A Slices Perspective for Incremental Nonparametric Inference in High Dimensional State Spaces

Moshe Shienman[1], Ohad Levy-Or[1], Michael Kaess[2] and Vadim Indelman[3]

Abstract— We introduce an innovative method for incremental nonparametric probabilistic inference in high-dimensional state spaces. Our approach leverages *slices* from high-dimensional surfaces to efficiently approximate posterior distributions of any shape. Unlike many existing graph-based methods, our *slices* perspective eliminates the need for additional intermediate reconstructions, maintaining a more accurate representation of posterior distributions. Additionally, we propose a novel heuristic to balance between accuracy and efficiency, enabling real-time operation in nonparametric scenarios. In empirical evaluations on synthetic and real-world datasets, our *slices* approach consistently outperforms other state-of-the-art methods. It demonstrates superior accuracy and achieves a significant reduction in computational complexity, often by an order of magnitude.

I. INTRODUCTION

Modern mobile robots and autonomous systems are expected to reliably and efficiently perform various tasks including navigation, exploration, planning, manipulation and tracking. Furthermore, they must operate under different sources of uncertainty such as noisy measurements, imprecise actions, and dynamic environments in which some events are unpredictable. When uncertainty plays a significant role and probabilistic reasoning is essential for making safe and informed decisions, access to the full posterior distribution, for both the robot and the environment state, is crucial. As such, probabilistic inference algorithms, providing posterior distribution estimates, are commonly used to address these challenges. However, handling high-dimensional state spaces and operating in real-time is far from trivial.

Substantial research efforts have been made in recent decades to develop probabilistic inference algorithms that are robust, accurate and capable of real time performance. These efforts often rely on various assumptions to achieve their objectives. Perhaps the most frequently employed assumption is that the actual posterior distribution can be approximated using a parametric Gaussian model. However, in real-world problems, the posterior distribution is often non-Gaussian, having multiple modes or a nonparametric structure. Due to the complex, non-Gaussian nature of such

[1]Moshe Shienman and Ohad Levy-Or are with the Technion Autonomous Systems Program (TASP), Technion - Israel Institute of Technology, Haifa 32000, Israel, {smoshe,ohadlor}@campus.technion.ac.il

[2]Michael Kaess is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. kaess@cmu.edu

[3]Vadim Indelman is with the Department of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel. vadim.indelman@technion.ac.il
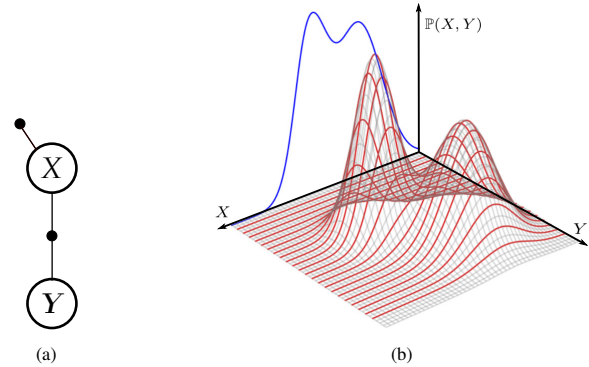
Fig. 1: (a) The factor graph of a small probabilistic inference problem containing two variable nodes X, Y and two factor nodes $f(X), f(X, Y)$ indicated as small solid black circles; (b) A high-dimensional surface representing the joint distribution $\mathbb{P}(X, Y)$. Each red *slice*, at a specific realization $Y = y$, represents a conditional distribution $\mathbb{P}(X|Y = y)$. The marginal distribution $\mathbb{P}(X)$, shown in blue, is calculated by integrating over all conditional *slices*.

posterior distributions, obtaining closed-form analytical solutions is challenging and frequently impractical.

To approximate and model such non-Gaussian posterior distributions, modern nonparametric methods use different variants of the *forward-backward* algorithm, exploiting graphical models such as factor graphs [19] and Bayes trees [17]. In these methods, samples are generated at each step to reconstruct intermediate distributions through techniques such as Kernel Density Estimators (KDE) and various learning procedures.

Our key observation is that these distributions can be directly reconstructed without the need for any additional learning techniques or KDE. This is accomplished by accessing *slices* from high-dimensional surfaces that represent partial joint distributions (Fig. 1). By circumventing these processes, our approach proves to be computationally more efficient, requires significantly less samples and produces more accurate results, as supported by our experimental findings.

Our main contributions in this paper are as follows:

1) We introduce a nonparametric inference approach which leverages *slices* from high-dimensional surfaces to approximate joint and marginal posterior distributions without any further intermediate reconstructions.
2) Unlike previous methods, ours does not require any iterative procedures nor breaking the underlying graphical model to generate samples in each step, even in cases where unary factors are not available.
3) We show how to utilize our *slices* perspective for nonparametric incremental inference and propose a

novel early stopping heuristic criteria to further speed up calculations.

4) Our approach requires less samples and consistently outperforms state-of-the-art nonparametric inference algorithms in terms of accuracy and computational complexity. This superiority is evident across evaluations conducted on both synthetic and real-world datasets, with improvements in time complexity reaching up to an order of magnitude.

Our proposed inference approach is general and can be effectively employed in various estimation problems such as tracking, sensor fusion, Bundle Adjustment (BA), Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM). In this particular study, we focus on demonstrating our approach within the framework of SLAM.

II. RELATED WORK

Research on inference algorithms, specifically used for SLAM, has predominantly focused on parametric solutions, often relying on approximating posterior distributions with parametric Gaussian models. One such notable work is iSAM2 [17], which introduced the Bayes tree graphical model. iSAM2 recovers posterior distributions using incremental upward and downward passes over the Bayes tree, in similar manner to the *forward-backward* algorithm. iSAM2 also incorporates an early stopping heuristic in the downward pass, aimed at enhancing computational efficiency.

