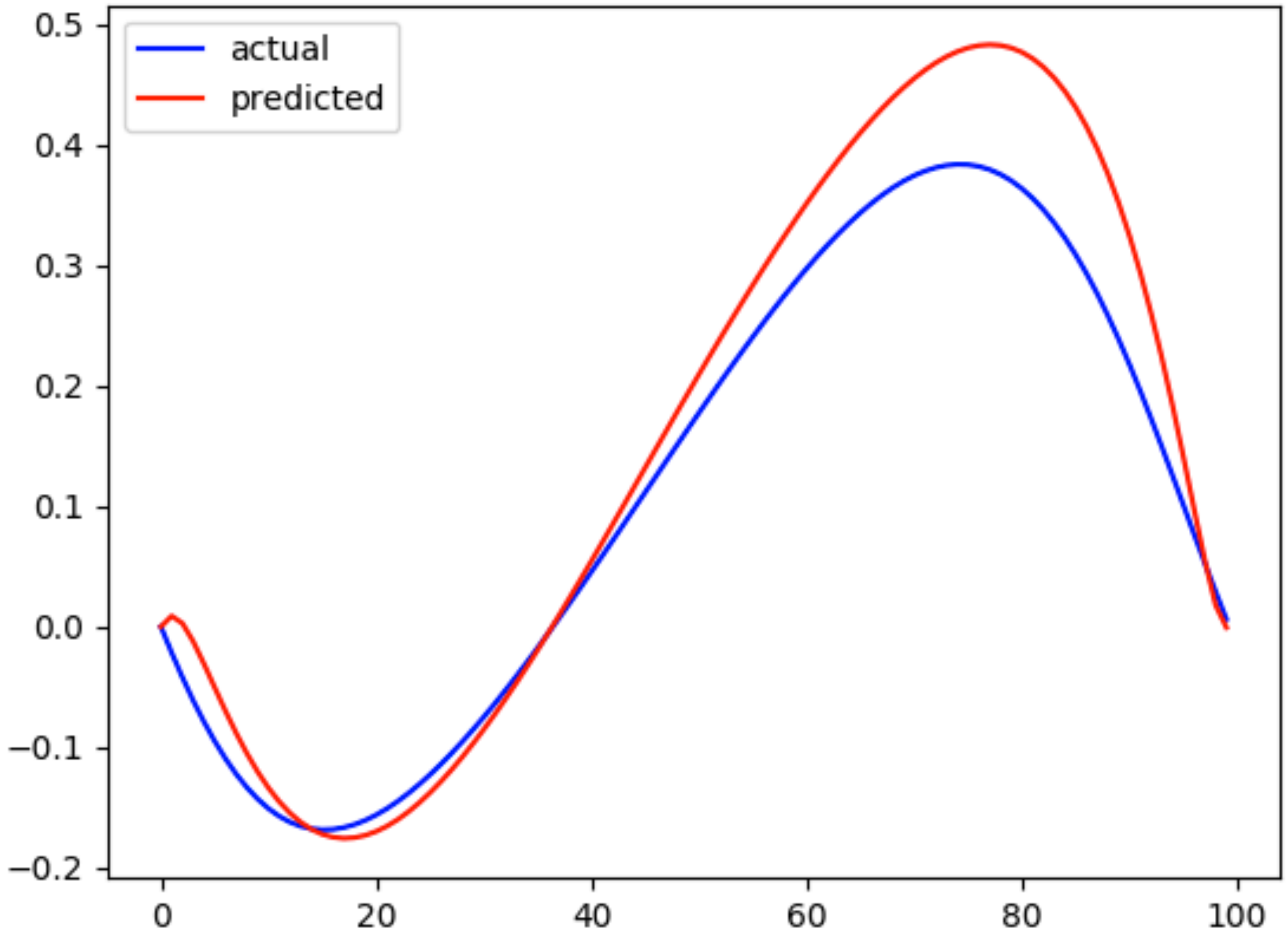
- Given a path points, predict the orientation of the path points, with condition that path points are from unknown fourbar/sixbar/sixbar linkage.

$$\{\theta_i\}_i^N = Q(\{x_i, y_i\}_i^N)$$

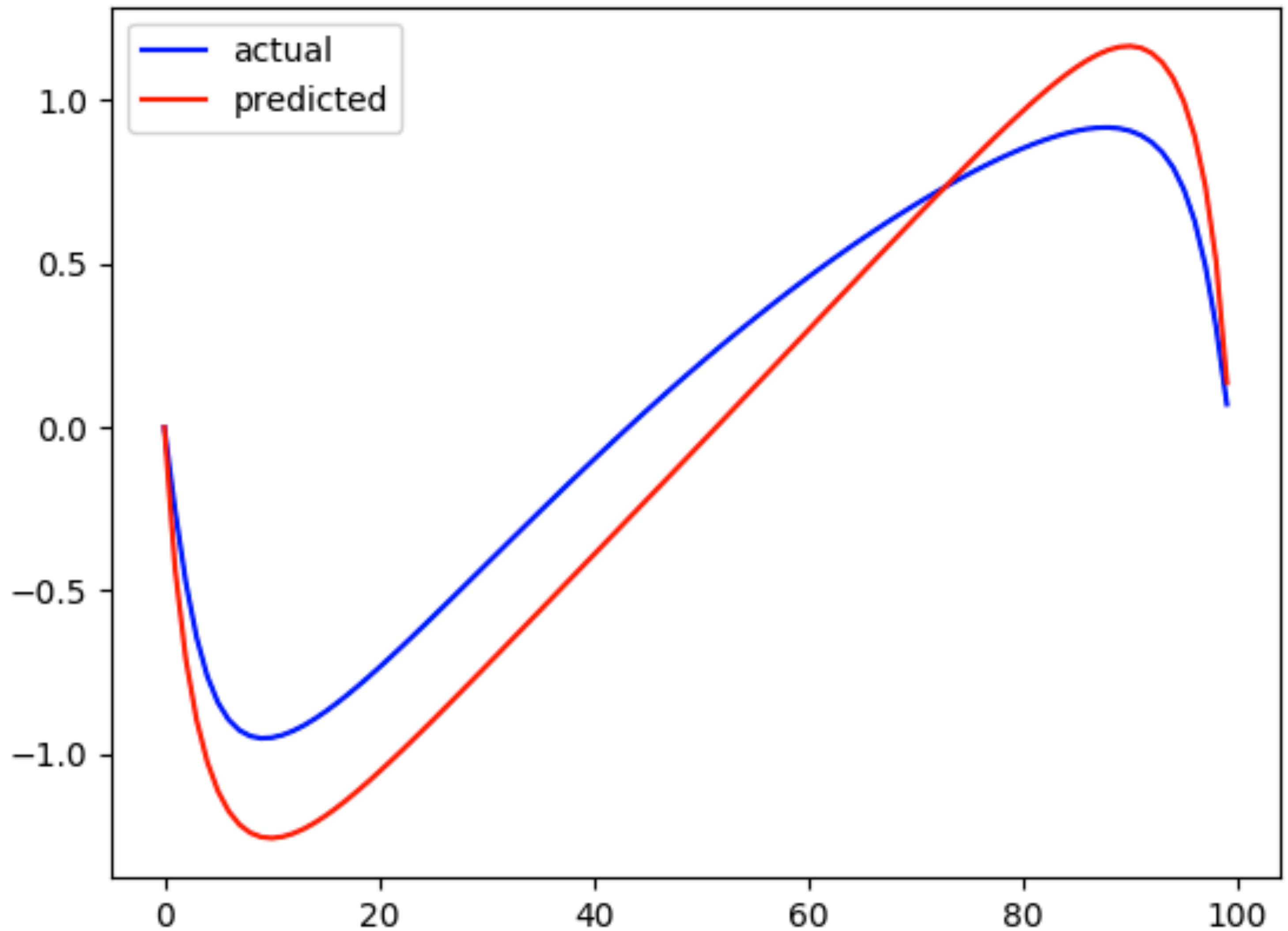
- LSTM (Long Short Term Memory) Recurrent Neural Networks are used to approximate Q

Input Augmentation: Orientation Prediction

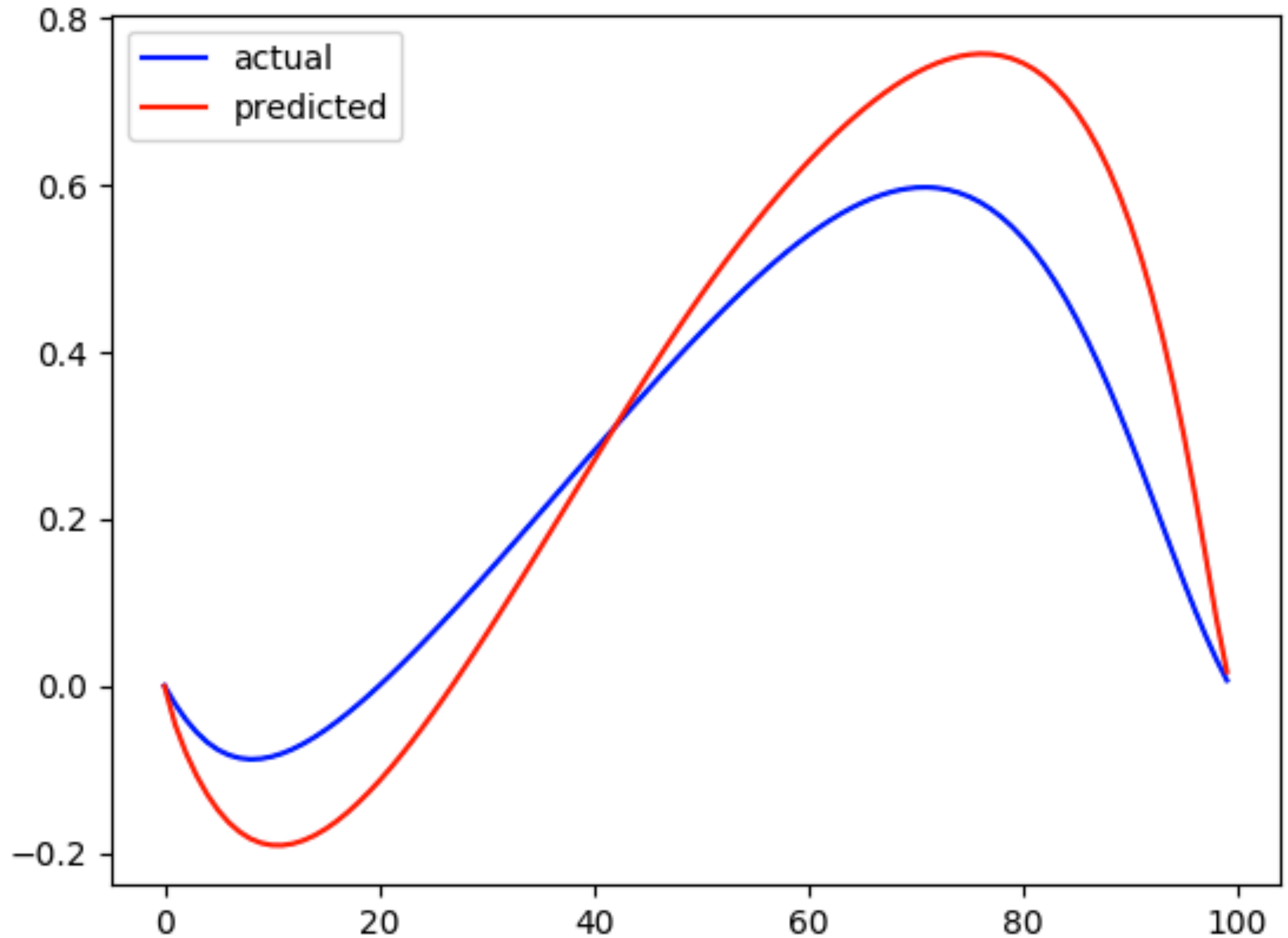
Input Augmentation: Orientation Prediction



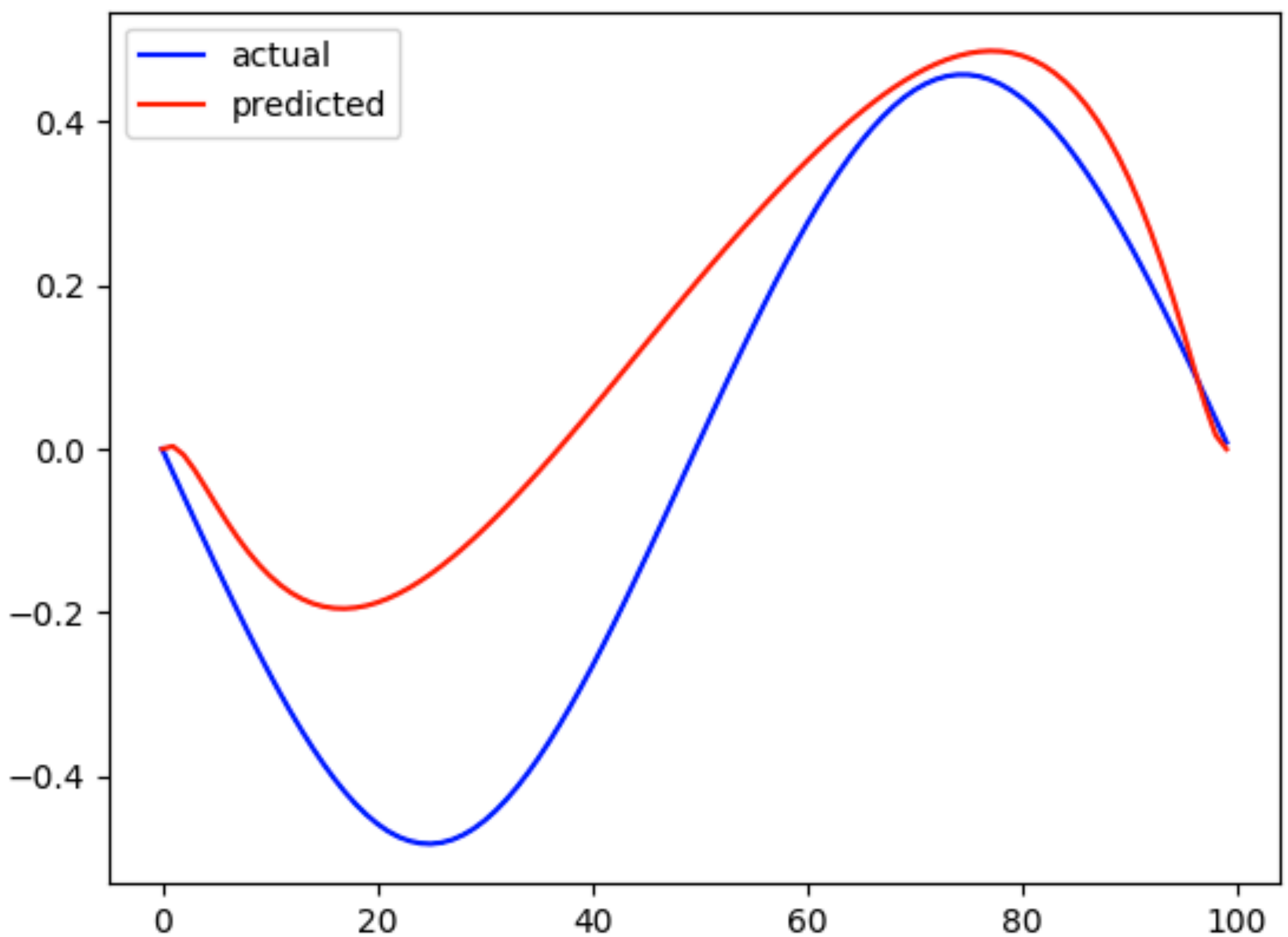
Input Augmentation: Orientation Prediction



