REACHABILITY PROJECT LOG :)

1. Problem Description

Our goal is to learn *where the robot should stand* given a desired grasp.

1.1. **Problem inputs.** Formulating the problem, we are given as inputs:

- $x_e \in SE(3)$: desired end-effector pose in world coordinates
- $b \in SE(2)$: wheeled mobile base pose (x, y, γ)
- $q \in \mathbb{R}^n$: arm joint configuration.

1.2. **Problem outputs.** We want to find a range of feasible whole-body configurations that satisfy the following requirements:

(1) **IK feasibility:** forward kinematics$(b, q) = x_e$
(2) **Visibility/sensing constraints:** The end-effector should lie within the camera's field of view and range (and un-occluded by the robot itself), i.e. visible$(b, q, x_e) = 1$
(3) **Ball of feasible configurations:** We want a region in joint space around (b, q) that keeps the hand in approximately the right pose and satisfies the above constraints.

Our target is to model some distribution $p(b, q \mid x_e)$ subject to these constraints ↑, ideally with some notion of pose quality (visibility, manipulability, clearance, etc.).

2. Simplified problem

2.1. **Setup.** Simple robot is a round thing with a fixed "stick" attached to it: a disk on the floor with a rigid stick of fixed length L glued to it, and the stick rotates with the disk.

- Configuration: $Q = x, y, \theta$ ($\mathbb{R}^2 \times S^1$)
- Hand target: $H = h_x, h_y$ (point in $\mathbb{R}^2$)

In this setup, $H \in \mathbb{R}^2$ is the desired 2D point on the floor where we want the tip of the stick to be.

2.2. **Forward kinematics.**

$$\text{hand}(x, y, \theta) \coloneqq f(Q) = \begin{bmatrix} x \\ y \end{bmatrix} + L \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}$$

So the IK constraint ("hand hits the target") is:

$$H = \begin{bmatrix} h_x \\ h_y \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + L \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$$

Rearranging shows that as θ varies, the base center (x, y) traces out a circle of radius L around the target point H: the set of feasible base positions for the robot to reach H is a circle centered at H.

Ground truth feasible set for a fixed $H \coloneqq \{Q : f(Q) = H\} = \{(h_x - L\cos\theta, h_y - L\sin\theta, \theta) : \theta \in [0, 2\pi)\}$

2.3. **Approaches.** We try the following approaches to model $p(Q \mid H)$:

(1) *Nearest neighbors* (baseline):
   - Given a dataset $\mathcal{D} = \{(H_i, Q_i)\}_{i=1}^N$: for a query H',
     (a) Find indices of the k nearest H_i to H' (under $||H_i - H'||$)
     (b) Return their associated Q_i's (either all or sample one etc.)
   - For a new H', we can return the Q_i associated to the nearest H_i, or sample from an empirical conditional distribution created based on the nearest neighbor graph: $\hat{p}_{\text{kNN}}(Q \mid H') = \sum_{j \in \mathcal{N}_k(H')} w_j(H')\delta(Q - Q_j)$ (where weight w_j is proportional to the distance between H' and H_i, i.e. $w_j \propto \exp(-||H_j - H'||^2/\sigma^2)$).
(2) *Conditional VAE:* Introduce latent $z \in \mathbb{R}^d$: $z \sim \mathcal{N}(0, I),\ Q \sim p_\theta(Q \mid H, z)$

Date: January 4, 2026.

- Train an encoder $q_\phi(z \mid Q, H)$ and decoder $p_\theta(Q \mid H, z)$ via ELBO loss:

$$\log p_\theta(Q \mid H) \geq \mathbb{E}_{z \sim q_\phi(z|Q,H)}[\log p_\theta(Q \mid H, z)] - \mathrm{KL}[q_\theta(z \mid Q, H) \,||\, \mathcal{N}(0, I)]$$

- Potential decoder choice: output $\mu_\theta(H, z)$ and diagonal $\Sigma_\theta(H, z)$; use Gaussian likelihood:

$$\log p_\theta(Q \mid H, z) = \log \mathcal{N}(Q; \mu_\theta(H, z), \Sigma_\theta(H, z))$$

