Linear Complexity Gibbs Sampling for Generalized Labeled Multi-Bernoulli Filtering

Changbeom Shim, Ba-Tuong Vo, *Member, IEEE*, Ba-Ngu Vo, *Fellow, IEEE*, Jonah Ong, and Diluka Moratuwage

***Abstract*—Generalized Labeled Multi-Bernoulli (GLMB) densities arise in a host of multi-object system applications analogous to Gaussians in single-object filtering. However, computing the GLMB filtering density requires solving NP-hard problems. To alleviate this computational bottleneck, we develop a linear complexity Gibbs sampling framework for GLMB density computation. Specifically, we propose a tempered Gibbs sampler that exploits the structure of the GLMB filtering density to achieve an $\mathcal{O}(T(P+M))$ complexity, where T is the number of iterations of the algorithm, P and M are the number hypothesized objects and measurements. This innovation enables the GLMB filter implementation to be reduced from an $\mathcal{O}(TP^2M)$ complexity to $\mathcal{O}(T(P+M+\log T)+PM)$. Moreover, the proposed framework provides the flexibility for trade-offs between tracking performance and computational load. Convergence of the proposed Gibbs sampler is established, and numerical studies are presented to validate the proposed GLMB filter implementation.**

***Index Terms*—Random finite sets, multi-object tracking, generalized labeled multi-Bernoulli, tempered Gibbs sampling.**

I. Introduction

The aim of multi-object tracking (MOT) is to estimate the number of objects and their trajectories from noisy sensor data. The challenges are the unknown and time-varying number of objects, accompanied by false alarms, misdetections, and data association uncertainty, culminating in computational bottlenecks for most real world applications. Notwithstanding this, numerous solutions have been developed, with multiple hypothesis tracking [1], joint probabilistic data association [2], and random finite set (RFS) [3] being the most widely used approaches. The RFS approach, in particular, is gaining substantial interest due to its versatile multi-object state space models [4], and efficient solutions [5]. Various tractable RFS multi-object filters have been devised ranging from the probability hypothesis density localization filters [4], [6] to the generalized labeled multi-Bernoulli (GLMB) tracking filter [7].

The GLMB filter [7] is an analytic solution to the RFS multi-object filter that provides tracks and their *provisional identities/labels* to meet MOT requirements [1]. The unique feature of this formulation is the provision for principled trajectory estimation (even with only single-scan filtering), capable of tracking millions of objects, online [5]. Moreover, the GLMB family of densities is furnished with elegant mathematical properties, offering a versatile set of tools including, conjugacy [7], closure under truncation with analytic truncation error [8], analytic approximation of general labeled multi-object density with minimal Kullback-Leibler divergence [9], analytic Cauchy-Schwarz divergence and void probabilities [10]. These properties enabled MOT with multiple sensors and scans [8], [11], [12], non-standard models [9], [13], [14], [15], unknown system parameters [16], multi-object control solutions [10], [17], as well as distributed implementations [18], [19], [20]. The GLMB filter is also amenable to parallelization that reduces computation times. For instance, in [5], spatial search was used to decompose the filtering density into independent GLMBs that are processed in parallel, while, in [21], a parallel centralized implementation for multiple sensors was developed. The GLMB filter has also found a host of applications from robotics [22], sensor networks [23], [24], cell biology [14], [25] to audio/video processing [15], [26].

The main computational bottleneck in GLMB filtering is the truncation of the multi-object filtering density [7]. Truncation by discarding terms with small weights minimizes the L_1-error [8], and can be posed as a *ranked assignment problem*, solvable by Murty's algorithm [27] with cubic complexity in both the number of measurements and hypothesized objects [28], [29]. A more efficient solution based on Gibbs sampling (GS) was proposed in [30], which incurs an $\mathcal{O}(TP^2M)$ complexity, where T, the number of iterates of the algorithm, dominates the number of hypothesized objects and measurements P and M. GS is an efficient Markov Chain Monte Carlo (MCMC) technique for sampling from complex probability distributions, popularized by the seminal work of Geman and Geman [31], which opened up applications in many disciplines ranging from statistics, engineering to computer science, and is still an active research topic.

Following the strategy of selecting significant GLMB components by random sampling [30], a number of GLMB truncation techniques have been developed. In [32], an approximate GLMB filter with linear complexity in the number of hypothesized objects was proposed by neglecting the standard data association requirement of at most one measurement per object. While this technique can be modified to accommodate the data

Manuscript received 18 November 2022; revised 12 April 2023; accepted 11 May 2023. Date of publication 17 May 2023; date of current version 9 June 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Lorenzo Galleani. This work was supported by the Australian Research Council under Linkage Project LP200301507 and Future Fellowship FT210100506. *(Corresponding author: Changbeom Shim.)*

The authors are with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Bentley, WA 6102, Australia (e-mail: changbeom.shim@curtin.edu.au; ba-tuong.vo@curtin.edu.au; ba-ngu.vo@curtin.edu.au; jonahosx25@gmail.com; diluka.moratuwage@curtin.edu.au).

Digital Object Identifier 10.1109/TSP.2023.3277220

association requirement, the complexity reverts to quadratic in number of hypothesized objects [32]. In [33], a herded Gibbs sampling implementation of the labeled multi-Bernoulli (LMB) filter [34]–a one-term approximation of the GLMB filter–was developed. However, this implementation is slower than the systematic-scan Gibbs sampling (SGS) GLMB implementation [30], not to mention that, compared to the (exact) GLMB filter, the LMB filter is more prone to track fragmentation and track switching. In [35] and [36], the cross-entropy method [37] was applied, respectively, to multi-sensor GLMB filtering and its distributed version, but with higher complexity than the SGS implementation in [11].

To alleviate the computational bottleneck in GLMB filtering, this article proposes a tempered Gibbs sampling (TGS) framework for selecting significant GLMB components, with linear complexity, i.e., $\mathcal{O}(T(P+M))$. Similar to the widely known random-scan Gibbs sampling (RGS), TGS randomly selects a coordinate to update, but provides the mechanism to improve mixing and sample diversity [38]. However, generic TGS incurs an $\mathcal{O}(TP^2M)$ complexity. To this end, we develop an innovative decomposition of the conditionals that enables TGS to be reduced to $\mathcal{O}(T(P+M))$ complexity, with negligible additional memory (compared to generic $\mathcal{O}(TP^2M)$ TGS). The samples generated by the proposed TGS algorithm converge to a tempered distribution (not necessarily the same as the original stationary distribution). Furthermore, keeping in mind that TGS is regarded as the combination of importance sampling and MCMC, the importance-weighted samples indeed converge to the original stationary distribution [38]. The drastic reduction in computational complexity facilitates the development of multi-object control solutions, which require fast computation of the multi-object filtering density for online operations [39], [40], [41], [42]. Moreover, our solution is not only restricted to GLMB truncation, but amenable to a wide range of applications of the ranked assignment problem [43].

The proposed TGS framework enables RGS to be implemented with $\mathcal{O}(T(P+M))$ complexity, whereas generic RGS incurs $\mathcal{O}(TPM)$. Further, we propose deterministic-scan Gibbs sampling (DGS), an efficient algorithm using deterministic coordinate selection, as opposed to RGS's completely random coordinate selection. DGS's better mixing and sample diversity compared to RGS are validated by numerical studies. In addition, using DGS, we show how an exact implementation of SGS can be accomplished with $\mathcal{O}(TPM)$ complexity. We also present numerical studies in MOT to discuss the trade-offs between tracking performance and computational load in the proposed TGS framework. Due to the computation of the cost matrix, the resultant GLMB filter implementation incurs an additional $\mathcal{O}(PM)$ complexity. In summary, our main contribution is a TGS framework for GLMB truncation that:

- Incurs a linear complexity of $\mathcal{O}(T(P+M))$, with negligible additional memory, by exploiting the structure of the GLMB filtering density; and
- Admits as special cases, RSG, DGS, and SGS implementations, with $\mathcal{O}(T(P+M))$, $\mathcal{O}(TM)$, and $\mathcal{O}(TPM)$ complexities, respectively.

TABLE I
SUMMARY OF FREQUENT NOTATIONS

Notation	Description
$1_S(\cdot)$	Indicator function for a given set S
$\|X\|$	Cardinality of a set X
$\langle f, g\rangle$	Inner product, $\int f(x)g(x)dx$ of two functions f and g
α	Mixture parameter of proposal distribution
β	Tempering parameter of proposal distribution
$\delta_X[Y]$	Kronecker-δ, $\delta_X[Y]=1$ if $X=Y$, 0 otherwise
$\Delta(\boldsymbol{X})$	Distinct label indicator $\delta_{\|\boldsymbol{X}\|}[\|\mathcal{L}(\boldsymbol{X})\|]$
$\mathcal{F}(S)$	Class of all finite subsets of a given set S
$f(x_{m:n})$	$f(x_m), ..., f(x_n), f: x_i \mapsto f(x_i)$
$f_B(x,\ell)$	Birth probability density of x given label ℓ
Γ	Space of all (extended) association map γ
γ	(Extended) association map
$\gamma\circ\mathcal{L}(\cdot)$	Function composition $\gamma(\mathcal{L}(\cdot))$
$g(z\|\boldsymbol{x})$	Single-object likelihood
$g(Z\|\boldsymbol{X})$	Multi-object likelihood
h^X	Multi-object exponential, $\prod_{x\in X} h(x)$, with $h^{\emptyset}=1$
$\mathcal{L}(\boldsymbol{x})$	Label of a (labeled) state $\boldsymbol{x}$, $\mathcal{L}((x,\ell))=\ell$
$\mathcal{L}(\boldsymbol{X})$	Labels of a (labeled) set $\boldsymbol{X}$, $\{\mathcal{L}(\boldsymbol{x}):\boldsymbol{x}\in\boldsymbol{X}\}$
$\mathcal{L}(\gamma)$	Live labels of γ, $\{\ell\in\mathbb{L}:\gamma(\ell)\geq 1\}$
M	Number of measurements (at the next time)
P	Number of hypothesized objects
$P_B(\ell)$	Birth probability of label ℓ
$P_D(\boldsymbol{x})$	Detection probability of labeled state $\boldsymbol{x}$
$P_S(\boldsymbol{x})$	Survival probability of labeled state $\boldsymbol{x}$
$\boldsymbol{\pi}$	Multi-object filtering density
π	Stationary distribution
ϕ	Proposal distribution
ρ	Coordinate probability distribution
T	Number of iterates
$x_{m:n}$	$x_m, x_{m+1}, ..., x_n$
ξ	History of association maps
Ξ	Some finite discrete space

We validate the proposed approach via a series of comprehensive numerical experiments, with considerations for computational efficiency, tracking accuracy, and sample diversity.

The rest of this article is organized as follows. Section II provides the necessary background on GLMB filtering and SGS-based GLMB truncation. In Section III, we present linear complexity GS for GLMB filtering density truncation. Numerical studies are presented in Section IV, and concluding remarks are given in Section V.

II. Preliminaries

This section presents the necessary background for the development of the main result of this article. We first outline the basics of generalized labeled multi-Bernoulli (GLMB) filtering in Section II-A, and then summarize the Gibbs sampling (GS) approach to GLMB truncation in Section II-B. Throughout this article, single-object states and multi-object states are respectively represented by lower case letters (e.g., x) and upper case letters (e.g., X). Further, boldfaced symbols denote labeled states or distributions (e.g., $\boldsymbol{x}$, $\boldsymbol{X}$, $\boldsymbol{\pi}$). Frequently used notations are summarized in Table I.

A. GLMB Filtering

1) Multi-object State: In GLMB filtering, the system state to be estimated at each time is the set $\boldsymbol{X}$ of labeled states of the underlying objects, called the *multi-object state* [7]. Each labeled single-object state $\boldsymbol{x} \in \boldsymbol{X}$ is an ordered pair (x, ℓ) in the product space $\mathbb{X} \times \mathbb{L}$, where $\mathbb{X}$ is a (finite dimensional) *state space*, and $\mathbb{L}$ is a discrete space called the *label space*. Let $\mathbb{B}_\tau$ denote the label space of objects born at time τ, then the space $\mathbb{L}$ of all labels up to time k is given by the disjoint union $\uplus_{\tau=0}^{k} \mathbb{B}_\tau$. The state x of an object varies with time, while its label or identity ℓ (usually consists of the object's time of birth and an index to distinguish those born at the same time) is time-invariant. The *trajectory* of an object is a sequence of consecutive labeled states with a common label.

The cardinality $|\boldsymbol{X}|$ (i.e., number of elements) of the multi-object state $\boldsymbol{X}$ varies with time due to the appearance and disappearance of objects. In addition, a multi-object state $\boldsymbol{X}$ at any time must have distinct labels. More concisely, let $\mathcal{L}(\boldsymbol{x})$ denote the label of $\boldsymbol{x}$, and for any finite $\boldsymbol{X} \subset \mathbb{X} \times \mathbb{L}$, define the label set $\mathcal{L}(\boldsymbol{X}) \triangleq \{\mathcal{L}(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{X}\}$ and *distinct label indicator* $\Delta(\boldsymbol{X}) \triangleq \delta_{|\boldsymbol{X}|}[|\mathcal{L}(\boldsymbol{X})|]$. Then for $\boldsymbol{X}$ to be valid multi-object state, we require $\Delta(\boldsymbol{X}) = 1$.

In Bayesian estimation, the single-object state and the measurement are modeled as random variables. Hence, for multi-object estimation, the multi-object state and measurement are modeled as Random Finite Sets (RFSs). In the following we describe the so-called standard multi-object models for the dynamics and observations in a multi-object system.

