2023 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2023, ROME, ITALY

AMORTIZED VARIATIONAL PEAK FITTING FOR SPECTROSCOPIC DATA

David Frich Hansen   *Tommy Sonne Alstrøm*   *Mikkel N. Schmidt*

Dept. of Applied Mathematics and Computer Science, Technical University of Denmark
Richard Petersens Plads 324, 2800 Kgs. Lyngby, Denmark
{dfha, tsal, mnsc}@dtu.dk

ABSTRACT

Spectroscopic analysis relies on identifying and understanding the spectral peaks that represent unique characteristics of an analyte. In high-speed real-time settings, current peak fitting techniques, particularly Bayesian methods involving MCMC or variational approximation, can be prohibitively expensive. We propose an unsupervised method using a convolutional neural network to estimate the number of peaks and their parameters along with posterior uncertainty, by amortizing variational inference in a classical parametric peak model. In a simulated data study, our method reliably determines the number of peaks, precisely estimates parameters similar to direct variational inference, and accurately captures uncertainty comparable to MCMC methods. Our novel, fast, and precise method for Bayesian spectral analysis opens new possibilities in real-time spectral data processing for high-speed monitoring and control.

Index Terms— Spectroscopy, GFlowNet, Variational autoencoder, Amortized inference

1. INTRODUCTION

An essential task in chemical science is determining the components and relative quantities of a compound analyte. This task is often achieved by spectroscopy, such as infrared (IR), near-infrared (NIR), or Raman [1]. These techniques work by irradiating the analyte and measuring the electromagnetic spectrum that results from the interaction. Identification and quantitation is typically based on the locations and intensities of one or more characteristic spectral peaks.

In a typical signal processing pipeline, a peak model is fitted to the observed spectrum using e.g. least squares or other statistical techniques such as Markov chain Monte Carlo (MCMC) [2, 3] or variational Bayes. As part of the fitting process, the number of peaks must also be estimated: Often, this parameter is either assumed to be known, or set to a sufficiently high value such that all relevant peaks are extracted. As an alternative to extracting peak parameters, the problem can be stated as an unmixing problem, and the unmixed components can then be matched to a known database [4]; however, this approach does not directly result in interpretable peak parameters.

In many use cases, uncertainty information about the peak parameters is of critical importance, for example, when studying chemical components at low concentrations or phenomena which give rise to minuscule peak shifts. Parameter uncertainty can be estimated in several ways, including traditional confidence intervals or by formulating the model in the Bayesian framework. In all cases, estimates are most often obtained using iterative methods such as gradient-based optimization, Markov chain Monte Carlo (MCMC), or stochastic variational inference (SVI); however, these methods are somewhat slow at inference time and might not be practical in time-sensitive applications, such as monitoring of in vivo drug delivery or control of chemical reaction processes.

1.1. Contributions

In this work, we seek to address these challenges: We propose a fully amortized Bayesian inference model that is faster than traditional inference methods at inference time, which can reliably estimate the number of components, and which can be trained in an unsupervised setting.

Based on a classical parametric peak model, we use the SVI framework for approximate Bayesian inference. We amortize the inference as an iterative peak-picking procedure by fitting a neural network inference model which outputs the approximate posterior parameters of one peak at a time as well as the probability that all peaks have been identified. Our work extends existing iterative peak-fitting procedures by combining ideas from generative flow networks (GFlowNets), stochastic variational inference (SVI), and variational autoencoders (VAE). Our approach is essentially a VAE with the parametric peak model as the decoder and the peak parameters as the latent space; however, since we learn the number of components in a sequential procedure and the dimensionality of the latent space thus varies, the encoder is similar to a GFlowNet. As in other VAEs with discrete latent com-

Thank you to the Independent Research Fund Denmark [grant no. 9131-00039B] for funding this work and to NVIDIA for supplying GPU's for computations.

ponents, we use a Gumbel-softmax relaxation to construct a differentiable loss. In the following, we briefly review these techniques before we describe the details of our approach.

