IEEE Signal Processing Society
IEEE Open Journal of
Signal Processing

Received 14 August 2023; revised 30 November 2023; accepted 30 November 2023. Date of publication 21 December 2023; date of current version 19 January 2024. The review of this article was arranged by Associate Editor Lotfi Chaari.

Digital Object Identifier 10.1109/OJSP.2023.3344076

Detection and Estimation of Gas Sources With Arbitrary Locations Based on Poisson's Equation

DMITRIY SHUTIN [1] (Senior Member, IEEE), THOMAS WIEDEMANN [1], AND PATRICK HINSEN [1]

Institute of Communications and Navigation, German Aerospace Center, 82234 Wessling, Germany

CORRESPONDING AUTHOR: DMITRIY SHUTIN (e-mail: dmitriy.shutin@dlr.de).

This work was supported in part by DLR Project STARE and in part by EU Project TEMA. The TEMA Project was supported by the European Commission — European Union through HORIZON EUROPE (HORIZON Research and Innovation Actions) and HORIZON-CL4-2022-DATA-01-01 under Grant 101093003.

ABSTRACT Accurate estimation of the number and locations of dispersed material sources is critical for optimal disaster response in Chemical, Biological, Radiological, or Nuclear accidents. This paper introduces a novel approach to Gas Source Localization that uses sparse Bayesian learning adapted to models based on Partial Differential Equations for modeling gas dynamics. Using the method of Green's functions and the adjoint state method, a gradient-based optimization with respect to source locations is derived, allowing superresolving (arbitrary) source locations. By combing the latter with sparse Bayesian learning, a sparse source support can be identified, thus indirectly assessing the number of sources. Simulation results and comparisons with classical sparse estimators for linear models demonstrate the effectiveness of the proposed approach. The proposed sparsity-constrained gas source localization method offers thus a flexible solution for disaster response and robotic exploration in hazardous environments.

INDEX TERMS Adjoint state method, PDE constrained optimization, sparse Bayesian learning, super-resolution sparsity.

I. INTRODUCTION

In situations involving Chemical, Biological, Radiological, or Nuclear incidents, accurately modeling and estimating the spatial and temporal progression of airborne released substances holds immense significance for optimizing disaster response efforts. The use of mobile robotic platforms, equipped with suitable sensors and capable of autonomous operation, emerges as an ideal solution for navigating and functioning within these hazardous settings. To enhance the autonomy of such robotic systems, it becomes imperative for the robots to acquire an understanding of their surroundings, which in turn enables the derivation of optimal operational and control strategies.

Accurate concentration mapping of the airbourne substances can be realized through the process of "learning" from data, employing data-driven methodologies. Various approaches have been proposed to address this objective, including support vector machines for concentration mapping [1], kernel methods [2], and deep-learning techniques, albeit predominantly applied within the visual domain [3], [4]. One notable limitation of these methodologies lies in their reliance on training data, particularly evident in the case of deep-learning methods. Additionally, these methods depend on visual sensors for gas plume registration, rendering them unsuitable when the gas is imperceptible to the camera. While chemical plume data can be simulated for training purposes [5], achieving realistic simulations necessitates the incorporation of domain knowledge pertaining to gas propagation. It is thus reasonable to ask why not directly utilize this domain knowledge?

Indeed, robotic platforms can take advantage of existing knowledge on gas propagation, such as a model for gas dispersion, and selectively acquire specific facets of this model from available data. This strategic approach mitigates the dependency on extensive training data, fostering more robust and versatile solutions for concentration mapping, in particular in scenarios involving chemical, rather than visual sensors. Such an approach in practice will thus serve a dual purpose: it will

compensate for the lack of data while also adjusting the model to align better with the gathered measurements. This paradigm forms the foundation of the approach discussed in this work.

In the broader context, the underlying physics governing the spatio-temporal dynamics of diffused substances can be effectively characterized through Partial Differential Equations (PDEs). For instance, the non-homogeneous advection-diffusion equation or scalar transport equation [6] serves as a means to simulate the dissemination of materials through the air, originating from multiple emission sources or being absorbed by designated material "sinks." Given the parameters of these equations, such as boundary and initial conditions as well as the count and positioning of emission sources, the spatio-temporal evolution of material concentration can be numerically computed. This is known as a *forward* modeling problem: it involves solving the corresponding PDE subject to boundary and initial conditions. However, real-world scenarios frequently involve instances where some, if not all, of these parameters are unknown: one might encounter undefined boundary conditions or an unknown count of emission sources. Their inference from measurements then becomes the objective of the robotic exploratory system, which constitutes an *inverse* problem. Notably, inverse problems are characterized by their ill-posed nature, demanding supplementary constraints or regularization techniques to ensure the stability and existence of a solution. In the present study, we direct our attention towards incorporating such constraints, specifically sparsity constraints. As we will demonstrate, this will allow us to reliably estimate both the count and locations of dispersed material sources, leveraging the information gathered through measurements.

The problem we consider is referred to as Gas Source Localization (GSL) [7] in the robotic olfaction community – determination of the spatial locations and number of gas sources. Early GSL techniques operated under quite restrictive assumptions. Specifically, they assumed a fixed count (typically one) of sources and integrated source estimation into the robot's movement strategy. For instance, approaches based on *chemotaxis* [8], [9] aimed to guide a robot or even a group of robots [10] along the gradient of chemical concentration. However, these methodologies confront inherent limitations posed by the intricacies of the actual chemical plume. Turbulence causes a high degree of variability of the concentration gradient, especially in regions distant from the source. This issue can be mitigated by accounting for wind patterns and adopting bioinspired algorithms [11], [12], referred to as *anemotaxis*. Standard *anemotaxis* can be advanced through the application of "higher order" method, which we term as *model-based anemotaxis*. In this approach, not only spatial gradients but also divergence and temporal variations of concentration values are considered [13]. In the latter work the authors employ collected data to estimate divergence and gradients, ensuring adherence to the advection-diffusion PDE. Subsequently, a navigation strategy for multiple robots is formulated to track the plume front. This method permits extension to inverse problems, assuming unknown wind or diffusion parameters [14], and addressing the GSL problem [15], [16] in lieu of front tracking. It is crucial to emphasize that, while [13], [14], [15] capitalize on domain knowledge to notably enhance standard anemotaxis performance, the approach does not directly solve (or invert) the advection-diffusion PDE. Instead, it relies on numerically estimated concentration gradients and divergence using historical observation data or data readings from multiple robots – an intricate task due to noise and the challenges associated with numerical differentiation. Consequently, the smoothness of gradients remains a significant constraint for these strategies. Furthermore, it should be mentioned that the number of gas sources remains fixed.

Nevertheless, the latter methods clearly highlight the benefits of leveraging the inherent mathematical structure of the dispersion process for GSL when compared to purely gradient-based strategies. Notably, more promising avenues in this context are revealed through the exploration of probabilistic methodologies. Instead of tracing gas concentration gradients, these methods treat the sources as latent parameters, which are subsequently estimated from the collected measurements. Particularly appealing are Bayesian approaches for parameter estimation [17]. These approaches permit accounting for uncertainties pertaining to source parameters [18], [19], [20], accommodating different models [18], [19], [20] and unknown environments [21], [22]. Furthermore, the Bayesian framework facilitates the adoption of *infotaxis*-based strategies [23] for autonomous robotic navigation, thereby enhancing GSL and directing robots toward sources through distinct information-theoretic criteria (see e.g., [23], [24], [25]). Nonetheless, the majority of such GSL approaches assumes the number of sources to be both fixed and known a priori. This situation arises mainly due to the computational complexity associated with integer optimization and inference pertaining to source counts. Pioneering work that relaxes this assumption was introduced in [26], [27]. There, a sparse Bayesian learning (SBL) approach [28], [29] coupled with a PDE-based dispersion model was employed, representing a departure from the fixed-source count assumption.

The key idea behind the SBL-based approach towards GSL is that a very large number of potential sources is initially assumed. However, the rates of the material release for these sources – the source weighs – are *constrained* to be sparse. In other words, it is assumed that only a few source weights are non-zero. This approach was shown to accurately estimate both the count and the locations of the sources in simulations [26], [27], [30], as well as in real experiments [31], [32] (see also [33] for a more extensive discussion). It thus allows "trading" integer optimization concerning source counts for a non-integer, but sparsity-constrained estimation of source parameters.

One of the features of such PDE-based approach towards GSL is a discretization of the equation required for its numerical solution. Standard approaches, like finite elements or

finite difference methods [34] discretize the equation both in space and time; as a result, the unknown functions – gas concentration and source distribution functions – are represented with a set of discrete parameters. As a consequence, the source locations become constrained to the locations of the vertices in the discretization mesh or centers of the discretization cells.[1] While in general this can be compensated again by imposing sparsity constraints on the resulting source estimate [27], [33], practically this leads to a high number of unknown parameters that have to be estimated – one parameter per each possible source. Also, the discretization has to be dense enough, in order not to miss any potential locations. Alternatively, expensive re-meshing has to be used. This inadvertently increases the number of measurements needed to reliably identify the parameters. In particular, in cases where the source signal is very sparse, i.e., when only a few sources are present, such over-parameterization negatively impact algorithm performance, especially in the early stages, when the number of measurements is low. Furthermore, the discretization limits the source localization accuracy to the size of the mesh elements closest to the actual source location.

The methodology introduced within this study alleviates the aforesaid constraint, allowing for arbitrary spatial positioning of sources within the exploration area. Specifically, we abstract from robotic aspects of GSL and focus on the underlying numerical algorithm. Our method uses a Poisson's equation as a specific (and simplified) model within the broader context of the advection-diffusion framework that governs gas propagation. The source term of this model is expressed as a linear combination of Dirac measures with unknown support and weights. By casting the resulting model in a Bayesian framework we then harness SBL to enforce sparsity constraints over the source weights.

The algorithmic strategy that we propose employs an iterative process that alternates between (i) estimation of source spatial support with source locations kept fixed, and (ii) updating locations of the sources that have non-zero weights. The latter involves a nonlinear optimization procedure, solved using a gradient descent technique [34]. We derive the corresponding gradient using a so-called adjoint state method [6], [35]. As a consequence, the proposed method allows for arbitrary source placement. In particular for scenarios with a number of well-distinguishable, separated sources this leads to an optimization problem of significantly lower dimensionality (of the number of "active" sources only). Moreover, such optimization approach brings additional advantages. Firstly, the presented scheme can be extended to distributed settings, which is particularly relevant for multiple cooperating robotic platforms employed for a GSL problem. Secondly, the adjoint-state method presents itself as a well-suited strategy for addressing general inverse problems subjected to PDE constraints. This quality makes it applicable across a broad spectrum of PDEs, encompassing more realistic scenarios that involve time-dependency and advection.

A. NOTATION

Throughout the paper vectors are represented as boldface lowercase letters, e.g., $\boldsymbol{x}$, and matrices as boldface uppercase letters, e.g., $\boldsymbol{X}$. Their transpose is denoted by $(\cdot)^\mathsf{T}$. We write $[\boldsymbol{X}]_l$ to denote an lth column of a matrix $\boldsymbol{X}$. The expression $\mathrm{diag}(\boldsymbol{x})$ stands for a diagonal matrix with the elements of $\boldsymbol{x}$ on the main diagonal. The expectation operator is denoted by $\mathbb{E}\{\cdot\}$, or $\mathbb{E}_q\{\cdot\}$ when the context requires to explicitly state the probability density function (pdf) q with respect to which expectation is taken. We denote the pdf of a Gaussian random vector with expectation $\boldsymbol{a} = \mathbb{E}\{\boldsymbol{x}\}$ and covariance matrix $\boldsymbol{B} = \mathbb{E}\{(\boldsymbol{x}-\boldsymbol{a})(\boldsymbol{x}-\boldsymbol{a})^\mathsf{T}\}$ as $\mathrm{N}(\boldsymbol{x}|\boldsymbol{a},\boldsymbol{B})$. We will use the notation $\delta_{\boldsymbol{x}}(\Omega)$ to specify a (multidimensional) Dirac measure over domain Ω with a support at $\boldsymbol{x} \in \Omega$; as an example for a 1D real line we have $\delta_a(\mathbb{R}) \equiv \delta(x-a)$ where $\forall(x,a) \in \mathbb{R}$, which is an often used alternative notation for Dirac measures on $\mathbb{R}$.