2) Multi-object Dynamic: For simplicity, we omit the time subscript "k", and use the subscript "+" for the next time $k+1$ when there is no ambiguity. Each element $\boldsymbol{x} = (x, \ell)$ of the current multi-object state $\boldsymbol{X}$ either survives with probability $P_S(\boldsymbol{x})$ and evolves to state $\boldsymbol{x}_+ = (x_+, \ell_+)$ at the next time according to the transition density $f_{S,+}(x_+|x, \ell)\delta_\ell[\ell_+]$, or dies with probability $1 - P_S(\boldsymbol{x})$ [7]. The term $\delta_\ell[\ell_+]$ in the transition density ensures the object retains the same label. In addition to the surviving objects, new objects can be born. New born objects are usually modeled by an LMB RFS, where an object with state $\boldsymbol{x}_+ = (x_+, \ell_+)$ is born at the next time with probability $P_{B,+}(\ell_+)$, and the state density $f_{B,+}(x_+, \ell_+)$. The multi-object state $\boldsymbol{X}_+$ at the next time is the union of surviving objects and new born objects, described by the *multi-object Markov transition density* $\boldsymbol{f}_+(\boldsymbol{X}_+|\boldsymbol{X})$ [7]. It is assumed that, conditional on the current multi-object state, objects survive and move independently of each other, and that new born objects and surviving objects are independent [3].

3) Multi-object Observation: Each element $\boldsymbol{x} \in \boldsymbol{X}$ is either detected with probability $P_D(\boldsymbol{x})$ and generates an observation z at the sensor with likelihood $g(z|\boldsymbol{x})$, or misdetected with probability $1 - P_D(\boldsymbol{x})$. In addition to the detections, the sensor also receives clutter, modeled by a Poisson RFS with intensity function κ. The multi-object observation Z is the union of detections and clutter. It is assumed that conditional on the multi-object state, objects are detected independently from each other and that clutter and detections are independent.

The association of objects with sensor measurements (at time k) is described by a *positive 1-1* mapping $\gamma : \mathbb{L} \to \{-1 : |Z|\}$, i.e., a mapping where *no two distinct arguments are mapped to the same positive value* [7]. The (extended) association map[1] γ specifies that object (with label) ℓ generates measurement $z_{\gamma(\ell)} \in Z$, with $\gamma(\ell) = 0$ if it is undetected, and $\gamma(\ell) = -1$ if it does not exist. The positive 1-1 property ensures each measurement comes from at most one object. Let $\mathcal{L}(\gamma) \triangleq \{\ell \in \mathbb{L} : \gamma(\ell) \geq 0\}$ denote the set of *live labels*[2] of γ, and Γ denote the space of all association maps, then the *multi-object likelihood function* can be written as [11]

$$g(Z|\boldsymbol{X}) \propto \sum_{\gamma \in \Gamma} \delta_{\mathcal{L}(\gamma)}[\mathcal{L}(\boldsymbol{X})][\psi_Z^{(\gamma \circ \mathcal{L}(\cdot))}(\cdot)]^{\boldsymbol{X}}, \qquad (1)$$

where $\gamma \circ \mathcal{L}(\cdot) = \gamma(\mathcal{L}(\cdot))$ and

$$\psi_{\{z_{1:M}\}}^{(j)}(\boldsymbol{x}) = \begin{cases} \frac{P_D(\boldsymbol{x}) g(z_j|\boldsymbol{x})}{\kappa(z_j)}, & j > 0 \\ 1 - P_D(\boldsymbol{x}), & j = 0 \end{cases}. \qquad (2)$$

4) The GLMB Filter: All statistical information about the underlying state is contained in the *filtering density*–probability density of the current state conditioned on all measurements up to (and including) the current time. Given the multi-object filtering density $\boldsymbol{\pi}$ at the current time, the propagation to the next time step is given by [3]

$$\boldsymbol{\pi}_+(\boldsymbol{X}_+|Z_+) \propto g\left(Z_+|\boldsymbol{X}_+\right) \int \boldsymbol{f}_+\left(\boldsymbol{X}_+|\boldsymbol{X}\right) \boldsymbol{\pi}(\boldsymbol{X})\, \delta \boldsymbol{X}. \qquad (3)$$

Under the standard multi-object system model, the multi-object filtering density takes on the GLMB form [7]:

$$\boldsymbol{\pi}\left(\boldsymbol{X}\right) = \Delta\left(\boldsymbol{X}\right) \sum_{\xi \in \Xi} \left[p^{(\xi)}\right]^{\boldsymbol{X}} \sum_{I \in \mathcal{F}(\mathbb{L})} w^{(\xi, I)} \delta_I[\mathcal{L}\left(\boldsymbol{X}\right)], \qquad (4)$$

where Ξ is some finite discrete space, $p^{(\xi)}(\cdot, \ell)$ is a probability density on $\mathbb{X}$, and $w^{(\xi,I)}$ is a non-negative weight satisfying the condition $\sum_{(\xi,I) \in \Xi \times \mathcal{F}(\mathbb{L})} w^{(\xi,I)} = 1$. To obtain a multi-object state estimate from the GLMB density (4), we first find the most probable cardinality n^* from the cardinality distribution

$$\Pr(|\boldsymbol{X}| = n) \triangleq \sum_{(\xi,I) \in \Xi \times \mathcal{F}(\mathbb{L})} \delta_n[|I|] w^{(\xi,I)}, \qquad (5)$$

and then the highest-weighted component (ξ^*, I^*) with cardinality $|I^*| = n^*$, see [7]. The state estimate for each object $\ell \in I^*$ can be taken as the mean (or mode) of $p^{(\xi^*)}(\cdot, \ell)$. Alternatively, the entire trajectory of object $\ell \in I^*$ can be estimated as described in [8], [17].

For compactness, we write a GLMB density in terms of its parameters:

$$\boldsymbol{\pi} \triangleq \left\{ \left(p^{(\xi)}, w^{(\xi,I)} \right) \right\}_{(\xi,I) \in \Xi \times \mathcal{F}(\mathbb{L})}. \qquad (6)$$

[1] Originally called extended association maps, herein referred to as association maps for brevity.

[2] Note the distinction from $\mathcal{L}(\boldsymbol{X})$, the labels of a labeled set.

Given a current GLMB filtering density of the form (6), its propagation to the next time is the GLMB [8], [30]

$$\boldsymbol{\pi}_+ = \left\{\left(p_{Z_+}^{(\xi,\gamma_+)}, w_{Z_+}^{(\xi,\gamma_+,I_+)}\right)\right\}_{(\xi,\gamma_+,I_+)\in\Xi\times\Gamma_+\times\mathcal{F}(\mathbb{L}_+)}, \tag{7}$$

where the new parameters are given by

$$p_{Z_+}^{(\xi,\gamma_+)}(\cdot,\ell) = \frac{\bar{p}_+^{(\xi,\gamma_+(\ell))}(\cdot,\ell)\,\psi_{Z_+}^{(\gamma_+(\ell))}(\cdot,\ell)}{\bar{\psi}_{Z_+}^{(\xi,\gamma_+)}(\ell)}, \tag{8}$$

$$w_{Z_+}^{(\xi,\gamma_+,I_+)} \propto \delta_{\mathcal{L}(\gamma_+)}[I_+]\sum_{I\in\mathcal{F}(\mathbb{L})} w^{(\xi,I)} 1_{\mathcal{F}(I\uplus\mathbb{B}_+)}(I_+)\, w_{Z_+}^{(\xi,I,\gamma_+)}, \tag{9}$$

$$\bar{p}_+^{(\xi,j)}(\cdot,\ell) = \begin{cases} \frac{\int f_+(\cdot|x,\ell)P_S(x,\ell)p^{(\xi)}(x,\ell)dx}{\bar{P}_S^{(\xi)}(\ell)}, & \ell\in I, j\geq 0 \\ f_{B,+}(\cdot,\ell), & \ell\in\mathbb{B}_+, j\geq 0 \end{cases}, \tag{10}$$

$$\bar{P}_S^{(\xi)}(\ell) = \left\langle P_S(\cdot,\ell), p^{(\xi)}(\cdot,\ell)\right\rangle, \tag{11}$$

$$\bar{\psi}_{Z_+}^{(\xi,\gamma_+)}(\ell) = \left\langle \bar{p}_+^{(\xi,\gamma_+(\ell))}(\cdot,\ell), \psi_{Z_+}^{(\gamma_+(\ell))}(\cdot,\ell)\right\rangle, \tag{12}$$

$$w_{Z_+}^{(\xi,I,\gamma_+)} = 1_{\Gamma_+}(\gamma_+)\prod_{\ell\in I\uplus\mathbb{B}_+}\eta_{Z_+,\ell}^{(\xi,I)}(\gamma_+(\ell)), \tag{13}$$

$$\eta_{Z_+,\ell}^{(\xi,I)}(j) = \begin{cases} 1-\bar{P}_S^{(\xi)}(\ell), & \ell\in I, j<0 \\ \bar{P}_S^{(\xi)}(\ell)\bar{\psi}_{Z_+}^{(\xi,j)}(\ell), & \ell\in I, j\geq 0 \\ 1-P_{B,+}(\ell), & \ell\in\mathbb{B}_+, j<0 \\ P_{B,+}(\ell)\bar{\psi}_{Z_+}^{(\xi,j)}(\ell), & \ell\in\mathbb{B}_+, j\geq 0 \end{cases}. \tag{14}$$

Note that each component (ξ, I) of the GLMB filtering density at time k generates a (very large) set $\{(\xi, I, \gamma_+, I_+) : I_+ \in \mathcal{F}(\mathbb{L}_+), \gamma_+ \in \Gamma_+\}$ of children components to the next time. Due to the terms $\delta_{\mathcal{L}(\gamma_+)}[I_+]$ and $1_{\mathcal{F}(I\uplus\mathbb{B}_+)}(I_+)$ in (9), we only need to consider components with $I_+ \subseteq I \uplus \mathbb{B}_+$ and $\mathcal{L}(\gamma_+) = I_+$. While this is a big reduction, in general the total number of GLMB components with non-zero weights still grows super-exponentially with time. Implementing the GLMB filter requires truncating the GLMB filtering density. Truncation by keeping the most highly weighted components minimizes the L_1 truncation error [8].

B. Gibbs Sampling for GLMB Truncation

Truncating the children of the GLMB component (indexed by (ξ, I)) amounts to selecting the γ_+'s with significant $w_{Z_+}^{(\xi,I,\gamma_+)}$. For a given component (ξ, I), let us enumerate $I = \{\ell_{1:R}\}, \mathbb{B}_+ = \{\ell_{R+1:P}\}$, and $Z_+ = \{z_{1:M}\}$, and abbreviate

$$\eta_i(j) \triangleq \eta_{Z_+,\ell_i}^{(\xi,I)}(j), \tag{15}$$

where $i \in \{1{:}P\}$, and $j \in \{\text{-}1{:}M\}$. Let π be a (discrete) probability distribution on $\{\text{-}1{:}M\}^P$ defined by

$$\pi(\gamma_+) \propto 1_{\Gamma_+}(\gamma_+)\prod_{i=1}^{P}\eta_i(\gamma_+(\ell_i)). \tag{16}$$

Note that due to the factor $1_{\Gamma_+}(\gamma_+)$, any sample from (16) is a valid association map. Further, it follows from (13) that the probability of sampling γ_+ is $\pi(\gamma_+) \propto w_{Z_+}^{(\xi,I,\gamma_+)}$. Hence, truncating the contribution from the parent component (ξ, I) can be accomplished by sampling from π.

GS is a computationally efficient Markov Chain Monte Carlo (MCMC) technique for sampling from complex probability distributions whose conditionals can be computed/sampled at low cost. In GLMB truncation, we aim to maximize the number of distinct significant samples, rather than focusing on the actual distribution of the samples as per MCMC inference. All distinct samples can be used regardless of their distribution, because each distinct sample constitutes a term in the approximant (the larger the weights, the smaller the approximation error) [30]. Hence, it is not necessary to discard burn-ins and wait for samples from the stationary distribution.

Systematic-scan GS (SGS) is the classical approach that samples from the stationary distribution π by constructing a Markov chain with transition kernel [31], [44]

$$\pi(\gamma_+'|\gamma_+) = \prod_{i=1}^{P}\pi_i(\gamma_+'(\ell_i)|\gamma_+'(\ell_{1:i\text{-}1}),\gamma_+(\ell_{i+1:P})),$$

where the i-th *conditional*, defined on $\{\text{-}1{:}M\}$, is given by

$$\pi_i(\gamma_+'(\ell_i)|\gamma_+'(\ell_{1:i\text{-}1}),\gamma_+(\ell_{i+1:P})) \propto \pi(\gamma_+'(\ell_{1:i}),\gamma_+(\ell_{i+1:P})).$$

This means, for a given γ_+, the next state γ_+' of the chain is generated one component after another, by sampling $\gamma_+'(\ell_i)$ from $\pi_i(\cdot|\gamma_+'(\ell_{1:i\text{-}1}),\gamma_+(\ell_{i+1:P})), i = 1, 2, \ldots, P$.

For GLMB truncation, the conditionals are the categorical distributions given by [30, Proposition 3], which is restated in a slightly different form as follows.