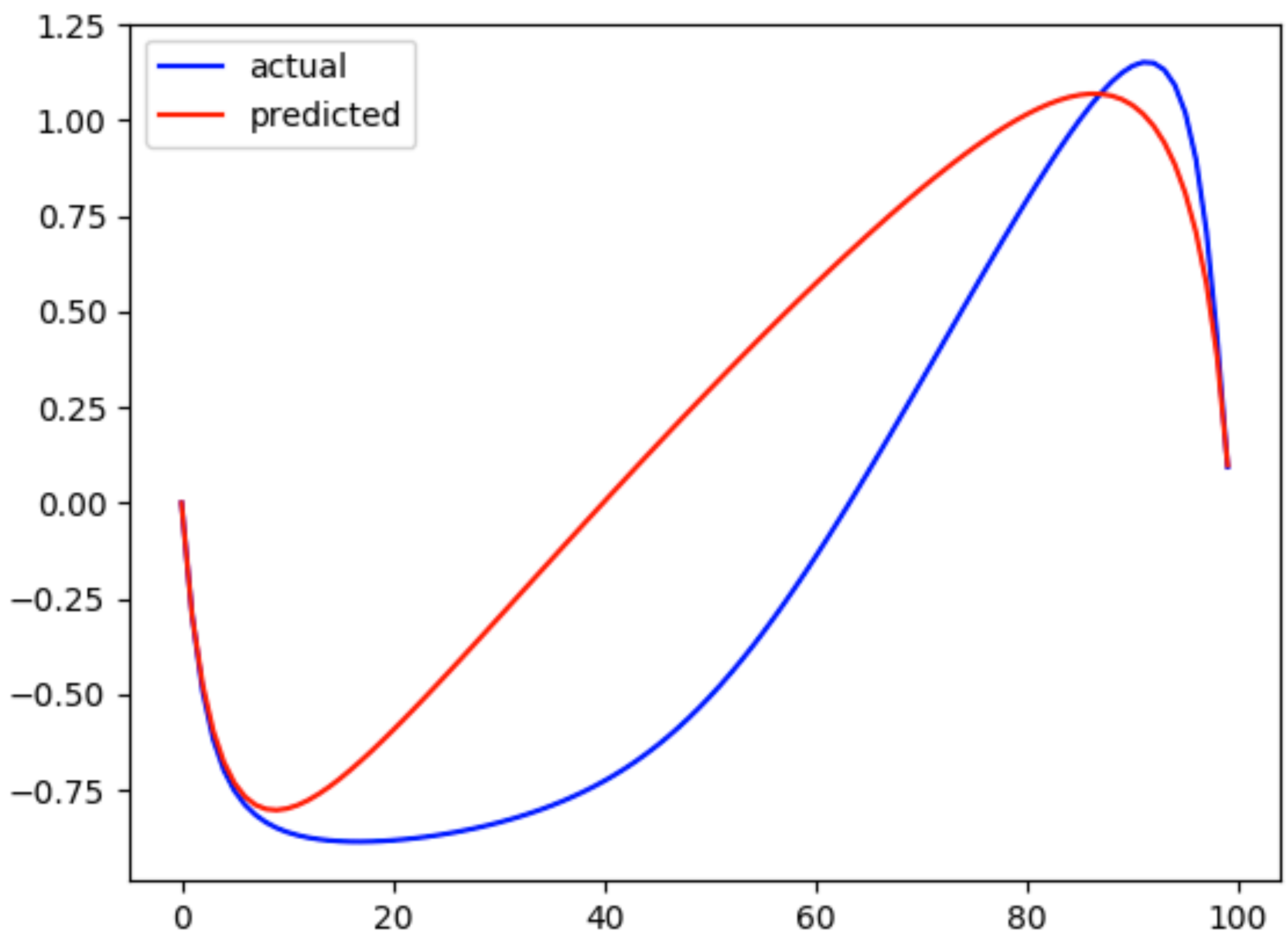
Input Augmentation: Orientation Prediction



Input Augmentation: Orientation Prediction



Input Augmentation: Orientation Prediction



Intelligent Assistant

- Assist Human Designer to generate viable concepts by
  - Conditioning the input
  - Augmenting the input
  - Leveraging and managing uncertainty
  - Making Free Choices (e.g. in Synthesis of Sixbars)
  - Exploring Solutions and presenting it the right way to designer

Input Augmentation and Conditioning

- Manual Input of m points (m > 3)

$$X_{manual}\,.\,dim = [m,2]\ldots(m \geq 4)$$

- Cubic Spline is fitted to input

$$X = Bspline(X_{manual})\ldots X\,.\,dim = [100,2]$$

- Assistant-VAE is trained to convert previous input into closed coupler path

$$p(z\,|\,X) = F_{encoder}(x)$$

$$\hat{X} = G(z)\ldots z \sim p(z\,|\,X)$$

- Assistant-RNN is trained to predict the orientations for closed coupler path received

$$\theta = Q_{RNN}(\hat{X})$$

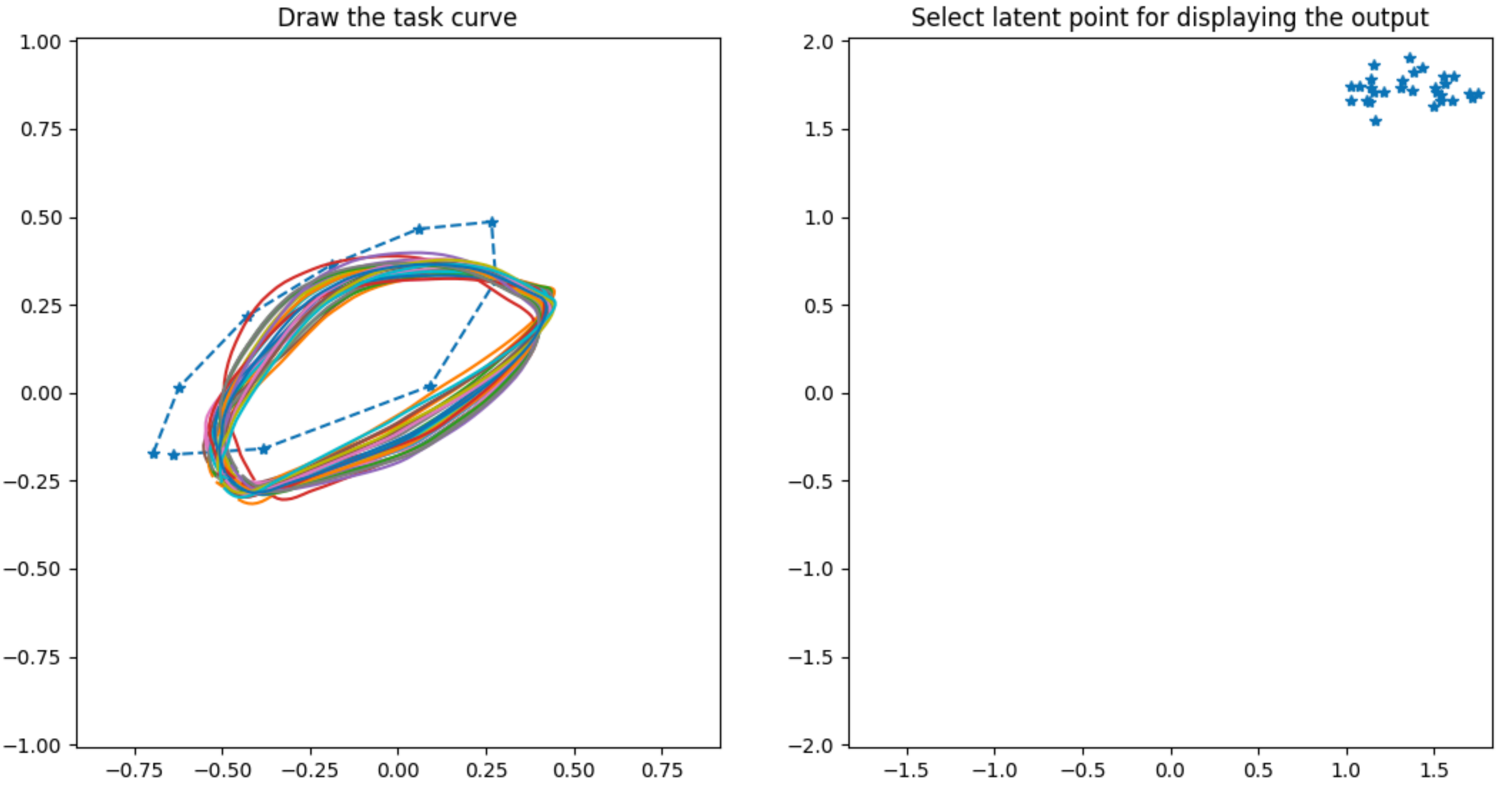
$$P = [\hat{X}, \theta]\ldots P\,.\,dim = [100,3]$$

- Assisted Input is fed to N-Pose Algebraic Fitting Algorithm

$$Solutions = \textbf{Algebraic Fitting}(P)$$

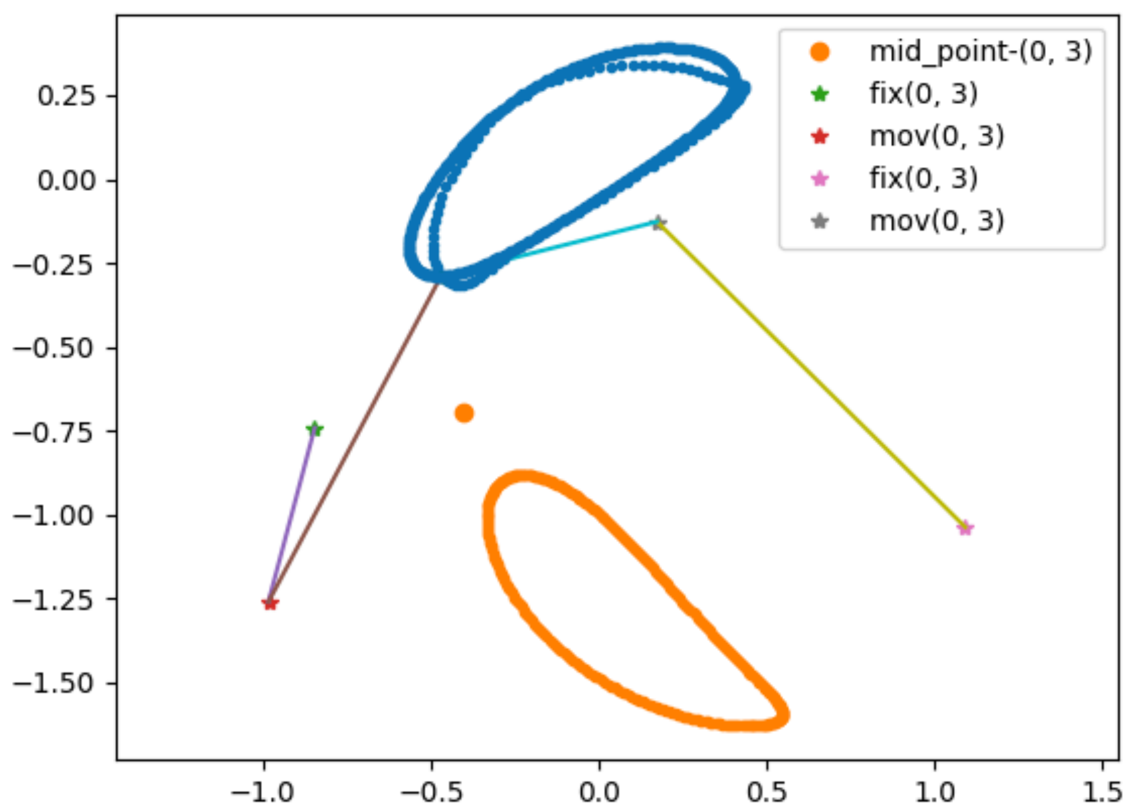
Input Augmentation and Conditioning Results