1.2. Background

Iterative peak picking: Our approach is most similar to the iterative peak picking algorithm of Park et al. [5], in which a convolutional neural network (CNN) is trained to estimate the number of peaks as well as the parameters of the single peak with the largest area. The trained network is then used to extract multiple peaks by sequentially estimating the strongest peak, subtracting it out, and running the algorithm again on the residual until the estimated number of peaks is zero. To improve the quality of the fit, an iterative optimization technique is subsequently used to finetune it. Since the training objective is supervised, the method requires access to a large number of spectra with annotated peak parameters.

GFlowNet: Inspired by reinforcement learning, in a GFlowNet [6], a stochastic policy is trained to sequentially generate discrete composite data, such that the probability of each datum is proportional to a given reward function (an unnormalized probability). A GFlowNet is structured as a directed acyclic graph (DAG) with a unique source state s_0. Data objects are constructed by taking a sequence of actions following the stochastic policy $P_F(s_k|s_{k-1};\phi)$, until a terminating state is reached, forming a trajectory $\tau = (s_0, s_1, \ldots, s_K)$. Each internal state has an associated backward policy $P_B(s_{k-1}|s_k;\phi)$ characterizing the probability that a given state s_k is reached from each of its parent states. Rewards at the terminal states constitute an out-flow of unnormalized probability, and since multiple distinctive paths can generate identical outputs, GFlowNet objectives seek to match the input and output flows at each intermediate node or by balancing flow trajectories.

Given a reward function $R(x)$ for terminal states x, a popular training objective is *trajectory balance* [7],

$$B = \left(\log \frac{Z_\phi \prod_{k=1}^{K} P_F(s_k|s_{k-1};\phi)}{R(s_K) \prod_{k=1}^{K} P_B(s_{k-1}|s_k;\phi)} \right)^2. \tag{1}$$

With Z_ϕ denoting the total flow, this objective states that the flow on a given trajectory from s_0 to s_K (numerator) should match the fraction of the reward that is attributed to that trajectory (denominator). Minimizing this objective with respect to the parameters ϕ results in a stochastic policy P_F that generates samples (reaches terminating states) with probability proportional to the reward.

Stochastic variational inference: SVI [8] is an approximate Bayesian method where the posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$ is approximated by a tractable distribution $q_\phi(\boldsymbol{\theta})$ with parameters ϕ by minimizing their Kullback-Leibler divergence,

$$L = KL\big(q_\phi(\boldsymbol{\theta}) \| p(\boldsymbol{\theta}|\boldsymbol{x})\big) \tag{2}$$

$$= \underbrace{-\,\mathbb{E}_{q_\phi}\left(\log \frac{p(\boldsymbol{\theta},\boldsymbol{x})}{q_\phi(\boldsymbol{\theta})}\right)}_{\text{ELBO}} + \log p(\boldsymbol{x}) \tag{3}$$

or equivalently maximizing the evidence lower bound (ELBO). The objective is minimized using stochastic gradient descent, where gradients are typically approximated using a single sample Monte Carlo estimate $\boldsymbol{\theta}^* \sim q_\phi(\boldsymbol{\theta})$, either using variants of the score function estimator [9] (a.k.a. REINFORCE)

$$\nabla_\phi L \approx \log \frac{p(\boldsymbol{\theta}^*,\boldsymbol{x})}{q_\phi(\boldsymbol{\theta}^*)} \nabla_\phi \log q_\phi(\boldsymbol{\theta}^*) \tag{4}$$

or the pathwise estimator [10, 9] (a.k.a the reparametrization trick)

$$\nabla_\phi L \approx \nabla_\phi \log \frac{p(g_\phi(\boldsymbol{\xi}^*),\boldsymbol{x})}{q_\phi(g_\phi(\boldsymbol{\xi}^*))}. \tag{5}$$

Here, $\boldsymbol{\theta}^* = g_\phi(\boldsymbol{\xi}^*)$ is a differentiable sampling path and $\boldsymbol{\xi}^* \sim p(\boldsymbol{\xi})$ is an auxillary random variable. For example, if $\theta \sim \mathcal{N}(\mu, \sigma^2)$ we can have $g(\xi) = \sigma \cdot \xi + \mu$ with $\xi \sim \mathcal{N}(0, 1)$.