Proposition 1: For each $i \in \{1{:}P\}$, let $\ell_{\bar{i}}$ denote $\ell_{1:i\text{-}1,i+1:P}$. Then the i-th conditional, defined on $\{\text{-}1{:}M\}$, is given by

$$\pi_i(\cdot|\gamma_+(\ell_{\bar{i}})) = \frac{\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))}{\langle\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}})), 1\rangle}, \tag{17}$$

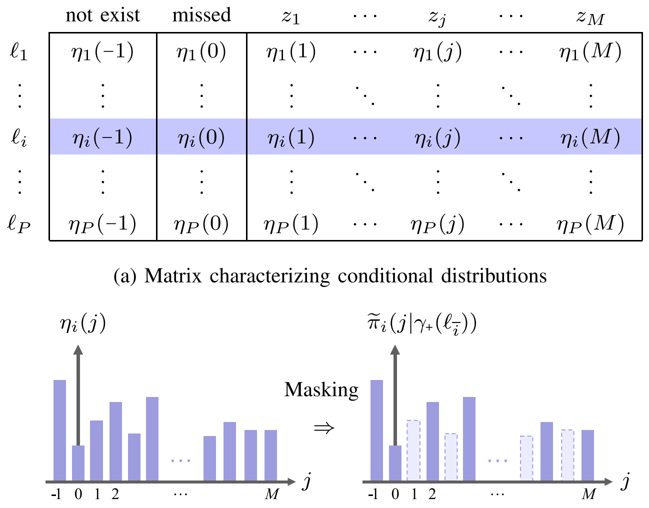
where

$$\widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}})) \triangleq \begin{cases} \eta_i(j), & j<1 \\ \eta_i(j)(1-1_{\{\gamma_+(\ell_{\bar{i}})\}}(j)), & j\in\{1{:}M\} \end{cases}. \tag{18}$$

Remark 1: The above result shows that the conditionals are completely characterized by the $P \times (M + 2)$ cost matrix in Fig. 1(a) (which can be pre-computed from the measurement Z_+) and the values of γ_+ on $\ell_{\bar{i}}$, i.e., $\{\gamma_+(\ell_{\bar{i}})\}$. Specifically, the unnormalized i-th conditional is simply given by the i-th row after entries with (positive) indices contained in $\{\gamma_+(\ell_{\bar{i}})\}$ have been zeroed (or masked) out, as illustrated in Fig. 1(b). Since evaluating $1_{\{\gamma_+(\ell_{\bar{i}})\}}(j)$ (and hence the mask $1 - 1_{\{\gamma_+(\ell_{\bar{i}})\}}(j)$) for each j incurs an $\mathcal{O}(P)$ complexity, computing $\pi_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ requires an $\mathcal{O}(PM)$ complexity.

Remark 2: If γ_+ is positive 1-1 on $\{\ell_{\bar{i}}\}$ and $\gamma_+(\ell_i)$ is set to any $j \sim \pi_i(\cdot|\gamma_+(\ell_{\bar{i}}))$, then γ_+ is also positive 1-1, i.e., a valid association map. Multiplication by the mask $1 - 1_{\{\gamma_+(\ell_{\bar{i}})\}}(j)$ ensures that any j violating the positive 1-1 condition has zero probability of being sampled.

Selecting significant GLMB components can be performed with $\mathcal{O}(TP^2M)$ complexity via SGS as shown in [30]. Noting that P (the number of hypothesized objects) is strongly



$$\begin{array}{c|c|c|ccccc|}
 & \text{not exist} & \text{missed} & z_1 & \cdots & z_j & \cdots & z_M \\
\hline
\ell_1 & \eta_1(-1) & \eta_1(0) & \eta_1(1) & \cdots & \eta_1(j) & \cdots & \eta_1(M) \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\ell_i & \eta_i(-1) & \eta_i(0) & \eta_i(1) & \cdots & \eta_i(j) & \cdots & \eta_i(M) \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
\ell_P & \eta_P(-1) & \eta_P(0) & \eta_P(1) & \cdots & \eta_P(j) & \cdots & \eta_P(M) \\
\hline
\end{array}$$

(a) Matrix characterizing conditional distributions

(b) Determining the unnormalized i-th conditional from the matrix

Fig. 1. Computing the unnormalized i-th conditional. Multiplying the i-th row of the matrix in (a) with a masking function (that depends on γ_+) and normalizing the resulting function in (b) yields the i-th conditional.

correlated with M (the number of measurements), this complexity translates roughly to a cubic complexity, i.e., $\mathcal{O}(TP^3)$ or $\mathcal{O}(TM^3)$. Nonetheless, SGS has been extended to address multi-dimensional ranked assignment problems in multi-scan and multi-sensor GLMB filtering [8], [11], [12].

III. Linear Complexity GS for GLMB Filtering

This section presents efficient linear complexity GS for selecting significant components in GLMB filtering. We begin with the widely-known random-scan GS (RGS) in Section III-A. Section III-B then presents tempered GS (TGS), a recent generalization that can overcome the drawbacks of RGS, but incurs an $\mathcal{O}(TP^2M)$ complexity. In Section III-C, we develop a decomposition of the conditionals (of the stationary distribution) allowing TGS to be implemented with a linear complexity of $\mathcal{O}(T(P+M))$. Salient special cases of the proposed linear complexity TGS, including deterministic-scan GS (DGS), are discussed in Section III-D. For completeness, the linear complexity TGS-based GLMB filter implementation is discussed in Section III-E.

A. RGS for GLMB Truncation

Whereas SGS generates the next iterate γ'_+ by traversing and updating all P coordinates of the current iterate γ_+, RGS only selects one coordinate at random to update [45], [46]. This means the transition kernel $\pi(\gamma'_+|\gamma_+)$ is given by $\frac{1}{P}\pi_i(\gamma'_+(\ell_i)|\gamma_+(\ell_{\bar{i}}))$ when γ_+ and γ'_+ differ at most in the i-th coordinate and 0 otherwise. A generic RGS implementation would incur an $\mathcal{O}(TPM)$ complexity because computing the conditionals requires $\mathcal{O}(PM)$. Further, RGS is inefficient in the sense that it generates less distinct significant samples than SGS, for the same number of iterates of the chain, leading to poorer GLMB approximations and tracking performance.

The two main factors affecting the efficiency of RGS are *mixing time* and *sample diversity*. Intuitively, mixing time is the number of iterations required for subsequent states to be treated as samples from the stationary distribution π. Sample diversity refers to the proportion of distinct samples in a given number iterates of the chain. While the actual distribution of the samples is not relevant for GLMB truncation, fast mixing is necessary for efficient generation of distinct significant samples. Furthermore, even if the chain converges to the stationary distribution, sample diversity can be poor due to frequent revisiting of previous states from successive iterations. RGS's notorious slow-mixing [38], [45], [47], together with observed poor sample diversity means that it could take many iterations for significant GLMB components to be generated.

B. Tempered Gibbs Sampling

Similar to RGS, TGS also generates the next iterate γ'_+ by randomly selecting a coordinate to update. However, TGS provides an additional mechanism to improve mixing and sample diversity [38]. Specifically, a coordinate $i \in \{1{:}P\}$ is chosen according to the distribution

$$\rho(i|\gamma_+) \propto \frac{\phi_i(\gamma_+(\ell_i)|\gamma_+(\ell_{\bar{i}}))}{\pi_i(\gamma_+(\ell_i)|\gamma_+(\ell_{\bar{i}}))}, \tag{19}$$

where $\phi_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ is a bounded proposal, defined on $\{-1{:}M\}$, with the same support as $\pi_i(\cdot|\gamma(\ell_{\bar{i}}))$. Further, given the selection of the i-th coordinate, its state is updated by sampling from the proposal, i.e.,

$$\gamma'_+(\ell_i) \sim \phi_i(\cdot|\gamma_+(\ell_{\bar{i}})), \tag{20}$$

(note that TGS reduces to RGS in the special case $\phi_i = \pi_i$). The proposal ϕ_i controls sample diversity, and determines coordinate selection that can influence mixing [46], [48], [49]. The transition kernel of TGS is given by $\phi(\gamma'_+|\gamma_+) \propto \rho(i)\phi_i(\gamma'_+(\ell_i)|\gamma_+(\ell_{\bar{i}}))$ when γ_+ and γ'_+ differ at most in the i-th coordinate and 0 otherwise.

In GLMB truncation, we are not interested in the importance weights since the goal is to generate distinct samples with significant GLMB weights (which are different from the importance weights). While TGS can circumvent the drawbacks of RGS by using fast mixing proposals that yield diverse samples, these may not be significant (in GLMB weights) because the tempered stationary distribution could be very different from π. One way to generate diverse and significant samples is to use proposals that approximate the conditional π_i, but are more diffuse, which can be achieved with a mixture consisting of the conditional and its tempered version, i.e.,

$$\phi_i(j|\gamma_+(\ell_{\bar{i}})) = \alpha\pi_i(j|\gamma_+(\ell_{\bar{i}})) + \frac{(1-\alpha)\pi_i^{\beta}(j|\gamma_+(\ell_{\bar{i}}))}{\left\langle \pi_i^{\beta}(\cdot|\gamma_+(\ell_{\bar{i}})), 1\right\rangle}, \tag{21}$$

where $\alpha, \beta \in (0, 1]$, and for any function $f, f^{\beta}(\cdot) \triangleq [f(\cdot)]^{\beta}$. This popular proposal preserves the modes of the conditional $\pi_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ to capture significant samples, and at the same time, increases sample diversity via the more diffuse tempered term [38], [50]. Moreover, the state informed coordinate selection strategy of TGS provides faster mixing [38], [51], [52].

```
Algorithm 1: Tempered Gibbs Sampling [38].
Input: $\gamma_+$, $\beta$, $[\eta_i(j)]_{i=1}^P$
Output: $\gamma'_+$
1 Compute $\rho(\cdot|\gamma_+)$;
2 Sample $n$ from $\rho(\cdot|\gamma_+)$;
3 Sample $\gamma'_+(\ell_n)$ from $\phi_n(\cdot|\gamma_+(\ell_{\bar{n}}))$;
```

While the TGS kernel (Algorithm 1) avoids traversing all coordinates, it still incurs an $\mathcal{O}(P^2M)$ complexity (and hence generating T iterates incurs $\mathcal{O}(TP^2M)$ complexity) because:

- Computing the conditionals $\pi_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ for all $i \in \{1{:}P\}$ incurs $\mathcal{O}(P^2M)$ complexity since each conditional requires performing the positive 1-1 checks using the set inclusion $1_{\{\gamma_+(\ell_{\bar{i}})\}}(j)$, which incurs an $\mathcal{O}(PM)$ complexity;
- Computing the proposals $\phi_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ for all $i \in \{1{:}P\}$ incurs $\mathcal{O}(PM)$ complexity due to β-th power operations and normalizations; and
- Computing the coordinate selection probabilities $\rho(i|\gamma_+)$ for all $i \in \{1{:}P\}$ and sampling the coordinate incurs $\mathcal{O}(P)$ complexity, while sampling from $\phi_n(\cdot|\gamma_+(\ell_{\bar{n}}))$ incurs $\mathcal{O}(M)$ complexity.

Nonetheless, it is possible to further exploit the particular structure of the problem through the positive 1-1 constraint to implement this kernel with $\mathcal{O}(P+M)$ complexity.

C. Linear Complexity TGS

To reduce the complexity of computing and sampling from the coordinate distribution $\rho(\cdot|\gamma_+)$, note that the proposal (21) can be rewritten in terms of the unnormalized conditional as

$$\phi_i(j|\gamma_+(\ell_{\bar{i}})) = \frac{\alpha\widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}}))}{\nu_i^{(1)}(\gamma_+)} + \frac{(1-\alpha)\widetilde{\pi}_i^{\beta}(j|\gamma_+(\ell_{\bar{i}}))}{\nu_i^{(\beta)}(\gamma_+)}, \quad (22)$$

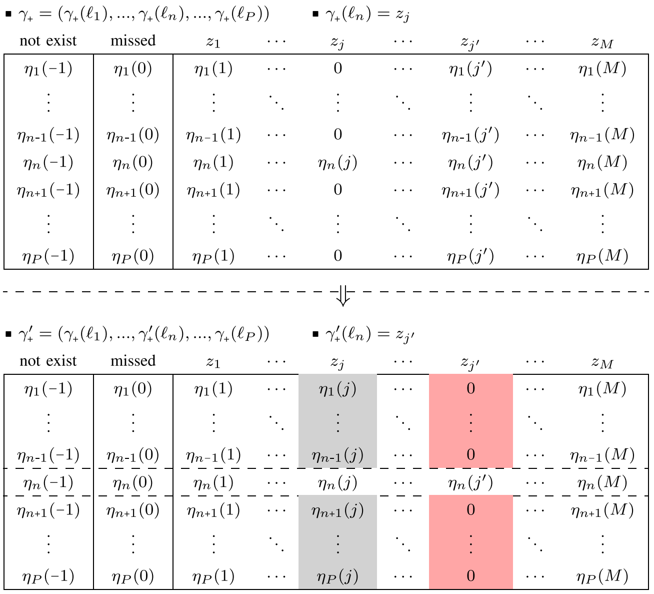
where $\nu_i^{(\beta)}(\gamma_+) = \left\langle \widetilde{\pi}_i^{\beta}(\cdot|\gamma_+(\ell_{\bar{i}})), 1 \right\rangle$ is the normalizing constant for $\widetilde{\pi}_i^{\beta}(\cdot|\gamma_+(\ell_{\bar{i}}))$ (hence $\nu_i^{(1)}(\gamma_+)$ is the normalizing constant for $\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))$). Further, the unnormalized conditional $\widetilde{\pi}_i(\cdot|\gamma'_+(\ell_{\bar{i}}))$ at the next GS iteration differs from $\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ at no more than two points on its domain $\{\text{-}1{:}M\}$. Fig. 2 provides an illustration.