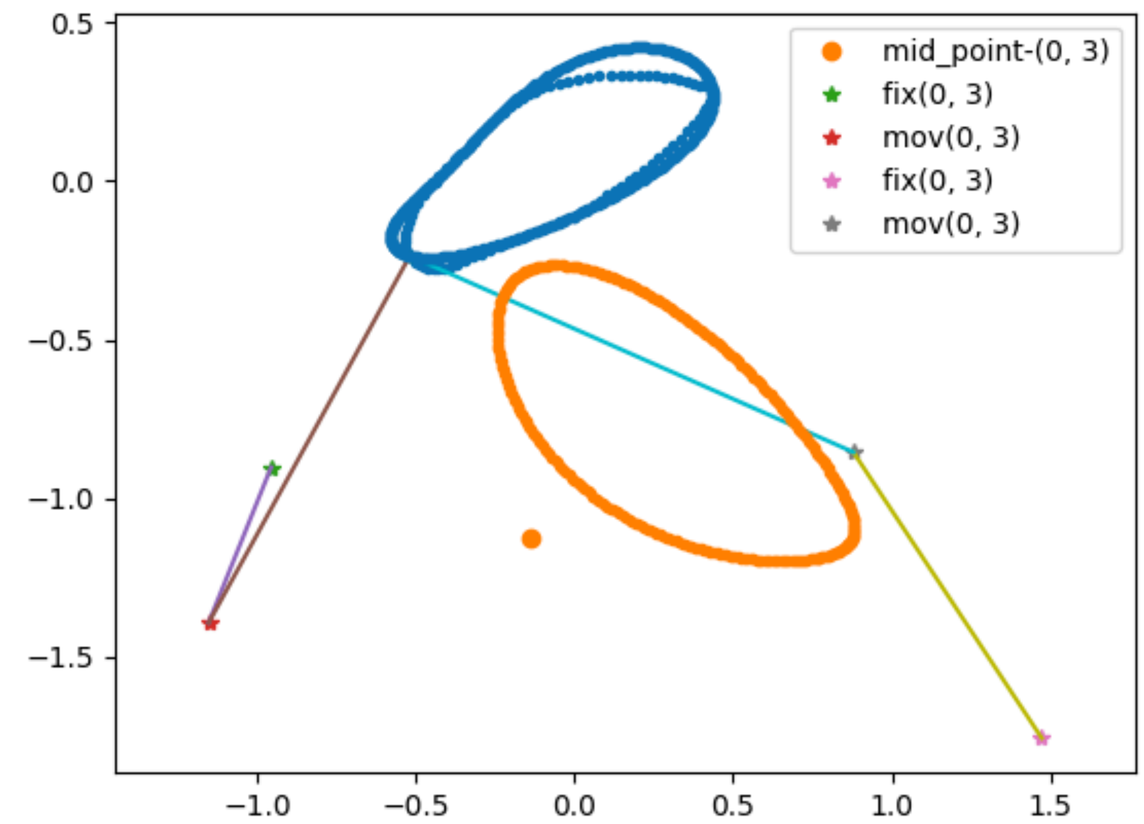
number of samples = 30



Input Augmentation and Conditioning Results

Concept 1

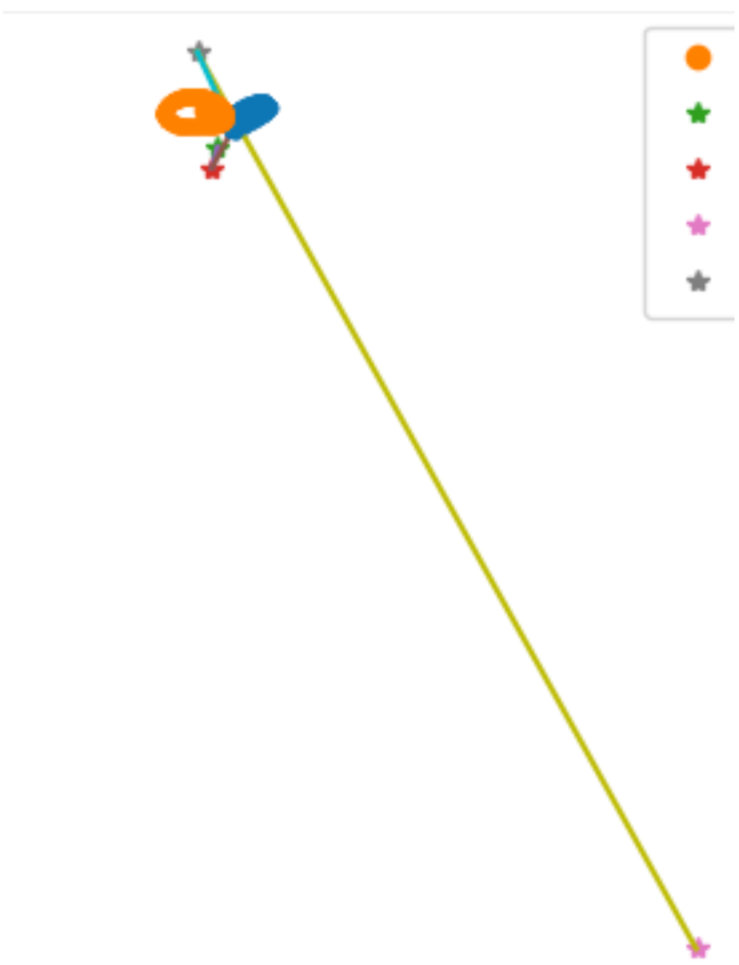


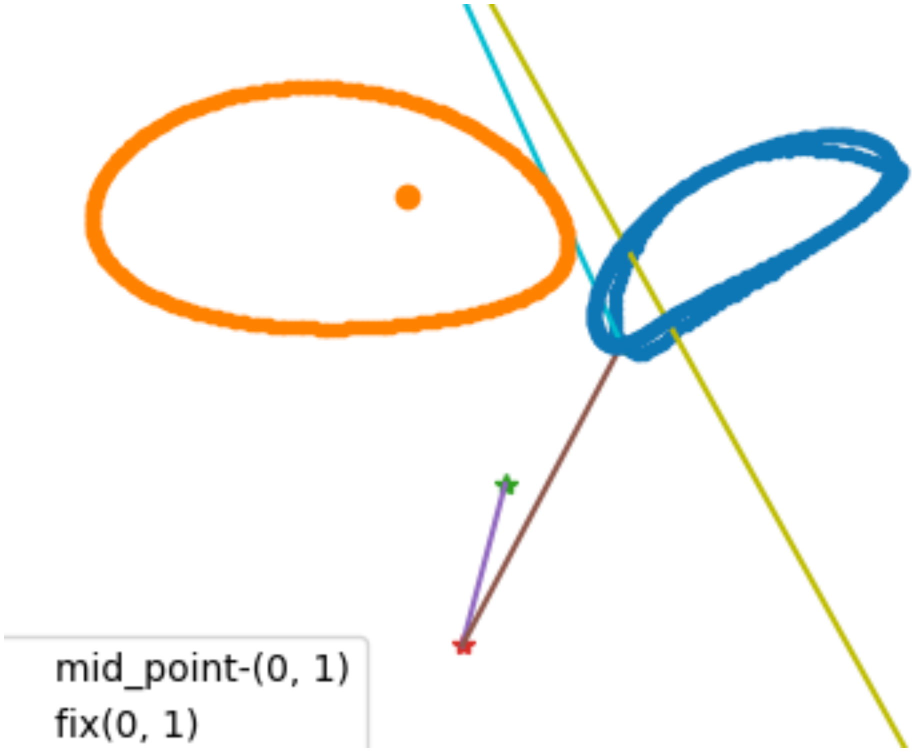


Similar Concepts, but very different link ratios

Input Augmentation and Conditioning Results

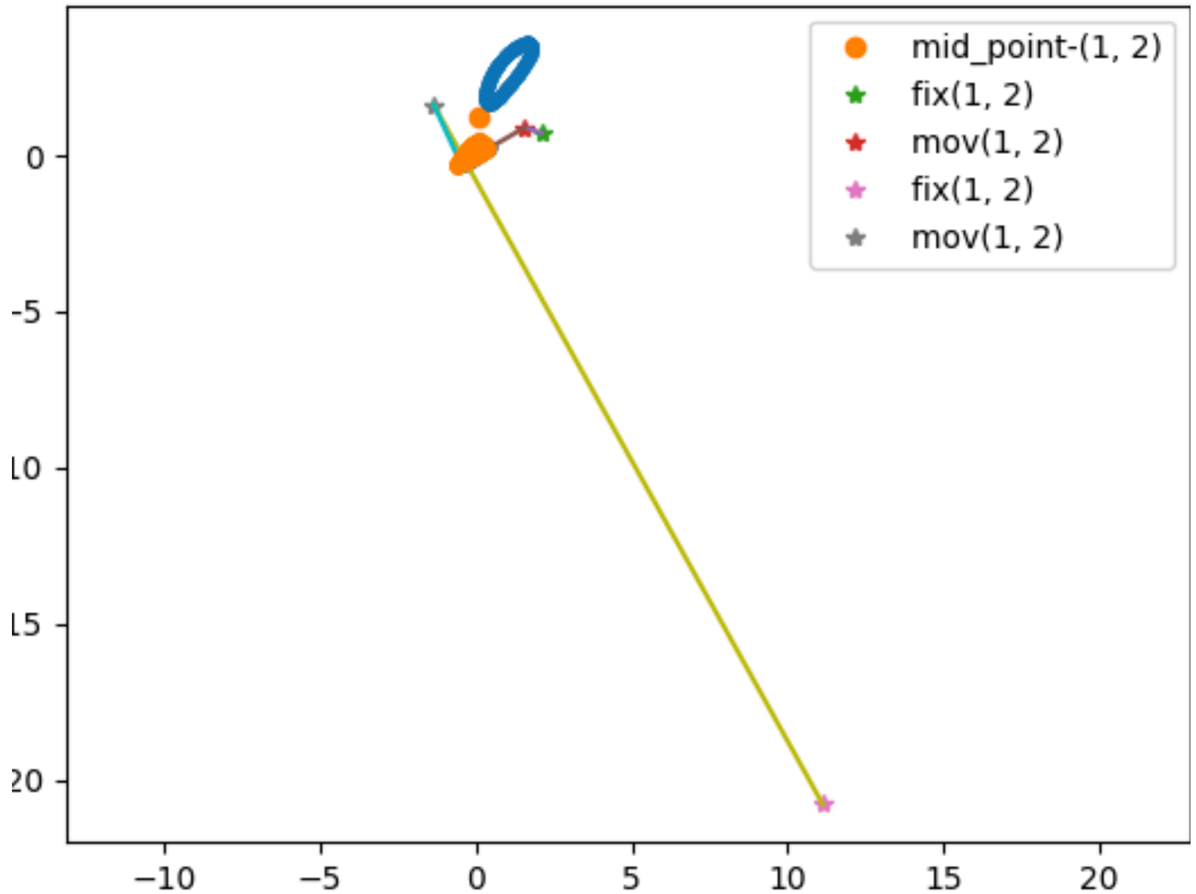
Concept 2

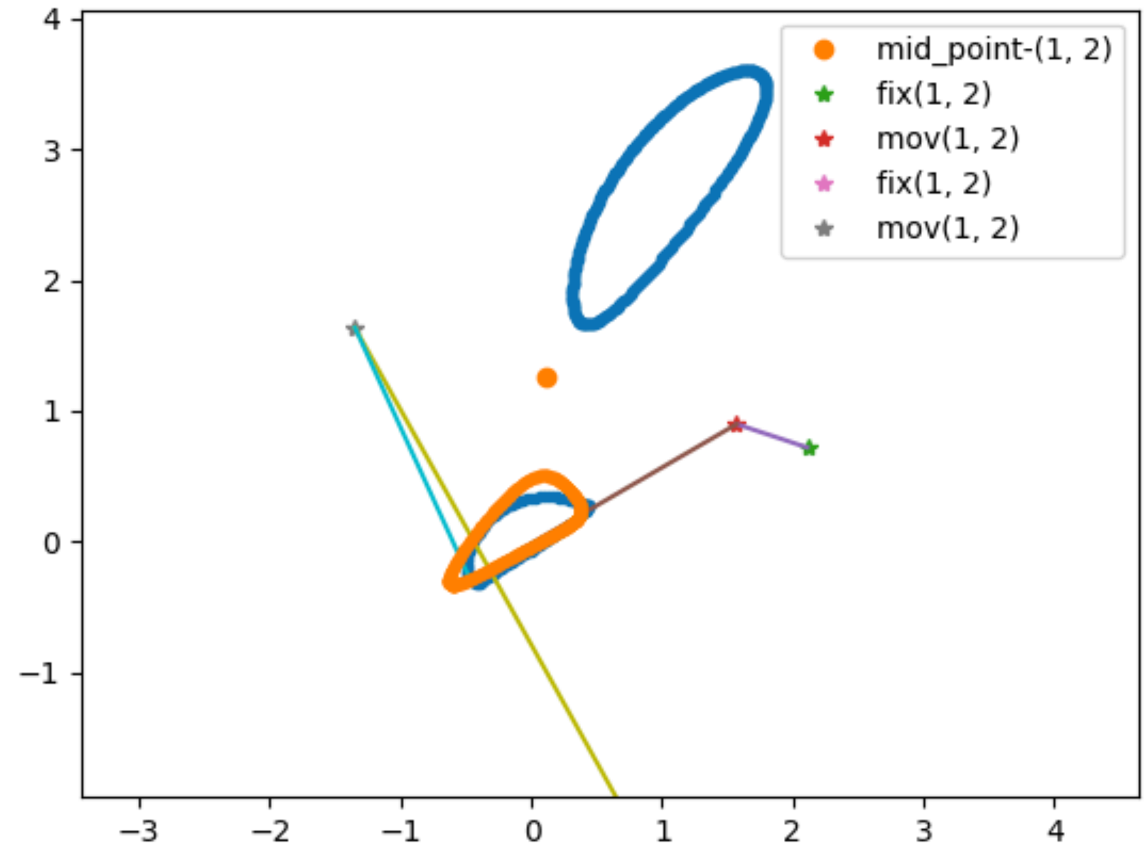




Input Augmentation and Conditioning Results

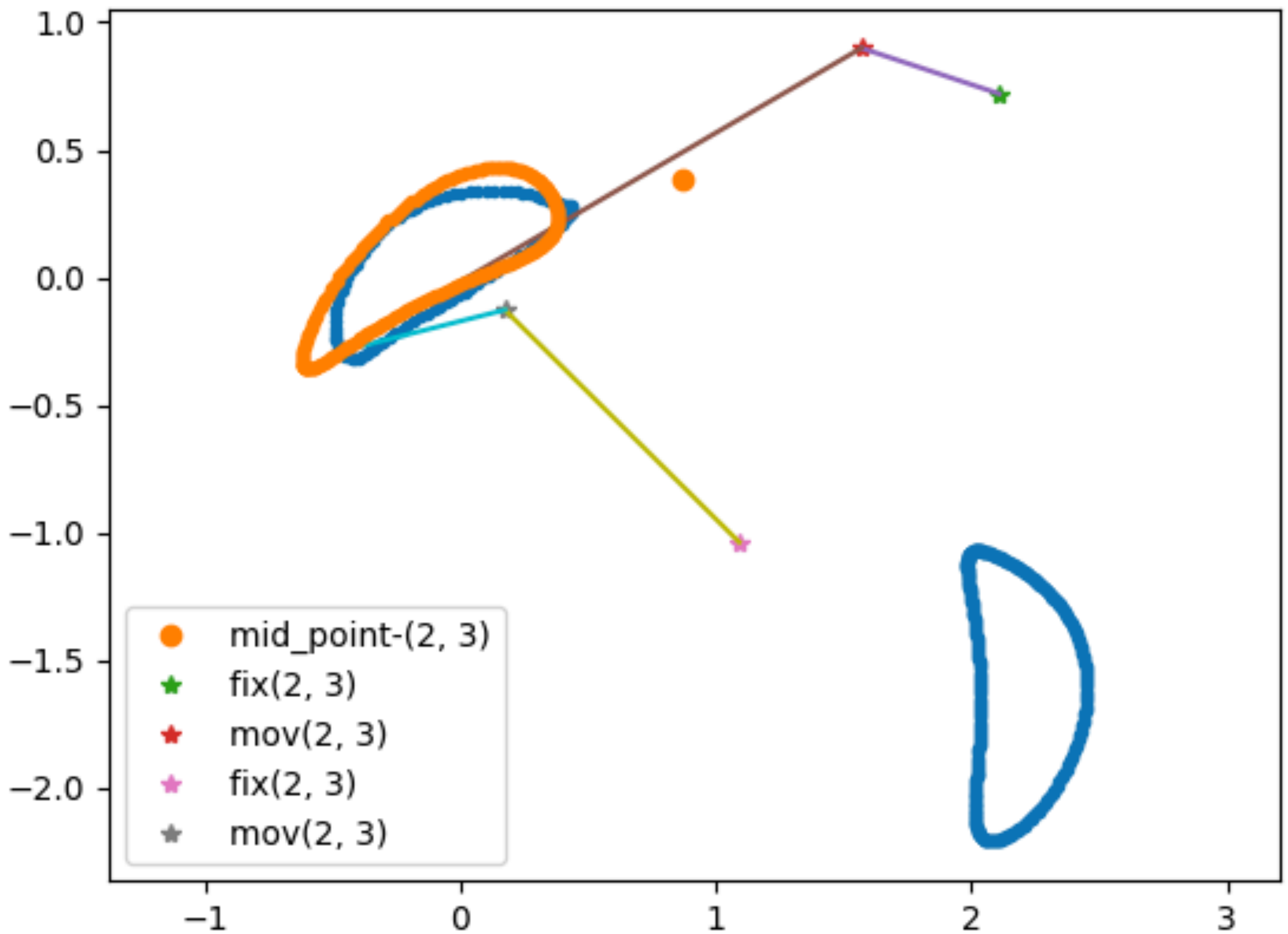
Concept 3





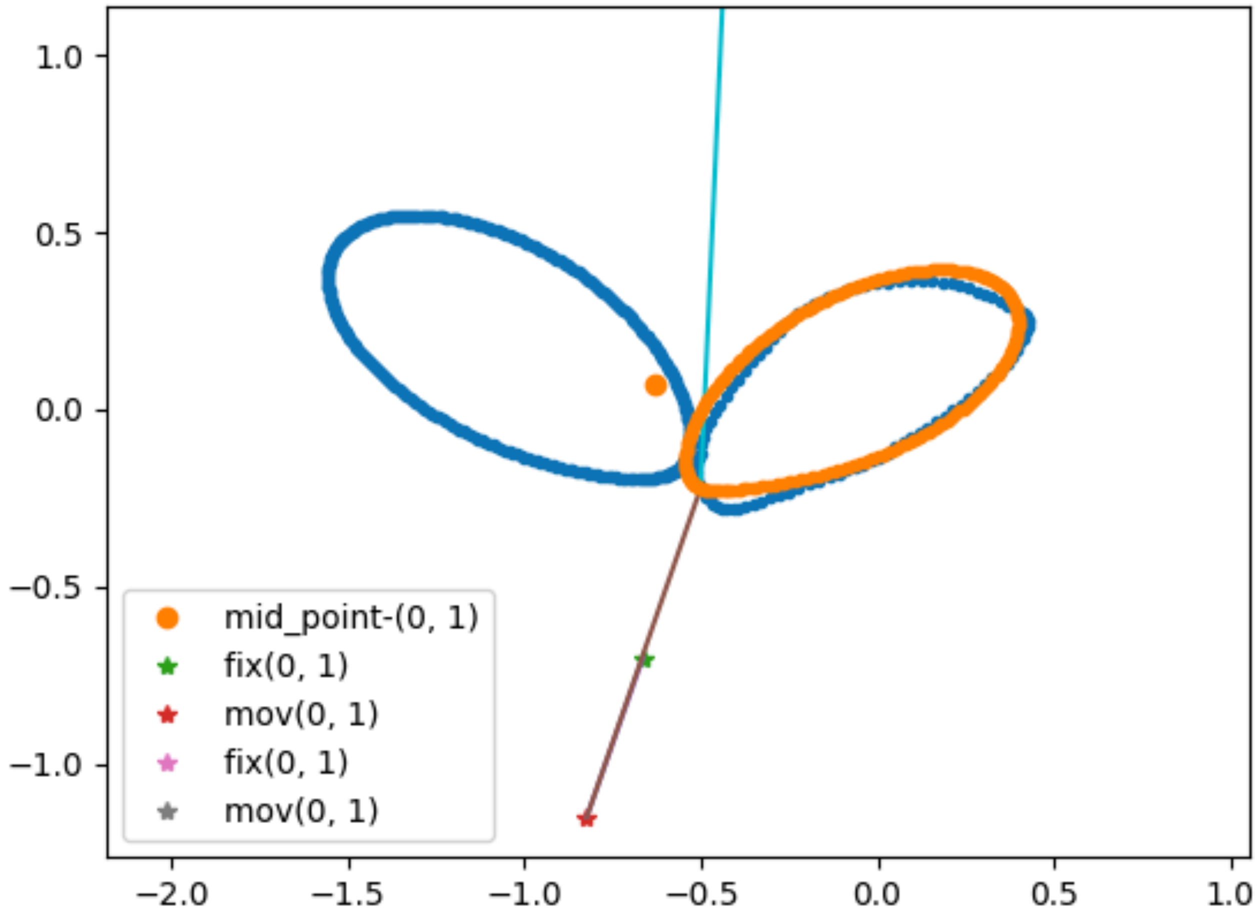
Input Augmentation and Conditioning Results