Parametric approaches have more recently also been adapted to improve robustness, addressing challenges like unresolved data associations, multi-modal factors, and outliers [8], [9], [13], [14], [21], [25]. Nevertheless, these methods cannot handle non-Gaussian posterior distributions.

Nonparametric inference methods utilize sampling techniques and can theoretically be used to approximate any posterior distribution. Markov Chain Monte Carlo (MCMC) algorithms [6] and particle filters [4] were among the initial paradigms explored in the development of such methods. However, these approaches often become computationally impractical in high-dimensional settings, such as those encountered in SLAM. FastSLAM [20] introduced a method that combines the strengths of particle filters and parametric techniques to address high-dimensional settings effectively. Nevertheless, it has limitations. FastSLAM uses extended Kalman filters to track landmarks and cannot represent general non-Gaussian distributions. Additionally, similar to many particle filters, FastSLAM can encounter the issue of particle depletion, which can result in degenerate estimates.

Several recently proposed state-of-the-art methods leverage the conditional independence structure within factor graphs [19] to handle nonparametric inference in SLAM. Among them, mm-iSAM [3] and NF-iSAM [11] take advantage of the Bayes tree [16] to reduce computational complexity. In contrast, NSFG [10], which employs nested sampling [24], sacrifices computational efficiency for higher accuracy. We concentrate on these three approaches because they align conceptually with our own method, given that we also utilize the factor graph model.

mm-iSAM approximates non-Gaussian posterior distributions through the use of samples and KDEs. More specifically, in each step, the mm-iSAM approximation process employs an iterative nested Gibbs sampling approach [12] to generate samples from partial posteriors associated with each clique in the Bayes tree. These generated samples are then employed to construct new approximations of posterior distributions using KDEs. This distinguishes it conceptually from our approach, as we circumvent the need for an additional intermediate approximation with KDEs by directly accessing high-dimensional surfaces through sampled *slices*.

NF-iSAM exploits the expressive power of neural networks, and trains normalizing flows [23] to model conditionals that factorize non-Gaussian posterior distributions during the upward pass. Once all cliques have learned their conditional samplers, NF-iSAM conducts a downward pass, drawing samples from posterior distributions in a root-to-leaf manner. However, this learning approach is computationally intensive and requires many samples for convergence.

NSFG leverages nested sampling techniques to sample the posterior distribution directly across iterations until convergence. In this iterative process, the factor graph is initially divided into two components: an acyclic graph, referred to as the prior factor set, and a likelihood factor set. NSFG subsequently employs ancestral sampling to produce samples from the prior factor set, while likelihood evaluations are provided based on the likelihood factor set. Although NSFG delivers more precise estimates of non-Gaussian posterior distributions when compared to mm-iSAM and NF-iSAM, its iterative nature imposes significant computational demands. Consequently, it struggles to address high-dimensional, large-scale problems and cannot be employed in real-time applications.

Our proposed approach, which utilizes the *slices* perspective, demonstrates superior computational efficiency by eliminating the need for iterative procedures such as Gibbs sampling [3], training neural networks [11], or NSFG [10], which inherently involves iterations. Moreover, our approach exhibits accuracy on par with NSFG, which is considered as an offline algorithm.

III. BACKGROUND AND NOTATIONS

Consider a SLAM framework in which an autonomous agent is operating in a partially known environment. At each time step k the agent takes an action u_k and acquires an observation $Z_k \triangleq \{z_k^1, .., z_k^n\}$ based on n measurements. The motion and observation models are given by

$$x_{k+1} = f\left(x_k, u_k, w_k\right) \ , \ z_k = h\left(x_k, x^l, v_k\right), \quad (1)$$

where x^l denotes a landmark pose and w_k, v_k are noise terms, sampled from known motion and measurement distributions, respectively. In this work specifically, while the motion and measurement distributions are known, they can have any arbitrary shape, i.e. not necessarily Gaussian.

We denote the set of all state variables, including all agent poses and observed landmarks, by Θ. Given all actions

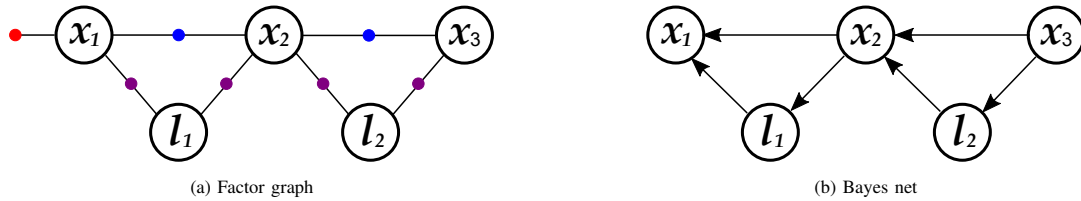(a) Factor graph   (b) Bayes net

Fig. 2: (a) A factor graph formulation of a SLAM problem with five variable nodes and seven factor nodes. Factor nodes represent probabilistic information over random variables. In this example, factor nodes include: a prior $f(x_1)$, odometry measurements $f(x_1, x_2), f(x_2, x_3)$ and landmark measurements $f(x_1, l_1), f(x_2, l_1), f(x_2, l_2), f(x_3, l_2)$. The factor graph represents a factorization of the joint distribution as a product of all factors; (b) The corresponding Bayes net after performing variable elimination on the factor graph using the elimination order $\mathcal{O} = \{\theta_1 = x_1, \theta_2 = l_1, \theta_3 = x_2, \theta_4 = l_2, \theta_5 = x_3\}$. The joint distribution is expressed as the product of conditionals produced in each step as $\mathbb{P}(x_1, x_2, x_3, l_1, l_2) = \mathbb{P}(x_1 | x_2, l_1) \cdot \mathbb{P}(l_1 | x_2) \cdot \mathbb{P}(x_2 | x_3, l_2) \cdot \mathbb{P}(l_2 | x_3) \cdot \mathbb{P}(x_3)$.