Proposition 2: Suppose that, γ'_+ and γ_+ differ only at the n-th coordinate, i.e., $\gamma'_+(\ell_{\bar{n}}) = \gamma_+(\ell_{\bar{n}})$ and $\gamma'_+(\ell_n) \neq \gamma_+(\ell_n)$. Then, for $i \in \{1{:}P\}$, $j \in \{\text{-}1{:}M\}$, and $\beta > 0$,

$$\widetilde{\pi}_i^{\beta}(j|\gamma'_+(\ell_{\bar{i}})) - \widetilde{\pi}_i^{\beta}(j|\gamma_+(\ell_{\bar{i}})) = \\ 1_{\{\bar{n}\}}(i)1_{\{1:M\}}(j)\eta_i^{\beta}(j)\left(\delta_{\gamma_+(\ell_n)}[j] - \delta_{\gamma'_+(\ell_n)}[j]\right).$$

Proof: See Appendix (Section A). ■

Remark 3: Given successive TGS iterates γ_+ and γ'_+ that differ at the n-th coordinate, Proposition 2 (with $\beta = 1$) states that their n-th conditionals are the same, i.e., $\pi_n(\cdot|\gamma_+(\ell_{\bar{n}})) = \pi_n(\cdot|\gamma'_+(\ell_{\bar{n}}))$. Moreover, for $i \neq n$, the unnormalized i-th conditional $\widetilde{\pi}_i(\cdot|\gamma'_+(\ell_{\bar{i}}))$ is the same as $\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))$, except at:



▪ $\gamma_+ = (\gamma_+(\ell_1),...,\gamma_+(\ell_n),...,\gamma_+(\ell_P))$ ▪ $\gamma_+(\ell_n) = z_j$

not exist	missed	z_1	$\cdots$	z_j	$\cdots$	$z_{j'}$	$\cdots$	z_M
$\eta_1(-1)$	$\eta_1(0)$	$\eta_1(1)$	$\cdots$	0	$\cdots$	$\eta_1(j')$	$\cdots$	$\eta_1(M)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$\eta_{n\text{-}1}(-1)$	$\eta_{n\text{-}1}(0)$	$\eta_{n\text{-}1}(1)$	$\cdots$	0	$\cdots$	$\eta_{n\text{-}1}(j')$	$\cdots$	$\eta_{n\text{-}1}(M)$
$\eta_n(-1)$	$\eta_n(0)$	$\eta_n(1)$	$\cdots$	$\eta_n(j)$	$\cdots$	$\eta_n(j')$	$\cdots$	$\eta_n(M)$
$\eta_{n+1}(-1)$	$\eta_{n+1}(0)$	$\eta_{n+1}(1)$	$\cdots$	0	$\cdots$	$\eta_{n+1}(j')$	$\cdots$	$\eta_{n+1}(M)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$\eta_P(-1)$	$\eta_P(0)$	$\eta_P(1)$	$\cdots$	0	$\cdots$	$\eta_P(j')$	$\cdots$	$\eta_P(M)$

▪ $\gamma'_+ = (\gamma_+(\ell_1),...,\gamma'_+(\ell_n),...,\gamma_+(\ell_P))$ ▪ $\gamma'_+(\ell_n) = z_{j'}$

not exist	missed	z_1	$\cdots$	z_j	$\cdots$	$z_{j'}$	$\cdots$	z_M
$\eta_1(-1)$	$\eta_1(0)$	$\eta_1(1)$	$\cdots$	$\eta_1(j)$	$\cdots$	0	$\cdots$	$\eta_1(M)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$\eta_{n\text{-}1}(-1)$	$\eta_{n\text{-}1}(0)$	$\eta_{n\text{-}1}(1)$	$\cdots$	$\eta_{n\text{-}1}(j)$	$\cdots$	0	$\cdots$	$\eta_{n\text{-}1}(M)$
$\eta_n(-1)$	$\eta_n(0)$	$\eta_n(1)$	$\cdots$	$\eta_n(j)$	$\cdots$	$\eta_n(j')$	$\cdots$	$\eta_n(M)$
$\eta_{n+1}(-1)$	$\eta_{n+1}(0)$	$\eta_{n+1}(1)$	$\cdots$	$\eta_{n+1}(j)$	$\cdots$	0	$\cdots$	$\eta_{n+1}(M)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
$\eta_P(-1)$	$\eta_P(0)$	$\eta_P(1)$	$\cdots$	$\eta_P(j)$	$\cdots$	0	$\cdots$	$\eta_P(M)$

Fig. 2. Difference between unnormalized conditionals in successive TGS iterations. If γ'_+ differs from γ_+ only at the n-th coordinate, i.e., $\gamma_+(\ell_n) = j > 0$ and $\gamma'_+(\ell_n) = j' > 0$, with $j \neq j'$, then for $i \in \{\bar{n}\}$, $\tilde{\pi}_i(\cdot|\gamma'_+(\ell_{\bar{i}}))$ is the same as $\tilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ on $\{\text{-}1{:}M\}$ except at j (gray) and j' (red).

1) $j = \gamma_+(\ell_n) > 0$, where $\eta_i(j)$ is added to $\widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}}))$ so that $\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = \eta_i(j)$ (because this j is now freed up); and
2) $j = \gamma'_+(\ell_n) > 0$, where $-\eta_i(j)$ is added to $\widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}}))$ so that $\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = 0$ (because this j is now taken).

This also means the normalizing constant for $\widetilde{\pi}_i(\cdot|\gamma'_+(\ell_{\bar{i}}))$ can be computed from that of $\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ by simply adding $\eta_i(\gamma_+(\ell_n))$ when $\gamma_+(\ell_n) > 0$, and subtracting $\eta_i(\gamma'_+(\ell_n))$ when $\gamma'_+(\ell_n) > 0$. Consequently, the unnormalized conditional $\widetilde{\pi}_i(\cdot|\cdot)$ and its normalizing constant can be propagated to the next given TGS iteration with at most two additions.

The above discussion also holds for the tempered unnormalized conditional $\widetilde{\pi}_i^{\beta}(\cdot|\cdot)$, and is stated more concisely in the following.

Corollary 1: Suppose that the successive TGS iterates γ_+ and γ'_+ differ at the n-th coordinate. Then, for $\beta > 0$,

$$\widetilde{\pi}_n^{\beta}(\cdot|\gamma'_+(\ell_{\bar{n}})) = \widetilde{\pi}_n^{\beta}(\cdot|\gamma_+(\ell_{\bar{n}})),$$
$$\left\langle \widetilde{\pi}_n^{\beta}(\cdot|\gamma'_+(\ell_{\bar{n}})), 1 \right\rangle = \left\langle \widetilde{\pi}_n^{\beta}(\cdot|\gamma_+(\ell_{\bar{n}})), 1 \right\rangle,$$

and for $i \in \{\bar{n}\}$, $j \in \{\text{-}1{:}M\}$,

$$\widetilde{\pi}_i^{\beta}(j|\gamma'_+(\ell_{\bar{i}})) = \begin{cases} \eta_i^{\beta}(j), & j = \gamma_+(\ell_n) > 0 \\ 0, & j = \gamma'_+(\ell_n) > 0\,, \\ \widetilde{\pi}_i^{\beta}(j|\gamma_+(\ell_{\bar{i}})), & \text{otherwise} \end{cases}$$

$$\left\langle \widetilde{\pi}_i^{\beta}(\cdot|\gamma'_+(\ell_{\bar{i}})), 1 \right\rangle = \left\langle \widetilde{\pi}_i^{\beta}(\cdot|\gamma_+(\ell_{\bar{i}})), 1 \right\rangle \\ + \eta_i^{\beta}(\gamma_+(\ell_n))1_{\{1:M\}}(\gamma_+(\ell_n)) \\ - \eta_i^{\beta}(\gamma'_+(\ell_n))1_{\{1:M\}}(\gamma'_+(\ell_n)).$$

Algorithm 2: TGS^+.

```
Input  : $\gamma_+^{(0)}$, $T$, $\alpha$, $\beta$, $[\eta_i(j)]_{i=1}^P$, $\rho(\cdot|\gamma_+^{(0)})$,
         $[\widetilde{\pi}_i(\cdot|\gamma_+^{(0)}(\ell_{\bar{i}}))]_{i=1}^P$, $[\nu_i^{(1)}(\gamma_+^{(0)})]_{i=1}^P$, $[\nu_i^{(\beta)}(\gamma_+^{(0)})]_{i=1}^P$
Output : $[\gamma_+^{(t)}]_{t=1}^T$

1  $M \leftarrow \mathsf{size}([\eta_i(j)]_{i=1}^P, \mathsf{col}) - 2$;
2  foreach $t = 1 : T$ do
3      $n \sim \mathsf{Categorical}([1{:}P], \rho(\cdot|\gamma_+^{(t\text{-}1)}))$;
4      $\phi_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})) \leftarrow \frac{\alpha\widetilde{\pi}_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))}{\nu_n^{(1)}(\gamma_+^{(t\text{-}1)})} + \frac{(1\text{-}\alpha)\widetilde{\pi}_n^{\beta}(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))}{\nu_n^{(\beta)}(\gamma_+^{(t\text{-}1)})}$;
5      $\gamma_+^{(t)}(\ell_n) \sim \mathsf{Categorical}([-1{:}M], \phi_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})))$;
6      $\gamma_+^{(t)} \leftarrow [\gamma_+^{(t\text{-}1)}(\ell_{1:n\text{-}1}), \gamma_+^{(t)}(\ell_n), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})]$;
7      foreach $i = 1 : P$ do
8          Compute $\widetilde{\pi}_i(\cdot|\gamma_+^{(t)}(\ell_{\bar{i}})), \nu_i^{(1)}(\gamma_+^{(t)}), \nu_i^{(\beta)}(\gamma_+^{(t)})$
               via Corollary 1;
9          $\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}})) \leftarrow \frac{\widetilde{\pi}_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}{\nu_i^{(1)}(\gamma_+^{(t)})}$;
10         $\phi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}})) \leftarrow \alpha\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))$
               $+ \frac{(1\text{-}\alpha)\widetilde{\pi}_i^{\beta}(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}{\nu_i^{(\beta)}(\gamma_+^{(t)})}$;
11         $\widetilde{\rho}(i|\gamma_+^{(t)}) \leftarrow \frac{\phi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}{\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}$;
12     $\rho(\cdot|\gamma_+^{(t)}) \leftarrow \frac{\widetilde{\rho}(\cdot|\gamma_+^{(t)})}{\langle\widetilde{\rho}(\cdot|\gamma_+^{(t)}),1\rangle}$;
```

The above result means propagating each tempered conditional and its normalizing constant to the next iterate of the Markov chain can be performed with a constant time complexity. Consequently, $\mathcal{O}(T(P+M))$ complexity TGS can be developed for selecting significant GLMB components. The pseudocode in Algorithm 2, herein referred to as TGS^+, outlines the steps for generating iterates $\gamma_+^{(1)}, \ldots, \gamma_+^{(T)}$, from initial chain state $\gamma_+^{(0)}$, and initial coordinate distribution $\rho(\cdot|\gamma_+^{(0)})$. Note that:

- Sampling n from the categorical distribution $\rho(\cdot|\gamma_+^{(t\text{-}1)})$, defined on P categories, incurs $\mathcal{O}(P)$ complexity (line 3);
- Computing the proposal $\phi_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))$ incurs $\mathcal{O}(M)$ complexity (line 4) since $\widetilde{\pi}_i(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{i}}))$, $\nu_i^{(1)}(\gamma_+^{(t\text{-}1)})$, and $\nu_i^{(\beta)}(\gamma_+^{(t\text{-}1)})$ were generated as by-products of computing $\rho(\cdot|\gamma_+^{(t\text{-}1)})$ in the previous iteration;
- Sampling $\gamma_+^{(t)}(\ell_n)$ from $\phi_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))$ incurs $\mathcal{O}(M)$ complexity (line 5);
- Computing $\widetilde{\rho}(\cdot|\gamma_+^{(t)})$ for the $(t+1)$-th iteration is only needed when $\gamma_+^{(t)}(\ell_n) \neq \gamma_+^{(t-1)}(\ell_n)$, and requires $\mathcal{O}(P)$ complexity (lines 7–11), because for each $i \in \{1{:}P\}$, evaluating (19) from the available $\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))$ and $\phi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))$ only requires a constant time complexity (Corollary 1); and
- Normalizing $\widetilde{\rho}(\cdot|\gamma_+^{(t)})$ incurs $\mathcal{O}(P)$ complexity (line 12).

TGS^+ also requires the initial coordinate distribution $\rho(\cdot|\gamma_+^{(0)})$ as an input, which can be pre-computed via the initialization routine described in Algorithm 3. The recursive construct (lines 2–6, 9–12) reduces the $\mathcal{O}(P^2M)$ complexity [30] to an $\mathcal{O}(PM)$ complexity. Specifically, for each $i \in \{1{:}P\}$, computing $\widetilde{\pi}_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}}))$, $\nu_i^{(1)}(\gamma_+^{(0)})$, $\nu_i^{(\beta)}(\gamma_+^{(0)})$ and $\widetilde{\rho}(i|\gamma_+^{(0)})$ incurs $\mathcal{O}(M)$ complexity.