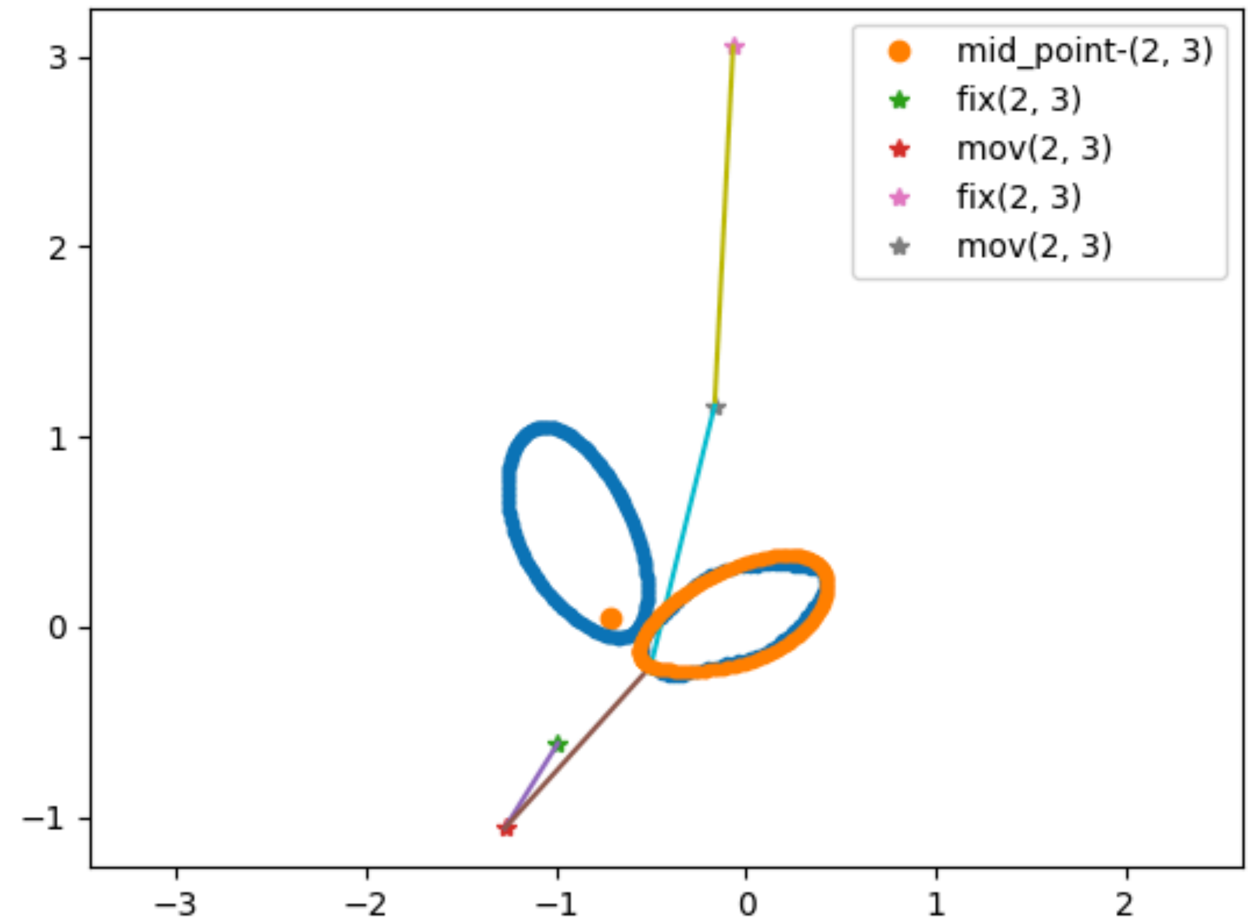
Concept 4



Input Augmentation and Conditioning Results

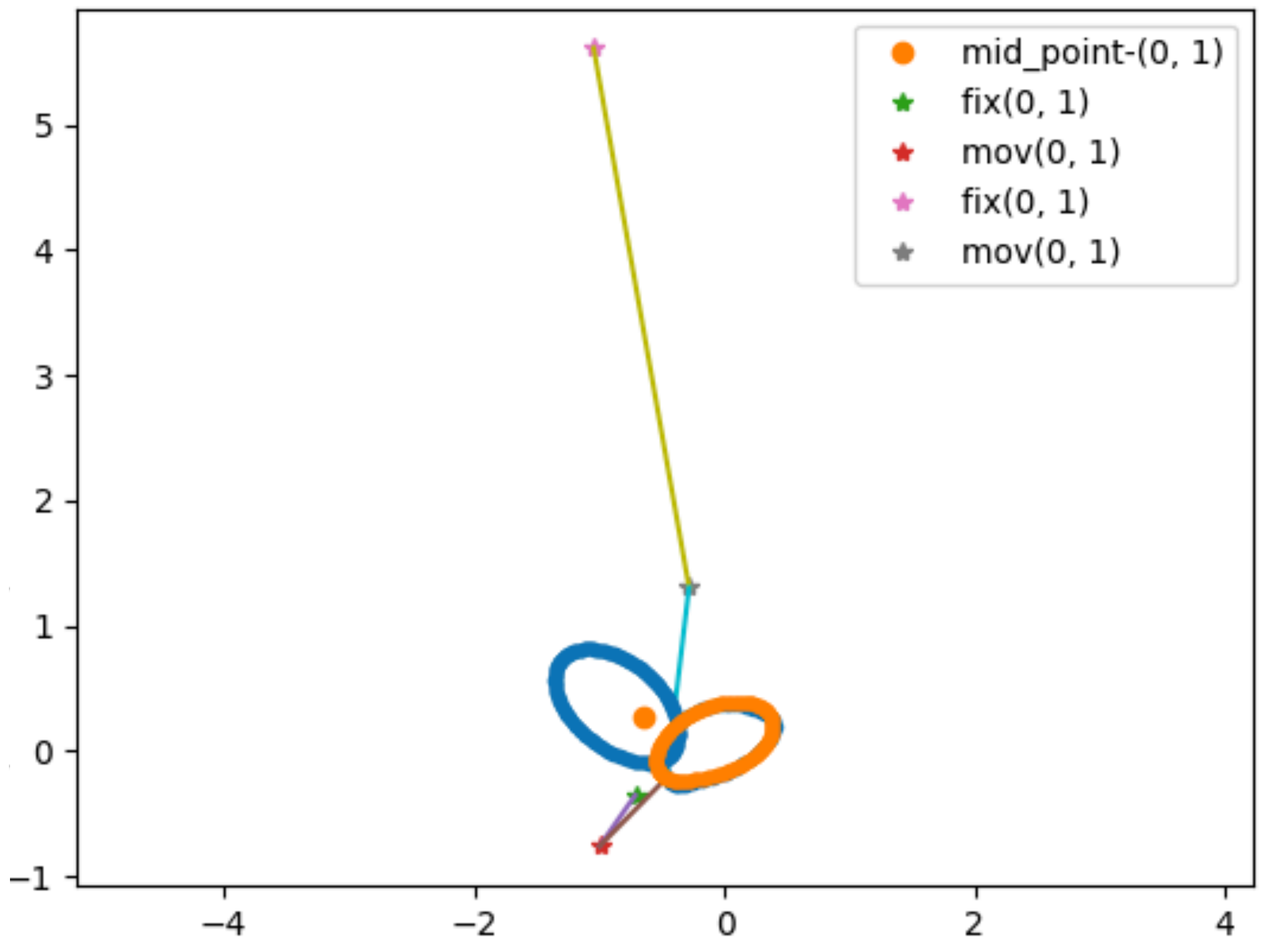
Concept 5

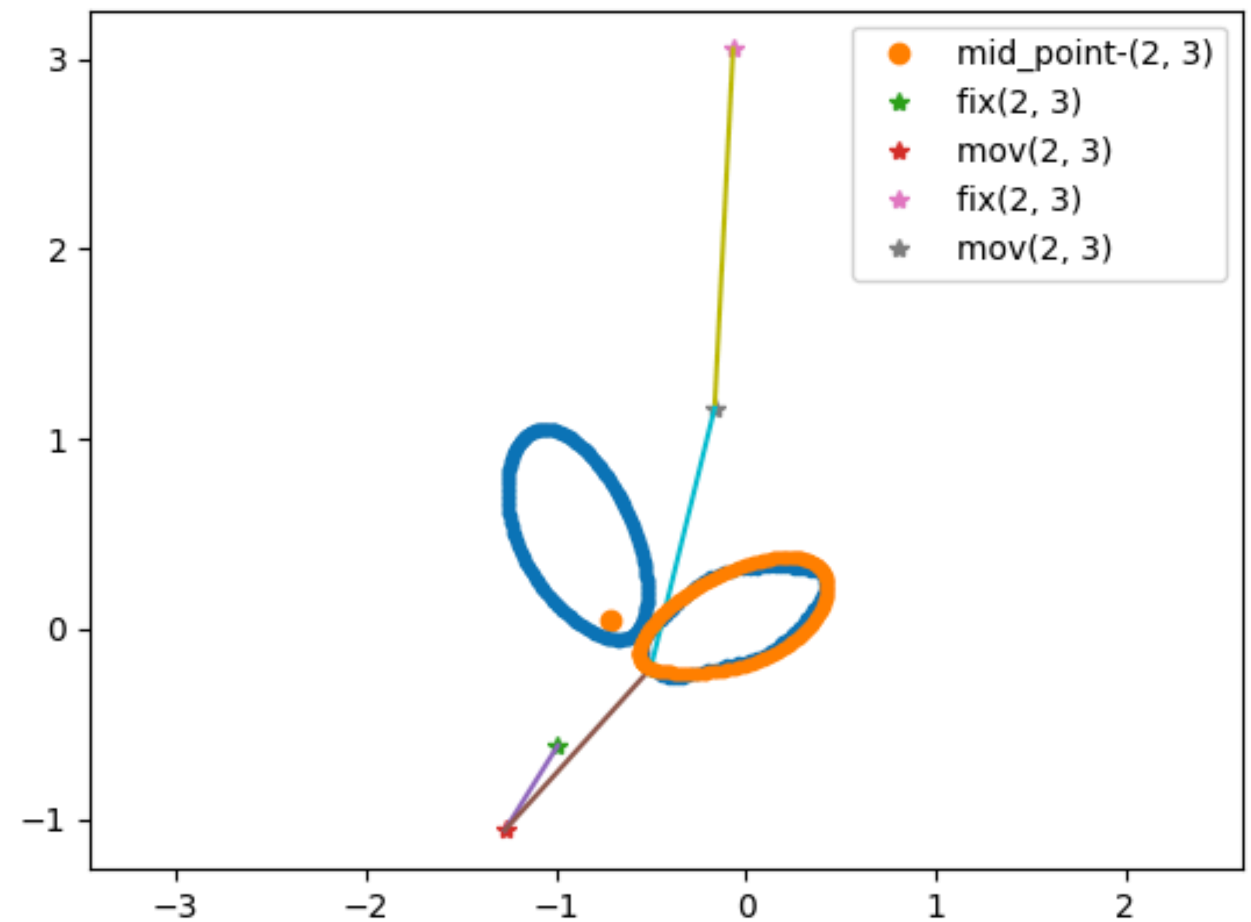




Input Augmentation and Conditioning Results