II. SIGNAL MODEL

Let us consider a GSL problem over some d-dimensional polygonal exploration area $\Omega \subset \mathbb{R}^d$. We assume that the spatial gas concentration over Ω can be described by a time-invariant diffusion PDE (also known as Poisson's equation):

$$-\kappa \Delta f(\boldsymbol{x}) = \sum_{i=1}^{K} w_i \delta_{\boldsymbol{\theta}_i}(\Omega), \quad \boldsymbol{x} \in \Omega, \tag{1}$$

$$\text{s.t.} \quad f(\boldsymbol{x}) = 0, \ \boldsymbol{x} \in \partial\Omega, \tag{2}$$

where κ is a diffusion coefficient, $f(\boldsymbol{x})$ is a spatial gas concentration intensity, and Δ is a Laplace operator. The right-hand side (RHS) of (1) defines a superposition of K static gas sources, each with an amplitude (or release rate) w_i, located at $\boldsymbol{\theta}_i \in \Omega$. (1) is augmented with boundary condition (2) that defines concentration values at the boundary $\partial\Omega$ of the domain Ω. We consider an open boundary $\partial\Omega$, which allows concentration to flow off (Dirichlet boundary condition). The model (1)-(2) represents a steady-state gas distribution in the absence of wind. As such, it is a special case of a more general advection-diffusion equation. Although simple, it is sufficient for outlining the proposed method by abstracting from realistic gas propagation effects.

A. SURROGATE PROCESS MODEL

The model (1) states that the concentration distribution $f(\boldsymbol{x})$ is governed by a few distinct gas sources. In practice the number of sources K, as well as the source parameters w_i and $\boldsymbol{\theta}_i$, $i = 1, \ldots, K$, are rarely known and have to be estimated from the measurements. A joint estimator of these parameters would lead to a numerically challenging combinatorial problem. To circumvent this, we approximate (1) with a surrogate model

$$-\kappa \Delta f(\boldsymbol{x}) = \sum_{l=1}^{L} w_l \delta_{\boldsymbol{\theta}_l}(\Omega), \quad \boldsymbol{x} \in \Omega \tag{3}$$

[1] This depends on whether the finite element or finite difference method is used for discretizing the PDE.

where $L > K$ and some of the weights w_l, $l \in \mathcal{L} \triangleq \{1, \ldots, L\}$, are zero. In other words, the source weights are assumed to be sparse. Note that although at first glance the difference between the model (1) and the surrogate (3) is minimal, it is has a major consequence for the estimation approach discussed later: in the former K is unknown, while L is known and fixed in the latter. Such an approach is also sometimes referred to as *max-search* approach [36].

Assume for a moment that both w_l and $\boldsymbol{\theta}_l, l \in \mathcal{L}$ are known. Then, the concentration $f(\boldsymbol{x})$ can be obtained by solving the forward problem. The approach we use in this work considers a solution for $f(\boldsymbol{x})$ using the method of Green's functions [37], [38]. The latter is appealing since the RHS of (3) is a linear combination of Dirac measures. Specifically, the method states that the solution $f(\boldsymbol{x})$ of (3) can be founds as

$$f(\boldsymbol{x}) = \int_{\Omega} G(\boldsymbol{x}, \boldsymbol{\theta}) \sum_{l=1}^{L} w_l \delta_{\boldsymbol{\theta}_l}(\Omega) \mathrm{d}\boldsymbol{\theta} = \sum_{l=1}^{L} w_l G(\boldsymbol{x}, \boldsymbol{\theta}_l) \quad (4)$$

where $G(\boldsymbol{x}, \boldsymbol{\theta})$ is a Green's function defined as a solution to

$$\begin{aligned} &-\kappa \Delta G(\boldsymbol{x}, \boldsymbol{\theta}) = \delta_{\boldsymbol{\theta}}(\Omega), \quad \boldsymbol{x}, \boldsymbol{\theta} \in \Omega \\ &\text{s.t.} \quad G(\boldsymbol{x}, \boldsymbol{\theta}) = 0, \, \boldsymbol{x} \in \partial\Omega. \end{aligned} \quad (5)$$

In other words, $G(\boldsymbol{x}, \boldsymbol{\theta})$ is a "response" of the equation to a single source at the location $\boldsymbol{\theta}$. We will assume that the solution to (5) exists and can be determined (see e.g., [38]). In general, Green's functions, as required for our purposes, can be approximated with e.g., a neural network [39]. We, however, leave the problem of finding or approximating Green's functions outside the scope of this work.

Now, consider a discretization of (3). To this end we partition Ω into N smaller subdomains or cells, which form a (not necessarily regular) grid. For each cell with center coordinates $\boldsymbol{x}_i$, $i \in \mathcal{N} \triangleq \{0, \ldots, N-1\}$, we then assume a constant concentration value $f(\boldsymbol{x}_i) = \text{const}$. This discretization can be interpreted as finite difference approximation of $f(\boldsymbol{x})$. The corresponding concentrations are aggregated into a vector $\boldsymbol{f} \triangleq [f(\boldsymbol{x}_0), \ldots, f(\boldsymbol{x}_{N-1})]^\mathsf{T}$. Also, define a discretization of Green's function as follows:

$$\boldsymbol{G}(\boldsymbol{\Theta}) \triangleq \begin{bmatrix} G(\boldsymbol{x}_0, \boldsymbol{\theta}_1) & \cdots & , G(\boldsymbol{x}_0, \boldsymbol{\theta}_L) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{x}_{N-1}, \boldsymbol{\theta}_1) & \cdots & , G(\boldsymbol{x}_{N-1}, \boldsymbol{\theta}_L) \end{bmatrix}$$

where $\boldsymbol{\Theta} \triangleq [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L]$. This allows us to rewrite (4) in a matrix vector form as

$$\boldsymbol{f} = \boldsymbol{G}(\boldsymbol{\Theta})\boldsymbol{w} \quad (6)$$

with $\boldsymbol{w} \triangleq [w_1, \ldots, w_L]^\mathsf{T}$.

Let us point out that the method we will propose can easily account for other discretization approaches, such as finite elements, to represent $f(\boldsymbol{x})$. Also, despite the fact that we discretize Ω, the location of the sources in the model (6) are not restricted to this grid and can take arbitrary values.

Assume now that M noisy samples of the concentration $f(\boldsymbol{x})$ have been obtained at locations $\boldsymbol{x}_m$, $m \in \mathcal{M} \triangleq \{1, \ldots, M\}$. Without loss of generality we will assume that $N \gg M$ and that $\{\boldsymbol{x}_m, m \in \mathcal{M}\}$ is a subset of discretizaton cells. Note that we can always select such a discretization of Ω that will include locations where measurements are taken. Now collect the measurements in a vector $\boldsymbol{z} \in \mathbb{R}^M$ as

$$\boldsymbol{z} = \boldsymbol{M}\boldsymbol{f} + \boldsymbol{\xi} \quad (7)$$

where $\boldsymbol{M} \in \mathbb{R}^{M \times N}$ is a selection matrix that selects elements of $\boldsymbol{f}$ that correspond to the cells where measurement are taken, and $\boldsymbol{\xi}$ is a random, zero-mean Gaussian perturbation with precision matrix $\lambda_\xi \boldsymbol{I}$ and some $\lambda_\xi > 0$. Without loss of generality we will assume that all measurements $\boldsymbol{z}$ are collected simultaneously at different locations of the domain Ω.

Our goal now is to use $\boldsymbol{z}$ to estimate a sparse vector $\boldsymbol{w}$, locations $\boldsymbol{\Theta}$, and recover $\boldsymbol{f}$ from (6). To this end we pursue a Bayesian approach towards parameter estimation as detailed in the following.

B. BAYESIAN FORMULATION OF THE INFERENCE MODEL

To cast the estimation problem into a probabilistic framework, consider the following posterior pdf of the variables of interest:

$$p(\boldsymbol{f}, \boldsymbol{w}, \boldsymbol{\Theta}|\boldsymbol{z}) \propto p(\boldsymbol{z}|\boldsymbol{f}) p(\boldsymbol{f}|\boldsymbol{w}, \boldsymbol{\Theta}) p(\boldsymbol{w}) p(\boldsymbol{\Theta}), \quad (8)$$

where we explicitly assume that source locations $\boldsymbol{\Theta}$ and the corresponding weights $\boldsymbol{w}$ are independent. Let us discuss the factors in (8) in more detail.

Based on (7) we immediately see that $p(\boldsymbol{z}|\boldsymbol{f}) = \mathrm{N}(\boldsymbol{z}|\boldsymbol{M}\boldsymbol{f}, \lambda_\xi^{-1}\boldsymbol{I})$. The pdf $p(\boldsymbol{f}|\boldsymbol{w}, \boldsymbol{\Theta})$ represents the relationship between sources and concentration $\boldsymbol{f}$ based on (6). This deterministic relationship can be represented with a Dirac distribution $p(\boldsymbol{f}|\boldsymbol{w}, \boldsymbol{\Theta}) = \delta_{\boldsymbol{G}(\boldsymbol{\Theta})\boldsymbol{w}}(\mathbb{R}^N)$. The other two terms $p(\boldsymbol{\Theta})$ and $p(\boldsymbol{w})$ are the prior distributions of the source parameters. Concerning the former we will assume a uniform prior over Ω, i.e., $p(\boldsymbol{\Theta}) \propto \text{const}$. In case of $p(\boldsymbol{w})$ we instead employ a modeling approach used in SBL.

The basic idea of SBL is to assign an appropriate prior to the L-dimensional vector $\boldsymbol{w}$ such that the resulting MAP estimate $\widehat{\boldsymbol{w}}$ is sparse, i.e. many of its entries are zero. Typically, SBL specifies a hierarchical factorable prior $p(\boldsymbol{w}|\boldsymbol{\alpha})p(\boldsymbol{\alpha}) = \prod_{l=1}^{L} p(w_l|\alpha_l)p(\alpha_l)$, where $p(w_l|\alpha_l) = \mathrm{N}(w_l|0, \alpha_l^{-1})$, $l \in \mathcal{L}$ [28], [40], [41]. For each $l \in \mathcal{L}$ the hyperparameter $\alpha_l \geq 0$, also called sparsity parameter, regulates the "width" of $p(w_l|\alpha_l)$; the product $p(w_l|\alpha_l)p(\alpha_l)$ defines a Gaussian scale mixture.[2] Note that the sparsity parameters $\boldsymbol{\alpha}$ encode the support of the weights $\boldsymbol{w}$: large entries in $\boldsymbol{\alpha}$ will drive the corresponding posterior weight estimate towards zero, thus ensuring a sparse posterior mode.

Bayesian inference on a linear model with such a hierarchical prior is commonly realized via two types of techniques:

[2] In [42] the authors extend the framework by generalizing $p(w_l|\alpha_l)$ to be the pdf of a power exponential distribution, which makes the hierarchical prior a power exponential scale mixture distribution.

MAP estimation of $\boldsymbol{w}$ (Type I estimation[3]) and MAP estimation of $\boldsymbol{\alpha}$ (Type II estimation). Type II estimation commonly demonstrates improved performance in contrast to Type I estimation. A contributing factor to this superiority lies in the nature of the objective function associated with Type II estimation: it usually possesses a notably reduced number of local minima compared to its corresponding Type I counterpart, which in turn promotes better sparsity estimation [42], [46], [47].