Algorithm 3: Initialization.

```
Input  : $\gamma_+^{(0)}$, $\alpha$, $\beta$, $[\eta_i(j)]_{i=1}^P$
Output : $[\widetilde{\pi}_i(\cdot|\gamma_+^{(0)}(\ell_{\bar{i}}))]_{i=1}^P$, $[\nu_i^{(1)}(\gamma_+^{(0)})]_{i=1}^P$, $[\nu_i^{(\beta)}(\gamma_+^{(0)})]_{i=1}^P$,
         $\rho(\cdot|\gamma_+^{(0)})$

1  $M \leftarrow \mathsf{size}([\eta_i(j)]_{i=1}^P, \mathsf{col}) - 2$;
2  foreach $j = -1 : M$ do
3      $\widetilde{\pi}_P(j|\gamma_+^{(0)}(\ell_{\bar{P}})) \leftarrow 1$;
4  foreach $i = 1 : P$ do
5      if $\gamma_+^{(0)}(\ell_i) > 0$
6          $\widetilde{\pi}_P(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{P}})) \leftarrow 0$;
7  foreach $i = 1 : P$ do
8      foreach $j = -1 : M$ do
9          if $\widetilde{\pi}_P(j|\gamma_+^{(0)}(\ell_{\bar{P}})) = 0$ and $j \neq \gamma_+^{(0)}(\ell_i)$
10             $\widetilde{\pi}_i(j|\gamma_+^{(0)}(\ell_{\bar{i}})) \leftarrow 0$;
11         else
12             $\widetilde{\pi}_i(j|\gamma_+^{(0)}(\ell_{\bar{i}})) \leftarrow \eta_i(j)$;
13     $\pi_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}})) \leftarrow \frac{\widetilde{\pi}_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}}))}{\nu_i^{(1)}(\gamma_+^{(0)})}$;
14     $\phi_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}})) \leftarrow \alpha\pi_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}}))$
           $+ \frac{(1\text{-}\alpha)\widetilde{\pi}_i^{\beta}(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}}))}{\nu_i^{(\beta)}(\gamma_+^{(0)})}$;
15     $\widetilde{\rho}(i|\gamma_+^{(0)}) \leftarrow \frac{\phi_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}}))}{\pi_i(\gamma_+^{(0)}(\ell_i)|\gamma_+^{(0)}(\ell_{\bar{i}}))}$;
16 $\rho(\cdot|\gamma_+^{(0)}) \leftarrow \frac{\widetilde{\rho}(\cdot|\gamma_+^{(0)})}{\langle\widetilde{\rho}(\cdot|\gamma_+^{(0)}),1\rangle}$;
```

Proposition 3: Starting from any positive 1-1 initialization, the Markov Chain $\{\gamma_+^{(t)}\}_{t=1}^{\infty}$ generated by the TGS kernel (Algorithm 2) is ergodic, and converges to a stationary distribution (not necessarily π). Additionally, taking into account the importance weight $w^{(t)} \propto P/\left\langle \rho(\cdot|\gamma_+^{(t)}), 1 \right\rangle$, the weighted samples converge to the stationary distribution π in the sense that for any bounded function $h : \Gamma_+ \to \mathbb{R}$

$$\lim_{T\to\infty} \sum_{t=1}^{T} w^{(t)} h(\gamma_+^{(t)}) = \sum_{\gamma_+ \in \Gamma_+} \pi(\gamma_+) h(\gamma_+), \tag{23}$$

where the coordinate probability distribution $\rho(\cdot|\gamma_+^{(t)})$ is given by (19). Further, the variance of the importance weights is stable, i.e., does not grow with the number of coordinates.

Proof: See Appendix (Section B). ■

Remark 4: In TGS, the variance of the weights does not grow with the number of coordinates, and hence the algorithm scales gracefully with P. Combining tempering with importance sampling is a strategy used to improve slow mixing in generic MCMC [38]. However, the variance of the weights grows exponentially with the number of coordinates [53], [54], which means that the performance of a generic MCMC method deteriorates with large values of P. Unlike generic tempering in MCMC, TGS only tempers the conditional of the selected coordinate, but still inherits the benefit of improved mixing [38].

Algorithm 4: RGS^{+}.

```
Input  : $\gamma_+^{(0)}$, $T$, $[\eta_i(j)]_{i=1}^{P}$, $[\widetilde{\pi}_i(\cdot|\gamma_+^{(0)}(\ell_{\bar{i}}))]_{i=1}^{P}$
Output : $[\gamma_+^{(t)}]_{t=1}^{T}$
1 $M \leftarrow \mathsf{size}([\eta_i(j)]_{i=1}^{P}, \mathsf{col}) - 2$;
2 foreach $t = 1 : T$ do
3     $n \sim \mathcal{U}(1, P)$;
4     $\gamma_+^{(t)}(\ell_n) \sim \mathsf{Categorical}([-1{:}M], \widetilde{\pi}_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})))$;
5     $\gamma_+^{(t)} \leftarrow [\gamma_+^{(t\text{-}1)}(\ell_{1:n\text{-}1}), \gamma_+^{(t)}(\ell_n), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})]$;
6     foreach $i = 1 : P$ do
7         Compute $\widetilde{\pi}_i(\cdot|\gamma_+^{(t)}(\ell_{\bar{i}}))$ via Corollary 1;
```

D. Salient Special Cases

1) Random-scan GS: Under the proposed TGS framework, setting $\beta = 1$ in the proposal (21) translates to uniformly random coordinate selection. This special case, described by Algorithm 4, is indeed RGS, herein referred to as RGS^{+} to distinguish it from generic $\mathcal{O}(TPM)$ complexity RGS. Since $\rho(\cdot|\gamma_+)$ is a uniform distribution, it requires no computation. However, the masking steps for $\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))$ are still needed. Nonetheless the complexity of RGS^{+} reduces to $\mathcal{O}(T(P+M))$ because:

- Sampling n uniformly from $\{1{:}P\}$ incurs $\mathcal{O}(P)$ complexity (line 3);
- Sampling $\gamma_+^{(t)}(\ell_n)$ from $\widetilde{\pi}_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))$ incurs $\mathcal{O}(M)$ complexity (line 4); and
- Computing $[\widetilde{\pi}_i(\cdot|\gamma_+(\ell_{\bar{i}}))]_{i=1}^{P}$ (based on Corollary 1) takes $\mathcal{O}(P)$ complexity (lines 6–7).

2) Deterministic-scan GS: The other extreme of complete randomness, namely deterministic coordinate selection can also be accommodated by TGS^{+}. The simplest deterministic scan is to traverse the coordinates according to the periodic sequence $1, 2, \ldots, P$, i.e., the selected coordinate at t-th iteration is given by $c(t) = 1 + \mathrm{mod}(t\text{-}1, P)$. To realize this in the TGS framework, a proposal of the form

$$\varphi_i(j|\gamma_+^{(t)}(\ell_{\bar{i}})) = \delta_{c(t)}[i]\phi_i(j|\gamma_+^{(t)}(\ell_{\bar{i}})) + \varepsilon(1 - \delta_{c(t)}[i]),$$

where ε is a very small positive number. Note that the corresponding coordinate distribution is given by

$$\rho(i|\gamma_+) \propto \frac{\delta_{c(t)}[i]\phi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}{\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))} + \frac{\varepsilon(1 - \delta_{c(t)}[i])}{\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}.$$

Hence, in the limiting case as ε tends to zero, only coordinate $c(t)$ can be selected at the t-th iterate.

In principle, this special case has $\mathcal{O}(T(P+M))$ complexity. However, since the coordinate is effectively selected according to the prescribed sequence $c(t)$, the sampling step is not needed. Moreover, using the recursive construct in the initialization (Algorithm 3), it is possible to implement this so-called DGS^{+} special case as described in Algorithm 5, which only incurs an $\mathcal{O}(TM)$ complexity. A similar approach based on the reverse scan order, i.e., $P, P-1, \ldots, 1$, is possible by setting $c(t) = P - \mathrm{mod}(t\text{-}1, P)$ in Algorithm 5 (line 3).

Algorithm 5: DGS^{+}.

```
Input  : $\gamma_+^{(0)}$, $T$, $\alpha$, $\beta$, $[\eta_i(j)]_{i=1}^{P}$, $[\widetilde{\pi}_i(\cdot|\gamma_+^{(0)}(\ell_{\bar{i}}))]_{i=1}^{P}$,
         $[\nu_i^{(1)}(\gamma_+^{(0)})]_{i=1}^{P}$, $[\nu_i^{(\beta)}(\gamma_+^{(0)})]_{i=1}^{P}$
Output : $[\gamma_+^{(t)}]_{t=1}^{T}$
1  $M \leftarrow \mathsf{size}([\eta_i(j)]_{i=1}^{P}, \mathsf{col}) - 2$;
2  foreach $t = 1 : T$ do
3      $n \leftarrow c(t)$;   $m \leftarrow c(t-1)$;
4      if $\gamma_+^{(t\text{-}1)}(\ell_m) > 0$
5          $\widetilde{\pi}_m(\gamma_+^{(t\text{-}1)}(\ell_m)|\gamma_+^{(t\text{-}1)}(\ell_{\bar{m}})) \leftarrow 0$;
6      foreach $j = -1 : M$ do
7          if $\widetilde{\pi}_m(j|\gamma_+^{(t\text{-}1)}(\ell_{\bar{m}})) = 0$ and $j \neq \gamma_+^{(t\text{-}1)}(\ell_n)$
8              $\widetilde{\pi}_n(j|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})) \leftarrow 0$;
9          else
10             $\widetilde{\pi}_n(j|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})) \leftarrow \eta_n(j)$;
11     $\phi_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})) \leftarrow \frac{\alpha\widetilde{\pi}_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))}{\nu_n^{(1)}(\gamma_+^{(t\text{-}1)})} + \frac{(1\text{-}\alpha)\widetilde{\pi}_n^{\beta}(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}}))}{\nu_n^{(\beta)}(\gamma_+^{(t\text{-}1)})}$;
12     $\gamma_+^{(t)}(\ell_n) \sim \mathsf{Categorical}([-1{:}M], \phi_n(\cdot|\gamma_+^{(t\text{-}1)}(\ell_{\bar{n}})))$;
13     $\gamma_+^{(t)} \leftarrow [\gamma_+^{(t\text{-}1)}(\ell_{1:n\text{-}1}), \gamma_+^{(t)}(\ell_n), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})]$;
```

Algorithm 6: SGS^{+}.

```
Input  : $\gamma_+^{(0)}$, $T$, $[\eta_i(j)]_{i=1}^{P}$, $[\widetilde{\pi}_i(\cdot|\gamma_+^{(0)}(\ell_{\bar{i}}))]_{i=1}^{P}$
Output : $[\gamma_+^{(t)}]_{t=1}^{T}$
1  $M \leftarrow \mathsf{size}([\eta_i(j)]_{i=1}^{P}, \mathsf{col}) - 2$;
2  foreach $t = 1 : T$ do
3      $\gamma_+^{(t)}(\ell_P) \leftarrow \gamma_+^{(t\text{-}1)}(\ell_P)$;
4      foreach $n = 1 : P$ do
5          $m \leftarrow 1 + \mathsf{mod}(P\text{-}2\text{+}n, P)$;
6          if $\gamma_+^{(t)}(\ell_m) > 0$
7              $\widetilde{\pi}_m(\gamma_+^{(t)}(\ell_m)|\gamma_+^{(t)}(\ell_{1:n\text{-}1}), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})) \leftarrow 0$;
8          foreach $j = -1 : M$ do
9              if $\widetilde{\pi}_m(j|\gamma_+^{(t)}(\ell_{1:m\text{-}1}), \gamma_+^{(t\text{-}1)}(\ell_{m+1:P})) = 0$
10                                                  and $j \neq \gamma_+^{(t\text{-}1)}(\ell_n)$
11                 $\widetilde{\pi}_n(j|\gamma_+^{(t)}(\ell_{1:n\text{-}1}), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})) \leftarrow 0$;
12             else
13                 $\widetilde{\pi}_n(j|\gamma_+^{(t)}(\ell_{1:n\text{-}1}), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})) \leftarrow \eta_n(j)$;
14         $\gamma_+^{(t)}(\ell_n) \sim \mathsf{Categorical}([-1{:}M],$
                          $\widetilde{\pi}_n(\cdot|\gamma_+^{(t)}(\ell_{1:n\text{-}1}), \gamma_+^{(t\text{-}1)}(\ell_{n+1:P})))$;
15     $\gamma_+^{(t)} \leftarrow [\gamma_+^{(t)}(\ell_{1:P})]$;
```

Note that in the literature, the terms "Systematic-scan GS", "Sequential-scan GS", and "Deterministic-scan GS" all refer to "SGS". The proposed DGS^{+} is clearly not the same as SGS, but is closely related to it. Specifically, the sequence of every P-th iterate of DGS^{+}, with $\alpha = 1$ or $\beta = 1$, is indeed a sequence of SGS iterates. However, this approach to SGS (summarized in Algorithm 6), referred to as SGS^{+}, has to traverse P coordinates to generate one iterate. Hence, for T iterations of DGS^{+}, SGS^{+} effectively has T/P iterations. This also means that the complexity per iteration increases by P, and consequently SGS^{+} incurs an effective $\mathcal{O}(TPM)$ complexity, a drastic reduction from $\mathcal{O}(TP^2M)$ as in [30].

Remark 5: Apart from the forward-scan coordinate selection discussed above, other deterministic scan orders are possible.

For efficiency and sample diversity, it is important that the periodic coordinate sequence $c(t)$ scans all P coordinates in one period.

E. TGS-GLMB Filter Implementation

The TGS-based implementation of the GLMB filter is the same as the SGS-based implementation in [30, Algorithm 2], with two key changes:

- SGS-based truncation is replaced by the proposed $\mathcal{O}(T(P+M))$ TGS algorithm (Algorithm 2); and
- There is an additional $\mathcal{O}(PM)$ complexity step for computing the initial coordinate distribution (Algorithm 3).

