Appendices

A Terminal Losses

This section provides detailed derivations of the different terminal loss functions, $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G})$, compared in our generalized Bayesian inference planner. Recall, the loss function $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G})$ measures the discrepancy between the terminal state distribution and the distribution of goal states, represented implicitly by samples $\tilde{X}_T$ and $\tilde{G}$ respectively.

Kernel Maximum Mean Discrepancy (MMD). The squared MMD [1] measures discrepancy between two distributions $p(x)$ and $q(y)$, and is defined as:

$$\text{MMD}^2(p, q) = \mathbb{E}[k(x, x')] - 2\mathbb{E}[k(x, y)] + \mathbb{E}[k(y, y')] \tag{1}$$

where $k(\cdot, \cdot)$ is a kernel function. In the case where distributions p and q are implicit, an unbiased two-sample approximation of the squared MMD is given by:

$$\text{MMD}_u^2(p, q) = \frac{1}{m-1}\sum_{i=1}^{m}\sum_{j\neq i}^{m} k(x_i, x_j) - \frac{2}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} k(x_i, y_j) + \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}^{n} k(y_i, y_j) \tag{2}$$

where $x_i \sim p$, $1 \leq i \leq m$ and $y_i \sim q$, $1 \leq i \leq n$. When applied to our setting we simply define $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G}) = \text{MMD}^2(\tilde{X}_T, \tilde{G})$ where $\tilde{X}_T = \{x_T^{(i)}\}_{i=1}^M$ correspond to samples from the terminal state associated with the corresponding trajectory particle $\tau^{(i)}$.

KL Divergence. A differentiable two-sample KL divergence approximation can be obtained through density ratio estimation via classification [2, 3]. The KL divergence between distributions p and q is defined as:

$$D^{\text{KL}}(p, q) = \mathbb{E}_{x\sim p(x)}\left[\log \frac{p(x)}{q(x)}\right]. \tag{3}$$

This quantity can be approximated over samples $x^{(i)} \sim p(x)$ if the ratio $r(x) = p(x)/q(x)$ can be computed analytically:

$$\mathbb{E}_{x\sim p(x)}\left[\log r(x)\right] \approx \frac{1}{M}\sum_{i=1}^{M} \log r(x^{(i)}) \tag{4}$$

In the case where the goal distribution and posterior distribution are implicit, the ratio $r(x)$ cannot be evaluated. Instead, the ratio can be approximated through classification, where binary label $y = 1$ indicates that a sample x was drawn from p and $y = 0$ indicates a sample was drawn from q. The ratio can then be approximated as:

$$r^*(x) = \frac{P(x|y=1)}{P(x|y=0)}, \tag{5}$$

which can be further simplified to:

$$r^*(x) = \exp(\sigma^{-1}(P(y = 1 \mid x))). \tag{6}$$

This approximation reduces the density ratio estimation to a classification problem. Thus, the ratio is computed by evaluating a binary classifier for each sample which can be performed efficiently.

Intuitively, this approach assumes that the KL divergence should be highest when the classifier cannot distinguish between the samples from each distribution. The classifier must be trained individually for each estimated trajectory distribution in order to use this divergence. To improve computational efficiency in practice, the classifier can be initialized with the result from the previous iteration. Finally, we can define $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G}) = D^{\text{KL}}(\tilde{X}_T, \tilde{G})$ where the KL divergence is approximated following Eq. (4).

Smooth K-Nearest Neighbor. It is possible to define a differentiable two-sample test based on the well known k-NN algorithm as demonstrated in [4]. The Smooth K-Nearest Neighbor test possesses

important statistical properties such as consistency and convergence of its statistics to f-divergence. This is despite the complexity of having to solve a combinatorial optimization problem (nearest neighbor match) required by the k-NN method. Let n_1 be the number of samples of p, and n_2 be the number of samples of q. The k-NN divergence can be defined as

$$D_\alpha^{\text{NN}} = \int \frac{\alpha^2 p^2(x) + (1-\alpha)^2 q^2(x)}{\alpha p(x) + (1-\alpha) q(x)} dx, \tag{7}$$

for $n_1/(n_1+n_2) \to \alpha \in (0,1)$.