Popular selections of the hyperpriors $p(\alpha_l)$, $l \in \mathcal{L}$, are Gamma [28], [48], [49], non-informative, i.e., $p(\alpha_l) \propto \alpha_l^{-1}$, [28], [50], and uniform, i.e. $p(\alpha) \propto 1$ [47], [51], [52]. Inference methods using the latter are called evidence maximization procedures [28], [29], [53]. Their virtue compared to other methods relying on other hyperprior selections is twofold [42]: they typically demonstrate superior (or similar) performance and very efficient implementations exist that can be analyzed [47], [51], [52], [54]. In the following we will exploit the third choice for the hyperprior.

With the sparsity parameters $\boldsymbol{\alpha}$, the resulting parameter posterior pdf becomes

$$p(\boldsymbol{f}, \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\Theta}|z) \propto p(z|\boldsymbol{f})p(\boldsymbol{f}|\boldsymbol{w}, \boldsymbol{\Theta})p(\boldsymbol{w}, \boldsymbol{\alpha})p(\boldsymbol{\Theta}). \tag{9}$$

Note that in (9) $\boldsymbol{f}$ is treated as a random variable, yet due to the form of $p(\boldsymbol{f}|\boldsymbol{w}, \boldsymbol{\Theta})$ it is actually deterministically dependent on $\boldsymbol{w}$ and $\boldsymbol{\Theta}$. This fact will be used in the proposed algorithm to maximize (9), which we discuss in the following.

III. SUPER-RESOLUTION SBL FOR GAS SOURCE LOCALIZATION PROBLEM

In order to find parameters of interest $\boldsymbol{f}$, $\boldsymbol{w}$, $\boldsymbol{\alpha}$ and $\boldsymbol{\Theta}$ we propose an approximative optimization scheme that maximizes (9) by alternating between (i) estimation of the support vector $\boldsymbol{\alpha}$ and sparse vector $\boldsymbol{w}$ under assumption that $\boldsymbol{\Theta}$ is fixed, and (ii) estimation of $\boldsymbol{\Theta}$ and $\boldsymbol{w}$ with the source support $\boldsymbol{\alpha}$ fixed. Note that at each of these steps the concentration $\boldsymbol{f}$ can be recovered from (6) using current estimates of $\boldsymbol{\Theta}$ and $\boldsymbol{w}$.

A. SOURCE SUPPORT ESTIMATION

Let us begin with estimation of $\boldsymbol{\alpha}$. To this end we assume that $\boldsymbol{\Theta}$ is fixed at an estimate $\widehat{\boldsymbol{\Theta}}$ and marginalize (9) over $\boldsymbol{f}$ which leads to

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}|z) &= \int p(\boldsymbol{f}, \boldsymbol{w}, \boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}|z)\mathrm{d}\boldsymbol{f} \\ &\propto p(z|\boldsymbol{w}, \widehat{\boldsymbol{\Theta}})p(\boldsymbol{w}|\boldsymbol{\alpha}), \end{aligned} \tag{10}$$

where

$$p(z|\boldsymbol{w}, \widehat{\boldsymbol{\Theta}}) \propto \exp\left(-\frac{\lambda_\xi}{2}\|z - \boldsymbol{M}\boldsymbol{G}(\widehat{\boldsymbol{\Theta}})\boldsymbol{w}\|^2\right).$$

[3]Note that many traditional "non-Bayesian" methods for learning sparse representations such as basis pursuit de-noising or re-weighted ℓ_p-norm regressions [43], [44], [45] can be interpreted as Type I estimation within the above Bayesian framework (see [42] for more details on this link).

The consequences of this step are the following. First, with $\widehat{\boldsymbol{\Theta}}$ fixed, we effectively linearize the estimation problem, as now parameters $\boldsymbol{w}$ enter the likelihood function linearly. Second, and most importantly, the form of the posterior (10) coincides with that used in a classical SBL. As such, we can reuse existing algorithms to maximize it.

Specifically, we will use a Fast Marginal Likelihood Maximization (FMLM) method to maximize $p(\boldsymbol{w}, \boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}|z)$ [52]. The approach is two-fold: first we consider the factorization $p(\boldsymbol{w}, \boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}|z) = p(\boldsymbol{w}|\boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}, z)p(\boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}|z)$. The second factor is a key to the Type II SBL support estimation: it considers the joint $p(\boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}|z) \propto p(z|\boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}})$ and computes a Maximum Likelihood estimate $\widehat{\boldsymbol{\alpha}}$ as a maximizer of Type II likelihood function [28]

$$p(z|\boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}) = \int p(z|\boldsymbol{w}, \widehat{\boldsymbol{\Theta}})p(\boldsymbol{w}|\boldsymbol{\alpha})\mathrm{d}\boldsymbol{w} \tag{11}$$

$$\propto |\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}(\widehat{\boldsymbol{\Theta}})|^{-\frac{1}{2}}\mathrm{e}^{-\frac{1}{2}z^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}(\widehat{\boldsymbol{\Theta}})^{-1}z}, \tag{12}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}(\widehat{\boldsymbol{\Theta}}) \triangleq \lambda_\xi^{-1}\boldsymbol{I} + \boldsymbol{M}\boldsymbol{G}(\widehat{\boldsymbol{\Theta}})\boldsymbol{A}^{-1}\boldsymbol{G}(\widehat{\boldsymbol{\Theta}})^{\mathsf{T}}\boldsymbol{M}^{\mathsf{T}}$, and $\boldsymbol{A} \triangleq \mathrm{diag}(\boldsymbol{\alpha})$. Define now negative log-likelihood function

$$\begin{aligned} \ell(\boldsymbol{\alpha}) &\triangleq -\log p(z|\boldsymbol{\alpha}, \widehat{\boldsymbol{\Theta}}) \\ &= \frac{1}{2}\log|\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}(\widehat{\boldsymbol{\Theta}})| + \frac{1}{2}z^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\alpha}}(\widehat{\boldsymbol{\Theta}})^{-1}z. \end{aligned} \tag{13}$$

Obviously, the minimizer of (13) also maximizes (12). The FMLM algorithm minimizes $\ell(\boldsymbol{\alpha})$ in a coordinate-wise descent manner, one component at a time, as follows. First, the vector $\boldsymbol{\alpha}$ is partitioned into two sets: in a single component α_l and the other entries $\boldsymbol{\alpha}_{\bar{l}}$. It was shown [52] that $\ell(\boldsymbol{\alpha})$ can be then partitioned as

$$\begin{aligned} \ell(\boldsymbol{\alpha}, \alpha_l) &= \mathrm{const}(\boldsymbol{\alpha}_{\bar{l}}) \\ &\quad + \frac{1}{2}\left[\log(\alpha_l) - \log\left(\alpha_l + \varsigma_l^{-1}\right) + \frac{\mu_l^2\varsigma_l^{-2}}{\alpha_l + \varsigma_l^{-1}}\right] \end{aligned} \tag{14}$$

where $\mathrm{const}(\boldsymbol{\alpha}_{\bar{l}})$ are the terms independent of α_l,

$$\varsigma_l \triangleq (\boldsymbol{g}_l^{\mathsf{T}}\boldsymbol{\Psi}_l^{-1}\boldsymbol{g}_l)^{-1}, \quad \mu_l \triangleq \varsigma_l\boldsymbol{g}_l^{\mathsf{T}}\boldsymbol{\Psi}_l^{-1}z, \tag{15}$$

with $\boldsymbol{g}_l \triangleq [\boldsymbol{M}\boldsymbol{G}(\widehat{\boldsymbol{\Theta}})]_l$ and

$$\boldsymbol{\Psi}_l \triangleq \lambda_\xi^{-1}\boldsymbol{I} + \sum_{m\in\mathcal{L};m\neq l}\alpha_m^{-1}\boldsymbol{g}_m\boldsymbol{g}_m^{\mathsf{T}}.$$

The minimum of $\ell(\boldsymbol{\alpha}, \alpha_l)$ with respect to α_l can now be found in the closed form at

$$\widehat{\alpha}_l = \begin{cases} (|\mu_l|^2 - \varsigma_l)^{-1} & \frac{|\mu_l|^2}{\varsigma_l} > 1 \\ \infty & \text{otherwise,} \end{cases} \tag{16}$$

The FMLM algorithm evaluates the expressions in (16) with l ranging in $\mathcal{L}$, in a round-robin fashion, until some stopping criterion is met.

Given $\widehat{\boldsymbol{\alpha}}$ that minimizes $\ell(\boldsymbol{\alpha})$, the FMLM procedure then infers a sparse weight vector $\boldsymbol{w}$ from the conditional marginal posterior pdf $p(\boldsymbol{w}|\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\Theta}}, z) \propto p(z|\boldsymbol{w}, \widehat{\boldsymbol{\Theta}})p(\boldsymbol{w}|\widehat{\boldsymbol{\alpha}})$. This pdf is Gaussian with mean $\widehat{\boldsymbol{w}}$ and covariance matrix $\widehat{\boldsymbol{\Sigma}}_w$ given by

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}_w &= \left(\lambda_\xi \boldsymbol{G}(\widehat{\boldsymbol{\Theta}})^{\mathsf{T}}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{G}(\widehat{\boldsymbol{\Theta}}) + \widehat{\boldsymbol{A}}\right)^{-1},\\ \widehat{\boldsymbol{w}} &= \lambda_\xi \widehat{\boldsymbol{\Sigma}}_w \boldsymbol{G}(\widehat{\boldsymbol{\Theta}})^{\mathsf{T}}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{z},\end{aligned} \tag{17}$$

where $\widehat{\boldsymbol{A}} \triangleq \operatorname{diag}(\widehat{\boldsymbol{\alpha}})$. Note that sources, i.e., the columns of $\boldsymbol{G}(\widehat{\boldsymbol{\Theta}})$, associated with infinite sparsity parameters α_l, $l \in \mathcal{L}$, can be safely removed from the model. Indeed, for a sparsity parameter $\alpha_l = \infty$, the corresponding lth column and row of $\widehat{\boldsymbol{\Sigma}}_w$, as well as the lth entry of $\widehat{\boldsymbol{w}}$ will be zero. It thus makes sense to reduce the dimensionality of (13) (and of (17)) by removing the corresponding contributions. As a consequence, the matrix $\widehat{\boldsymbol{A}}$ will include only finite entries, corresponding to sources with non-zero weights. The number of finite sparsity parameters thus indirectly provides an estimate of the number of "active" sources $\widehat{L}$.

B. LOCATION PARAMETER ESTIMATION

Once the support parameters $\widehat{\boldsymbol{\alpha}}$ are found, we fix them and maximize the posterior $p(\boldsymbol{f}, \boldsymbol{w}, \widehat{\boldsymbol{\alpha}}, \boldsymbol{\Theta}|z)$ with respect to $\boldsymbol{\Theta}$ and $\boldsymbol{w}$. Note that while dependency on $\boldsymbol{w}$ is linear[4], the dependency of the posterior on location parameters is nonlinear, which requires the use of numerical optimization techniques. One approach to address this is to discretize $\boldsymbol{\Theta}$, as in e.g., [33], [55], where the right-hand side of (3) is discretized over Ω. This naturally simplifies the whole problem, yet at the expense of a larger parameter space, which can be significantly higher than N depending on a desired accuracy. We, therefore, refrain from a discretization of $\boldsymbol{\Theta}$ and estimate these parameters using gradient-descent based techniques [34].