Thus, the resulting GLMB filter implementation incurs an overall complexity of $\mathcal{O}(T(P+M)+PM+T\log T)$. The additional factor of $T\log T$ arises from the need to remove duplicates from the TGS output. Noting that in practice, the number T of iterates is large, typically $T(P+M) >> PM$, nonetheless $\log T$ is small, and $P+M >> \log T$. This is a substantial reduction from the $\mathcal{O}(TP^2M)$ complexity of the SGS-based implementation.

IV. Experimental Results

This section presents numerical studies for GLMB filtering with truncation via the proposed TGS based schemes and the new recursive implementation of SGS:

1) TGS^+: $\mathcal{O}(T(P+M))$ TGS (Algorithm 2);
2) RGS^+: $\mathcal{O}(T(P+M))$ RGS (Algorithm 4);
3) $\overrightarrow{DGS}^+$: $\mathcal{O}(TM)$ forward-scan DGS (Algorithm 5);
4) $\overleftarrow{DGS}^+$: $\mathcal{O}(TM)$ backward-scan DGS (Algorithm 5);
5) SGS^+: $\mathcal{O}(TPM)$ SGS (Algorithm 6);
6) RGS: $\mathcal{O}(TPM)$ RGS (Section III-A); and
7) SGS: $\mathcal{O}(TP^2M)$ SGS (Algorithm 2a in [30]).

Note that, although all methods use the same initialization (Algorithm 3), only TGS^+ requires $\rho(\cdot|\gamma_+)$, whereas RGS^+/SGS^+ do not require the tempering related steps. No parallel implementations are used in our experiments.

A common linear Gaussian setup on a square 3 km × 3 km surveillance region over 100 time steps is used throughout this section. The number of objects is time varying due to various births and deaths. The single-object state is a 4D vector $[x,\, \dot{x},\, y,\, \dot{y}]^{\mathsf{T}}$ of 2D position and velocity which follows a constant velocity model with sampling period 1 s and process noise standard deviation $\sigma_p = 5$ ms^{-2} on each axis. Existing objects have a constant survival probability of $P_S = 0.99$. New objects appear according to an LMB birth density with $N_B = 50$ components. Each LMB component has a fixed birth probability $P_{B,+}(\ell_i) = 0.01$ and Gaussian birth density $f_{B,+}(x,\ell_i) = \mathcal{N}(\cdot; m_B^{(i)}, R_B)$ for $i \in \{1,\ldots,N_B\}$, where the means $m_B^{(i)} = [x^{(i)},\, 0,\, y^{(i)},\, 0]^{\mathsf{T}}$ are uniformly spaced in the surveillance region, and $R_B = \mathrm{diag}([10,10,10,10]^{\mathsf{T}})^2$. This implies a mean birth rate of $\lambda_B = 0.5$ per scan, combined with the constant death or survival probability, results in a total of approximately $N_X = 50$ trajectories, or a peak of approximately 33 objects simultaneously. Observations are noisy 2D positions with noise standard deviation $\sigma_m = 10$ m on each axis. All objects have a constant detection probability of $P_D = 0.86$. Clutter follows a Poisson model with a mean rate of $\lambda_c = 90$ returns per scan and a uniform density on the observation space. Each of the GS implementations are run for $T = 5000$ iterations, and the $TGS^+/\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ use default mixture and tempering parameters of $\alpha = \beta = 0.5$.

TABLE II
Parameter Settings

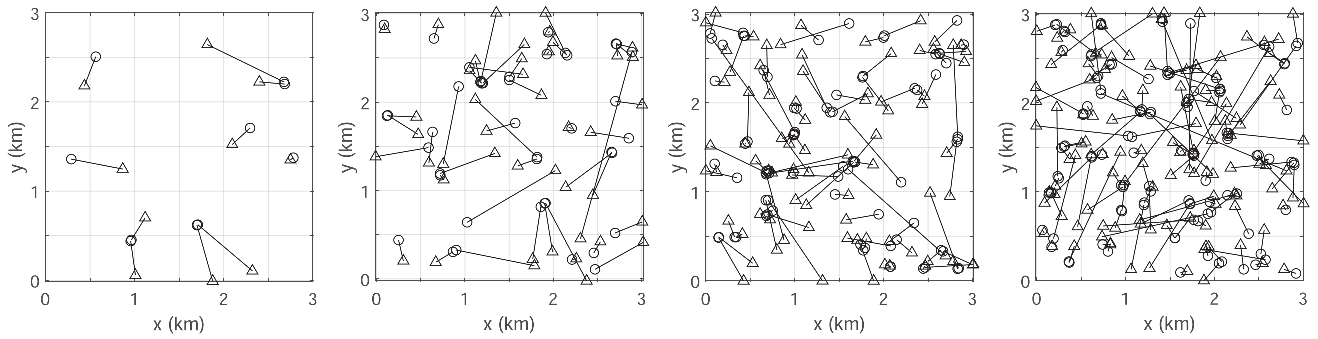
Parameter	Setting (default)
Number of Iterations T	1000 - 10000 (5000)
Total Trajectories N_X	10 - 100 (50)
Detection Rate P_D	0.78 - 0.96 (0.86)
Clutter Rate λ_c	50 - 140 (90)
Mixture Rate α	0.1 - 0.9 (0.5)
Tempering Level β	0.1 - 0.9 (0.5)

The effects of individually varying key parameters affecting T, P, and M are also examined. In addition to numerical studies with the default parameters described above, further experiments are also carried out according to Table II with varying number of GS iterations, total number of trajectories (or peak number of objects), plus detection and clutter rates. The effects of different mixture rates and tempering levels for TGS^+ are also studied.

Evaluations are undertaken for computation time, tracking accuracy, and (association) sample diversity via (i) measured wall-clock times, (ii) OSPA [55] and OSPA$^{(2)}$ [5] metrics, and (iii) the number of unique solutions per scan during GLMB filtering. Results on the tracking accuracy and sample diversity of RGS/SGS are the same as per RGS^+/SGS^+ and hence omitted. Note that each Monte Carlo trial generates random births, motions, deaths (see the example in Fig. 3) and measurements according to the multi-object model parameters. All empirical averages are then reported over 1000 Monte Carlo runs using MATLAB R2020a on a dual CPU machine with dual 64-Core AMD EPYC 7702 CPUs and 1 TB RAM.

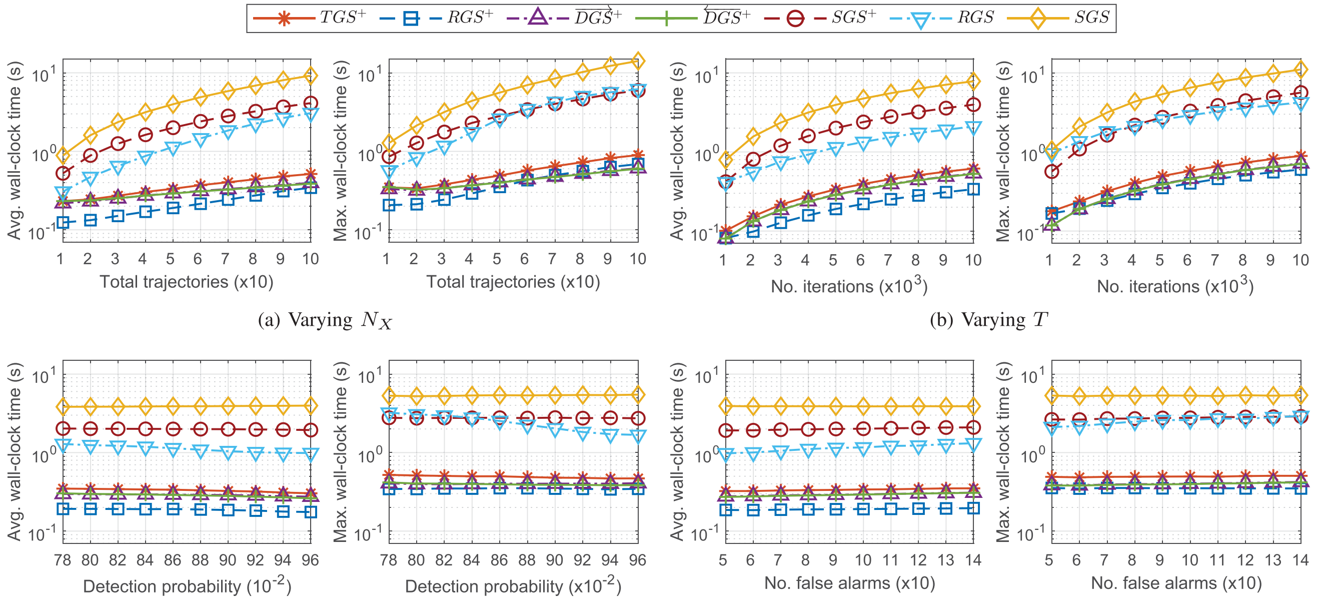
A. Computation Time

Fig. 4 shows (in log scale) the measured wall-clock times averaged per scan (*lower is better*). It is imperative to examine both average and maximum times particularly for latency sensitive real-time applications [56]. As expected, the results in Fig. 4(a) and (b) show increasing computation times with increasing numbers of trajectories and iterations. The results in Fig. 4(c) and (d) show generally flat trends of computation times with varying rates of detection and clutter, since the plots use the same default values for the number of iterations and number of trajectories. In all cases, it can be seen that all linear complexity GS implementations have similar run times, but all are significantly faster than the higher complexity methods $SGS^+/RGS/SGS$. Further, RGS^+/SGS^+ show better computational efficiency than RGS/SGS. Of the linear complexity truncation strategies, TGS^+ is generally the most expensive, due to the additional calculations involved with tempering and coordinate selection. The deterministic scan $\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ have more similar run



(a) Random scenario 1 (b) Random scenario 2 (c) Random scenario 3 (d) Random scenario 4

Fig. 3. Examples of randomly generated ground truths (○/△: start/end).



(a) Varying N_X (b) Varying T

(c) Varying P_D (d) Varying λ_c

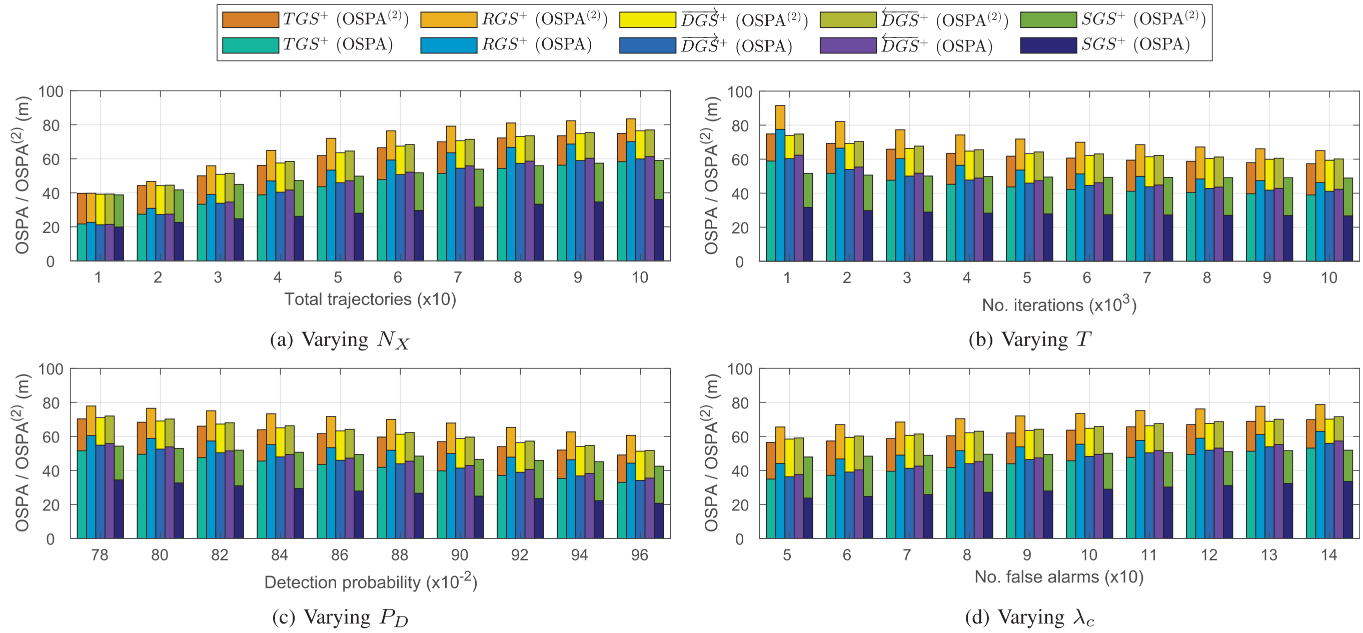
Fig. 4. Experimental results for computation time.

times to RGS^+, depending on the fraction of time spent in sampling versus overheads. Nonetheless, $TGS^+/\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ are observed to have improved tracking accuracy and improved sample diversity, when compared to RGS^+, as will be seen and further discussed in the next subsections.

B. Tracking Accuracy

Fig. 5 shows the OSPA and OSPA$^{(2)}$ metrics via overlay bar charts (*lower is better*). Specifically, an average OSPA error (taken over all time steps) and a single OSPA$^{(2)}$ error (computed over the entire scenario window) are calculated with parameters $p = 1$ and $c = 100$ m. In essence, the OSPA distance captures the error between two sets of point estimates (i.e., single-scan), whereas the OSPA$^{(2)}$ distance captures the errors between two sets of track estimates (i.e., multi-scan), both in a mathematically consistent and physically intuitive manner.