and measurements, the joint probability density function (the belief) is given by

$$b_k[\Theta] \triangleq \mathbb{P}(\Theta | u_{0:k-1}, Z_{0:k}) = \mathbb{P}(\Theta | D), \qquad (2)$$

where $D \triangleq \{b_0, u_{0:k-1}, Z_{0:k}\}$ represents all available data at time instant k. To reduce clutter, we omit time notations from hereon, and refer to the above joint density as $\mathbb{P}(\Theta | D)$. Our goal is to compute the joint posterior distribution (2) and marginal posterior distributions $\mathbb{P}(\theta | D)$, $\forall \theta \in \Theta$.

We use a factor graph model [19] to represent the joint posterior distribution (Fig. 2a). A factor graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ is a bipartite graph with two types of nodes: factor nodes $f \in \mathcal{F}$ and variable nodes $\theta \in \Theta$. The variable nodes represent the random variables in the estimation problem, whereas the factor nodes represent probabilistic information on those variables. The edges in $\mathcal{E}$ encode connectivity based on the variables associated with each factor, with each edge $e(f, \theta) \in \mathcal{E}$ linking a factor node to a variable node. This probabilistic graphical model represents a factorization of the posterior distribution in terms of process and measurement models

$$\mathbb{P}(\Theta | D) \propto \prod_i f_i(\Theta_i), \qquad (3)$$

where $\Theta_i = \{\theta \mid \exists e(f_i, \theta) \in \mathcal{E}\}$ is the set of all variables adjacent to factor f_i. In this work we assume for simplicity that only unary and pairwise factors exist in G, i.e. each $f \in \mathcal{F}$ can be connected with up to two variable nodes.

A common approach for computing the posterior distribution (2) and the posterior marginals $\mathbb{P}(\theta | D)$, $\forall \theta \in \Theta$ is the *forward-backward* algorithm which operates in two passes given a specific variable ordering $\mathcal{O} \triangleq \mathrm{ord}(\Theta)$.

During the *forward* pass, the factor graph is gradually transformed into a Bayes net [22] through a bipartite elimination game [7]. In each step, a single variable is eliminated from the factor graph following the elimination order $\mathcal{O}$, starting with a variable linked to a prior factor. Specifically, each step begins with a factor graph $G_{j-1} = (\mathcal{F}_{j-1}, \Theta_{j-1}, \mathcal{E}_{j-1})$, where $G_0 = G$ by convention. A variable node θ_j, representing the jth variable in $\mathcal{O}$, and all factors $\mathcal{F}_{j-1}(\theta_j) \triangleq \{f \mid \exists e(f, \theta_j) \in \mathcal{E}_{j-1}\}$ are first removed from G_{j-1} along with the corresponding edges, i.e. $\mathcal{E}_{j-1}(\theta_j) \triangleq \{e(f, \theta) \mid f \in \mathcal{F}_{j-1}(\theta_j), \theta \in \Theta_{j-1}\}$. All variables involved in $\mathcal{E}_{j-1}(\theta_j)$, except for θ_j, define a separator S_j representing the Markov blanket of variable

node θ_j in G_{j-1}. Next, a joint density is defined by the product of all removed factors as

$$\mathbb{P}_{joint}(\theta_j, S_j | D_j) = \eta_j^{-1} \prod_{f_i \in \mathcal{F}_{j-1}(\theta_j)} f_i(\Theta_i), \qquad (4)$$

where η_j^{-1} is a normalizing term and D_j represents all available data given by all factors removed up to and including the elimination of θ_j, i.e. all data in $\mathcal{F} \setminus \mathcal{F}_j$. Using the chain rule, the joint density (4) is then factorized as

$$\mathbb{P}_{joint}(\theta_j, S_j | D_j) = \mathbb{P}(\theta_j | S_j, D_j) f_{new}(S_j | D_j). \qquad (5)$$

The conditional $\mathbb{P}(\theta_j | S_j, D_j)$ is added as a new node to the Bayes net and the factor $f_{new}(S_j | D_j)$ is added into the factor graph G_j. Formally, each $G_j = (\mathcal{F}_j, \Theta_j, \mathcal{E}_j)$ is recursively defined by

$$\begin{aligned}
\mathcal{F}_j &= \mathcal{F}_{j-1} \setminus \mathcal{F}_{j-1}(\theta_j) \cup \{f_{new}(S_j | D_j)\}, \\
\Theta_j &= \Theta_{j-1} \setminus \{\theta_j\}, \\
\mathcal{E}_j &= \mathcal{E}_{j-1} \setminus \mathcal{E}_{j-1}(\theta_j) \cup \{e(f_{new}(S_j | D_j), \theta) \mid \theta \in S_j\}.
\end{aligned}$$

Note that in each intermediate step we have both an incomplete Bayes net and a reduced factor graph which defines a density on the remaining variables.