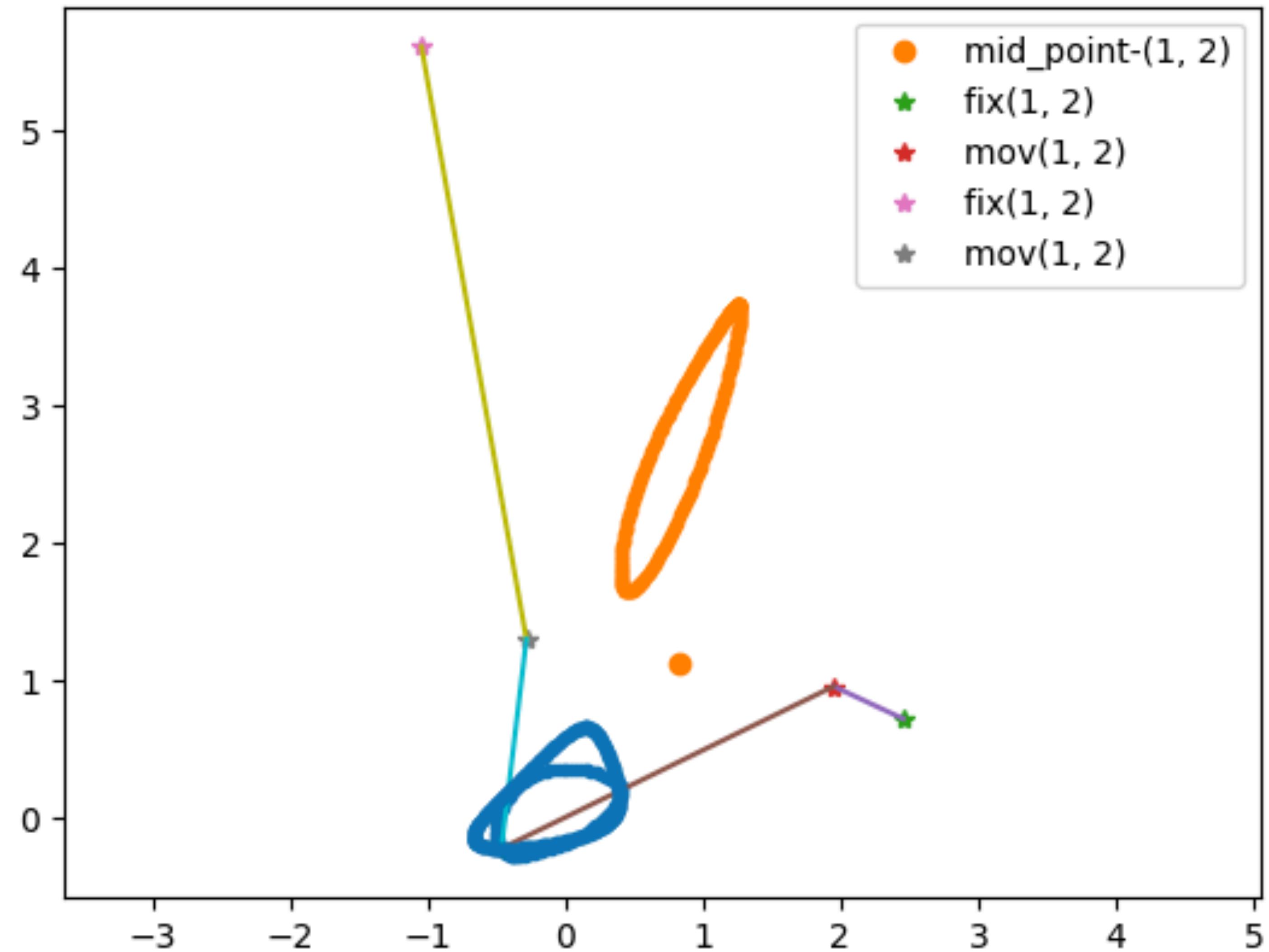
Concept 5





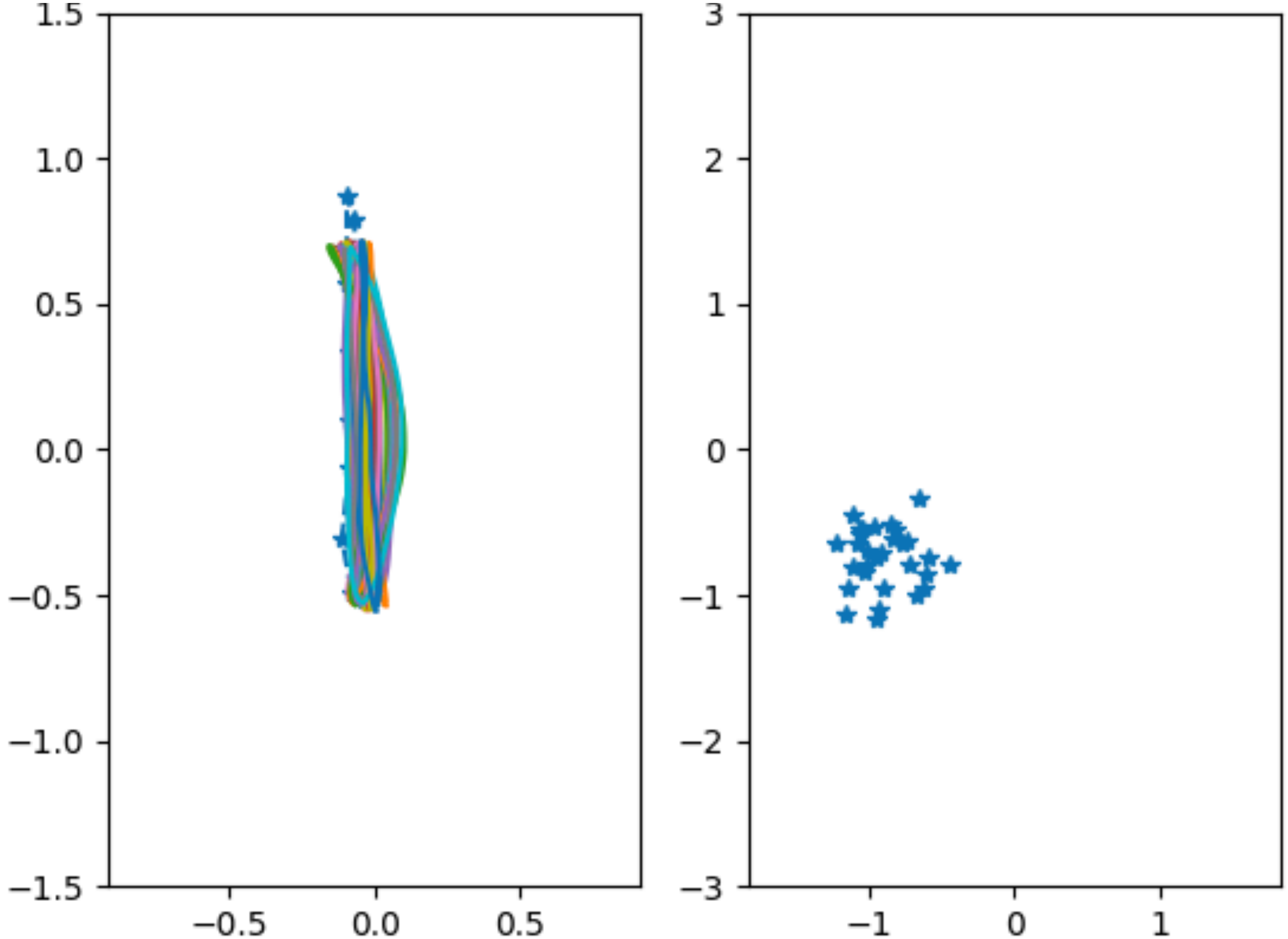
Input Augmentation and Conditioning Results

Concept 6



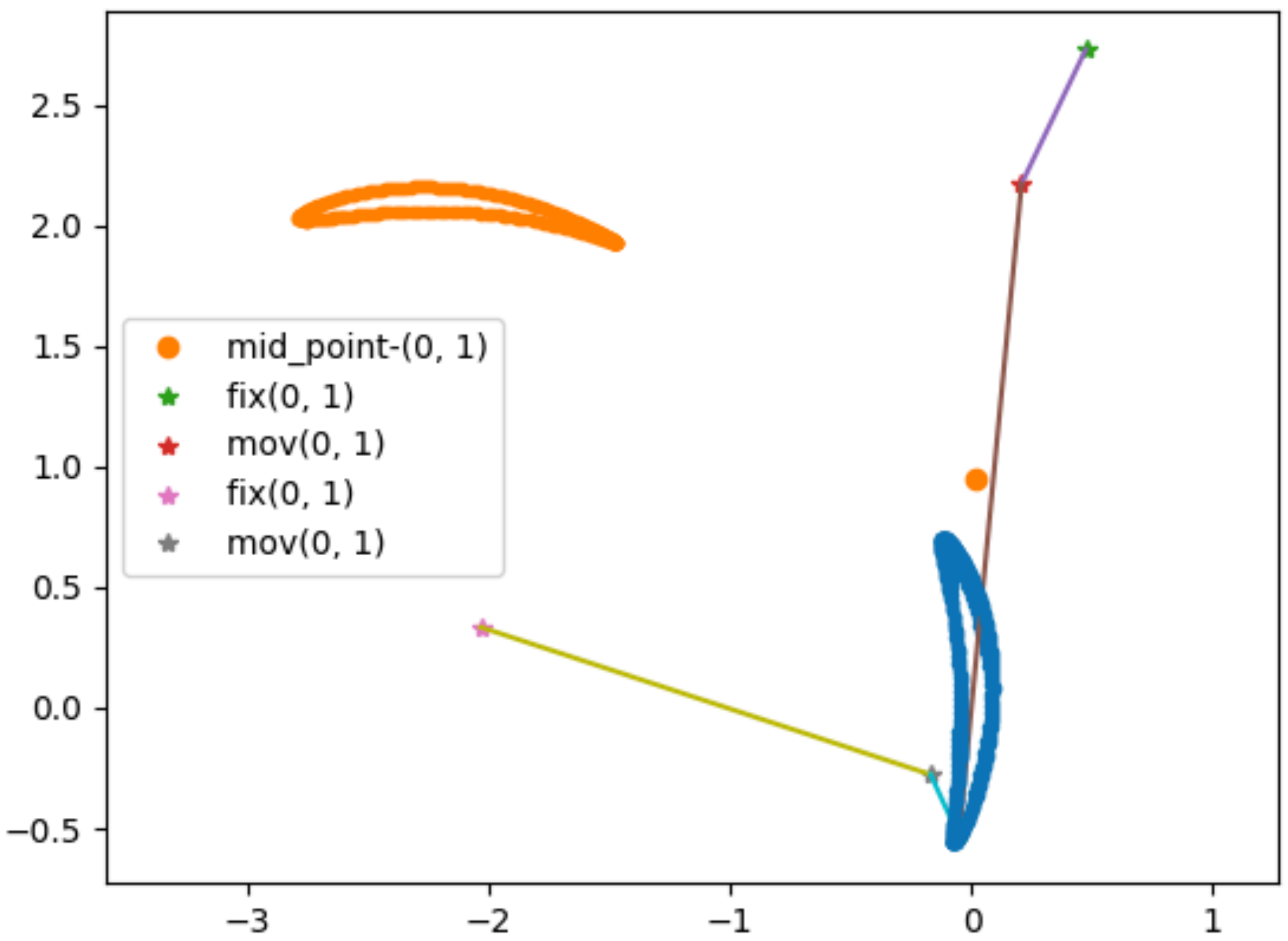
Input Augmentation and Conditioning Results