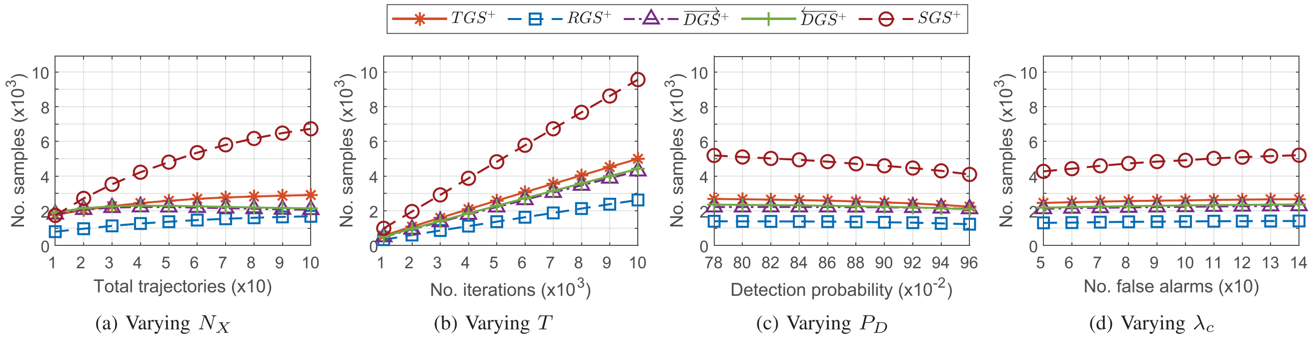
The resultant OSPA and OSPA$^{(2)}$ for varying numbers of trajectories and iterations are shown in Fig. 5(a) and (b). For a relatively small number of objects, the average filtering and tracking errors for all GS variants are virtually identical. As the number of objects is gradually increased, as expected, the corresponding errors for SGS^+ (the most expensive variant) exhibit the smallest increase of all variants. Conversely but also as expected, the errors for RGS^+ show the largest increase, due to slow mixing and insufficient diversity. The errors for TGS^+ are significantly better than those for RGS^+, even though both have the same linear complexity, due to the combination of tempering with smart coordinate selection. The errors for $\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ are higher than that of TGS^+ but are generally lower than for RGS^+. The plots for varying numbers of iterations similarly indicate that SGS^+ generally has the lowest error, while RGS^+ has the highest, and $TGS^+/\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ are comparatively robust. The effect of varying the detection and clutter rate is shown in Fig. 5(c) and (d).



(a) Varying N_X (b) Varying T

(c) Varying P_D (d) Varying λ_c

Fig. 5. Experimental results for tracking accuracy.



(a) Varying N_X (b) Varying T (c) Varying P_D (d) Varying λ_c

Fig. 6. Experimental results for sample diversity.

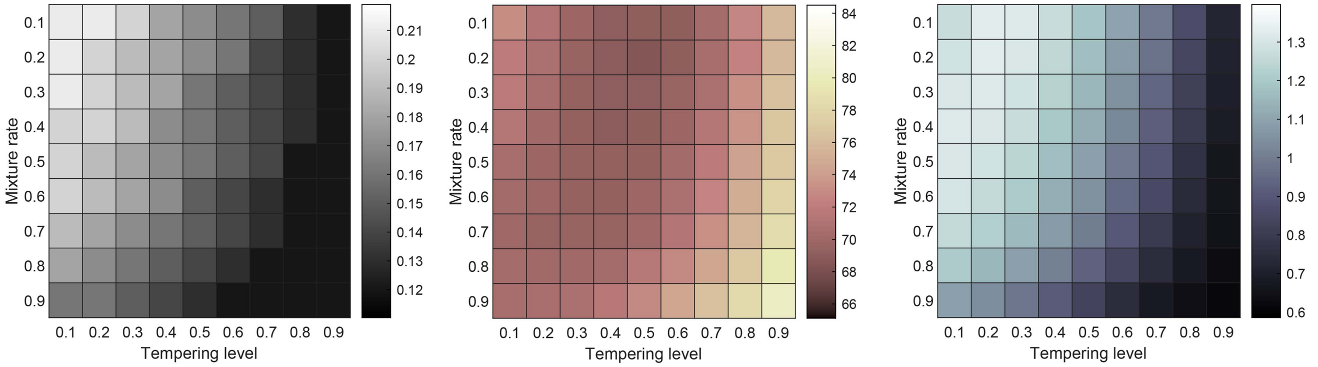
Though SGS^+ generally outperforms $TGS^+/\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$, the gap appears to narrow as the SNR increases. Again, TGS^+ outperforms $\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ which are better than RGS^+ in all cases.

Overall, the results suggest that $TGS^+/\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ generally outperform RGS^+ on tracking accuracy, even though they have similar complexities. In addition, there is small disparity between $\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$, suggesting that scan order can influence results. Unsurprisingly, it is observed that SGS^+ generally outperforms TGS^+, due to a significant disparity in complexity and hence running times. Consequently, the proposed linear complexity TGS^+ based solutions have the potential to offer a robust trade-off between tracking performance and computational load.

C. Sample Diversity

The average number of unique samples over the entire scenario is shown in Fig. 6 (*higher is better*). A higher number of unique samples means a higher number of association hypotheses, or mixture components in the estimated GLMB filtering density, which generally results in a lower GLMB truncation error.

The trend in Fig. 6(a) confirms that the number of unique samples follows the increase with the number of trajectories, which is necessary to capture the corresponding increase in the diversity of the components in the filtering density. In the case of the lowest number of objects, it can be seen in Fig. 5(a) that even though all the GS variants have the same tracking accuracy, they generally produce different numbers of unique solutions. This is due to the fact that a small number of samples is still sufficient to capture the GLMB filtering density. As the number of objects increases, it can be seen that SGS^+ produces more than twice the number of unique samples of TGS^+, but incurs a significant increase in time complexity, which is expected due to the exhaustive traversal of all coordinates in SGS^+. In addition, the informed (single) coordinate selection in TGS^+ produces more unique samples than RGS^+ even though both have the same complexity. Note also from Fig. 6(a) that the number of



(a) Avg. wall-clock time (s)

(b) OSPA$^{(2)}$

(c) No. samples ($\times 10^3$)

Fig. 7. Experimental results for mixture rate and tempering level.

unique samples from $\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ is also higher than that from RGS^+ but lower than that from TGS^+. The relative trends in the number of unique samples are generally consistent with those of the tracking error shown in Fig. 5(a).

Examination of Fig. 6(b) confirms that the number of iterations directly influences to the number of unique samples. It can also be seen that TGS^+ produces up to twice the number of unique samples as RGS^+. While $\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ produce slightly fewer unique samples than TGS^+, there is also a corresponding reduction in the observed run times. As expected, SGS^+ outperforms the other linear complexity variants, but at a significant increase in computational complexity. Again, it is observed in general that the relative increase in the number of unique samples is consistent with the increasing running times in Fig. 4(b) and with the decreasing OSPA and OSPA$^{(2)}$ values in Fig. 5(b). Inspection of the curves for increasing detection rates in Fig. 6(c) shows a decreasing trend due to lower number of components required to accurately represent the GLMB filtering density in higher SNR. Inspection of the curves for increasing clutter rates in Fig. 6(d) shows the converse trend.

It can be seen that $TGS^+/\overrightarrow{DGS}^+/\overleftarrow{DGS}^+$ generally outperform RGS^+ on increased sample diversity, and that the former has the same (or smaller) complexity than the latter. Compared to RGS^+, the experimental results in general suggest that the use of tempering and/or state informed coordinate selection can significantly improve slow mixing and sample diversity, but nonetheless retain the benefit of linear complexity. Furthermore, the improvement in tracking accuracy of each method levels off once enough significant samples have accumulated.

D. Mixture Rate and Tempering Level

Fig. 7 shows the experimental results for TGS^+, in 2D histograms, with varying mixture and tempering parameters, α and β. Recall from (21) that higher values α or β weaken the effect of tempering. Note from the averaged wall-clock times over all scenarios shown in Fig. 7(a) that weak tempering or mixing slightly reduces computation time. More unique solutions require slightly longer run times due to the removal of duplicated samples. Tracking accuracy (in OSPA$^{(2)}$) generally improves with stronger tempering or mixing (low values of α or β) as shown in Fig. 7(b). However, tracking performance may degrade with excessive tempering that over flattens the significant modes of the original distribution. From the average number of unique samples shown in Fig. 7(c), observe that stronger tempering or mixing can increase sample diversity, although excessive tempering can reduce sample diversity for a low memory budget implementation since the samples are no longer a good representation of the stationary distributions. It is, however, difficult to determine the optimal choices of α and β because they are heavily scenario-dependent.

V. Conclusion

A linear complexity Gibbs Sampling (GS) framework for GLMB filtering density computation has been developed. Specifically, we use the recently developed tempered GS approach to generate significant component of the GLMB filtering density based on measurements received. Our so-called TGS^+ framework tailors tempered GS to exploit the structure of the problem for scalable GLMB filtering, and offers trade-offs between tracking accuracy, computational efficiency, and memory load. Further, this framework enables the SGS algorithm in [30] to be implemented with one order of magnitude reduction in complexity, although this is still higher than linear. Comprehensive numerical experiments compare the performance of the linear complexity TGS^+, and the two special cases of deterministic and completely random-scan GS, namely DGS^+ and RGS^+. Our results indicate that, of the linear complexity solutions, TGS^+ provides the best tracking accuracy. DGS^+ is simple to implement and slightly faster, with a very small degradation in tracking performance. While RGS^+ is the cheapest, its tracking performance degrades significantly. Optimizing TGS^+ could yield better balance between computational load and tracking accuracy, and is a prospective venue for investigation.

Due to the effectiveness of the proposed approach, extension to the multi-dimensional assignment problem [37] could alleviate the computational bottlenecks in multi-scan and/or multi-sensor truncation. The recent multi-sensor multi-scan GLMB smoother proposed in [12] extends the SGS solution to the multi-dimensional assignment problem. While this is the first solution to address multi-dimension assignment problems of such large scale, we envisage that its time complexity can be drastically reduced using our proposed TGS approach.

Appendix

A. *Proof of Proposition 2*

Proof: If $i = n$, then $\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = \widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}}))$, because $\gamma'_+(\ell_{\bar{n}}) = \gamma_+(\ell_{\bar{n}})$. Suppose $i \neq n$, then we have the following.

1) For $j < 1$, it follows from Proposition 1 that $\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = \eta_i(j) = \widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}}))$.
2) For $j > 0$, note that $n \in \{\bar{i}\}$ implies $\gamma_+(\ell_n) \in \{\gamma_+(\ell_{\bar{i}})\}$, and $\gamma'_+(\ell_n) \in \{\gamma'_+(\ell_{\bar{i}})\}$, i.e.,

$$1_{\gamma_+(\ell_{\bar{i}})}(\gamma_+(\ell_n)) = 1, \tag{24}$$

$$1_{\gamma'_+(\ell_{\bar{i}})}(\gamma'_+(\ell_n)) = 1. \tag{25}$$

Further, since γ'_+ and γ_+ are positive 1-1 and differ only at the n-th coordinate wherein $\gamma'_+(\ell_n) \neq \gamma_+(\ell_n)$, we have

$$\{\gamma_+(\ell_{\bar{i}})\} = (\{\gamma'_+(\ell_{\bar{i}})\} \backslash \{\gamma'_+(\ell_n)\}) \cup \{\gamma_+(\ell_n)\}, \tag{26}$$

$$\{\gamma'_+(\ell_{\bar{i}})\} = (\{\gamma_+(\ell_{\bar{i}})\} \backslash \{\gamma_+(\ell_n)\}) \cup \{\gamma'_+(\ell_n)\}, \tag{27}$$

and hence $\gamma_+(\ell_n) \notin \{\gamma'_+(\ell_{\bar{i}})\}$, and $\gamma'_+(\ell_n) \notin \{\gamma_+(\ell_{\bar{i}})\}$, i.e.,

$$1_{\gamma'_+(\ell_{\bar{i}})}(\gamma_+(\ell_n)) = 0, \tag{28}$$

$$1_{\gamma_+(\ell_{\bar{i}})}(\gamma'_+(\ell_n)) = 0. \tag{29}$$

Now, apply Proposition 1 to each of the following cases.

(a) $j = \gamma_+(\ell_n)$:
$\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = \eta_i(j)(1 - 1_{\gamma'_+(\ell_{\bar{i}})}(j)) = \eta_i(j)$, using (28); and
$\widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}})) = \eta_i(j)(1 - 1_{\gamma_+(\ell_{\bar{i}})}(j)) = 0$, using (24).

(b) $j = \gamma'_+(\ell_n)$:
$\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = \eta_i(j)(1 - 1_{\gamma'_+(\ell_{\bar{i}})}(j)) = 0$, using (25); and
$\widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}})) = \eta_i(j)(1 - 1_{\gamma_+(\ell_{\bar{i}})}(j)) = \eta_i(j)$, using (29).

(c) $j \neq \gamma'_+(\ell_n)$ and $j \neq \gamma_+(\ell_n)$: it follows from (26), (27) that $j \in \{\gamma'_+(\ell_{\bar{i}})\}$ iff $j \in \{\gamma_+(\ell_{\bar{i}})\}$, i.e., $1_{\gamma'_+(\ell_{\bar{i}})}(j) = 1_{\gamma_+(\ell_{\bar{i}})}(j)$. Hence, $\widetilde{\pi}_i(j|\gamma'_+(\ell_{\bar{i}})) = \eta_i(j)\ (1 - 1_{\gamma'_+(\ell_{\bar{i}})}(j)) = \widetilde{\pi}_i(j|\gamma_+(\ell_{\bar{i}}))$.