To this end we define the following cost function:

$$J(\boldsymbol{f}, \boldsymbol{\Theta}) \triangleq -\log p(z|\boldsymbol{f}) = \frac{\lambda_\xi}{2}\|z - \boldsymbol{M}\boldsymbol{f}\|^2 \tag{18}$$

$$\text{s.t.} \quad g(\boldsymbol{f}, \boldsymbol{\Theta}) \triangleq \boldsymbol{f} - \boldsymbol{G}(\boldsymbol{\Theta})\boldsymbol{w} = \boldsymbol{0}, \quad \boldsymbol{\Theta} \in \Omega. \tag{19}$$

Given some estimate of $\widehat{\boldsymbol{\Theta}}^{[i]}$ at the iteration i, we aim to minimize $J(\boldsymbol{f}, \boldsymbol{\Theta})$ by following its gradient:

$$\widehat{\boldsymbol{\Theta}}^{[i+1]} = \widehat{\boldsymbol{\Theta}}^{[i]} - \rho \nabla_{\boldsymbol{\Theta}} J(\boldsymbol{f}, \widehat{\boldsymbol{\Theta}}^{[i]}) \tag{20}$$

where $\nabla_{\boldsymbol{\Theta}} J(\boldsymbol{f}, \widehat{\boldsymbol{\Theta}}^{[i]})$ is a gradient of (18) with respect to $\boldsymbol{\Theta}$ evaluated at $\widehat{\boldsymbol{\Theta}}^{[i]}$ and ρ is an appropriately chosen step size. We will discuss the choice of the latter further in the text. For now let us focus on the gradient term $\nabla_{\boldsymbol{\Theta}} J(\boldsymbol{f}, \widehat{\boldsymbol{\Theta}}^{[i]})$. To compute it we do not integrate $\boldsymbol{f}$ out as we did previously, but instead use it as an intermediate variable to simplify the gradient computation with respect to $\boldsymbol{\Theta}$. To be more specific, we use an adjoint state method [6], [35] to compute $\nabla_{\boldsymbol{\Theta}} J(\boldsymbol{f}, \widehat{\boldsymbol{\Theta}}^{[i]})$.

Let $\boldsymbol{\eta}$ be the adjoint variable defined as a solution to

$$\left(\frac{\partial g(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{f}}\right)^{\mathsf{T}} \boldsymbol{\eta} = -\left(\frac{\partial J(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{f}}\right)^{\mathsf{T}}. \tag{21}$$

It is rather straightforward to show that

$$\boldsymbol{\eta} = -\lambda_\xi \boldsymbol{M}^{\mathsf{T}}(\boldsymbol{M}\boldsymbol{f} - z). \tag{22}$$

Then, we can evaluate the gradient $\nabla_{\boldsymbol{\Theta}} J(\boldsymbol{f}, \boldsymbol{\Theta})$ as

$$\nabla_{\boldsymbol{\Theta}} J(\boldsymbol{f}, \boldsymbol{\Theta}) = \boldsymbol{\eta}^{\mathsf{T}} \frac{\partial g(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} \tag{23}$$

with

$$\frac{\partial g(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = \left[\frac{\partial g(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{\theta}_1}, \ldots, \frac{\partial g(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{\theta}_L}\right]^T. \tag{24}$$

Taking the structure of $\boldsymbol{G}(\boldsymbol{\Theta})$ into account, it is easy to see that

$$\frac{\partial g(\boldsymbol{f}, \boldsymbol{\Theta})}{\partial \boldsymbol{\theta}_l} = -\left[\widehat{w}_l^{[i]} \frac{\partial G(\boldsymbol{x}_0, \boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l}, \ldots, \widehat{w}_l^{[i]} \frac{\partial G(\boldsymbol{x}_{N-1}, \boldsymbol{\theta}_l)}{\partial \boldsymbol{\theta}_l}\right]^{\mathsf{T}}$$

for $l \in \mathcal{L}$. Note that the latter requires an estimate of the source weight $\widehat{w}_l^{[i]}$, which is computed from (17) using location estimate $\widehat{\boldsymbol{\Theta}}^{[i]}$ at which the gradient is computed, and a fixed support estimate $\widehat{\boldsymbol{\alpha}}$. The update (20) is then evaluated until some suitable convergence criterion is satisfied.

To ensure fast convergence of the gradient update (20) it is necessary to appropriately select the step size parameter ρ. Using a fixed small step size can result in a slow convergence; on the other hand, the gradient descent can be combined with a back-tracking schemes to line search for an optimal ρ based on e.g., Armijo–Goldstein rule. This, however, can also be time-consuming. In our implementation we used a limited memory Broyden–Fletcher–Goldfarb–Shanno algorithm for bound constrained optimization (L-BFGS-B) [56], which can be used to accelerate convergence by numerically approximating the Hessian matrix from a gradient, while also ensuring that location estimates remain in Ω.

C. ALGORITHM IMPLEMENTATION DETAILS

We are now ready to summarize the key steps of the proposed algorithm, to which we will refer as Super-resolution Gas Source Localization (SR-GSL). We begin with the initialization, followed by the implementation pseudo-code.

1) ALGORITHM INITIALIZATION

For initialization we first partition the location space Ω uniformly into a certain number $\widetilde{L}$ of grid points $\boldsymbol{\vartheta}_l$, $l = 1, \ldots, \widetilde{L}$; typically, selecting $\widetilde{L} \approx N$ suffices in practice. This partitioning will allow evaluating Green's functions at these locations

[4] Given $\widehat{\boldsymbol{\alpha}}$ and $\widehat{\boldsymbol{\Theta}}$, the weights $\boldsymbol{w}$ can always be computed from (17).

Algorithm 1: Initialization of the Support $\boldsymbol{\alpha}$.

1: Input: $\boldsymbol{R}, z, \boldsymbol{M}$
2: Initialize: $y_l \leftarrow \frac{1}{L}\text{trace}(\lambda_\xi \boldsymbol{R}^\mathsf{T}\boldsymbol{M}^\mathsf{T}\boldsymbol{M}\boldsymbol{R}),\ l = 1, \ldots, \widetilde{L}$
3: Solve for $\widehat{\boldsymbol{w}}$:

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \frac{\lambda_\xi}{2}\|z - \boldsymbol{M}\boldsymbol{R}\boldsymbol{w}\|^2 + \sum_{i=1}^{L'} \sqrt{y_i}|w_i| \quad (26)$$

4: Compute initial support $\boldsymbol{\alpha}$:

$$\widehat{\alpha}_l = \frac{\sqrt{y_l}}{|\widehat{w}_l|}, \quad l = 1, \ldots, L'$$

5: Compute weight covariance $\widehat{\boldsymbol{\Sigma}}_w$ as

$$\widehat{\boldsymbol{\Sigma}}_w = \left(\lambda_\xi \boldsymbol{R}^\mathsf{T}\boldsymbol{M}^\mathsf{T}\boldsymbol{M}\boldsymbol{R} + \widehat{\boldsymbol{A}}\right)^{-1}$$

as

$$\boldsymbol{R} \triangleq \begin{bmatrix} G(\boldsymbol{x}_0, \boldsymbol{\vartheta}_1) & \cdots & , G(\boldsymbol{x}_0, \boldsymbol{\vartheta}_{\widetilde{L}}) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{x}_{N-1}, \boldsymbol{\vartheta}_1) & \cdots & , G(\boldsymbol{x}_{N-1}, \boldsymbol{\vartheta}_{\widetilde{L}}) \end{bmatrix}.$$

This results in initial source locations – the initial search space. The next step is initialization of the support for this locations set. The approach we found to work most efficiently in the considered setting is based on the reformulated Automatic Relevance Determination (R-ARD) algorithm [47] – a version of SBL that maximizes the marginal likelihood in (12) (or equivalently minimizes $\ell(\boldsymbol{\alpha})$ in (13)) via a solution to a sequence of reweighted Least Absolute Shrinkage and Selection Operator (LASSO) [57] problems.

Specifically, in [47] the log-likelihood $\ell(\boldsymbol{\alpha})$ is optimized with respect to the variance parameters $\boldsymbol{\gamma} = \boldsymbol{\alpha}^{-1}$, where the inversion is applied element-wise. The resulting cost function is then upper bounded as

$$\ell(\boldsymbol{\gamma}) \leq \boldsymbol{y}^\mathsf{T}\boldsymbol{\gamma} - g^*(\boldsymbol{y}) + \boldsymbol{z}^\mathsf{T}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}\boldsymbol{z} \triangleq \ell(\boldsymbol{\gamma}, \boldsymbol{y}), \quad (25)$$

where $\boldsymbol{y} \triangleq [y_1, \ldots, y_L]^T$ is a dual variable and $g^*(\boldsymbol{y})$ is the concave conjugate of $\log|\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}|$ defined by the duality relationship $g^*(\boldsymbol{y}) = \min_{\boldsymbol{\gamma}} \boldsymbol{y}^T\boldsymbol{\gamma} - \log|\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}|$ [58]. Here $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \triangleq \lambda_\xi^{-1}\boldsymbol{I} + \boldsymbol{M}\boldsymbol{R}\boldsymbol{\Gamma}\boldsymbol{R}^\mathsf{T}\boldsymbol{M}^\mathsf{T}$, and $\boldsymbol{\Gamma} \triangleq \text{diag}(\boldsymbol{\gamma})$. The upper bounding function $\ell(\boldsymbol{\gamma}, \boldsymbol{y})$ is then minimized first with respect to the dual variable $\boldsymbol{y}$ with fixed $\boldsymbol{\gamma}$, and then with respect to $\boldsymbol{\gamma}$ with newly updated dual parameters $\boldsymbol{y}$ fixed. The latter can be cast as a LASSO problem and solved very efficiently. For more details on R-ARD we refer the reader to [47].

While the R-ARD algorithm implements the updates of $\boldsymbol{\gamma}$ and $\boldsymbol{y}$ multiple times, for the initialization we do this only once to compute an initial estimate of the support $\boldsymbol{\alpha} \equiv \boldsymbol{\gamma}^{-1}$. We found that this results in a numerically stable initialization of the algorithm across different parameters setting. The corresponding steps are now summarized for our particular problem in Algorithm 1.

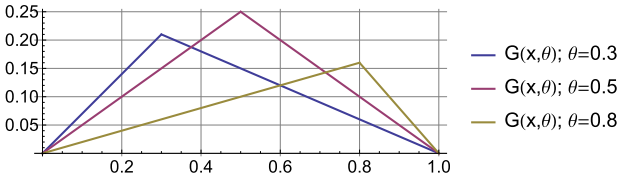


FIG. 1. Evaluated Green's functions for different values of the location parameter θ and $a = 1$.

The computation of the weights in (26) requires solving a convex, ℓ_l-norm constrained optimization problem; it is likely that this step has a stabilizing effect on the initial computation of the source support. In our implementation we used CVXPY library to solve (26)[5]

2) ALGORITHM STRUCTURE

Let us now outline the global structure of the SR-GSL algorithm. The main steps are summarized in Algorithm 2.

Essentially, the algorithm includes two iterative updates: in lines 3 - 7 a gradient-based location parameter estimation is realized. Lines 8 - 12 implement the FMLM algorithm for support estimation. These two "internal" iterations are repeated until some convergence criterion is met. In our implementation we consider these update loops as converged when the relative change of location parameters $\boldsymbol{\Theta}$ and sparsity parameters $\boldsymbol{\alpha}$ is below 1%. The alternations between support estimate and location updates – the outer iteration loop – are repeated either for a fixed number of runs, or when the parameter changes between outer iterations are also below 1%.