Once all variables were eliminated the *forward* pass is completed. The joint distribution (3) can then be expressed as the product of conditionals produced in each step as

$$\mathbb{P}(\Theta | D) = \prod_{j=1}^{|\Theta|} \mathbb{P}(\theta_j | S_j, D_j), \qquad (6)$$

which defines the Bayes net (Fig. 2b). The marginal distribution $\mathbb{P}(\theta_j | D)$, for each variable θ_j, is calculated using backsubstitution in a second, *backward* pass. Starting from the last eliminated variable and following the elimination order $\mathcal{O}$ in reverse, each marginal $\mathbb{P}(\theta_j | D)$ is calculated by integrating over S_j

$$\mathbb{P}(\theta_j | D) = \int_{S_j} \mathbb{P}(\theta_j, S_j | D) \, dS_j = \int_{S_j} \mathbb{P}(\theta_j | S_j, D) \cdot \mathbb{P}(S_j | D) \, dS_j.$$

Note that by definition, the separator for the last eliminated variable is an empty set, thereby yielding the direct marginal of the last eliminated variable. A key observation is that due to conditional independence $\mathbb{P}(\theta_j | S_j, D) = \mathbb{P}(\theta_j | S_j, D_j)$. As such, each marginal can be rewritten as

$$\mathbb{P}(\theta_j | D) = \int_{S_j} \mathbb{P}(\theta_j | S_j, D_j) \cdot \mathbb{P}(S_j | D) \, dS_j. \qquad (7)$$

In this work, we put forth a solution that extends to cases where no closed form solutions for calculating the marginal posterior distributions exist and the noise terms are non-Gaussian. In such cases, posterior distributions must be approximated.

IV. METHODOLOGY

Our key observation is that a joint probability function, for several random variables, can be seen as a *high-dimensional surface* from which conditional and marginal distributions can be calculated. For example, consider the joint distribution $\mathbb{P}(X, Y)$ over two random variables X and Y. Using the chain rule, we rewrite the joint distribution as

$$\mathbb{P}(X, Y) = \mathbb{P}(X|Y) \cdot \mathbb{P}(Y), \quad (8)$$

where the conditional $\mathbb{P}(X|Y)$ is given by a specific *slice* from the high-dimensional surface for each realization of Y. See illustration in Fig 1.

The marginal distribution $\mathbb{P}(X)$ is calculated using the Chapman-Kolmogorov transit integral

$$\mathbb{P}(X) = \int_Y \mathbb{P}(X|Y) \cdot \mathbb{P}(Y) \, dY. \quad (9)$$

As there is no closed form solution for (9) in the general case, it can be approximated using N samples of Y from $\mathbb{P}(Y)$. The corresponding estimated marginal is given by

$$\hat{\mathbb{P}}(X) = \hat{\mathbb{E}}_{y \sim \mathbb{P}(Y)}[\mathbb{P}(X|Y=y)] = \frac{1}{N} \sum_{i=1}^{N} \mathbb{P}(X|Y=y^i). \quad (10)$$

Put differently, evaluating (10) can be seen as accessing the high-dimensional surface at specific sample points of Y in order to extract a *mixture* of conditional *slices* over X. Moreover, when $N \to \infty$ this approximation is guaranteed to converge to the true analytical solution.

A. Inference Using Slices

For nonparametric inference, our proposed approach leverages the *forward-backward* algorithm presented in Sec. III. For each eliminated variable θ_j in the *forward* pass, the joint density defined in (4) is factorized into a conditional and a marginal via (5) which is rewritten once again for convenience

$$\mathbb{P}(\theta_j, S_j|D_j) = \mathbb{P}(\theta_j|S_j, D_j) f_{new}(S_j|D_j). \quad (11)$$

This factorization serves as the pivotal step through which information propagates across the algorithm. In strike contrast to existing approaches that depend on intermediate density reconstructions, our primary contribution is the direct use of *slices* to approximate both $f_{new}(S_j|D_j)$ and the conditional $\mathbb{P}(\theta_j|S_j, D_j)$. We rigorously describe how our *slices* perspective is systematically employed throughout the algorithm.

1) Forward Pass: We start by explicitly writing the term $f_{new}(S_j|D_j) = \int_{\theta_j} \mathbb{P}_{joint}(\theta_j, S_j|D_j) \, d\theta_j$ in (11) as

$$f_{new}(S_j|D_j) = \eta_j^{-1} \int_{\theta_j} \prod_{f_i \in \mathcal{F}_{j-1}(\theta_j)} f_i(\Theta_i) \, d\theta_j, \quad (12)$$

where η_j^{-1} is a normalizing term. As there is no closed form solution for (12), it must be approximated. While other approaches use additional intermediate approximations, such as KDE in the case of [3] or learning techniques as seen in [11], we employ *slices* as a more direct and effective means of approximation. We note that we assume the ability to access such high-dimensional *slices*, an assumption shared with mm-iSAM, NF-iSAM, and NSFG. By generating N samples of θ_j from some factor $f' \in \mathcal{F}_{j-1}(\theta_j)$, a standard Monte Carlo estimator of (12) is given by

$$\hat{f}_{new}(S_j|D_j) = \frac{\eta_j^{-1}}{N} \sum_{n=1}^{N} \prod_{f_i \in \mathcal{F}_{j-1}(\theta_j) \setminus \{f'\}} f_i(\theta_j^n, \Theta_i \setminus \{\theta_j\}), \quad (13)$$

where $\Theta_i \setminus \{\theta_j\} \subseteq S_j$ and each $f_i \in \mathcal{F}_{j-1}(\theta_j) \setminus \{f'\}$ is evaluated using the samples of θ_j. In a *slices* perspective, for each sample θ_j^n, a high-dimensional conditional *slice* over $\Theta_i \setminus \{\theta_j\}$ is retrieved from every non-unary factor $f_i \in \mathcal{F}_{j-1}(\theta_j)$. We acknowledge a slight misuse of notation, as D_j is no longer purely theoretical due to the fact that the conditioned data is approximated at each step by $\hat{f}_{new}$ (see **Example** below).
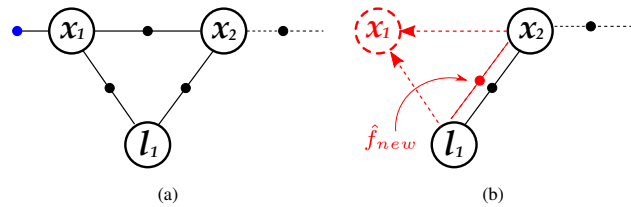

Fig. 3: Generating samples during variable elimination; (a) A subset of three variables x_1, x_2, l_1 from a larger factor graph. The given elimination order is $\mathcal{O} = \{\theta_1 = x_1, \theta_2 = l_1, ...\}$. When eliminating x_1, samples of x_1 are directly obtained from $f(x_1)$ to approximate the new factor $\hat{f}_{new}(l_1, x_2|D_1)$ with *slices* via (13); (b) When eliminating l_1, samples of l_1 are obtained according to Lemma 1 from $\hat{f}_{new}(l_1, x_2|D_1)$. Even without a unary factor on l_1, our *slices* approach provides a method to directly generate samples of l_1.