In the auto-encoding variational Bayes setting, a conditional inference model $q_\psi(\boldsymbol{\theta}|\boldsymbol{x})$ is used to approximate the posterior distribution. The inference model is typically chosen as a neural network with parameters ψ which takes the data as input and outputs parameters ϕ for the posterior approximation. Rather than having to fit the posterior approximation for each observation, the cost of inference is thus amortized in the training of the neural inference model.

Gumbel softmax: Pathwise gradient estimation is not directly applicable to discrete variables; however, the Gumbel-softmax can be used as a differentiable approximation [11]. In the case of a Bernoulli(π) variable, softmax relaxed samples $\widetilde{\pi} \in [0, 1]$ can be generated as

$$\widetilde{\pi} = \Big(1 + \exp\big[-T\big(\ell + \log \tfrac{\pi}{1-\pi}\big)\big]\Big)^{-1} \tag{6}$$

where T is a temperature that controls how close the sampled variables are to being discrete and ℓ is a standard logistic random variable or, equivalently, the difference between two independent standard Gumbel variables.

2. METHODS

2.1. Pseudo-Voigt spectral model

A spectrum can be represented as a non-negative vector, $\boldsymbol{x} \in \mathbb{R}_+^W$ of dimensionality W, which is the number of measured wavenumbers. Spectral models typically capture signal peaks of a given parametric form, and in some cases, baselines and other artifacts as well. We assume an additive noise model,

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{\epsilon} \tag{7}$$

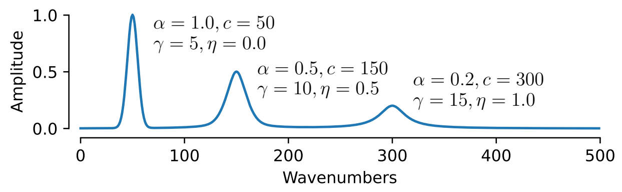


Fig. 1. Illustration of a noise-free spectrum with three peaks.

where $\boldsymbol{\theta}$ are model parameters and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$. We employ a K-peak pseudo-Voigt signal model,

$$\boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{\alpha}^\top \boldsymbol{V}(\boldsymbol{c}, \boldsymbol{\gamma}, \boldsymbol{\eta}), \tag{8}$$

where $\boldsymbol{\alpha} \in \mathbb{R}_+^K$ is a vector with individual peak amplitudes as its components α_k, and $\boldsymbol{V}(\boldsymbol{c}, \boldsymbol{\gamma}, \boldsymbol{\eta}) \in \mathbb{R}_+^{K \times W}$ is the so-called height-normalized K-peak pseudo-Voigt function evaluated at all observed wavenumbers ω,

$$\boldsymbol{V}(\boldsymbol{c}, \boldsymbol{\gamma}, \boldsymbol{\eta}) = \boldsymbol{\eta} \underbrace{\frac{1}{1+([\omega-\boldsymbol{c}]/\boldsymbol{\gamma})^2}}_{\text{Lorentzian}} + (1-\boldsymbol{\eta}) \underbrace{\exp\left[-\frac{(\omega-\boldsymbol{c})^2}{2\boldsymbol{\gamma}^2}\right]}_{\text{Gaussian}},$$