As proved in [5], the statistic

$$1 - \frac{T(X_1, X_2)}{(n_1+n_2)^k}, \tag{8}$$

where $T(X_1, X_2)$ refers to the number of edges connecting samples in a set $X_1 = \{x_i\}_{i=1}^{n_1}$ to a set $X_2 = \{x_i\}_{i=1}^{n_2}$ from a k-neighborhood graph created with points in X_1 and X_2, converges in probability to the D_α^{NN} divergence, and can be used as an efficient approximation. To make the computation differentiable, the authors of [6] define

$$T(X_1, X_2) = \sum_{i=1}^{n} \sum_{j=1}^{n} s_i(\{x_m\}_{m=1}^{n_1+n_2})_j, \tag{9}$$

where $s_i(\{x_m\}_{m=1}^{n_1+n_2})$ denotes the `softmax` function computed on the Euclidean distances between all points, except point i. As the `softmax` function is differentiable, the statistic in Eq. (8) becomes differentiable and can be used directly as our loss function $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G})$. To avoid specifying a particular value for α, our implementation computes the statistic for several values and averages them as the final result.

Energy Statistic. This two-sample test is based on Newton's gravitational potential energy which relates two entities by the Euclidean distance between them [7]. Given two distributions $p(x)$ and $q(y)$, the energy distance is defined as:

$$D^{\text{E}}(p, q) = 2\mathbb{E}[||x-y||^2] - \mathbb{E}[||x-x'||^2] - \mathbb{E}[||y-y'||^2], \tag{10}$$

where x and y are independent random variables. The corresponding two-sample statistic given two sets of samples $X = \{x_i\}_{i=1}^{n_1}$ and $Y = \{y_j\}_{j=1}^{n_2}$ can then be written as:

$$D^{\text{E}}(X, Y) = \frac{2}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{m=1}^{n_2} \|x_i - y_m\|^2 - \frac{1}{n_1^2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} \|x_i - x_j\|^2 - \frac{1}{n_2^2} \sum_{l=1}^{n_2} \sum_{m=1}^{n_2} \|y_l - y_m\|^2. \tag{11}$$

This provides a computationally efficient statistic which can be directly used as our loss $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G}) = D^{\text{E}}(\tilde{X}_T, \tilde{G})$.

A.1 Practical Considerations

The statistics considered in this work are good local approximations of distribution divergences. In the case of trajectory optimization, when the terminal states in early planning iterations are far from the goal set, the goal loss gradients can be uninformative. We therefore include a prior in our set planning method consisting of a smooth uniform distribution constructed by placing a bounding box around the goal samples. This can be included in our framework by multiplying a prior over the terminal state $p(x_T)$ with the goal likelihood in Eq. (6). This mitigates the poor divergence approximation in early iterations. Note that the uniform prior is insufficiently informative on its own, particularly in cases where the goal set is multi-modal. Furthermore, the prior needs to be differentiable, which is not the case for a standard uniform distribution. Therefore we define a smooth uniform prior in the region $R = x_T \colon a \leq x_T \leq b$ as

$$p(x_T) \propto \exp\left(-d(x_T, R)^2 / \sqrt{(2\sigma^2)}\right) \tag{12}$$

where $d(x, R) = \min |x - x'|,\ x' \in R$ is a distance function, and σ controls the *sharpness* of the approximation.

Once inference over the trajectory distributions converges, we must select a single trajectory estimate to execute. A common approach to accomplish this is by taking the mean, or weighted mean, of the particle set. This method is ineffective when the trajectory distribution is multi-modal. An alternative approach is to pick the maximum weighted particle. Our proposed set-based terminal losses yield a single score over the whole distribution, which does not enable weighing individual particles based on terminal loss. To select our final sample, we instead select the lowest cost trajectory. In practice, we also include the prior in the weight computation to avoid local minima with very low-cost trajectories.

B Experiment Details

B.1 Planar Navigation

The agent state x_t is composed of a 2D position and velocity, and the control signal u_t is a 2D acceleration. We use known, linear dynamics in a fully observed environment for the controller rollouts.

Losses. For each method, the running cost is summed over each timestep, where the cost for one timestep is:

$$c_t(x_t, u_t, z) = x_t^\top Q x_t + u_t^\top R u_t + \alpha\, c_{\text{SDF}}(x_t, z) \tag{13}$$

where Q and R are quadratic cost parameters for the state and action, and $c_{\text{SDF}}(x_t, z)$ is the obstacle avoidance term, computed using the Signed Distance Function (SDF) over the environment z. The goal loss $\mathcal{L}_{\text{goal}}(\tilde{X}_T, \tilde{G})$ is a set goal loss which is differentiable with respect to the trajectory τ.

Implementation Details. For each of our goal set planner ablations, $N = 50$ samples are randomly selected from goal distribution, except for *KL (Ratio Estimation)*, which uses $N = 100$ samples. This method involves training a learned classifier so is aided by a higher sample size. For the closest point methods, the goal sample with the smallest Euclidean distance from the start state is selected. For all the methods, $M = 50$ particles are used to represent the trajectory distribution. The particles represent the discrete control signals, u_t, which are 2D accelerations at each 0.1 second timestep over a horizon of 3 seconds. All planners are initialized with the distribution from the previous timestep, shifted to the current timestep, and run for 50 iterations. We use the Adam optimizer [8] to select the step size in the SVGD update rule.