Obtaining samples of θ_j is essential for approximating (12). However, generating these samples becomes non-trivial when no unary factor $f \in \mathcal{F}_{j-1}(\theta_j)$ is available for direct sampling. To handle this issue, mm-iSAM employs multi-scale Gibbs sampling, which is computationally expensive. In contrast, both NF-iSAM and NSFG break the factor graph to chain like structures and generate samples using ancestral sampling, which does not leverage loop closure factors. Through direct approximation of (12) with *slices* in each step, our approach eliminates the need for iterative procedures or factor graph decomposition into chain-like structures for generating samples of θ_n. Unlike previous methods, this holds true even in the absence of unary factors.

Lemma 1: Given a factor graph $G = (\mathcal{F}, \Theta, \mathcal{E})$ and an elimination order $\mathcal{O}$, if each eliminated variable $\theta_j \in \Theta$ either has a unary factor connected to it, i.e. $\exists f(\theta_j) \in$

$\mathcal{F}_{j-1}(\theta_j)$ or, $\exists \theta_i \in \Theta$ such that θ_i was previously eliminated and $f(\theta_i, \theta_j) \in \mathcal{F}$, then samples of θ_j can be drawn from one of the factors $\mathcal{F}_{j-1}(\theta_j)$ (see proof in appendix).

Example: *We demonstrate how Lemma 1 is utilized to draw samples when there are no unary factors using the example in Fig. 3. After eliminating the first variable $\theta_1 = x_1$, a new factor $\hat{f}_{new}(l_1, x_2 | D_1)$ is added to the factor graph. Consequently, the second variable to be eliminated, $\theta_2 = l_1$, lacks any unary factors at this stage. According to* (4)*, the joint density $\mathbb{P}_{joint}(l_1, x_2 | D_2)$ is given by*

$$\mathbb{P}_{joint}(l_1, x_2 | D_2) = \hat{f}_{new}(l_1, x_2 | D_1) \cdot f(l_1, x_2), \quad (14)$$

and is factorized, according to (5)*, as*

$$\mathbb{P}_{joint}(l_1, x_2 | D_2) = \mathbb{P}(l_1 | x_2, D_2) \cdot f_{new}(x_2 | D_2), \quad (15)$$

given the separator $S_2 = \{x_2\}$. As previously stated, it is important to note that D_2 does not encompass all raw data up to this point. Instead of $D_2 = \{b_0, u_1, z_1, z_2\}$, where $b_0 \equiv f(x_1)$ is given by the prior, we have $D_2 = \{\hat{f}_{new}(l_1, x_2 | D_1), z_2\}$ due to the approximation from the previous elimination step. We write $f_{new}(x_2 | D_2)$ explicitly, following (12) *and considering the estimator $\hat{f}_{new}(l_1, x_2 | D_1)$ obtained via* (13)*. Denoting it intermediately by $\tilde{f}_{new}(x_2 | D_2)$ and changing the order of the integral and summation, yields*

$$\tilde{f}_{new}(x_2 | D_2) = \frac{\eta^{-1}}{N} \sum_{n=1}^{N} f(x_1^n, x_2) \int_{l_1} f(x_1^n, l_1) \cdot f(l_1, x_2) \ dl_1. \quad (16)$$

We can now exploit the unary structure of $f(x_1^n, l_1)$, for each x_1^n, to generate samples of l_1 and approximate (16) *as*

$$\hat{f}_{new}(x_2 | D_2) = \frac{\eta^{-1}}{N^2} \sum_{n_1=1}^{N} f(x_1^{n_1}, x_2) \sum_{n_2=1}^{N} f(l_1^{n_1, n_2}, x_2).$$

Unlike other methods, as previously discussed, our slices approach exclusively utilizes structure for sample generation when there are no unary factors.

We next turn our attention to the conditional term $\mathbb{P}(\theta_j | S_j, D_j)$ in (11). By utilizing $\hat{f}_{new}$, given by (13), and replacing (4), an estimator of the conditional is obtained

$$\hat{\mathbb{P}}(\theta_j | S_j, D_j) = \frac{\prod_{f_i \in \mathcal{F}_{j-1}(\theta_j)} f_i(\Theta_i)}{\frac{1}{N} \sum_{n=1}^{N} \prod_{f_i \in \mathcal{F}_{j-1}(\theta_j) \setminus \{f'\}} f_i(\theta_j^n, \Theta_i \setminus \{\theta_j\})}. \quad (17)$$

We note that the normalization term η_j^{-1} cancels out. Moreover, there is no need to explicitly calculate η_j^{-1} in (13) as intermediate new factors created during the *forward* pass are only used to generate samples of S_j. During the *backward* pass, *slices* from the high-dimensional surface representing the numerator, are used to approximate the conditional (17) as described next.