where all operations are element-wise. Here, $\boldsymbol{c} \in \mathbb{R}_+^K$ are the positions, $\boldsymbol{\gamma} \in \mathbb{R}_+^K$ are the full-width-at-half-maximum, and $\boldsymbol{\eta} \in [0, 1]^K$ are the Lorentzianities (shapes) of the K peaks, with components denoted by c_k, γ_k, and η_k respectively. Thus, the set of model parameters are $\boldsymbol{\theta} = \boldsymbol{\theta}_{1:K} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K\}$ where $\boldsymbol{\theta}_k = \{\alpha_k, c_k, \gamma_k, \eta_k\}$. An illustrative example of a noise-free spectrum is given in Fig. 1. With this model, the likelihood is given as $p(\boldsymbol{x}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{f}(\boldsymbol{\theta}), \sigma^2\boldsymbol{I})$ and the joint $p(\boldsymbol{\theta}, \boldsymbol{x})$ is found by multiplying with appropriate priors $p(\boldsymbol{\theta})$.

2.2. Stochastic variational inference

As a variational approximation, we adopt a mean field family for each peak given by

$$q_{\phi k}(\boldsymbol{\theta}_k) = q(\alpha_k) q(c_k) q(\gamma_k) q(\eta_k), \tag{9}$$
$$q(\alpha_k) = \mathcal{N}(\alpha_k|, \mu_{\alpha k}, \sigma_{\alpha k}^2), \quad q(c_k) = \mathcal{N}(c_k|\mu_{ck}, \sigma_{ck}^2),$$
$$q(\gamma_k) = \mathcal{N}(\gamma_k|\mu_{\gamma k}, \sigma_{\gamma k}^2), \quad q(\eta_k) = \mathcal{B}(\eta_k|a_{\eta k}, b_{\eta k}),$$

where $\mathcal{N}$ and $\mathcal{B}$ are univariate normal and beta distributions, such that we have

$$q_\phi(\boldsymbol{\theta}) = \prod_{k=1}^{K} q_{\phi k}(\boldsymbol{\theta}_k). \tag{10}$$

SVI entails minimizing the loss in Eq. 2 with respect to the variational parameters $\phi = \{\phi_1, \ldots, \phi_K\}$,

$$\phi_k = \{\mu_{\alpha k}, \sigma_{\alpha k}^2, \mu_{ck}, \sigma_{ck}^2, \mu_{\gamma k}, \sigma_{\gamma k}^2, a_{\eta k}, b_{\eta k}\},$$

with the number of components K assumed known. Non-negativity constraints on variance and shape parameters are easily implemented as unconstrained optimization on the log-transformed parameters.

2.3. Amortized inference

There are two major drawbacks of the described SVI approach: 1) The number of components K needs to be specified or estimated separately, and 2) the method requires running an iterative optimization procedure, which might be prohibitively slow in a real-time setting.

We propose an amortized procedure that alleviates both problems: The idea is to utilize a construction similar to a GFlowNet as a conditional inference model. We let the initial state s_0 denote an "empty" model and design the stochastic policy P_F such that it either adds a single pseudo-Voigt peak or terminates. Mimicking the trajectory balance criterion in Eq. (1), we write

$$q_\psi(\boldsymbol{\theta}|\boldsymbol{x}) = \pi_K(\psi, \boldsymbol{x}_K) \cdot \prod_{k=1}^{K} \frac{P_F(s_k|s_{k-1}; \psi, \boldsymbol{x}_{k-1}) \cdot \pi_{k-1}(\psi, \boldsymbol{x}_{k-1})}{P_B(s_{k-1}|s_k)}, \tag{11}$$

where π_k denotes the probability of terminating at step k. The policy is conditioned on $\boldsymbol{x}_{k-1}$, which denotes the residual at step k with $\boldsymbol{x}_0 = \boldsymbol{x}$. The forward policy and termination probability are constructed as a neural network $\boldsymbol{x}_{k-1} \to \{\phi_k, \pi_k\}$, which inputs a residual and output parameters of the variational family in Eq. (9) as well as the termination probability. Finally, we assign equal weight to all parents in the backward policy,

$$P_B(s_{k-1}|s_k) = \frac{1}{(k+1)!}. \tag{12}$$

In a full sequential forward pass, we first input the spectrum to the neural network yielding a variational distribution for a single peak and a stopping probability. We sample the Bernoulli stop variable, and if it is false, we sample the peak parameters and subtract the peak from the signal to yield the first residual. We then repeat the process using the residual as input, and continue until the sampled stop variable is true. The process is illustrated in Fig. 2.