IV. SIMULATION RESULTS

To simplify the analysis of the method and demonstrate its performance we will consider a Poisson's equation in 1D, setting $\Omega = [0, a]$. Without loss of generality we will also set the diffusion coefficient to $\kappa = 1$ and $a = 1$. In this case the Green's function of the corresponding equation can be computed in closed form as

$$G(x, \theta) = \begin{cases} \frac{x}{a}(a - \theta), & 0 \leq x \leq \theta, \\ -\frac{\theta}{a}(x - a), & \theta < x \leq a. \end{cases} \quad (27)$$

In Fig. 1 we show the evaluated Green's functions for location parameters θ set to 0.3, 0.5 and 0.8. It can be seen that the Green's function matrix $\boldsymbol{G}(\boldsymbol{\Theta})$ in this case, which is can also be referred to as a dictionary matrix, will be highly coherent, especially for closely spaced sources.

Let us now generate synthetic data for experiments. To generate the right-hand side of (1) we set $K = 3$ and compute $\theta_1 = 0.2 + 0.05a\epsilon_1$, and $\theta_i = \theta_{i-1} + 0.1 + 0.2a\epsilon_i$, $i = 2, \ldots, K$, where ϵ_i, $i = 1, \ldots, K$ are uniformly and independently drawn from a unit interval. Such selection ensures that generated impulses are well separated and also remain within

[5]The code can be found at https://www.cvxpy.org/.

Algorithm 2: SR-GSL Algorithm Structure.

1: Initialization: use Algorithm 1
2: **while** not converged **do**
3: *Update 1: Estimation of* $\widehat{\boldsymbol{\Theta}}$ *with* $\boldsymbol{\alpha}$ *fixed*
4: **while** $\boldsymbol{\Theta}$ has not converged **do**
5: Update $\widehat{\boldsymbol{w}}$ from (17) with $\boldsymbol{\alpha}$ fixed
6: Update $\boldsymbol{\Theta}$ using (20) : *Gradient update*
7: **end while**
8: *Update 2: Estimation of* $\widehat{\boldsymbol{\alpha}}$ *with* $\boldsymbol{\Theta}$ *fixed*
9: **while** $\boldsymbol{\alpha}$ has not converged **do**
10: Update $\boldsymbol{\alpha}$ using (16) : *FMLM algorithm*
11: Remove sources with $\alpha_l = \infty$, $l \in \mathcal{L}$
12: **end while**
13: **end while**

interval $[0, a]$. The amplitudes of the sources are set to $w_i = 1$, $\forall i = 1, \ldots, K$.

To generate the measurements we draw M random locations from the interval $[0, a]$. These locations are then augmented with $N - M$ additional regular sampling points in the interval $[0, a]$, which form the discretization grid for concentration $\boldsymbol{f}$. Such selection ensures that rows in the selection matrix $\boldsymbol{M}$ in (7) have 1's at column indices corresponding to the measurement locations. Then, using generated parameters θ_i, w_i, $i = 1, \ldots, K$, we compute the ground truth concentration $\boldsymbol{f}$ from (6) and generate measurements $\boldsymbol{z}$ using (7).

To benchmark the SR-GSL algorithm, we will compare its performance to sparse estimators applied to a linearized version of the GSL problem. To this end we will partition the location space Ω uniformly in a sufficiently large number $\widetilde{L} \gg N$ of grid points ϑ_l, $l = 1, \ldots, \widetilde{L}$ and reformulate (3) as

$$-\kappa \Delta f(\boldsymbol{x}) = \sum_{i=1}^{\widetilde{L}} w_i \delta_{\vartheta_i}(\Omega), \quad \boldsymbol{x} \in \Omega \tag{28}$$

$$\text{s.t.} \quad f(\boldsymbol{x}) = 0, \; \boldsymbol{x} \in \partial\Omega, \tag{29}$$

This partitioning will allow evaluating Green's functions at these locations as

$$\boldsymbol{R} \triangleq \begin{bmatrix} G(\boldsymbol{x}_0, \vartheta_1) & \cdots & , G(\boldsymbol{x}_0, \vartheta_{\widetilde{L}}) \\ \vdots & \ddots & \vdots \\ G(\boldsymbol{x}_{N-1}, \vartheta_1) & \cdots & , G(\boldsymbol{x}_{N-1}, \vartheta_{\widetilde{L}}) \end{bmatrix}$$

and express concentration $\boldsymbol{f} = \boldsymbol{R}\boldsymbol{w}$, which now becomes linear in weights $\boldsymbol{w}$. Thus, the GSL problem can be reduced to finding a sparse estimate of the weights $\boldsymbol{w}$. In simulations we will set $\widetilde{L} = 4N$, which should provide sufficiently accurate discretization of Ω, albeit at the expense of significantly increased problem size.

To find a sparse estimate we will use two algorithms that we will compare to SR-GSL. First, we will use a standard LASSO estimator [57] that finds $\boldsymbol{w}$ by solving the following optimization problem:

$$\min_{\boldsymbol{w}} \; J_{\text{lasso}}(\boldsymbol{w}) \triangleq \frac{1}{2}\lambda_\xi \|\boldsymbol{z} - \boldsymbol{M}\boldsymbol{R}\boldsymbol{w}\|^2 + \alpha_{\text{lasso}}\|\boldsymbol{w}\|_1 \tag{30}$$

A single regularization parameter α_{lasso} here regulates the amount of sparsity in the resulting estimate of $\boldsymbol{w}$. Its selection naturally impacts the number of estimated sources and thus requires some tuning.

To circumvent selection of the regularization parameter, a re-weighted version of the LASSO problem can be used, which we will use as a second benchmark algorithm. The re-weighting is realized by modifying (30) as

$$\min_{\boldsymbol{w}} \; J_{\text{rlasso}}(\boldsymbol{w}) \triangleq \frac{1}{2}\lambda_\xi \|\boldsymbol{z} - \boldsymbol{M}\boldsymbol{R}\boldsymbol{w}\|^2 + \sum_{i=1}^{\widetilde{L}} \alpha_{\text{rlasso},i}|w_i| \tag{31}$$

with multiple regularization parameters $\alpha_{\text{rlasso},i}$, $i = 1, \ldots \widetilde{L}$. In contrast to (30), the problem (31) has to be solved iteratively several times: first with fixed parameters α_{rlasso}, $i = 1, \ldots \widetilde{L}$, and then updating the latter following a certain rule, see e.g., [43], [44] and [42, Section II]. We will use as a re-weighting strategy the R-ARD algorithm [47] that we proposed for the initialization of SR-GSL. As we mentioned, the R-ARD finds a sparse estimate of $\boldsymbol{w}$ via a sequence of re-weighted LASSO problems, with the weights $\alpha_{\text{rlasso},i}$ selected as $\alpha_{\text{rlasso},i} = \sqrt{\eta_i}$, $i = 1, \ldots \widetilde{L}$, where

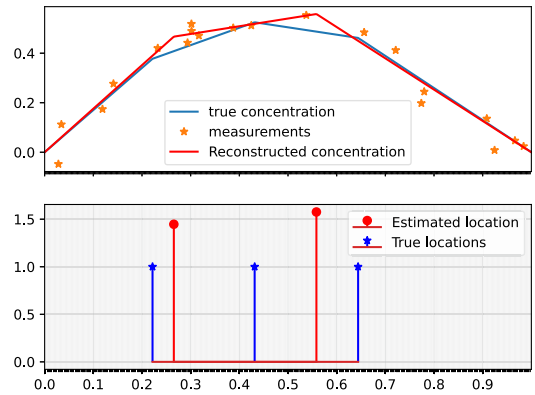
$$\boldsymbol{\eta} = \text{diag}\left(\boldsymbol{R}^{\mathsf{T}}\boldsymbol{M}^{\mathsf{T}}\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}\boldsymbol{M}\boldsymbol{R}\right), \tag{32}$$

$\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \triangleq \lambda_\xi^{-1}\boldsymbol{I} + \boldsymbol{M}\boldsymbol{R}\boldsymbol{\Gamma}\boldsymbol{R}^{\mathsf{T}}\boldsymbol{M}^{\mathsf{T}}$, $\boldsymbol{\Gamma} = \text{diag}(\boldsymbol{\gamma})$, and $\gamma_i = \eta_i^{-\frac{1}{2}}|w_i|$, $i = 1, \ldots, \widetilde{L}$ (see also [47] for more details).

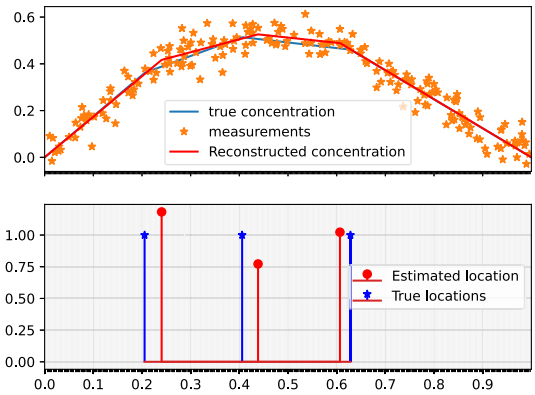
In the comparisons we will refer to the last two algorithms as LASSO and re-weighted LASSO (rLASSO), respectively. Note that neither LASSO nor re-weighted Least Absolute Shrinkage and Selection Operator (rLASSO) has explicit threshold mechanism, as realized in SR-GSL by (16). As such, the resulting estimates of $\widehat{\boldsymbol{w}}$ can include small, but numerically non-zero entries. To account for this we apply an additional threshold τ to the magnitude squared of the weights in order to remove such small components, i.e., when $|\widehat{w}_i|^2 < \tau$ we set $\widehat{w}_i = 0$ for $i = 1, \ldots \widetilde{L}$. Furthermore, for the LASSO algorithm we select the value of the regularization parameter α_{lasso} using cross-validation for different values of signal-to-noise ratio (SNR).[6]

Let us also mention that the computational complexity of both rLASSO and LASSO algorithms strongly depends on

[6]We have observed that the optimal value of α_{lasso} is almost independent of M.

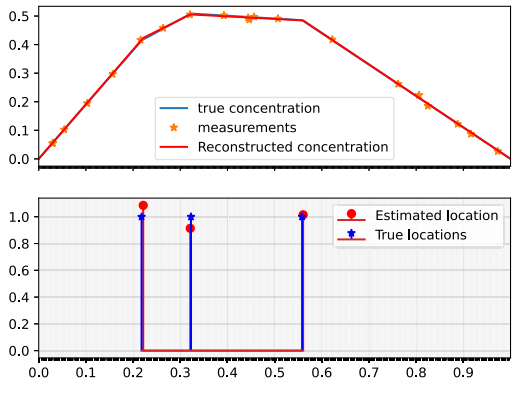


(a) $M = 20$ measurements

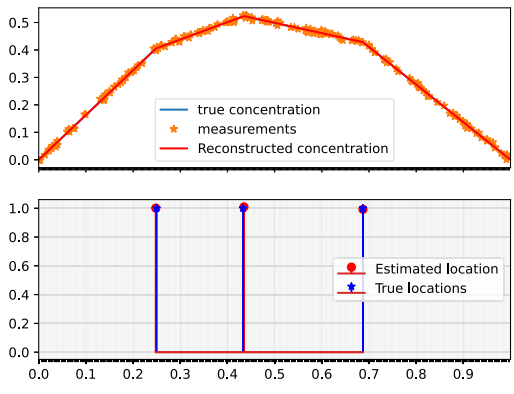


(b) $M = 200$ measurements

FIG. 2. Estimated concentration $f(x)$ and location parameters for SNR = 10 dB and different number of measurements M.