2) Backward Pass: Once the *forward* pass concludes, we utilize a similar *slices* perspective, leveraging the constructed Bayes net, to retrieve the joint and marginal posterior distributions. Following the elimination order $\mathcal{O}$ in reverse, we obtain an estimator of the marginal (7) for each variable

θ_j. This is achieved by utilizing the estimator of the conditional $\mathbb{P}(\theta_j | S_j, D_j)$, which was already evaluated during the *forward* pass (17), and samples of S_j directly obtained from the marginal $\hat{\mathbb{P}}(S_j | D)$ which is given by a *mixture* of *slices*. Specifically, given N samples of $S_j \sim \hat{\mathbb{P}}(S_j | D)$, the estimator is given by

$$\hat{\mathbb{P}}(\theta_j | D) = \frac{1}{N} \sum_{n=1}^{N} \hat{\mathbb{P}}(\theta_j | S_j^n, D_j), \quad (18)$$

which is by itself also a *mixture* of high-dimensional *slices*. For each sample S_j^n, a high-dimensional conditional *slice* over θ_j is retrieved from every non-unary factor $f_i \in \mathcal{F}_{j-1}(\theta_j)$. The weight of each *slice* is given by evaluating the denominator of (17) at sample point S_j^n. We note that (18) represents the marginal as a distribution. As such, we can both sample from the marginal and evaluate the likelihood for a given sample.

Similarly, the joint posterior distribution (2) can also be approximated. By replacing each conditional in (6) with (17), we can explicitly represent the joint distribution from which we can also both sample and evaluate the likelihood for a given sample. Sampling from the joint distribution is done by following the elimination order $\mathcal{O}$ in reverse. We first generate a sample of the last eliminated variable directly from the corresponding marginal. Next, we recursively sample each variable from the corresponding conditional (17), given a sample of the separator, to obtain a single sample from the joint distribution.

Up to this point, our approach has focused on a batch processing method applied to a given factor graph. However, in real-world applications, a real-time online solution is needed. This requires continuous updates and providing estimates whenever new measurements are added.

B. Incremental Inference with Early Stopping Heuristic

Incremental inference was initially developed for the Gaussian case [1], [18], [17]. The underlining observation in these approaches is that the addition of new measurements affects only specific parts the original factor graphs. As a result, there's no need to redo the entire *forward-backward* inference process. Instead, an incremental *forward* pass is conducted, focusing computations solely on the affected variables. This process, termed re-elimination, significantly reduces computational costs by reusing previously computed information and selectively updating relevant sections of the factor graph (See Algorithm 1 in [15] for a simplified version). Furthermore, performing full backsubstitution in every iteration can be computationally expensive, particularly in large-scale problems with complex factor graphs. Consequently, iSAM2 [17] incorporated an early stopping heuristic in the downward pass to significantly reduce computational cost in the non-linear Gaussian case.

By utilizing the Bayes tree [17], both mm-iSAM [3] and NF-iSAM [11] achieve nonparametric incremental inference capabilities. Nevertheless, these algorithms restrict the application of incremental aspects solely to the upward pass,

while the entire downward pass is executed anew. Moreover, training new conditional samplers for affected variables in NF-iSAM, during re-elimination, must be performed from scratch, resulting in computational inefficiency.

To integrate incremental aspects into both the *forward* and *backward* passes, we suggest the adoption of re-elimination and an early stopping heuristic. This combination, which serves to notably diminish computational complexity, is a novel concept introduced for the first time in a nonparametric setting to the best of our knowledge.

We employ the Maximum Mean Discrepancy (MMD) [5] metric as a heuristic for early stopping of the *backward* pass. The MMD is a metric used to assess the dissimilarity between two probability distributions, relying on samples. During the *backward* pass of each incremental step, we calculate and cache the most recent marginal distributions. Given a new marginal and a previously found marginal for a given variable, we use the MMD metric to evaluate the distance between the two distributions. If this distance is below a user defined threshold, we stop the *backward* pass. Notably, we have two hyper-parameters available for fine-tuning, providing a balance between accuracy and efficiency: the number of samples used for evaluating the MMD distance and the threshold criteria.

V. Experiments

We evaluate our proposed *slices* approach using both synthetic and real-world datasets. To assess its performance, we compare our results with those obtained using mm-iSAM [3], NF-iSAM [11], and NSFG [10], employing the open-source code provided by the respective authors. Following [10] and [11], we use the root mean square error (RMSE) metric to measure the discrepancies between ground truth and samples from posterior distributions. For robustness, all reported results for each method were averaged over ten independent runs.

A. Synthetic Dataset - Multi Modal Four Doors

The synthetic four doors localization example [3] is a one-dimensional SLAM problem. The robot is aware of a map with four identical doors but initially observes only one of them, without knowing which specific door it is. Consequently, the prior distribution in this scenario is multi-modal, with four modes of equal weight. As the robot progresses through several steps, it gathers loop closure measurements related to a landmark within its environment. Additionally, it acquires two more measurements, each corresponding to different doors. These latter measurements play a critical role in resolving the ambiguity, leading to the collapse of the posterior distribution into a single mode.

This one dimensional problem is interesting as the marginal posterior distributions for all robot poses and the landmark can be analytically calculated (as discussed in [3]). Our *slices* approach directly approximates these marginal distributions via (18). In mm-ISAM these marginal distributions are approximated with KDEs while in NS-iSAM and NSFG they are only available through samples (Fig. 4). Note
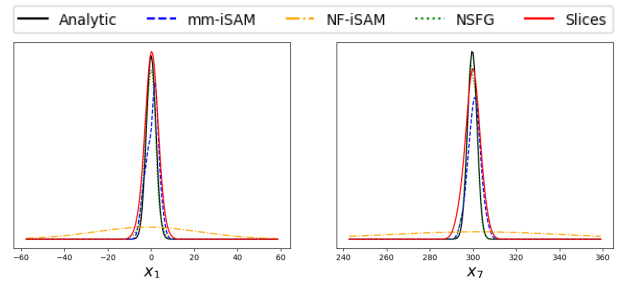