To train the neural network, we use pathwise gradient estimates; however, since the sampling path for discrete variables is not differentiable, we relax samples from the Bernoulli variables π_k with the Gumbel-softmax. We then compute the loss as a weighted average of the joint target distribution. Let $\mathcal{L}_k = \log p(\boldsymbol{\theta}_{0:k}, \boldsymbol{x})$ denote the unnormalized log-posterior of a model containing the first k components. We relax the target in the ELBO as

$$\log p(\boldsymbol{\theta}, \boldsymbol{x}) \approx \sum_{k=0}^{K_{\text{max}}} \rho_k \mathcal{L}_k, \tag{13}$$

with weights

$$\rho_k = \tilde{\pi}_k \prod_{i=0}^{k-1} (1 - \tilde{\pi}_i), \tag{14}$$

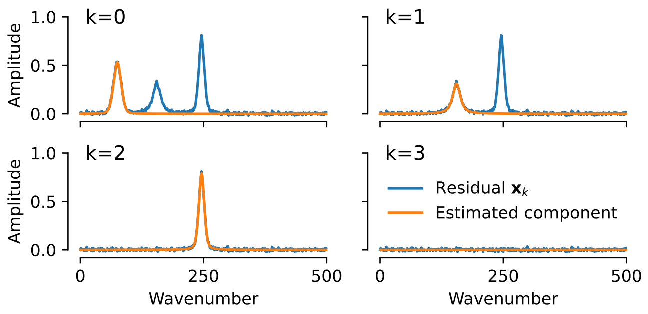


Fig. 2. Example of the iterative peak picking procedure originally suggested by [5] with three components. Components are sequentially subtracted from the signal, yielding a residual $\boldsymbol{x}_k$, until no more components are estimated.

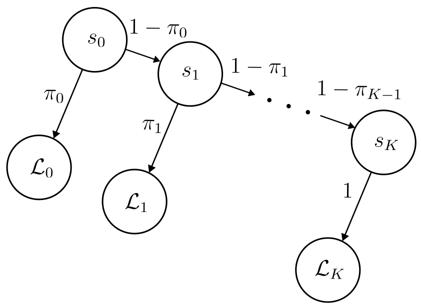


Fig. 3. The relaxation of the stopping variable $\widetilde{\pi}$ requires a weighting of each possible state in the model with the relaxed variable. While in theory, this could continue infinitely, in practice we set a maximum number of allowed components, K_{max} after which the component generation procedure stops, regardless of π_{K-1}

and relax the log-variational distribution similarly. The procedure is schematically illustrated in Fig. 3.

The final loss is the relaxed negative ELBO,

$$L = -\mathbb{E}_{q_\psi}\left[\sum_{k=0}^{K_{\text{max}}} \rho_k \log \frac{p(\boldsymbol{\theta}_{0:k}, \boldsymbol{x})}{q_\phi(\boldsymbol{\theta}_{0:k}|\boldsymbol{x})}\right] \tag{15}$$

which we optimize using the single sample pathwise gradient estimator.

3. EXPERIMENTS

We demonstrate our method on a simulated data problem. Our primary purpose is to compare the amortized approach to direct SVI in terms of how well peak parameters and posterior uncertainties are estimated. For the latter, we further benchmark against MCMC initialized at ground truth.

3.1. Experimental settings

Data generation: Spectra were simulated according to Eq. 7 using integer wavenumbers $\omega = 0, \ldots, 499$, $K \sim U(0,4)$ peaks located at $c \sim U(0,499)$ with amplitude $\alpha_k \sim \mathcal{U}(0.2, 0.8)$, width $\gamma_k \sim \mathcal{U}(5,15)$, and Lorentzianity $\eta \sim \mathcal{U}(0,1)$, where U and $\mathcal{U}$ denote discrete and continuous uniform distributions. The locations and amplitudes of peaks were further constrained to be separated by at least 50 and 0.2 units, respectively, to avoid strong overlap/symmetry. We used a noise variance of $\sigma^2 = 0.01^2$.