(a) $M = 20$ measurements



(b) $M = 200$ measurements

FIG. 3. Estimated concentration $f(x)$ and location parameters for SNR = 30 dB and different number of measurements M.

the selection of the parameter τ. Note that both variants include a solution to an ℓ_1 regularized problem (31) and (30) respectively, which is polynomial in $\widehat{L}$ (typically $\mathcal{O}(\widehat{L}^2)$ per iteration [59]) where $\widehat{L}$ is the number of active sources with non-zero weights. In case of rLASSO the problem (31) has to be solved multiple times following the R-ARD algorithm [47]. Thus, for low threshold τ, $\widehat{L}$ will tend to be high, increasing the computational complexity of the methods. Increasing the threshold τ, on the other hand, will lead to a possible omission of the sources. Thus, in general, τ remain a free parameter of LASSO and rLASSO, requiring a proper tuning.

We begin simulation studies by analyzing SR-GSL performance for some selected parameter configurations. We set $N = 400$ and run the algorithm with SNR set at 10 dB and 30 dB for $M = 20$ samples and $M = 200$ samples. The results are summarized in Figs. 2 and 3. It can be seen that in low SNR (Fig. 2) the proposed SR-GSL algorithm performs reasonably well, even when the number of measurements is low. Due to low SNR, the sources are found at slightly shifted locations, yet the estimated concentration values match closely true concentrations. When M is low, SR-GSL tends to be conservative, generally underestimating the number of sources. With increased SNR (Fig. 3) both the number and locations of the sources are accurately estimated even when the number of measurements is small.

However, we observed that the number of estimated components does vary over the range of SNR values and available measurements samples M. To study this variability we now run SR-GSL, rLASSO and LASSO algorithms for SNR set at 5 dB, 10 dB, 20 dB and 30 dB, and M varying between $M = 5$ and $M = 380$. In case of rLASSO and LASSO methods we also set the threshold for source detection at $\tau = 10^{-4}$ and $\tau = 10^{-9}$: higher threshold would thus result in sparser estimates. The corresponding results averaged over 500 Monte Carlo runs are shown in Fig. 4.

As we can notice, all algorithms on averaged tend to overestimate the number of sources. For SR-GSL it grows slightly with increasing M to an averaged estimate of 4 components; SR-GSL results in the sparsest estimate of the gas sources. It is closely followed by rLASSO with $\tau = 10^{-4}$, especially in low SNR regime, yet its sparsity reduces as SNR grows.

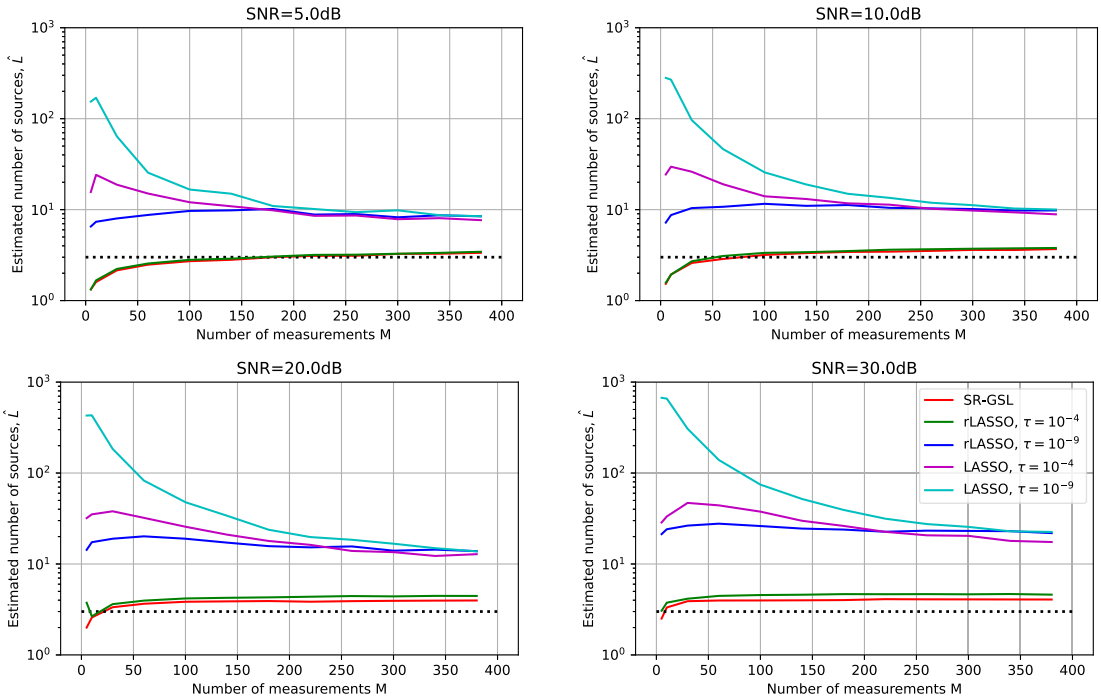


FIG. 4. Estimated averaged number of components $\widehat{L}$. The dotted line represent the true number of components $K = 3$.

Both LASSO versions as well as rLASSO with $\tau = 10^{-9}$ significantly overestimate the number of components. This implies that especially in high SNR regime additional (and quite numerous) artifact sources are recovered. Although with increasing M the number of artifacts drops, it does not approach the true number of components $K = 3$. Of course, the sparsity of rLASSO and LASSO methods can be controlled with appropriate choice of regularization parameters. Yet this requires choosing an optimal threshold for each selected parameter configuration. In contrast for SR-GSL this is done automatically via update (16), with variables μ_l and ς_l in (15) accounting for SNR, N as well as available number of measurements M.

We also see that both SR-GSL and rLASSO algorithms depend rather weakly on the number of measurement samples M. In the low SNR regime, 100–150 measurements are sufficient to determine the source support. In the high SNR regime only 20–50 measurements are sufficient. In case of LASSO this is not the case: the number of estimated sources depends strongly on M, especially with the low threshold of 10^{-9}, resulting in many artifacts. Moreover, with increasing SNR the performance of LASSO decreases, as more artifacts are discovered. This clearly demonstrates the overfitting effect of the LASSO algorithm.

We now compare the methods using the mean squared error (MSE) $\frac{1}{N}\|\widehat{\boldsymbol{f}} - \boldsymbol{f}_{\text{true}}\|^2$ between the estimated concentration $\widehat{\boldsymbol{f}}$ and the true concentration value $\boldsymbol{f}_{\text{true}}$. The latter is computed based on synthesized (true) location parameters θ_i, $i = 1, \ldots, K$. Additionally, we will compare the estimates of the sources using the Earth Mover's Distance (EMD) [60] metric. The latter is a discrete equivalent of the Wasserstein metric used as a distance between probability distributions and is particularly suited to measure distance between sparse signals. The EMD can thus provide an indication of how well the source locations – the signal support – as well as the source weights are estimated. MSE performance, on the other hand, reveals how accurately the concentration values are estimated. The results of the comparison averaged over 500 Monte-Carlo realizations are shown in Figs. 5 and 6 for MSE and EMD metrics, respectively. As we can see, the MSE of SR-GSL, rLASSO, as well as of the LASSO algorithm with 10^{-9} source detection threshold perform almost indistinguishably – the corresponding curves practically overlay each other. The MSE drops with a growing number of measurement samples M, as one would expect. This also implies that an increased number of detected gas sources observed in Fig. 4 for these algorithms does not affect the reconstruction of the concentration. In other words, the introduced artifacts tend to have relatively insignificant impact on the quality of concentration estimation.

The LASSO algorithm with $\tau = 10^{-4}$ is an exception here. While it results in a lower number of estimated sources as we see in Fig. 4 due to a higher threshold, this causes an increase of the MSE and worse reconstruction performance. This can be attributed to the fact that sparsity of the LASSO cost function in (30) is regulated with a single parameter α_{lasso}. Thus, its poor choice has a stronger effect on all sources at the same time. In particular, this leads to appearance of source clusters. The additional thresholding destroys these clusters and thus diminishes the resulting reconstruction performance

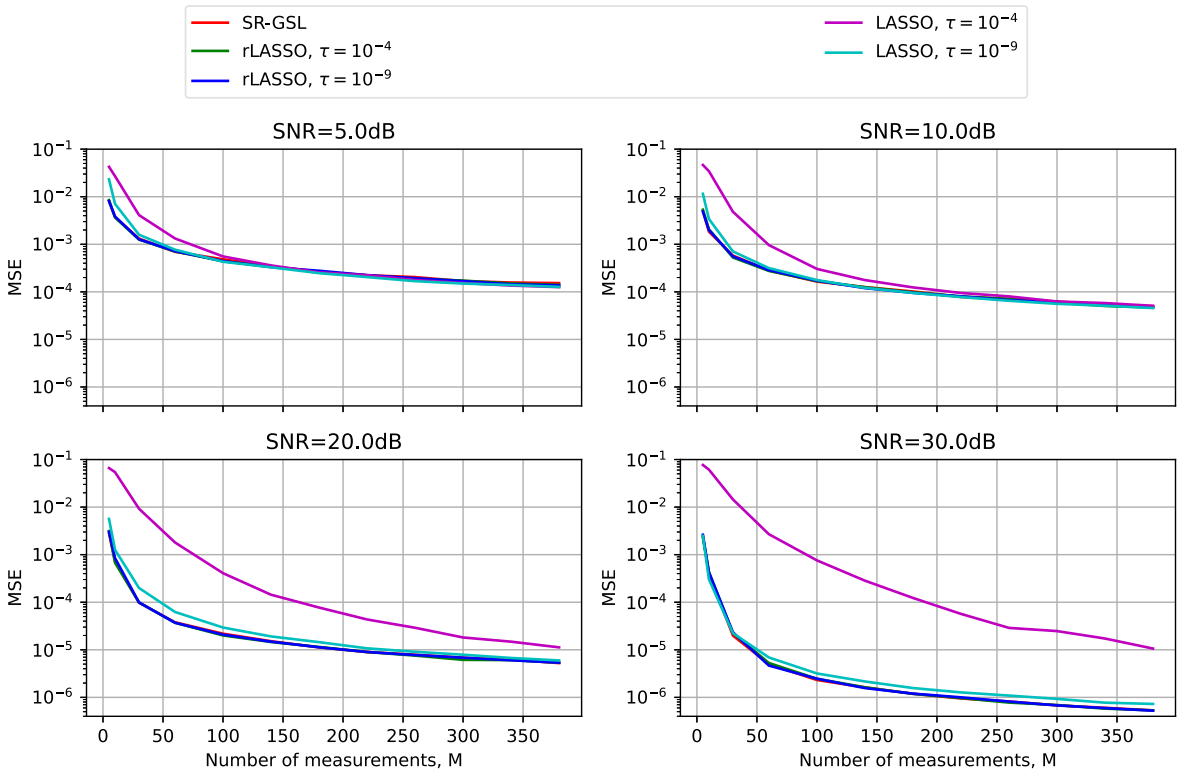


FIG. 5. The MSE value of the estimated concentration $f(x)$ for different SNR values.