Fig. 4: Four doors synthetic SLAM dataset [3] marginal posterior distributions for the first and last robot poses. A black solid line indicates the analytic solution when considering only the mode which corresponds to the ground truth. Our *slices* approach directly approximates these marginal via (17). In mm-ISAM, KDEs are used to approximate these marginal distributions, whereas in NF-iSAM and NSFG, they are derived solely from samples and approximated using KDEs based on these samples, as demonstrated in [10]. In each method, 200 samples were utilized to approximate the distributions.
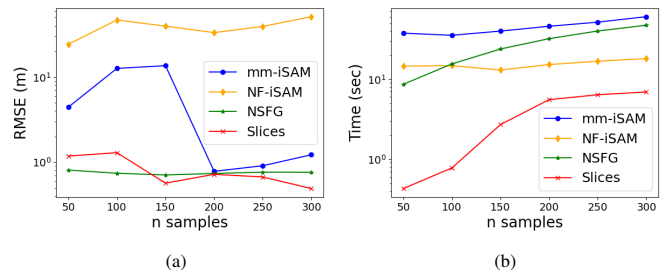

Fig. 5: Four doors synthetic SLAM dataset [3] results. We report RMSE (m) and run time (sec) as functions of the number of samples used by all methods.

how NF-ISAM fails to recover the correct mode, even in this small scenario, due to its difficulty in converging when the number of samples is relatively small.

In this experiment, we assess run time and RMSE as functions of the number of samples generated by the algorithms in each step, employing a batch computation approach across the entire factor graph. Our *slices* approach demonstrates superior accuracy compared to both mm-iSAM and NF-iSAM, achieving results on par with NSFG (Fig. 5), despite the latter being recognized as an offline algorithm. Furthermore, our approach shows significantly improved computational efficiency when using the same number of samples.

B. Real World Dataset - Plaza

The Plaza2 dataset [2] comprises a sequence of odometry and range measurements to four unknown landmarks gathered by a vehicle navigating a planar environment. This range measurement SLAM problem is challenging because it involves both nonlinear measurements and non-Gaussian likelihood models.

In this experiment, we perform incremental inference and evaluate both the run time and RMSE at each step. We adopt the same noise models across all methods, following the specifications outlined in [11], for a fair compression. We also set our hyperparameters, for the early stopping heuristic, to $N_M = 100$ and $\delta = 1e^{-4}$, representing the number

of samples used for evaluating the MMD distance and the threshold criteria, respectively. The chosen re-elimination order for affected variables, in each incremental step, is such that landmark variables are eliminated last.

We configured our *slices* approach to generate 150 samples in each step. For NF-iSAM, we followed the recommended setting of 2000 samples and used the same hyperparameters as published by the authors in [11]. As for mm-iSAM, we employed the default value of 100 samples, as advised in the open source implementation of [3]. Notably, when we attempted to use 200 samples, the mm-iSAM run time became prohibitively long for this specific problem. The scale and dimensionality of this problem were too demanding for NSFG, which is also confirmed in [10], where the authors only presented results for a limited number of time steps. Notably, our *slices* approach achieves superior accuracy compared to NF-iSAM and mm-iSAM (Fig. 7). Furthermore, our method significantly enhances computational efficiency, reducing complexity by an order of magnitude compared to NF-iSAM and mm-iSAM. An illustration depicting the marginal distribution across different time steps for the second landmark is shown in Fig. 6.
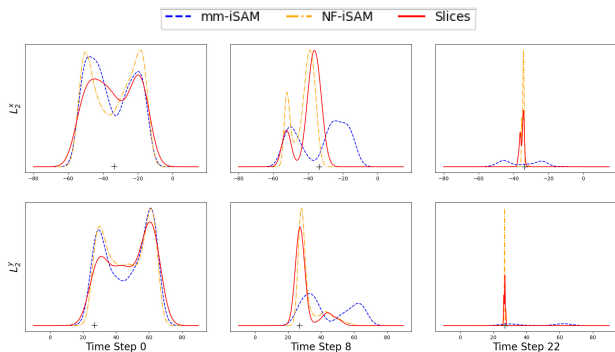

Fig. 6: Plaza2 dataset [2] marginal posterior distributions for landmark number 2 across different time steps, presented separately for the x and y axes. The robot captures the first, fifth and ninth range measurement to L_2 at time steps 0, 8 and 22 respectively. The ground truth position of L_2 is denoted by a black $+$ sign. The hyperparameters used for each method are described in Section V.B .
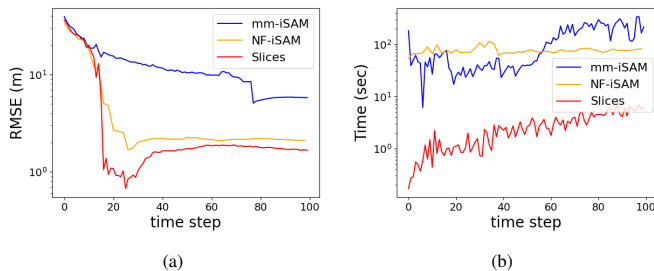

Fig. 7: Plaza2 dataset [2] with range measurements. We report RMSE (m) and run time (sec) for each incremental step.

VI. CONCLUSIONS

In this work, we introduced a novel approach that utilizes *slices* from high-dimensional surfaces to efficiently approximate nonparametric posterior distributions. Unlike existing methods that use generated samples to reconstruct intermediate distributions through learning procedures and KDEs, our *slices* perspective avoids such additional intermediate approximations. Moreover, our approach does not use any iterative procedures as in [3], [10], [11].

We have also introduced a novel early stopping heuristic, a first in nonparametric scenarios, during the *backward* pass to significantly reduce computational complexity and enable real-time operation. In future research, we plan to investigate the impacts of different thresholds for the early stopping heuristic.