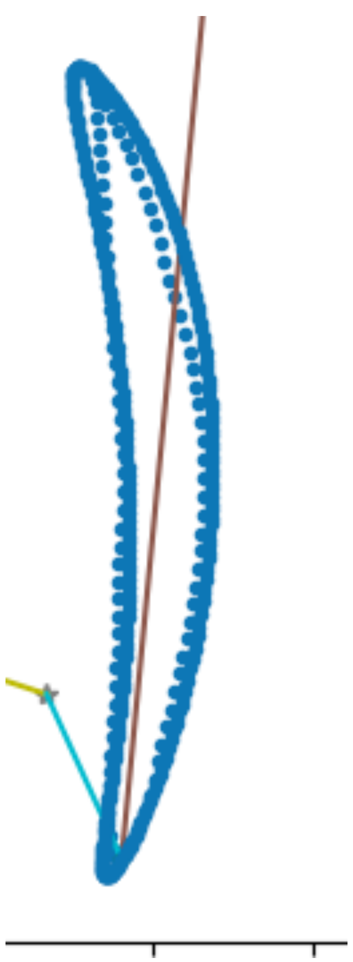
number of samples = 30



Input Augmentation and Conditioning Results

Concept 1



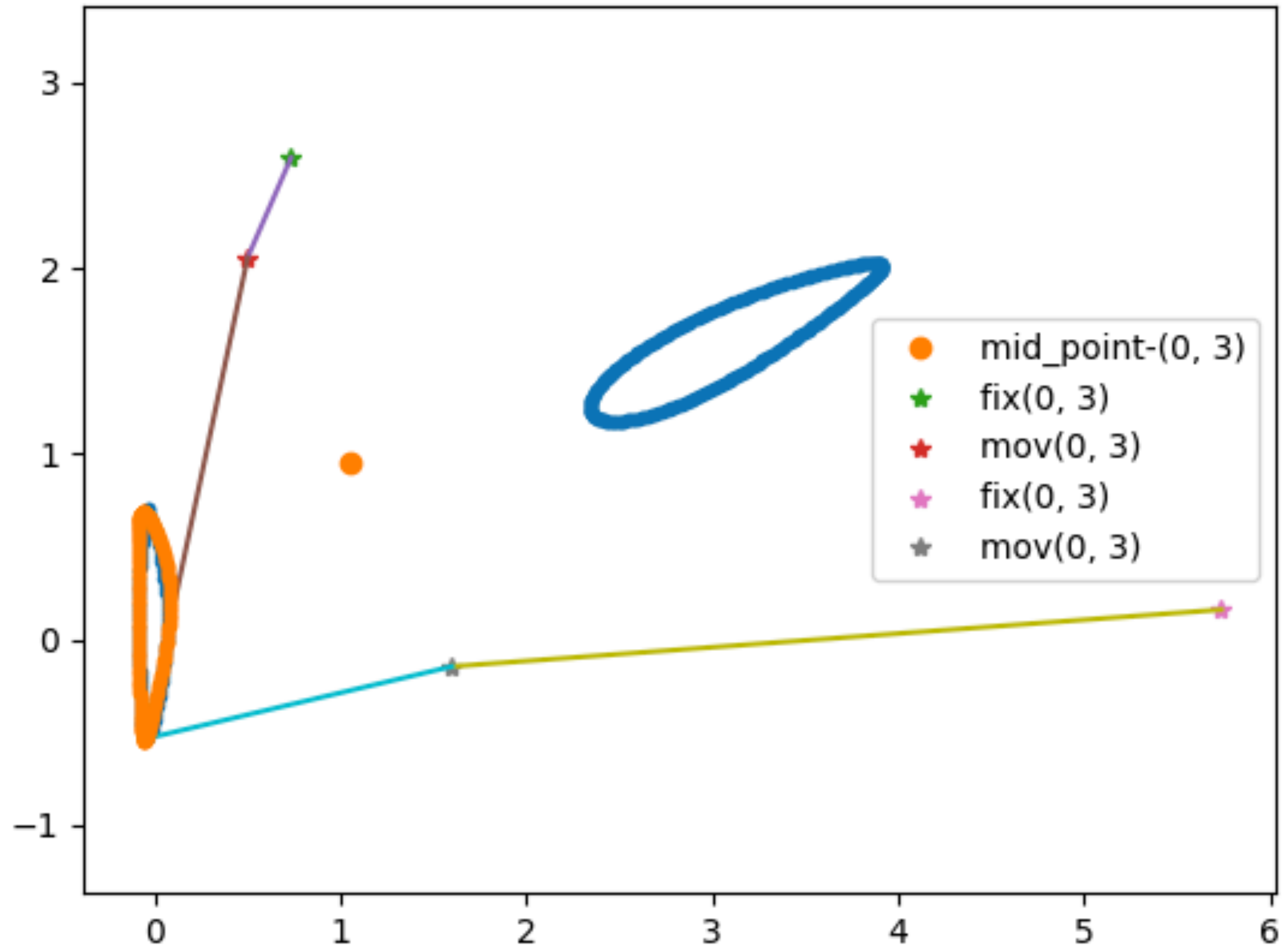


Input Augmentation and Conditioning Results

Concept 2

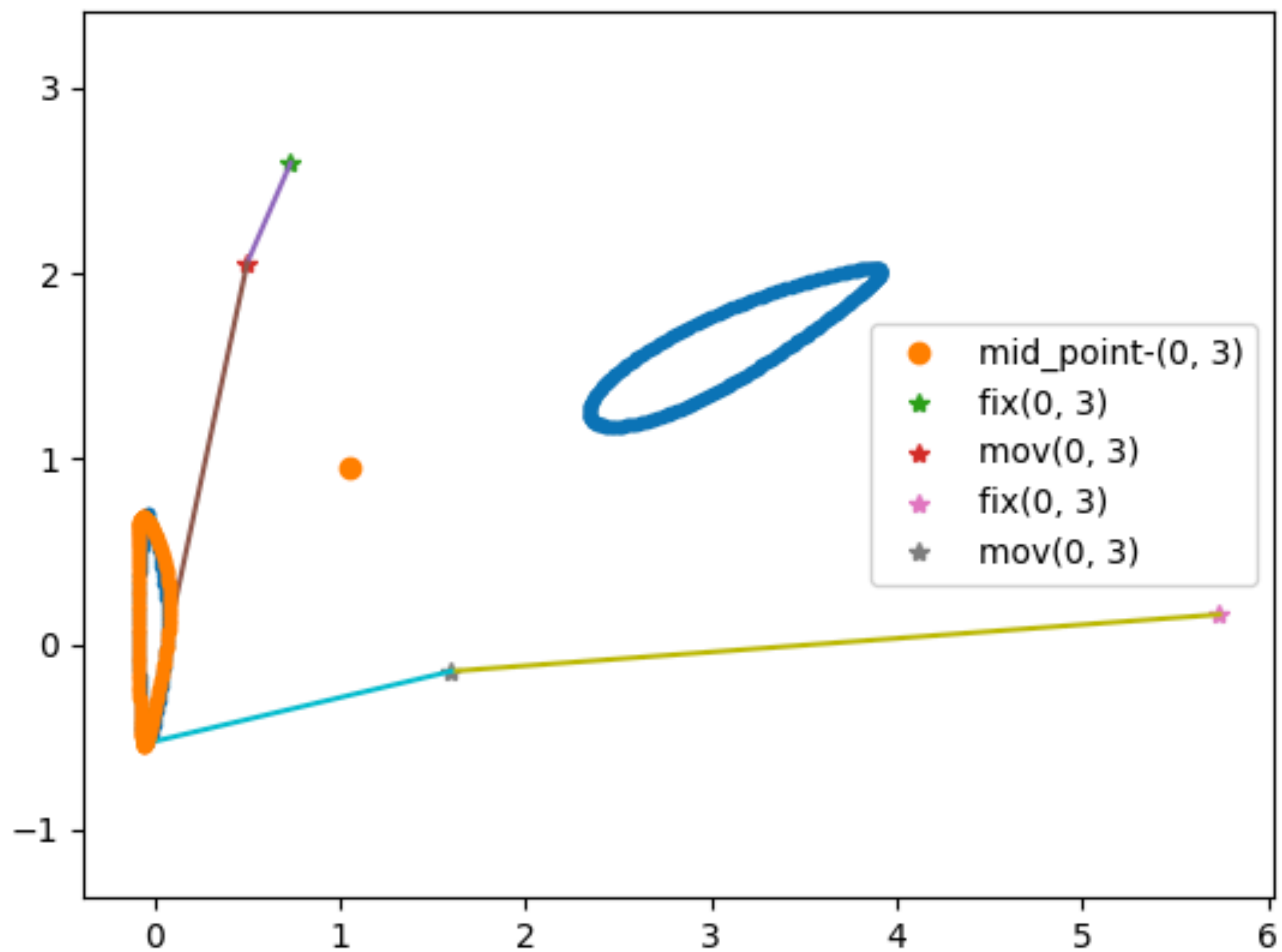
Input Augmentation and Conditioning Results