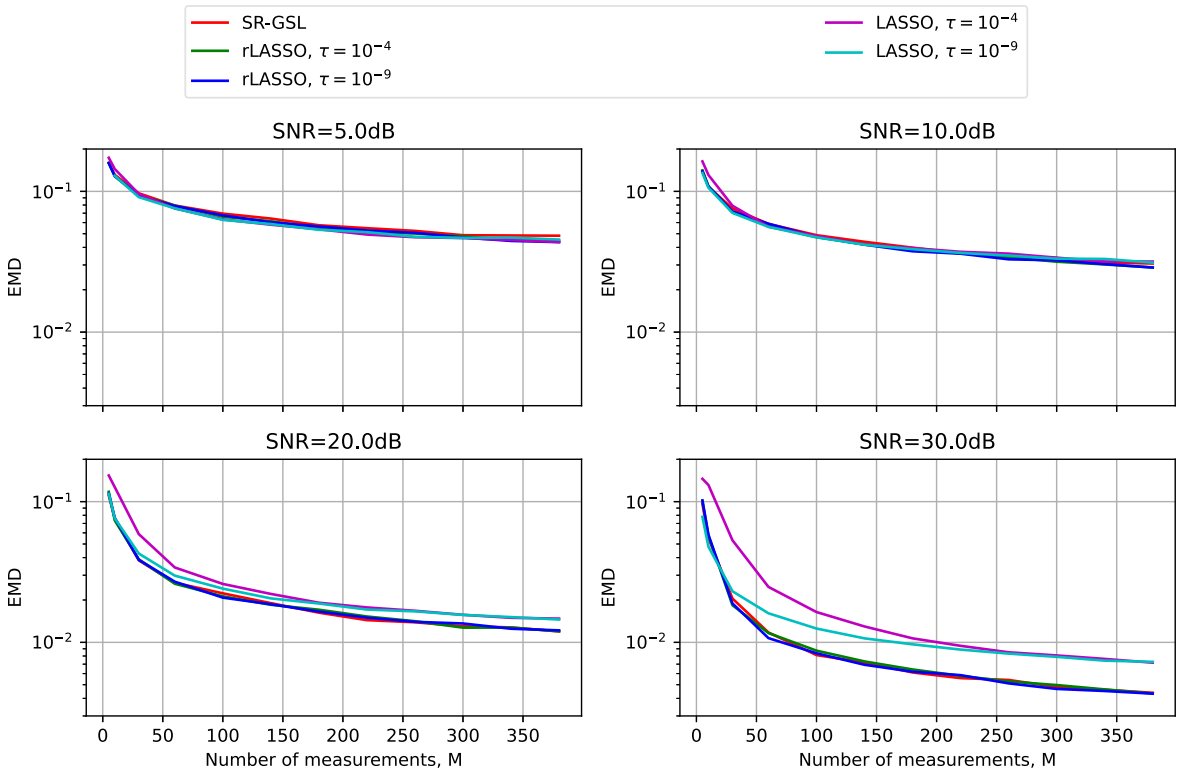


FIG. 6. Error between the true and estimated source distribution as measured by the EMD criterion.

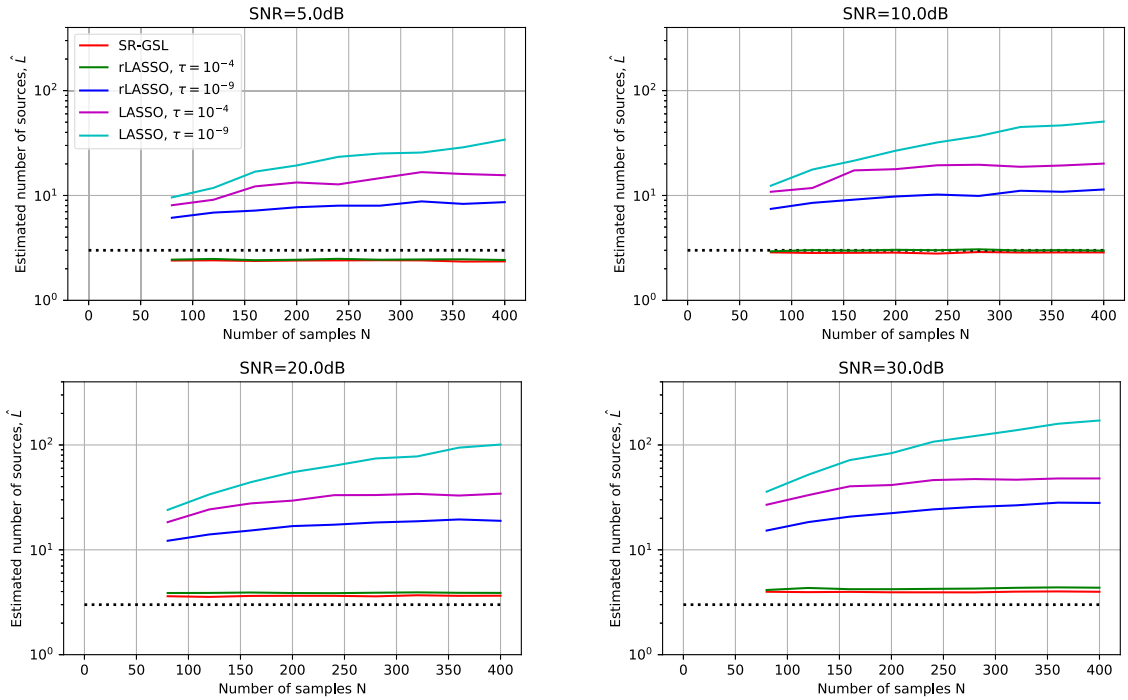


FIG. 7. Estimated averaged number of components $\widehat{L}$ as a function of the discretization accuracy N for fixed number of measurements $M = 50$. The dotted line represent the true number of components $K = 3$.

of LASSO-based estimators. In contrast, the SR-GSL and rLASSO have multiple sparsity or regularization parameters, which allows more flexibility in the "control" of individual sources. In case of rLASSO the additional thresholding does not have such an impact on MSE of the concentration estimation.

Let us now consider the EMD performance of the methods. As we can see from Fig. 6, for 5 dB and 10 dB SNR all methods perform almost indistinguishably. In case of SR-GSL and rLASSO with $\tau = 10^{-4}$ this implies that the estimated sources correspond well to the true locations and few artifacts are inserted. In case of other algorithms this means that despite artifact sources are detected, they tend to have low amplitude that does not impact the EMD performance. However, for the SNR above 20 dB we see that the EMD performance of LASSO algorithms worsens. This again can be attributed to the fact that the LASSO algorithm with a single regularization parameter tends to produce clusters rather that single source estimates. Thus, several sources are detected to approximate a single true gas source. Applying the threshold on the estimated source weights does lead to fewer detections and sparser results, yet also distorts amplitudes of the remaining sources, leading to an increase of EMD error. SR-GSL and rLASSO with a high threshold here again perform best, having the lowest EMD error.

Let us now study the sensitivity of the algorithm on the discretization accuracy N. To this end we fix the number of collected measurement at $M = 50$ and evaluate the estimated number of sources, as well as the MSE criterion for N ranging between $N = 80$ and $N = 400$. The corresponding results averaged over 500 Monte Carlo runs are shown in Figs. 7 and 8.

First, let us study the number of estimated components in Fig. 7. As we see, the SR-GSL as well as rLASSO with the threshold $\tau = 10^{-4}$ do not depend on N. Thus, provided enough measurement samples are collected, the discretization accuracy does not play a significant role for these methods. Let us re-iterate that in case of rLASSO, which is a linearized version of SR-GSL, this is achieved at the expense of high initial number of sources $\widetilde{L}$. It thus requires tuning the threshold τ to properly assess the number of source. With $\tau = 10^{-9}$ rLASSO clearly overestimates the number of components, and the number of artifacts grows with N. A similar trend is observed for the LASSO solver. Also note that with increasing SNR the number of artifacts grows, indicating an overfitting. This effect can be compensated by increasing the threshold τ for rLASSO and LASSO algorithms, which, however, has an effect on the MSE performance as the following results indicate.

The MSE performance of the methods is summarized in Fig. 8. Here we see that with an exception of the LASSO method with $\tau = 10^{-4}$, the MSE performance remains independent of N over the tested SNR range. The LASSO algorithms with $\tau = 10^{-4}$ performs poorly simply due to the fact that with a high threshold the relevant sources are being removed from the model. This does lead to sparser model estimates, with fewer sources, as can be see in Fig. 7, but at the expense of growing reconstruction error. A similar behavior is also observed for EMD criterion: the results show

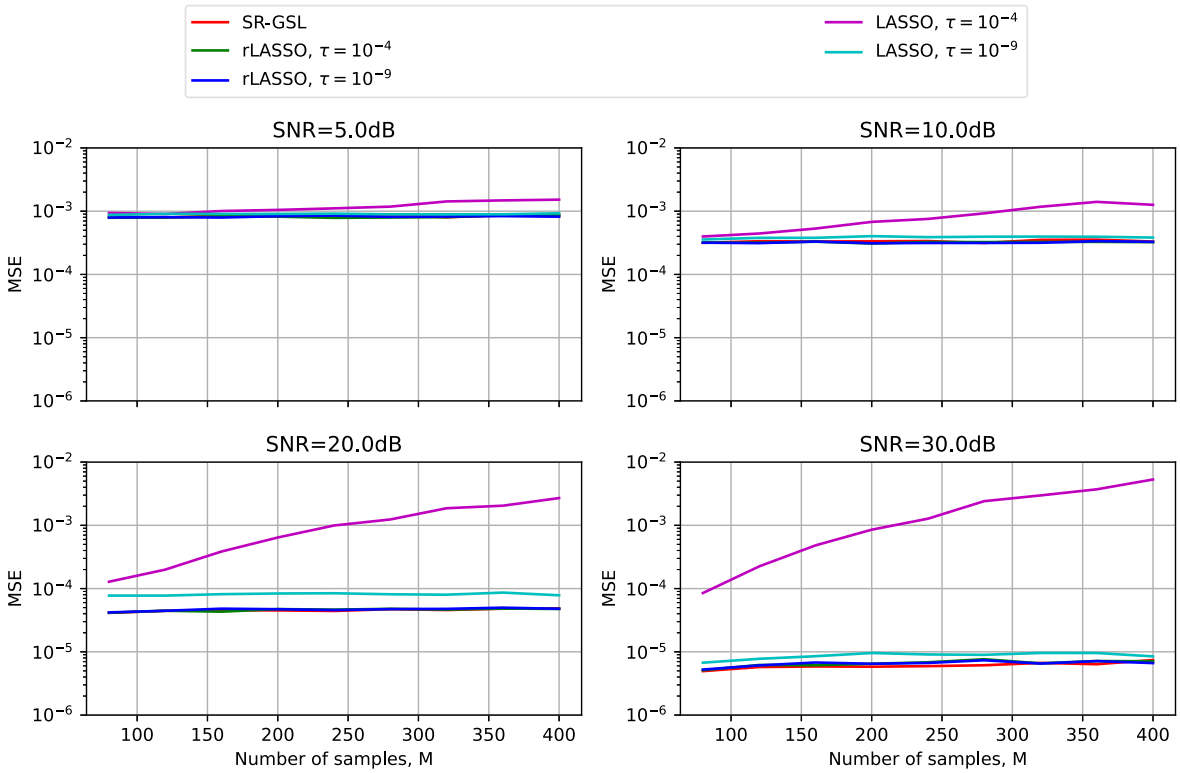


FIG. 8. **The MSE value of the estimated concentration $f(x)$ as a function of discretization accuracy N for different SNR values and fixed number of measurements $M = 50$.**

independence of N, with an exception of the LASSO algorithms. These results are however not shown here, as they follow those of MSE evaluations quite closely.

V. CONCLUSION

The proposed method addresses the problem of nonlinear estimation of the parameters of partial differential equations. Specifically, a linear Poisson's equation has been considered, with the right-hand side – the source signal – represented as a linear combination of Dirac measures with unknown support and weights. The standard approaches to identify parameters of the source signals would imply suitable discretization of the equation, followed by an estimator that imposes sparsity constraints on the source weights of the discretized model using, e.g., ℓ_1 constraints.