[TODO: Read into math more], notes in 3.2

- Advantages of cVAE for this problem:
  - Multi-modality via latent variable z (different z's can map to different valid configurations on the circle) $\rightarrow$ addresses mode collapse relative to deterministic nets [assuming we avoid posterior collapse (decoder ignores z)]
  - Smooth generalization in H: theoretically, the encoder/decoder can learn continuous maps in H-space, allowing better interpolation than the nearest neighbors approach.
- Note: θ is periodic, so training a Gaussian likelihood $p_\theta(Q \mid H, z)$ (in the decoder) on raw $\theta \in [0, 2\pi)$ is awkward because $\theta \approx 0$ and $\theta \approx 2\pi$ are close physically but far numerically ($\hat{\theta} = 2\pi - 0.01$ is a pretty good prediction for ground truth $\theta = 0.01$). To avoid this, instead of modeling $\theta \sim \mathcal{N}(\mu, \sigma^2)$, we can model a Gaussian over Cartesian coordinates in $\mathbb{R}^2$:

$$(\cos\theta, \sin\theta) \sim \mathcal{N}(\mu_{cs}, \mathrm{diag}(\sigma_{cs}^2))$$

  We can recover θ via atan2 at inference time.

(3) *Invertible NN* (FrEIA): [TODO: Read into math more], notes in 3.3

- Invertibility requires equal dimensions $\rightarrow$ in the toy problem, $\dim(Q) = 3 \neq \dim(H) = 2$, so add a latent $z \in \mathbb{R}^1$ s.t. $(H, z) \in \mathbb{R}^3$
- Learn an invertible map: $F_\theta : Q \leftrightarrow (\hat{H}, z)$ where z is encouraged to be $\mathcal{N}(0, 1)$ and $\hat{H}$ matches H
- $\uparrow$ Training objectives:
  - Forward consistency: $\hat{H} \approx H \rightarrow \mathcal{L}_H = ||\hat{H} - H||^2$
  - Latent regularization: make $z \sim \mathcal{N}(0, 1)$
- At test time, the inverse we want is:

$$Q = F_\theta^{-1}(H, z), \quad z \sim \mathcal{N}(0, 1)$$

- Added FK constraint (optional loss regularization term):

$$\mathcal{L} = \mathcal{L}_{\mathrm{NLL}} + \boxed{\lambda_{\mathrm{FK}} \mathbb{E}_{z \sim \mathcal{N}}[\mathrm{FK\_error}(f^{-1}(z; H), H)]}$$

  The hope is that when we penalize FK on generated samples x conditioned on H, we directly tell the model that "when you sample from $p_\theta(q \mid H)$, those samples should actually reach H."

2.4. **Evaluation.** Assuming that the ground-truth $p^*(Q \mid H)$ is a uniform distribution over the feasible circle set, i.e.: $\theta \sim \mathrm{Unif}[0, 2\pi) \rightarrow x = h_x - L\cos\theta$, $y = h_y - L\sin\theta$:

- Accuracy: $e_{\mathrm{hand}}(Q, H) \coloneqq ||f(Q) - H||$
- Diversity/coverage: convert each sample to its implied angle on the circle:

$$\theta_{\mathrm{implied}}^{(s)} \coloneqq \mathrm{atan2}\left(h_y - y^{(s)}, h_x - x^{(s)}\right)$$

  Check how well the empirical distribution over θ matches the target (here we assume $\theta \sim \mathrm{Unif}[0, 2\pi)$): for histogram $\hat{p}(\theta)$, we can compute:
  - KL divergence to uniform: $D_{\mathrm{KL}}(\hat{p}(\theta) \,||\, \mathrm{Unif})$
  - Max angle gap: sort angles $\theta^{(s)}$ around the circle, compute max gap $\Delta_{\max} = \max_i(\theta_{i+1} - \theta_i)$ Large $\Delta_{\max}$ = model is missing big arcs (collapse)
- "Uses latent" test (for cVAE/cINN): fix H, sample many z's, then measure output variance $\mathrm{Var}(Q \mid H)$. If the model ignores z, this variance will be small.
- Generalization in H: make a test set where H is (1) in-distribution, (2) near boundary, (3) out-of-distribution (slightly outside training workspace). Compare success and coverage!
- Inference speed per sample

2.5. **Extending to more general cases.** Ideally, would be nice if this toy problem gave us an idea of the advantages/disadvantages of different approaches in regards to the following challenges:

- *Mode collapse:* Test: fixed H, sample 1k solultions, compute $\Delta_{\max}$, KL-to-uniform, etc.
- *Generalization:* Test: distribution of $e_{\text{hand}} = ||f(Q) - H||$
- *Bias from data collection* (forward vs. inverse sampling):

2.6. **Potential next-step extensions.**

- Disconnected feasible sets: modify toy with a "visibility" constraint such as

$$\text{visible}(Q, H) = 1 \iff \theta \in [\alpha_1, \beta_1] \cup [\alpha_2, \beta_2]$$

  Now the true $p^*(Q \mid H)$ is two separated modes. **Test:** cluster sampled angles into arcs and compute mode recall: fraction of samples that land in each arc.
- Scalability with dimension: add DoF to the toy (e.g., 2-link arm, variable stick length). **Test:** track how coverage/error degrades with dimension.