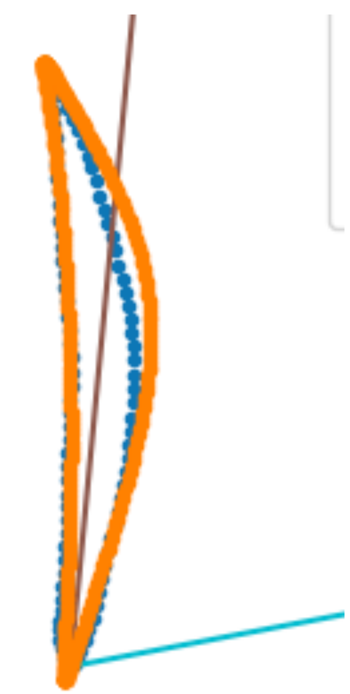
Concept 2



Input Augmentation and Conditioning Results

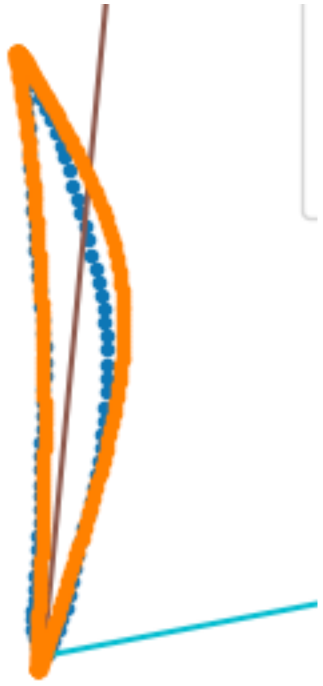
Concept 2





Input Augmentation and Conditioning Results

Concept 2

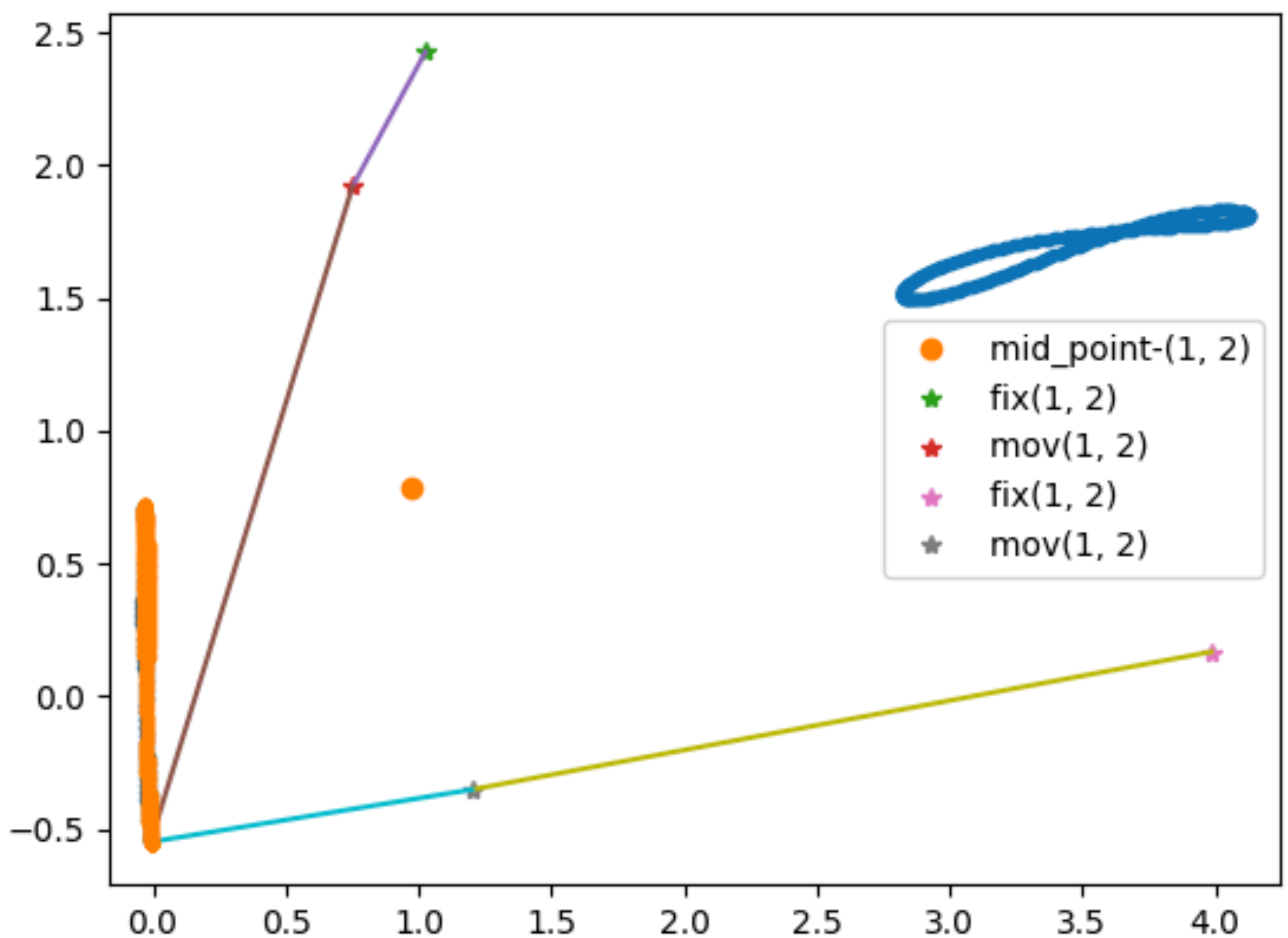


Input Augmentation and Conditioning Results

Concept 2

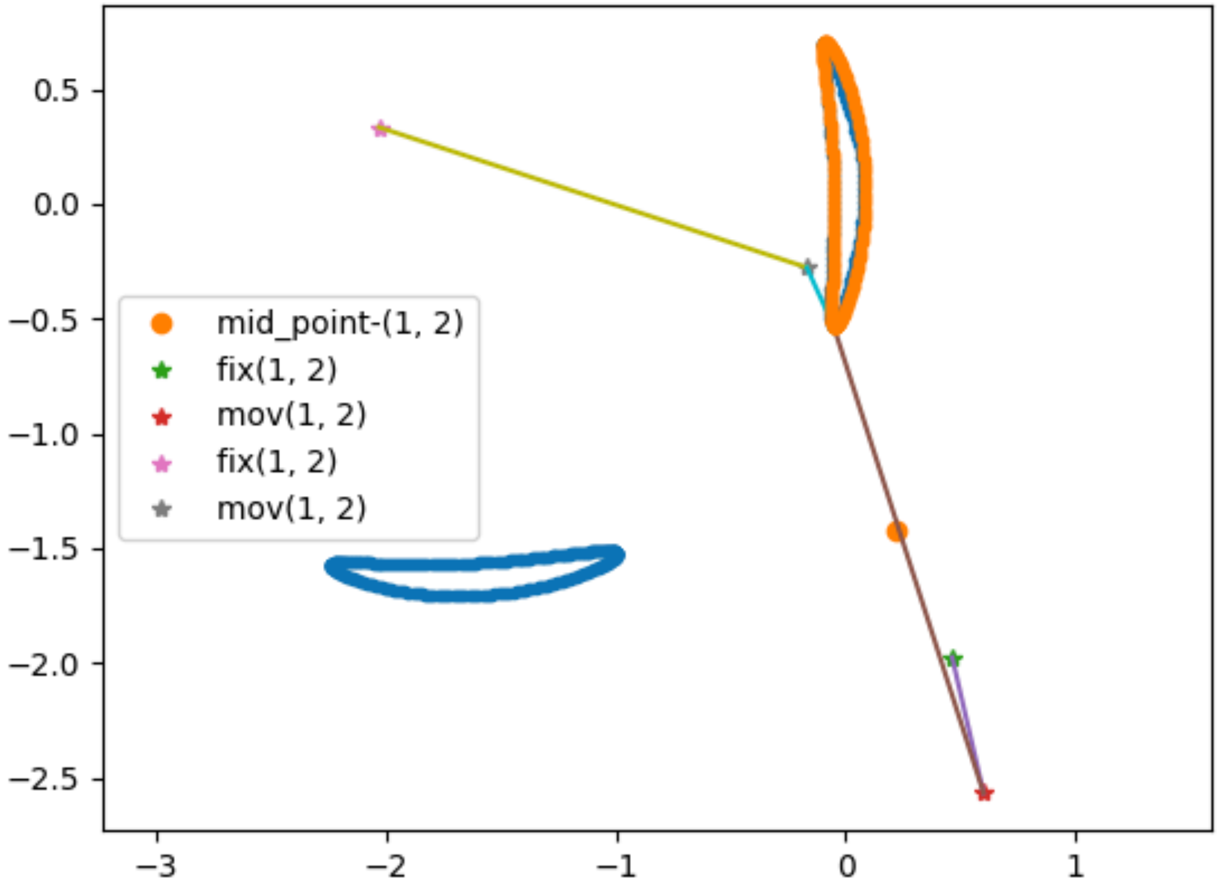
Input Augmentation and Conditioning Results