Model settings: In all experiments we used flat, improper priors, $p(\boldsymbol{\theta}) \propto 1$. We used a temperature of $T = 1$ for the Gumbel-softmax relaxation, and the maximum number of components was set to $K_{\text{max}} = 4$.

Training and test: As a common test set used across all settings, we generated 1000 random spectra. SVI and MCMC were run directly on the test set, and as training data for the amortized method, we generated data on the fly.

SVI training: SVI was trained with 100 000 iterations of the ADAM optimizer with a learning rate of 10^{-3}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. All variables were transformed by either a log transformation or a sigmoid transformation to an unconstrained optimization problem using the mean field variational approximation in Eq. (10).

MCMC reference: We ran MCMC on the test set initialized with the ground truth parameters, using a Metropolis-within-Gibbs procedure [2] with acceptance rates tuned to around 25% on a pilot run. We ran 100 000 update sweeps for each spectrum and used the entire chain to compute statistics.

Neural network architecture: For the amortized model, we used a neural network consisting of a) a 1D-convolutional layer with kernel size 100, 128 output channels, and tanh-activation, b) a 1D-convolutional layer with kernel size 1, 128 output channels, and tanh-activation, c) flattening followed by a linear layer with 128 output channels and tanh-activation, and finally d) a linear layer with 9 output channels. The first eight outputs correspond to distribution parameters for location, amplitude, width, and Lorentzianity, and the ninth output is the stop probability. A sigmoid or exponential function was applied to each output to map it onto a suitable domain (see code for details).

Amortized training: The neural network was trained using the Adam optimizer for 2M steps with batch size 256 and a triangular cyclic learning rate between 10^{-4} and 10^{-6} with a period of 4 000 steps, followed by 1M steps with the learning rates reduced by a factor 10.

3.2. Results

Model fit: Fig. 4 shows histograms of the achieved log-likelihood using SVI and our amortized method. When SVI does not fail, fits are close to optimal; however, in our experiments, SVI failed in around 20% of the cases, but only rarely when there is a single peak. We found empirically that the majority of these failures are when one or more peaks are completely missed in the fit. We note that the failure of SVI is likely fixed by running multiple restarts with different

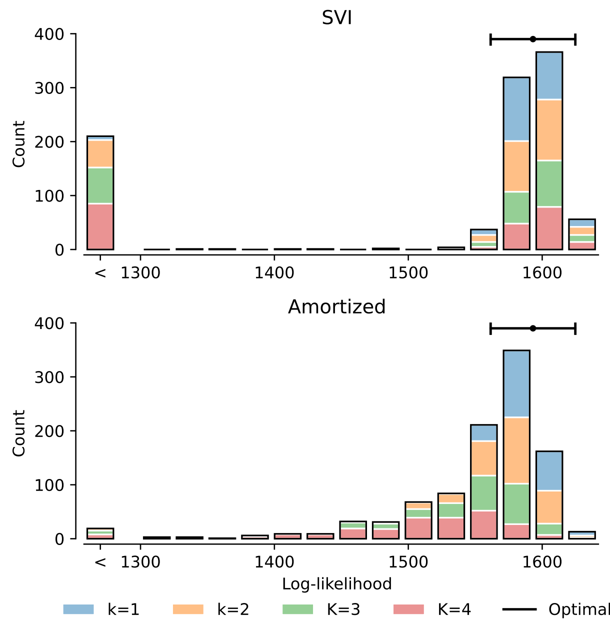
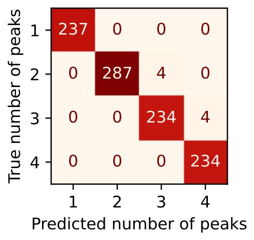


Fig. 4. Log-likelihood on test data. An optimal fit yields log-likelihoods around 1600 nats (95% interval indicated). Log-likelihoods around 1300 nats correspond to a slightly poor fit in our assessment, and we lumped together values below that threshold as failures.