3. Extra notes

3.1. **Mode collapse.** A generative model has mode collapse if, for a fixed condition H, the samples $Q \sim \hat{p}(Q \mid H)$ cover only a small subset of the true support of $p^*(Q \mid H)$.

- In this toy problem, the support is the full circle parameterized by θ. Mode collapse looks like the model always outputting $\theta \approx 0$ (i.e. stands in one favorite location) or outputting only a few discrete angles, ignoring the rest the circle, etc.

3.2. **Conditional VAE theory.** The goal is to model a multi-modal conditional distribution over robot configurations Q given target H: $p^*(Q \mid H)$. In our simple robot setup, the true conditional has a 1D manifold of solutions (a circle in (x, y) paired with corresponding θ), so it's not unimodal in the usual sense (rather than being a single blob in $\mathbb{R}^d$, for fixed H, probability mass $p^*(Q \mid H)$ is concentrated along a ring in (x, y) with a corresponding orientation θ at each point).

3.2.1. *Motivation.* Because $p^*(Q \mid H)$ is not unimodal, if we model $p(Q \mid H)$ as a single Gaussian/predict a single mean, we face mean collapse: the mean of points around a circle is the center of the circle, but the center is invalid. So the "best" unimodal prediction is often physically invalid/low-probability under the true distribution.

3.2.2. *Model architecture.* We introduce latent $z \in \mathbb{R}^{d_z}$, which ideally will be the variable that "chooses" *which* solution the model means among the many possible solutions on the ring for a given H. Define a prior $p(z) = \mathcal{N}(0, I)$. There are two parts to the model (decoder and encoder):

- Decoder (generative model): The decoder is a conditional distribution $p_\theta(Q \mid H, z)$. For the architecture of the decoder, we use the diagonal Gaussian (potential next step – try other architectures):

$$p_\theta(Q \mid H, z) = \mathcal{N}(Q; \mu_\theta(H, z), \text{diag}(\sigma^2_\theta(H, z)))$$

  where $\mu_\theta(H, z)$ and σ^2_θ are the predicted mean/variance of each coordinate of Q.
- *Interlude:* $p_\theta(Q, z \mid H) = p(z \mid H)p(Q \mid z, H) = p(z)p_\theta(Q \mid H, z)$ [assuming $p(z \mid H) = p(z)$]. Bayes' rule gives:

$$p_\theta(z \mid Q, H) = \frac{p_\theta(Q, z \mid H)}{p_\theta(Q \mid H)} = \frac{p(z)p_\theta(Q \mid H, z)}{p_\theta(Q \mid H)}$$

  Expanding the denominator: $p_\theta(Q \mid H) = \int p(z')p_\theta(Q \mid H, z')\, dz'$ (consider all possible latent explanations z', weight how likely each would produce Q, then sum them up). Plugging this in,

$$p_\theta(z \mid Q, H) = \frac{p(z)p_\theta(Q \mid H, z)}{\int p(z')p_\theta(Q \mid H, z')\, dz'}$$

  The denominator $\int p(z')p_\theta(Q \mid H, z')\, dz'$ is intractable: $p_\theta(Q \mid H, z') = \mathcal{N}(Q; \mu_\theta(H, z'), \Sigma_\theta(H, z'))$, so we effectively have

$$p(z')p_\theta(Q \mid H, z') \Rightarrow \mathcal{N}(z'; 0, 1) \times \mathcal{N}(Q; \mu_\theta(H, z'), \Sigma_\theta(H, z'))$$

For fixed Q and H, $\mathcal{N}(Q; \mu_\theta(H, z'), \Sigma_\theta(H, z'))$ becomes some complicated nonnegative function of $z' \to g(z')$. Then,

$$p_\theta(Q \mid H) = \int \mathcal{N}(z'; 0, I) g(z')\, dz'$$

This is very hard to compute analytically: recall the multivariate Gaussian density is given by:

$$f_{\mu,\Sigma}(Q) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\tfrac{1}{2}(Q-\mu)^\top \Sigma^{-1}(Q-\mu)\right)$$

Here, we plug in $\mu = \mu_\theta(H, z'), \Sigma = \Sigma_\theta(H, z')$. These are neural-net outputs, i.e. complicated nonlinear functions of z', which makes the integral difficult to compute.