Raising the unnormalized conditionals in (a), (b), and (c) to the power of $\beta > 0$, and taking their difference, i.e., $\widetilde{\pi}_i^{\beta}(j|\gamma'_+(\ell_{\bar{i}})) - \widetilde{\pi}_i^{\beta}(j|\gamma_+(\ell_{\bar{i}}))$, give the desired result. ■

B. *Proof of Proposition 3*

Proof: Recall from (19) that $\rho(i|\gamma_+) \propto \frac{\phi_i(\gamma_+(\ell_i)|\gamma_+(\ell_{\bar{i}}))}{\pi_i(\gamma_+(\ell_i)|\gamma_+(\ell_{\bar{i}}))}$ and let $\varpi(\gamma_+) = \frac{1}{P}\sum_{i=1}^{P} \rho(i|\gamma_+)$. Since Γ_+ is a discrete space then $\pi(\gamma_+) \in [0, 1]$. Furthermore, note from [30] the assumption that $\eta_i(j) \in (0, \infty)$ and from (16) it follows for any positive 1-1 $\gamma_+ \in \Gamma_+$ that $\pi(\gamma_+) \in (0, 1)$. By similar reasoning it follows from (21) for any positive 1-1 $\gamma_+ \in \Gamma_+$ that $\phi_i \in (0, 1)$. Hence, $\varpi(\gamma_+) \in (0, \infty)$ for any positive 1-1 $\gamma_+ \in \Gamma_+$.

In [30], it was shown that the standard Gibbs sampler (i.e., SGS) is π-irreducible. It was also shown in [57] that the TGS extension to generate the Markov Chain $\{\gamma_+^{(t)}\}_{t=1}^{\infty}$ is $\pi\varpi$-irreducible. Hence, it follows from [38, Proposition 1] that TGS is reversible with respect to $\pi\varpi$. Furthermore, since the importance weight is $w^{(t)} = [\varpi(\gamma_+^{(t)})]^{-1}$ see [38, (1)],

$$\lim_{T\to\infty} \frac{\sum_{t=1}^{T} w^{(t)} h(\gamma_+^{(t)})}{\sum_{t=1}^{T} w^{(t)}} = \sum_{\gamma_+ \in \Gamma_+} \pi(\gamma_+) h(\gamma_+),$$

for any bounded test function h. The convergence result follows noting that $[\varpi(\gamma_+^{(t)})]^{-1} \propto P/\left\langle \rho(\cdot|\gamma_+^{(t)}), 1 \right\rangle$. The stability of the importance weights follows from [38, Proposition 2], i.e., the variance of the weights does not grow with the number of coordinates. Specifically,

$$var(w^{(t)}) \leq \max_{i,\gamma_+^{(t)}} \frac{\pi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))}{\phi_i(\gamma_+^{(t)}(\ell_i)|\gamma_+^{(t)}(\ell_{\bar{i}}))} - 1.$$

■

References

[1] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA, USA: Artech House, 1999.

[2] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion*, vol. 11. Storrs, CT, USA: YBS Publishing Storrs, 2011.

[3] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*, Norwood, MA, USA: Artech House, 2014.

[4] R. P. Mahler, “Multitarget Bayes filtering via first-order multitarget moments,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

[5] M. Beard, B. T. Vo, and B.-N. Vo, “A solution for large-scale multi-object tracking,” *IEEE Trans. Signal Process.*, vol. 68, pp. 2754–2769, 2020.

[6] R. Mahler, “PHD filters of higher order in target number,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, Oct. 2007.

[7] B.-T. Vo and B.-N. Vo, “Labeled random finite sets and multi-object conjugate priors,” *IEEE Trans. Signal Process.*, vol. 61, no. 13, pp. 3460–3475, Jul. 2013.

[8] B.-N. Vo and B.-T. Vo, “A multi-scan labeled random finite set model for multi-object state estimation,” *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4948–4963, Oct. 2019.

[9] F. Papi, B.-N. Vo, B.-T. Vo, C. Fantacci, and M. Beard, “Generalized labeled multi-Bernoulli approximation of multi-object densities,” *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5487–5497, Oct. 2015.

[10] M. Beard, B.-T. Vo, B.-N. Vo, and S. Arulampalam, “Void probabilities and Cauchy-Schwarz divergence for generalized labeled multi-Bernoulli models,” *IEEE Trans. Signal Process.*, vol. 65, no. 19, pp. 5047–5061, Oct. 2017.

[11] B.-N. Vo, B.-T. Vo, and M. Beard, “Multi-sensor multi-object tracking with the generalized labeled multi-Bernoulli filter,” *IEEE Trans. Signal Process.*, vol. 67, no. 23, pp. 5952–5967, Dec. 2019.

[12] D. Moratuwage, B.-N. Vo, B.-T. Vo, and C. Shim, “Multi-scan multi-sensor multi-object state estimation,” *IEEE Trans. Signal Process.*, vol. 70, pp. 5429–5442, 2022.

[13] S. Li, W. Yi, R. Hoseinnezhad, B. Wang, and L. Kong, “Multiobject tracking for generic observation model using labeled random finite sets,” *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 368–383, Jan. 2018.

[14] T. T. D. Nguyen, B.-N. Vo, B.-T. Vo, D. Y. Kim, and Y. S. Choi, “Tracking cells and their lineages via labeled random finite sets,” *IEEE Trans. Signal Process.*, vol. 69, pp. 5611–5626, 2021.

[15] J. Ong, B.-T. Vo, B.-N. Vo, D. Y. Kim, and S. Nordholm, “A Bayesian filter for multi-view 3D multi-object tracking with occlusion handling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2246–2263, May 2022.

[16] A. Trezza, D. J. Bucci, and P. K. Varshney, “Multi-sensor joint adaptive birth sampler for labeled random finite set tracking,” *IEEE Trans. Signal Process.*, vol. 70, pp. 1010–1025, 2022.

[17] H. Van Nguyen, H. Rezatofighi, B.-N. Vo, and D. C. Ranasinghe, "Online UAV path planning for joint detection and tracking of multiple radio-tagged objects," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5365–5379, Oct. 2019.

[18] B. Wang, W. Yi, S. Li, M. R. Morelande, L. Kong, and X. Yang, "Distributed multi-target tracking via generalized multi-Bernoulli random finite sets," in *Proc. IEEE 18th Int. Conf. Inf. Fusion*, 2015, pp. 253–261.

[19] S. Li, W. Yi, R. Hoseinnezhad, G. Battistelli, B. Wang, and L. Kong, "Robust distributed fusion with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 66, no. 2, pp. 278–293, Jan. 2018.

[20] S. Li, G. Battistelli, L. Chisci, W. Yi, B. Wang, and L. Kong, "Computationally efficient multi-agent multi-object tracking with labeled random finite sets," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 260–275, Jan. 2019.

[21] M. Herrmann, C. Hermann, and M. Buchholz, "Distributed implementation of the centralized generalized labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 69, pp. 5159–5174, 2021.

[22] H. Deusch, S. Reuter, and K. Dietmayer, "The labeled multi-bernoulli SLAM filter," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1561–1565, Oct. 2015.

[23] A. K. Gostar et al., "Cooperative sensor fusion in centralized sensor networks using Cauchy-Schwarz divergence," *Signal Process.*, vol. 167, 2020, Art. no. 107278.

[24] A. K. Gostar et al., "Centralized cooperative sensor fusion for dynamic sensor network with limited field-of-view via labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 69, pp. 878–891, 2021.

[25] W. J. Hadden et al., "Stem cell migration and mechanotransduction on linear stiffness gradient hydrogels," *Proc. Nat. Acad. Sci.*, vol. 114, no. 22, pp. 5647–5652, 2017.

[26] J. Ong, B. T. Vo, S. E. Nordholm, B. N. Vo, D. Moratuwage, and C. Shim, "Audio-visual based online multi-source separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 1219–1234, 2022.

[27] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Res.*, vol. 16, no. 3, pp. 682–687, 1968.

[28] M. L. Miller, H. S. Stone, and I. J. Cox, "Optimizing Murty's ranked assignment method," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 3, pp. 851–862, Jul. 1997.

[29] C. R. Pedersen, L. R. Nielsen, and K. A. Andersen, "An algorithm for ranking assignments using reoptimization," *Comput. Operations Res.*, vol. 35, no. 11, pp. 3714–3726, 2008.

[30] B.-N. Vo, B.-T. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1975–1987, Apr. 2017.

[31] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[32] B. Yang, J. Wang, and W. Wang, "An efficient approximate implementation for labeled random finite set filtering," *Signal Process.*, vol. 150, pp. 215–227, 2018.

[33] L. M. Wolf and M. Baum, "Deterministic Gibbs sampling for data association in multi-object tracking," in *Proc. IEEE Int. Conf. Multi. Fusion Integ. Intell. Syst.*, 2020, pp. 291–296.

[34] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3246–3260, Jun. 2014.

[35] J. Y. Yu, A.-A. Saucan, M. Coates, and M. Rabbat, "Algorithms for the multi-sensor assignment problem in the δ-generalized labeled multi-Bernoulli filter," in *Proc. IEEE 7th Int. Workshop Comput. Adv. Multi-Sensor Adaptive Process.*, 2017, pp. 1–5.

[36] A.-A. Saucan and P. K. Varshney, "Distributed cross-entropy δ-GLMB filter for multi-sensor multi-target tracking," in *Proc. IEEE Int. Conf. Inf. Fusion*, 2018, pp. 1559–1566.

[37] D. M. Nguyen, H. A. Le Thi, and T. Pham Dinh, "Solving the multidimensional assignment problem by a cross-entropy method," *J. Combinatorial Optim.*, vol. 27, no. 4, pp. 808–823, 2014.

[38] G. Zanella and G. Roberts, "Scalable importance tempering and Bayesian variable selection," *J. Roy. Stat. Soc.: Ser. B. (Stat. Methodol.)*, vol. 81, no. 3, pp. 489–517, 2019.

[39] X. Wang, R. Hoseinnezhad, A. K. Gostar, T. Rathnayake, B. Xu, and A. Bab-Hadiashar, "Multi-sensor control for multi-object Bayes filters," *Signal Process.*, vol. 142, pp. 260–270, 2018.

[40] H. Van Nguyen, H. Rezatofighi, B.-N. Vo, and D. C. Ranasinghe, "Multi-objective multi-agent planning for jointly discovering and tracking mobile objects," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7227–7235.

[41] S. Panicker, A. K. Gostar, A. Bab-Hadiashar, and R. Hoseinnezhad, "Tracking of targets of interest using labeled multi-Bernoulli filter with multi-sensor control," *Signal Process.*, vol. 171, 2020, Art. no. 107451.

[42] W. Wu, H. Sun, Y. Cai, and J. Xiong, "MM-GLMB filter-based sensor control for tracking multiple maneuvering targets hidden in the Doppler blind zone," *IEEE Trans. Signal Process.*, vol. 68, pp. 4555–4567, 2020.

[43] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems: Revised Reprint*. Philadelphia, PA, USA: SIAM, 2012.

[44] G. Casella and E. I. George, "Explaining the Gibbs sampler," *Amer. Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[45] G. O. Roberts and J. S. Rosenthal, "On convergence rates of Gibbs samplers for uniform distributions," *Ann. Appl. Probability*, vol. 8, pp. 1291–1302, 1998.

[46] D. B. He, M. C. De Sa, I. Mitliagkas, and C. Ré, "Scan order in Gibbs sampling: Models in which it matters and bounds on how much," *Adv. Neural Inf. Process. Syst.*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., vol. 29, 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/e4da3b7fbbce2345d7772b0674a318d5-Paper.pdf

[47] C. Bélisle, "Slow convergence of the Gibbs sampler," *Can. J. Statist.*, vol. 26, no. 4, pp. 629–641, 1998.

[48] G. O. Roberts and S. K. Sahu, "Updating schemes, correlation structure, blocking and parameterization for the gibbs sampler," *J. Roy. Stat. Soc.: Ser. B. (Stat. Methodol.)*, vol. 59, no. 2, pp. 291–317, 1997.

[49] P. Diaconis, K. Khare, and L. Saloff-Coste, "Gibbs sampling, exponential families and orthogonal polynomials," *Stat. Sci.*, vol. 23, no. 2, pp. 151–178, 2008.

[50] R. Gramacy, R. Samworth, and R. King, "Importance tempering," *Statist. Comput.*, vol. 20, no. 1, pp. 1–7, 2010.

[51] J. Griffin, K. Łatuszyński, and M. Steel, "In search of lost mixing time: Adaptive Markov chain Monte Carlo schemes for Bayesian variable selection with very large p," *Biometrika*, vol. 108, no. 1, pp. 53–69, 2021.

[52] Q. Zhou and A. Smith, "Rapid convergence of informed importance tempering," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2022, pp. 10939–10965.

[53] S. T. Tokdar and R. E. Kass, "Importance sampling: A review," *Wiley Interdiscipl. Rev.: Comput. Statist.*, vol. 2, no. 1, pp. 54–60, 2010.

[54] A. B. Owen, "Monte Carlo theory, methods and examples," 2013.

[55] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.

[56] S. Reuter, A. Danzer, M. Stübler, A. Scheel, and K. Granström, "A fast implementation of the labeled multi-Bernoulli filter using Gibbs sampling," in *Proc. IEEE Intell. Veh. Symp.*, 2017, pp. 765–772.

[57] G. O. Roberts and A. F. Smith, "Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms," *Stochastic Processes Appl.*, vol. 49, no. 2, pp. 207–216, 1994.