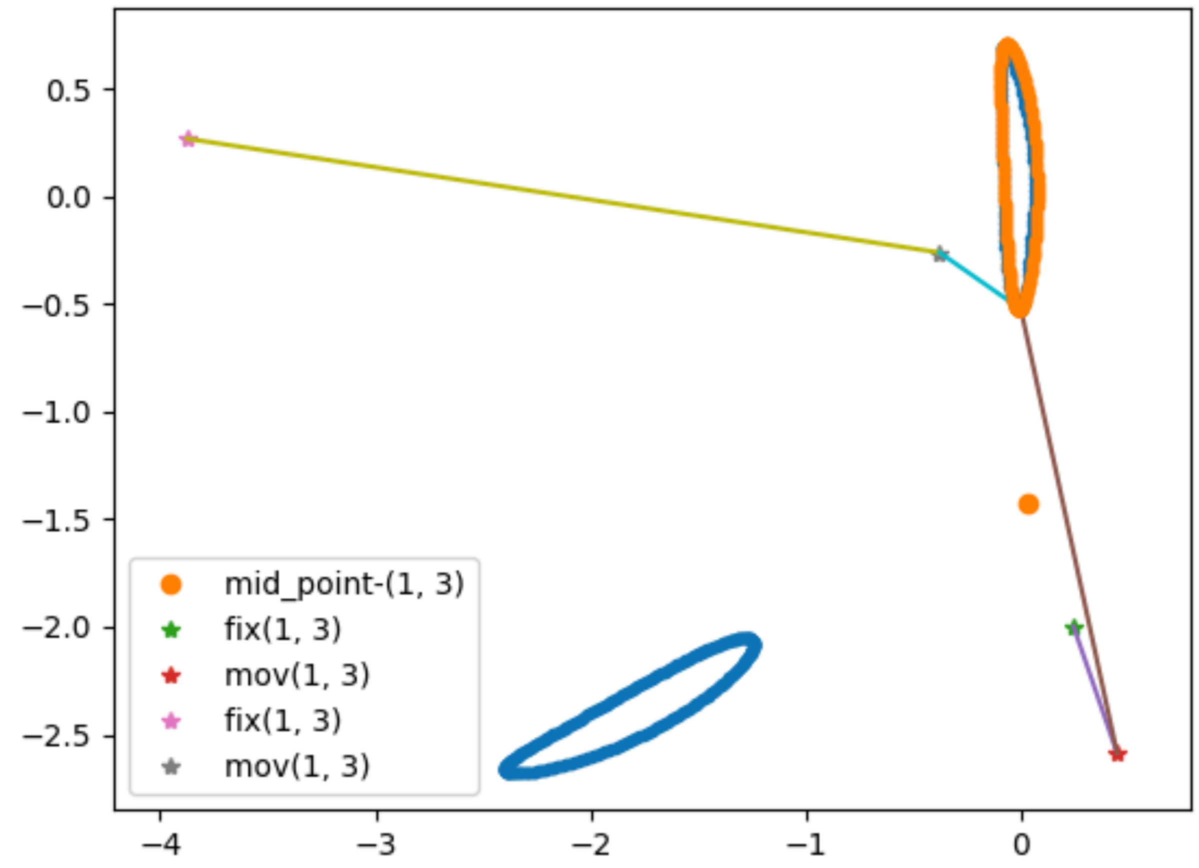
Concept 2



Input Augmentation and Conditioning Results

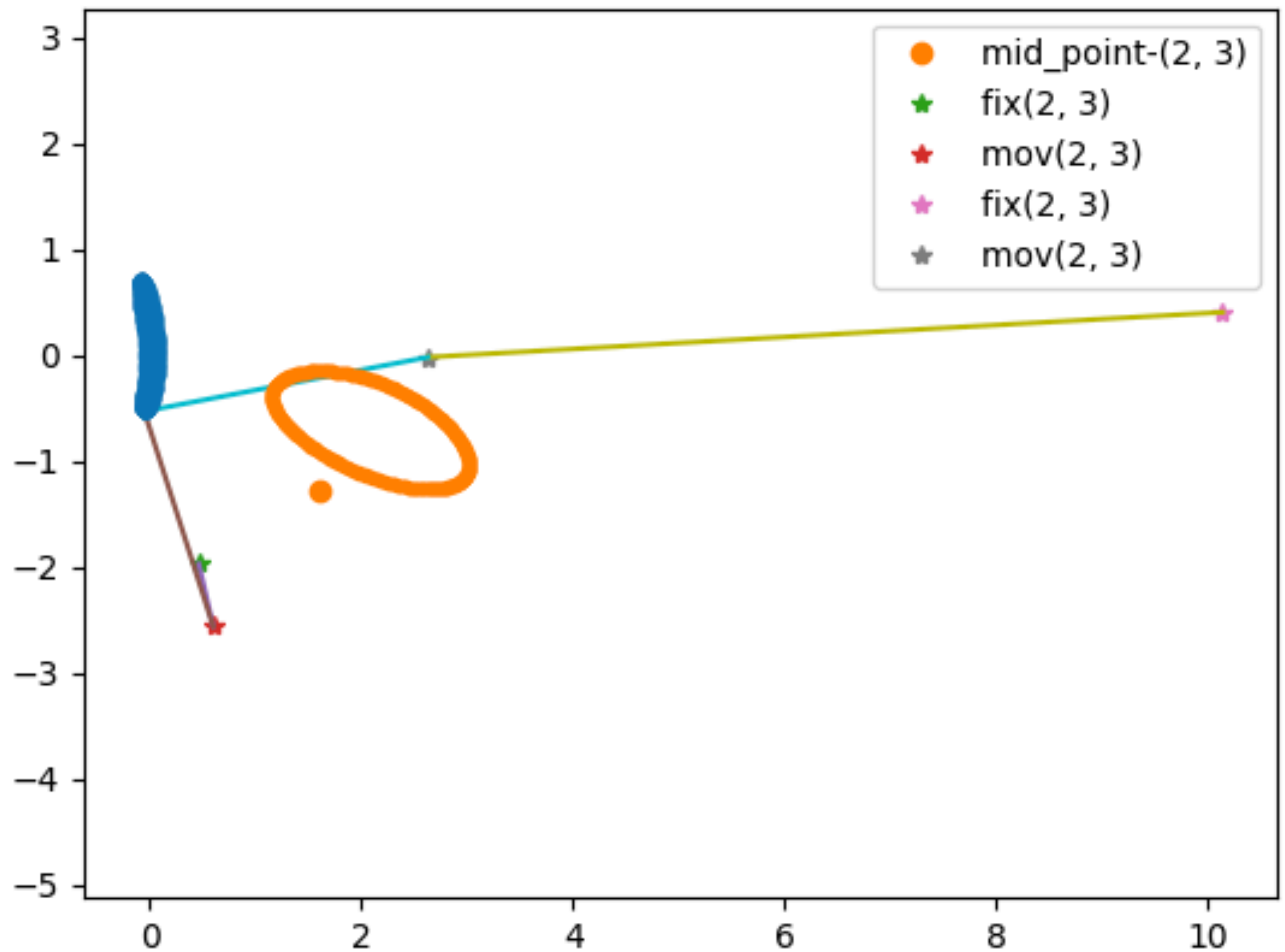
Concept 3





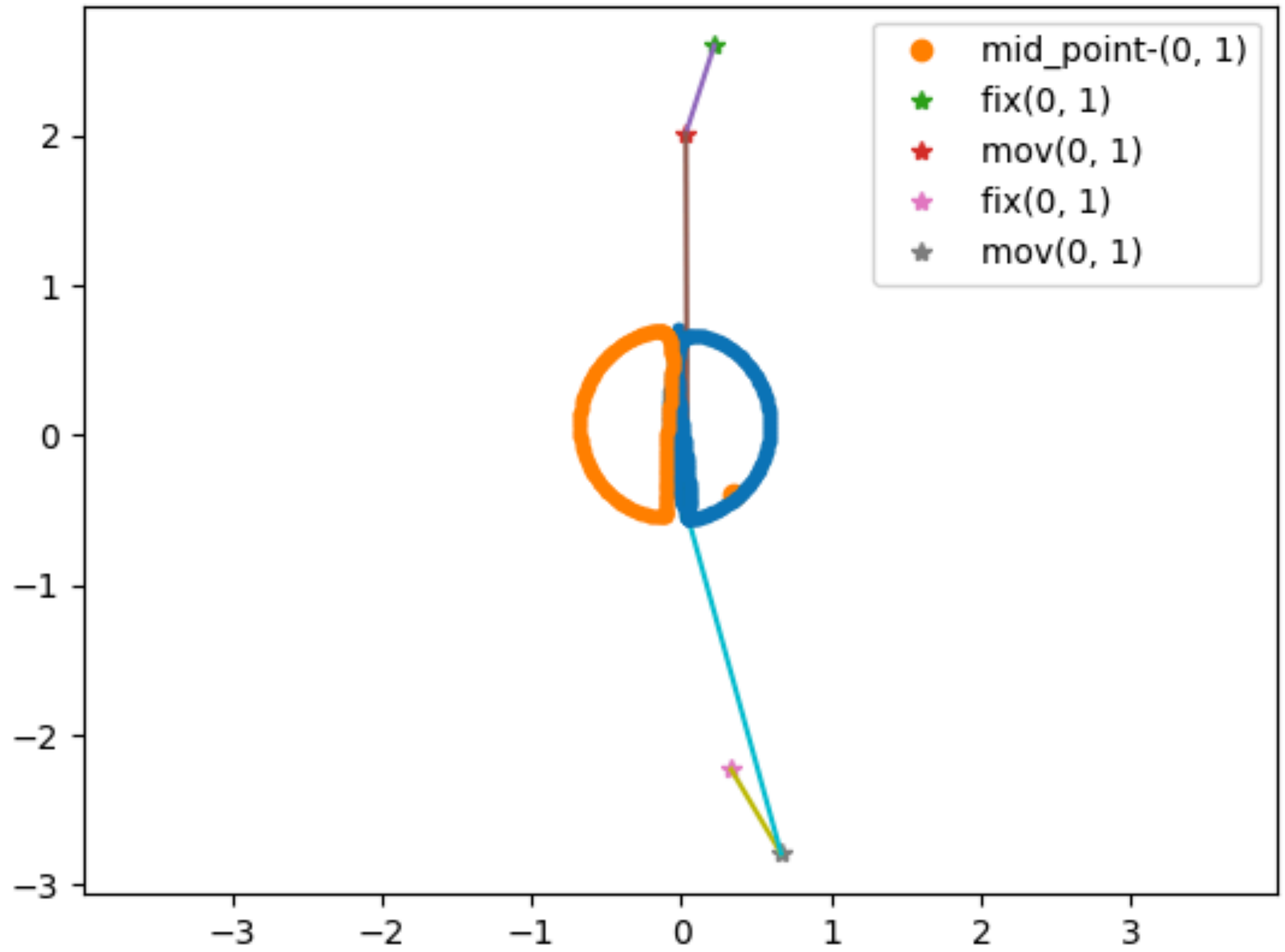
Input Augmentation and Conditioning Results

Concept 4



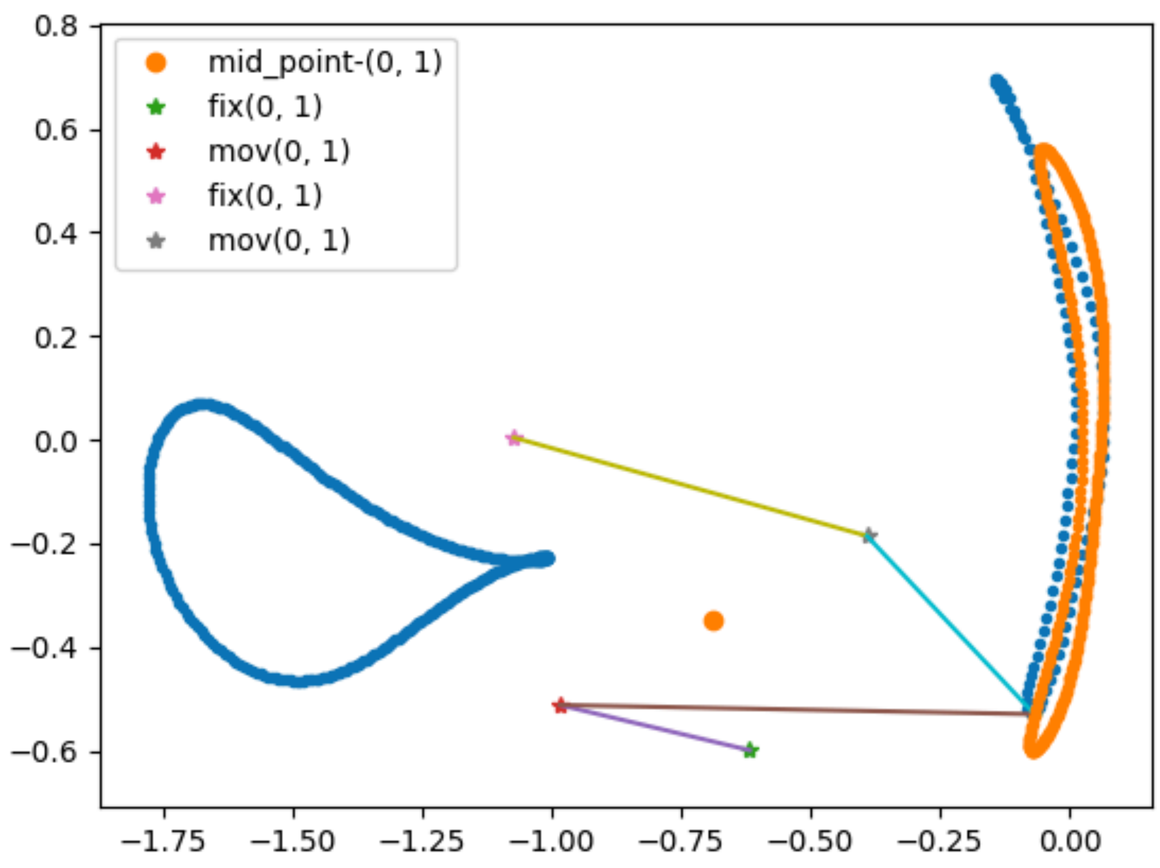
Input Augmentation and Conditioning Results

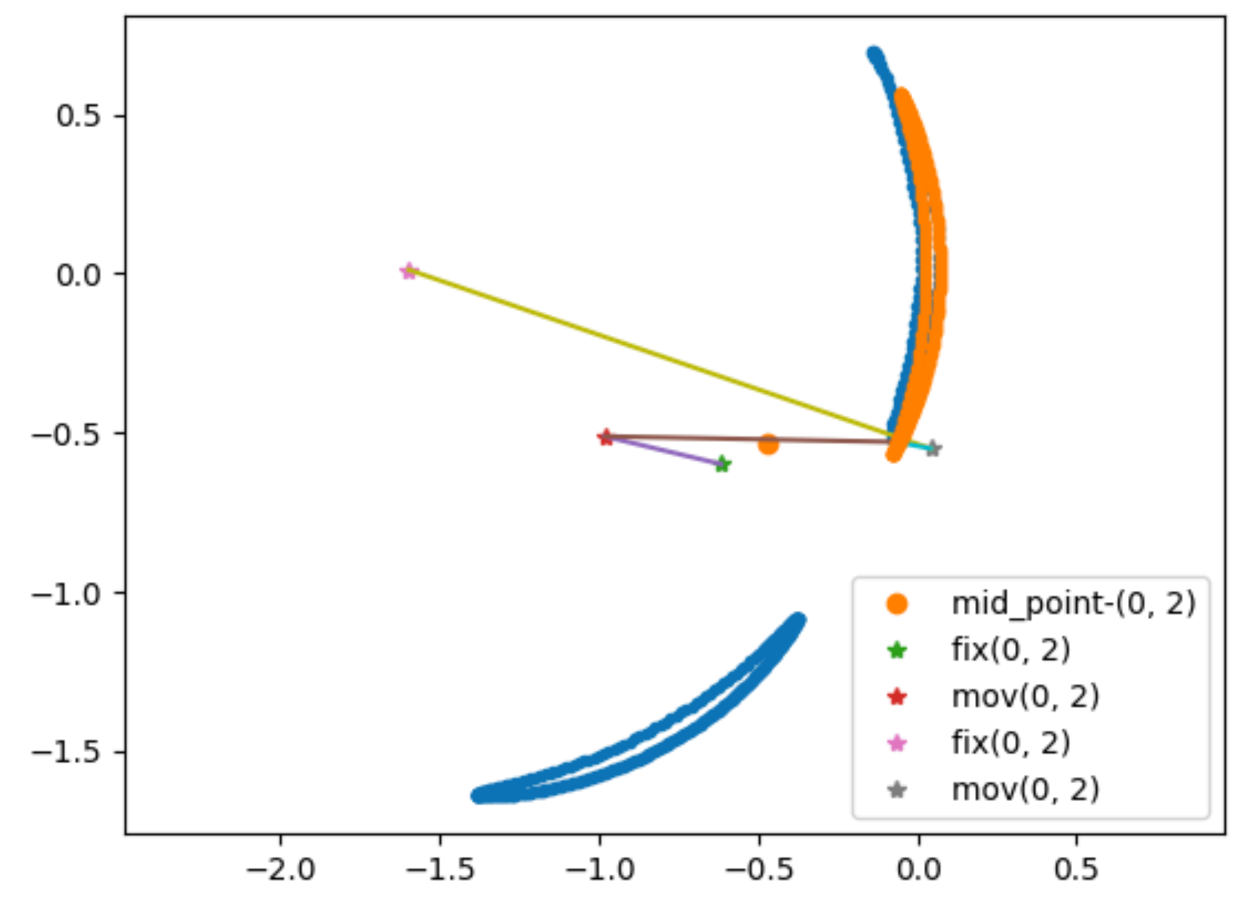
Concept 5



Input Augmentation and Conditioning Results

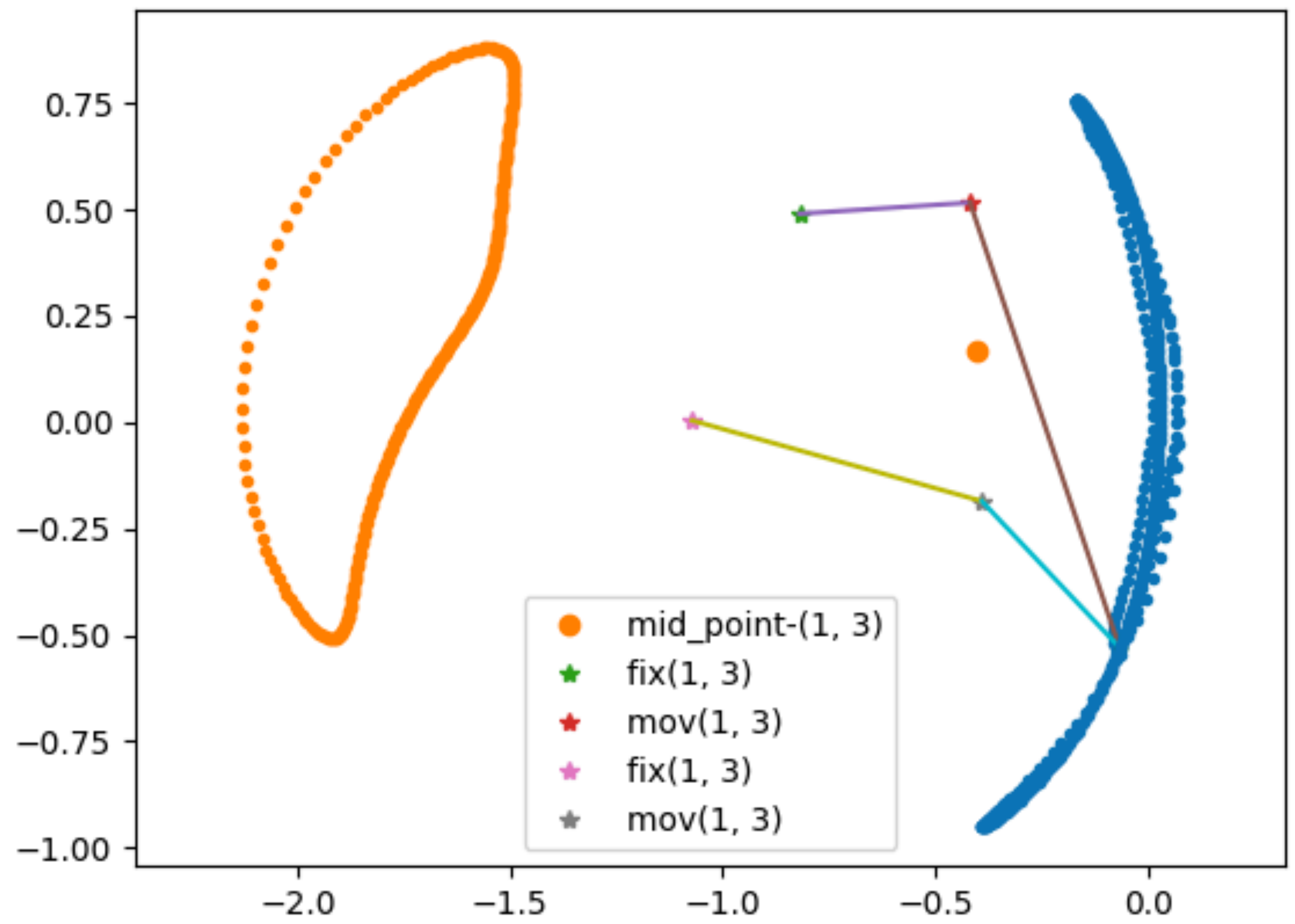
Concept 6





Input Augmentation and Conditioning Results

Concept 7



Conclusion

- Enhances Concept Generation efficacy of previously developed kinematic framework
- Greatly improves user's experience with input augmentation and conditioning
- Path synthesis problem was attempted for the approach validation, but the goal is to fill the gaps in problem setup
- Once a good problem is set, solvers can churn out meaningful solutions.

Conclusion

- Computations Aspects:
  - Solutions are computed in real-time
  - Entire Computation can be parallelize, and accelerated with GPU