Our *slices* perspective not only demonstrated superior accuracy compared to other online nonparametric inference methods [3], [11], but also matched the performance of an offline, state-of-the-art, nonparametric inference method [10]. Additionally, our approach achieved a remarkable reduction in computational complexity, as evidenced in both synthetic and real-world datasets.

VII. APPENDIX: PROOF OF LEMMA 1

We prove Lemma 1 by induction.

base case: Let θ_1 be the first eliminated variable. Due to the properties of the elimination order given in Sec. III, $\exists f(\theta_1) \in \mathcal{F}$ from which samples of θ_1 can be generated.

induction step: Eliminating θ_j, if $\exists f(\theta_j) \in \mathcal{F}$ then according to the elimination algorithm it holds that $f(\theta_j) \in \mathcal{F}_{j-1}$ from which samples of θ_j can be generated. Else, according to the given elimination order $\exists \theta_i \in \Theta$ such that θ_i was previously eliminated and $f(\theta_i, \theta_j) \in \mathcal{F}$. Since $f(\theta_i, \theta_j) \in \mathcal{F}$, according to the elimination algorithm it holds that $f(\theta_i, \theta_j) \in \mathcal{F}_{i-1}$ and thus $\theta_j \in S_i$. As such, according to (12), when θ_i was eliminated, a new factor

$$f_{new}(S_i|D_i) = \eta^{-1} \int_{\theta_i} \prod_{f_k \in \mathcal{F}_{i-1}(\theta_i)} f_k(\Theta_k) \, d\theta_i, \quad (19)$$

was added. Using the induction assumption, samples of θ_i were generated from one of the factors in $\mathcal{F}_{i-1}(\theta_i)$ to approximate this integral. Without loss of generality, we denote the factor from which samples of θ_i were generated as $f_{\bar{k}}$ and write the approximation to (19) as

$$\hat{f}_{new}(S_i|D_i) = \frac{\eta^{-1}}{N} \sum_{n=1}^{N} f(\theta_i^n, \theta_j) \prod_{f_k \in \mathcal{F}_{i-1}(\theta_i) \setminus \{f(\theta_i^n, \theta_j), f_{\bar{k}}\}} f_k\left(\theta_i^n, \Theta_k^{-i}\right). \quad (20)$$

According to the elimination algorithm, for any variable $\theta_m \in S_i$ eliminated after θ_i and before θ_j, it must hold that the new factor $f_{new}(S_m|D_m)$ must contain $f(\theta_i^n, \theta_j)$. Thus, $\mathcal{F}_{j-1}$ also contains $f(\theta_i^n, \theta_j)$ from which samples of θ_j can be generated. ∎

REFERENCES

[1] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.

[2] Joseph Djugash, Bradley Hamner, and Stephan Roth. Navigating with ranging radios: Five data sets with ground truth. *Journal of Field Robotics*, 26(9):689–695, 2009.

[3] D. Fourie, J. Leonard, and M. Kaess. A nonparametric belief solution to the bayes tree. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[4] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Procedings F*, 140(2):107–113, 1993.

[5] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. of Machine Learning Research*, 13(1):723–773, 2012.

[6] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[7] P. Heggernes and P. Matstoms. Finding good column orderings for sparse QR factorization. In *Second SIAM Conference on Sparse Matrices*, 1996.

[8] M. Hsiao and M. Kaess. Mh-isam2: Multi-hypothesis isam using bayes tree and hypo-tree. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2019.

[9] Guoquan Huang, Michael Kaess, John J Leonard, and Stergios I Roumeliotis. Analytically-selected multi-hypothesis incremental map estimation. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6481–6485, 2013.

[10] Qiangqiang Huang, Alan Papalia, and John J Leonard. Nested sampling for non-gaussian inference in slam factor graphs. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):9232–9239, 2022.

[11] Qiangqiang Huang, Can Pu, Kasra Khosoussi, David M Rosen, Dehann Fourie, Jonathan P How, and John J Leonard. Incremental non-gaussian inference for slam using normalizing flows. *IEEE Trans. Robotics*, 39(2):1458–1475, 2023.

[12] Alexander Ihler, Erik Sudderth, William Freeman, and Alan Willsky. Efficient multiscale sampling from products of gaussian mixtures. *Advances in Neural Information Processing Systems (NIPS)*, 16, 2003.

[13] V. Indelman, E. Nelson, J. Dong, N. Michael, and F. Dellaert. Incremental distributed inference from arbitrary poses and unknown data association: Using collaborating robots to establish a common reference. *IEEE Control Systems Magazine (CSM), Special Issue on Distributed Control and Estimation for Robotic Vehicle Networks*, 36(2):41–74, 2016.

[14] V. Indelman, E. Nelson, N. Michael, and F. Dellaert. Multi-robot pose graph localization and data association from unknown initial relative poses via expectation maximization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014.

[15] V. Indelman, R. Roberts, and F. Dellaert. Incremental light bundle adjustment for structure from motion and robotics. *Robotics and Autonomous Systems*, 70:63–82, 2015.

[16] M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Intl. Workshop on the Algorithmic Foundations of Robotics*, Dec 2010.

[17] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31(2):217–236, Feb 2012.

[18] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[19] F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2):498–519, February 2001.

[20] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. 19th AAAI National Conference on AI*, Edmonton, Alberta, Canada, 2002.

[21] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *Intl. J. of Robotics Research*, 32(7):826–840, 2013.

[22] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[23] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Intl. Conf. on Machine Learning (ICML)*, pages 1530–1538. PMLR, 2015.

[24] John Skilling. Nested sampling for general bayesian computation. *Bayesian analysis 1(4)*, page 833–859, 2006.

[25] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.