Since the true posterior is intractable, introduce a tractable approximation $q_\phi(z \mid Q, H) : (Q, H) \mapsto p(z)$ which essentially answers "If this Q came from some latent z, what must that z have been?"

- Encoder (inference model): The encoder is the conditional distribution $q_\phi(z \mid Q, H)$.

3.3. **FrEIA INN theory.** The goal is to learn $p(Q \mid H)$, where $Q = (x, y, \cos\theta, \sin\theta) \in \mathbb{R}^4$ (equivalent form of (x, y, θ)) and $H = (h_x, h_y) \in \mathbb{R}^2$. A normalizing flow models an invertible map (for each fixed condition h):

$$f_\theta(\cdot; h) : \mathbb{R}^4 \to \mathbb{R}^4, \quad z = f_\theta(q; h), \quad q = f_\theta^{-1}(z; h)$$

We pick the base density for z to be the multivariate standard normal:

$$p_Z(z) = \mathcal{N}(0, I_4)$$

Then, $p_\theta(q \mid h)$ can be defined via "push q through f to get z, and measure how likely that z is, corrected by volume change" $\to$ correction is the Jacobian determinant.

3.3.1. *Hand-wavy intuition.* For each condition h, the true data q lives in some weird, potentially multi-modal shape (ring of solutions in our simplified problem). A Gaussian cannot represent that shape directly.

- *Core idea:* Instead of trying to fit a complicated distribution directly, we learn a warp of space that makes the complicated shape look Gaussian.
  - Forward direction: $z = f_\theta(q; h)$ takes real samples q and maps them into "Gaussian space"
  - Training enforces: "after mapping, these z's should look like samples from $\mathcal{N}(0, I)$"
  - Warping space changes volume $\to$ probability densities change:
    - If f expands a region of q-space, then points there become less dense in q-space.
    - If f compresses a region, points become more dense.

After learning an invertible NN model, we can generate realistic q by:

(1) Sample $z \sim \mathcal{N}(0, I)$
(2) Invert: $q = f_\theta^{-1}(z; h)$

Fun way of thinking about it: Imagine data for q's is shaped like a bent pretzel in $\mathbb{R}^D$. A flow learns a continuous, invertible deformation of space that morphs the pretzel into a D-dim. Gaussian blob. Forward pass = "unbend the pretzel into a Gaussian." Inverse pass = "bend a Gaussian blob back into a pretzel." The Jacobian term is bookkeeping of how the deformation stretches space.

3.3.2. *Why layers need to be invertible.*

(1) Invertible layers allow flow models to compute an exact probability for any observed training point: if Q is a real solution from the dataset for some H, we can compute exactly how likely the model thinks it is (flows are the only model out of GANs, VAEs, and diffusion models that can compute the exact log-likelihood of a new sample)

$$\Pr(\mathbf{f}(\mathbf{Q}) \mid \mathbf{H}, \phi) = \left|\frac{\partial \mathbf{f}(\mathbf{z}, \mathbf{H}, \phi)}{\partial \mathbf{z}}\right|^{-1} \cdot \Pr(\mathbf{z}) \text{ where } \mathbf{z} = \mathbf{f}^{-1}(\mathbf{x}, \mathbf{H}, \phi) \text{ and } \phi = \text{learned model params}$$

(2) Invertibility allows sampling to be one inverse pass rather than iterative denoising, sampling chains, etc.: once trained, we can directly (1) sample $z \sim \mathcal{N}(0, I)$ (2) compute $Q = f_\theta^{-1}(z; H)$

3.3.3. *More concrete math.* In separate PDF (`flow model FrEIA implementation notes.pdf`).

3.4. **Normalizing flow.** In separate PDF (`normalizing flow notes.pdf`).

3.5. **Conceptual comparison of different methods.**

3.5.1. *Flow.*

3.5.2. *cVAE.*

3.5.3. *Diffusion.*

3.6. **Invertible neural networks vs. normalizing flow.**

References