The method proposed here does not require such discretization. The algorithm is realized within the framework of Sparse Bayesian learning, which allows seamlessly combining the nonlinear support estimation and weight estimation. The weights are found with the support of Dirac measures kept fixed. This step can be implemented using standard sparse Bayesian learning inference algorithms. The estimation of the source locations, on the other hand, requires a nonlinear optimization and is implemented using the adjoint state method. The latter allows computing the gradient of the corresponding cost function very efficiently. Specifically, optimization is done over the support parameters with non-zero weights, which for highly sparse source signals reduces the dimensionality of the optimization problem. The advantage of such an approach is that it can be extended beyond Poisson's equation to more general Partial Differential Equations with sparse right-hand side, including time-dependency of the PDE and advection effects. It requires the solution of a Partial Differential Equation to be expressed in terms of Green's functions. The latter, however, are not always available in closed form for general boundary conditions; moreover, some solutions might include discontinuities, as is the case for a 2D Poisson's equation, which might complicate the application of the proposed technique. A promising approach in this perspective is to approximate the Green's function through learning, e.g., with a physics-informed neural network, which is an approach currently being investigated. This study remains however outside the scope of this work.

The performance of the proposed method has been compared in simulations to solutions obtained with other sparse estimators utilizing LASSO and re-weighted LASSO schemes applied to a linearized model. The latter is obtained by finely discretizing the search space for possible source locations, and then recovering sparse source weight estimates. We found such re-weighted LASSO with an additional, relatively high threshold can perform comparably with the proposed SR-GSL algorithm. Yet other tested estimation methods perform worse, often introducing false sources – artifacts. While these artifact sources do tend to have low weights and have

relatively small impact on the concentration estimation, they are nonetheless detected and cannot be easily distinguished from potential true gas sources. The results clearly demonstrate the potential of the sparse Bayesian learning method for estimating parameters of Partial Differential Equations with additional sparsity constraints by using non-linear optimization techniques, as compared to classical approaches based on, e.g., ℓ_1 constraint.

ACKNOWLEDGMENT

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union—European Commission. Neither the European Commission nor the European Union can be held responsible for them.

REFERENCES

[1] W. Khalaf et al., "Gas detection via machine learning," *Int. J. Comput. Elect. Autom. Control Inf. Eng*, vol. 2, no. 1, pp. 61–65, 2008.
[2] M. Reggente and A. J. Lilienthal, "Using local wind information for gas distribution mapping in outdoor environments with a mobile robot," in *Proc. Sensors*, 2009, pp. 1715–1720.
[3] D. Badawi, I. Bassi, S. Ozev, and A. E. Cetin, "Deep-learning-based gas leak source localization from sparse sensor data," *IEEE Sensors J.*, vol. 22, no. 21, pp. 20999–21008, Nov. 2022.
[4] A. S. A. Yeon, A. Zakaria, S. M. M. S. Zakaria, R. Visvanathan, K. Kamarudin, and L. M. Kamarudin, "Gas source localization via mobile robot with gas distribution mapping and deep neural network," in *Proc. IEEE 2nd Int. Conf. Electron. Elect. Eng. Intell. Syst.*, 2022, pp. 120–124.
[5] H. Kim, M. Park, C. W. Kim, and D. Shin, "Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation," *Comput. Chem. Eng.*, vol. 125, pp. 476–489, 2019.
[6] G. Strang, *Computational Science and Engineering*.Cambridge, MA, USA:Wellesley-Cambridge Press, 2007.
[7] A. Francis, S. Li, C. Griffiths, and J. Sienz, "Gas source localization and mapping with mobile robots: A review," *J. Field Robot.*, vol. 39, no. 8, pp. 1341–1373, 2022.
[8] H. Ishida, Y. Wada, and H. Matsukura, "Chemical sensing in robotic applications: A review," *IEEE Sensors J.*, vol. 12, no. 11, pp. 3163–3173, Nov. 2012.
[9] A. Lilienthal and T. Duckett, "Experimental analysis of gas-sensitive braitenberg vehicles," *Adv. Robot.*, vol. 18, no. 8, pp. 817–834, 2004.
[10] S. Li and Y. Guo, "Distributed source seeking by cooperative robots: All-to-all and limited communications," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 1107–1112.
[11] A. Rutkowski, S. Edwards, M. Willis, R. Quinn, and G. Causey, "A robotic platform for testing moth-inspired plume tracking strategies," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2004, pp. 3319–3324.
[12] X. xing Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future Outlook," *Robot. Auton. Syst.*, vol. 112, pp. 123–136, 2019.
[13] S. Li, Y. Guo, and B. Bingham, "Multi-robot cooperative control for monitoring and tracking dynamic plumes," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 67–73.
[14] X. Jiang and S. Li, "Plume front tracking in unknown environments by estimation and control," *IEEE Trans. Ind. Inform.*, vol. 15, no. 2, pp. 911–921, Feb. 2019.
[15] X. Jiang, S. Li, B. Luo, and Q. Meng, "Source exploration for an underactuated system: A control-theoretic paradigm," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 3, pp. 1100–1107, May 2020.
[16] Z. Shen, Z. He, S. Li, Q. Wang, and Z. Shao, "A multi-quadcopter cooperative cyber-physical system for timely air pollution localization," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 3, pp. 1–23, Apr. 2017.
[17] M. Hutchinson, H. Oh, and W.-H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Inf. Fusion*, vol. 36, pp. 130–148, 2017.
[18] J. R. Bourne, E. R. Pardyjak, and K. K. Leang, "Coordinated Bayesian-based bioinspired plume source term estimation and source seeking for mobile robots," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 967–986, Aug. 2019.
[19] M. Hutchinson, C. Liu, and W.-H. Chen, "Source term estimation of a hazardous airborne release using an unmanned aerial vehicle," *J. Field Robot.*, vol. 36, no. 4, pp. 797–817, 2019.
[20] B. Ristic, D. Angley, B. Moran, and J. L. Palmer, "Autonomous multi-robot search for a hazardous source in a turbulent environment," *Sensors*, vol. 17, no. 4, 2017, Art. no. 918.
[21] B. Ristic, A. Skvortsov, and A. Walker, "Autonomous search for a diffusive source in an unknown structured environment," *Entropy*, vol. 16, no. 2, pp. 789–813, 2014.
[22] C. Rhodes, C. Liu, P. Westoby, and W.-H. Chen, "Autonomous search of an airborne release in urban environments using informed tree planning," *Auton. Robots*, vol. 47, no. 1, pp. 1–18, Jan. 2023.
[23] M. Vergassola, E. Villermaux, and B. I. Shraiman, "Infotaxis as a strategy for searching without gradients," *Nature*, vol. 445, pp. 406–409, 2007.
[24] X.-Y. Dai, J.-Y. Wang, and Q.-H. Meng, "An infotaxis-based odor source searching strategy for a mobile robot equipped with a tdlas gas sensor," in *Proc. IEEE Chin. Control Conf.*, 2019, pp. 4492–4497.
[25] M. Hutchinson, H. Oh, and W.-H. Chen, "Entrotaxis as a strategy for autonomous search and source reconstruction in turbulent conditions," *Inf. Fusion*, vol. 42, pp. 179–189, 2018.
[26] T. Wiedemann, C. Manss, D. Shutin, A. J. Lilienthal, V. Karolj, and A. Viseras, "Probabilistic modeling of gas diffusion with partial differential equations for multi-robot exploration and gas source localization," in *Proc. IEEE Eur. Conf. Mobile Robots*, 2017, pp. 1–7.
[27] T. Wiedemann, C. Manss, and D. Shutin, "Multi-agent exploration of spatial dynamical processes under sparsity constraints," *Auton. Agents Multi-Agent Syst.*, vol. 32, pp. 134–162, Jul. 2017.
[28] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.
[29] D. Wipf, J. Palmer, and B. Rao, "Perspectives on sparse Bayesian learning," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst.*. 2003, pp. 249–256.
[30] T. Wiedemann, A. Lilienthal, and D. Shutin, "Analysis of model mismatch effects for a model-based gas source localization strategy incorporating advection knowledge," *Sensors*, vol. 19, no. 3, Jan. 2019, Art. no. 520.
[31] T. Wiedemann, D. Shutin, and A. Lilienthal, "Model-based gas source localization strategy for a cooperative multi-robot system - a probabilistic approach and experimental validation incorporating physical knowledge and model uncertainties," *Robot. Auton. Syst.*, vol. 118, pp. 66–79, 2019.
[32] T. Wiedemann, D. Shutin, and A. Lilienthal, "Experimental validation of domain knowledge assisted robotic exploration and source localization," in *Proc. IEEE Int. Conf. Auton. Syst.*, 2021, pp. 1–5.
[33] T. Wiedemann, "Domain knowledge assisted robotic exploration and source localization," Ph.D. dissertation, Sch. Sci. Technol., Örebro Univ., Örebro, Sweden, 2020.
[34] J. Nocedal and S. Wright, *Numerical Optimization, Ser. Springer Series in Operations Research and Financial Engineering*. New York, Ny, USA:Springer, 2006.
[35] R.-E. Plessix, "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications," *Geophysical J. Int.*, vol. 167, no. 2, pp. 495–503, Nov. 2006.
[36] P.-J. Chung, "A max-search approach for DOA estimation with unknown number of signals," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 3, pp. 612–619, Jun. 2010.
[37] G. Korn and T. Korn, *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review, Ser. Dover Civil and Mechanical Engineering Series*. New York, NY, USA: Dover Pub., 2000.
[38] P. Kythe, Green's *Functions and Linear Differential Equations: Theory, Applications, and Computation*. Boca Raton, FL, USA:CRC Press, 2011.

[39] V. S. P. Ruiz, "Neural network surrogate models for the localization of sources in semilinear elliptic PDEs" Master's thesis, TUM Sch. Computation., Inf. Technol. Tech. Univ. Munich, Munich,Germany, 2023.
[40] D. P. Wipf and B. D. Rao, "Sparse bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.
[41] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, Nov. 2008.
[42] R. Giri and B. Rao, "Type I and type II Bayesian methods for sparse signal recovery using scale mixtures," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3418–3428, Jul. 2016.
[43] E. J. Candes, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, Dec. 2008.
[44] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Sig. Process.*, 2008, pp. 3869–3872.
[45] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 187–200, Jan. 1999.
[46] D. Wipf, B. Rao, and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6236–6255, Sep. 2011..
[47] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*. Vancouver, BC, Canada, Dec. 2007, pp. 1–8. [Online]. Available: https://papers.nips.cc/paper_files/paper/2007/hash/9c01802ddb981e6bcfbec0f0516b8e35-Abstract.html
[48] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, San Francisco, CA, USA, 2000, pp. 46–53.
[49] L. Hu, Z. Shi, J. Zhou, and Q. Fu, "Compressed sensing of complex sinusoids: An approach based on dictionary refinement," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3809–3822, Jul. 2012.
[50] D. Shutin, T. Buchgraber, S. R. Kulkarni, and H. V. Poor, "Fast variational sparse Bayesian learning with automatic relevance determination for superimposed signals," *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6257–6261, Dec. 2011.
[51] D. Shutin, S. R. Kulkarni, and H. V. Poor, "Incremental reformulated automatic relevance determination," *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4977–4981, Sep. 2012.
[52] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. 9th Int. Workshop Artif. Intell. Statist.*, Key West, FL, USA, 2003, pp. 276–283.
[53] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.
[54] T. L. Hansen, B. H. Fleury, and B. D. Rao, "Superfast line spectral estimation," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2511–2526, May 2018.
[55] T. Wiedemann, C. Manss, and D. Shutin, "Multi-agent exploration of spatial dynamical processes under sparsity constraints," *Auton. Agents Multi-Agent Syst.*, vol. 32, no. 1, pp. 134–162, 2018.
[56] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.*, vol. 16, no. 5, pp. 1190–1208, 1995.
[57] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Roy. Statist. Soc.*, vol. 58, pp. 267–288, 1996.
[58] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[59] Y. Zhao and X. Huo, "A survey of numerical algorithms that can solve the lasso problems," *WIREs Comput. Statist.*, vol. 15, no. 4, 2023, Art. no. e1602.
[60] Y. Rubner, C. Tomasi, and L. Guibas, "A metric for distributions with applications to image databases," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 59–66.