The KL divergence uses a 3 layer fully-connected network as the classifier, retrainined at each timestep. To mitigate computational complexity, we warm start the training with the weights from the previous timestep. The Kernel MMD uses an RBF kernel, with a bandwidth selected by applying the median heuristic over the goal samples [8]. The Smooth k-NN loss uses a value of $k = 1$.

We use the RBF kernel for SVGD, and set the bandwidth using the median heuristic, a popular technique for choosing the kernel bandwidth which yields a good estimate in many cases [9]. Without access to data consisting of trajectory samples, we assume that they are normally distributed with covariance $\sigma = 1$. Under these assumptions, it follows that the expected distance between samples drawn from the distribution is $2D$, where D is the trajectory dimension.

B.2 Grasping

Figure 1 shows a case where the goal set planner finds a reachable point from the set, but the closest point method may select goal points which are in collision in some of the example scenes. This phenomenon helps explain the discrepancy in grasp success at the cost of increasing distance error.

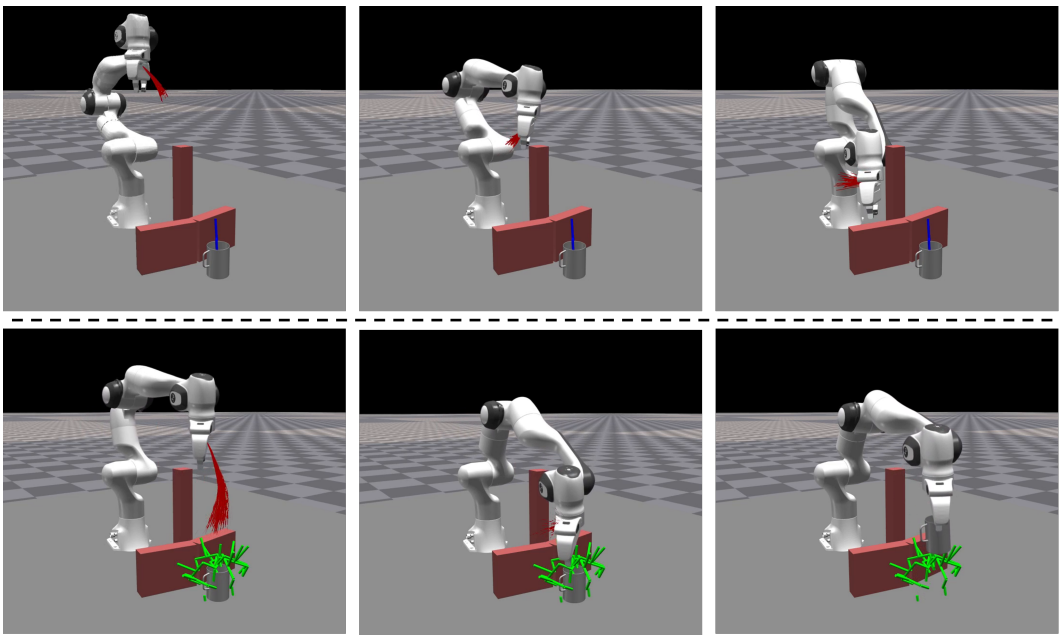

Figure 1: Example failure execution for the closest point planner (top) compared the the goal set planner (bottom). The closest point selected is shown in blue. The point is not reachable in the environment, causing the robot to fail to reach it (top right). The set planner considers all the grasp samples, shown in green. It finds a reachable grasp point and grasps successfully (bottom right).

References

[1] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[2] M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

[3] A. Menon and C. S. Ong. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pages 304–313. PMLR, 2016.

[4] M. F. Shilling. Multivariate two-sample tests based on nearest neighbours. *Journal of the American Statistical Association*, 81(395):799–806, 1986.

[5] N. Henze. A multivariate two-sample test based on the number of nearest neighbor type coincidences. *Annals of Statistics*, pages 772–783, 1988.

[6] J. Djolonga and A. Krause. Learning implicit generative models using differentiable graph tests. *arXiv preprint arXiv:1709.01006*, 2017.

[7] G. J. Székely and M. L. Rizzo. Energy statistics: A class of statistics based on distances. *Journal of statistical planning and inference*, 143(8):1249–1272, 2013.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[9] D. Garreau, W. Jitkrittum, and M. Kanagawa. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.