True number of peaks \ Predicted number of peaks	1	2	3	4
1	237	0	0	0
2	0	287	4	0
3	0	0	234	4
4	0	0	0	234

Fig. 5. Confusion matrix: Estimated number of components in the amortized method versus ground truth.

initializations. Fits using the amortized method are slightly worse but still very close to optimal, and the failure rate is significantly lower, around 2%. A possible explanation for the strong reduction of the failure rate could be that the neural network, by providing a mapping from the data to variational parameters, does not depend on any initialization and is thus not sensitive to local minima in the loss. The number of peaks also affects the amortized performance, which degrades with the number of peaks.

Number of components: Fig. 5 shows a confusion matrix for the number of components found by the amortized method compared with ground truth. The correct number of components was found in 99% of the cases, and in remaining cases the number of components was over estimated by one.

Parameter estimation: Fig. 6 shows histograms of the absolute error of the estimated components compared to ground truth. With 1000 test spectra containing 2.5 peaks each on average, statistics are shown for each of the approx. 2 500 peaks. Both SVI and our amortized method recover the parameters very accurately in most cases (SVI is slightly better); however, large errors occur in substantially more cases in SVI.

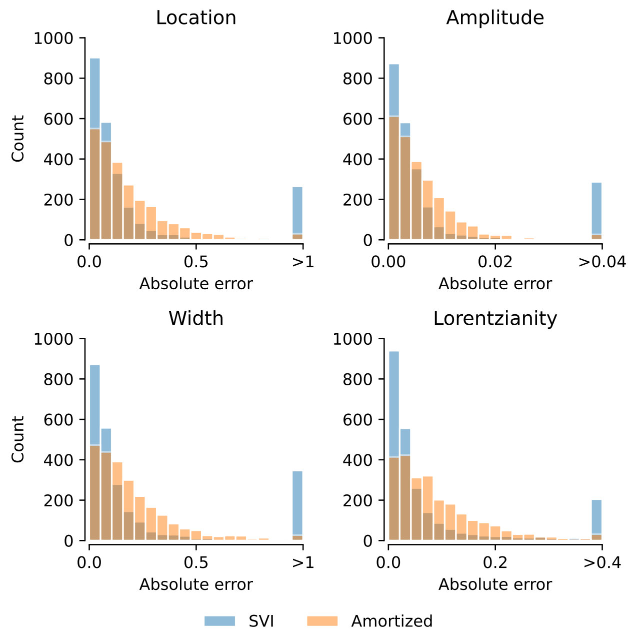


Fig. 6. Mean absolute error on peak parameters for SVI and the amortized method.

Uncertainty: Fig. 7 shows the estimated standard deviations of each fitted peak parameter using SVI and the amortized approach, benchmarked against the MCMC ground truth. In general, SVI and our amortized method are in almost perfect agreement with each other. On location, width, and Lorentzianity, agreement with MCMC is very good, whereas uncertainty on amplitudes appears to be slightly underestimated for both methods (mean field variational methods are known to often underestimate variance, so this is perhaps not surprising).

4. CONCLUSION

Based on our findings, amortized variational inference appears to be an attractive method for fast and precise probabilistic peak estimation in spectral data, given a sufficient amount of training data. The experiments reported herein correspond to a setting without strong noise, baselines, and contaminants, and further experiments are needed to determine if the results carry over to more realistic settings. While we have not conducted an exact timing experiment, we note that the amortized method is more robust and orders of magnitudes faster than iterative fitting using SVI at inference time while providing only slightly worse estimates. If needed, it is possible to improve the amortized estimates by finetuning with a small number of SVI updates.

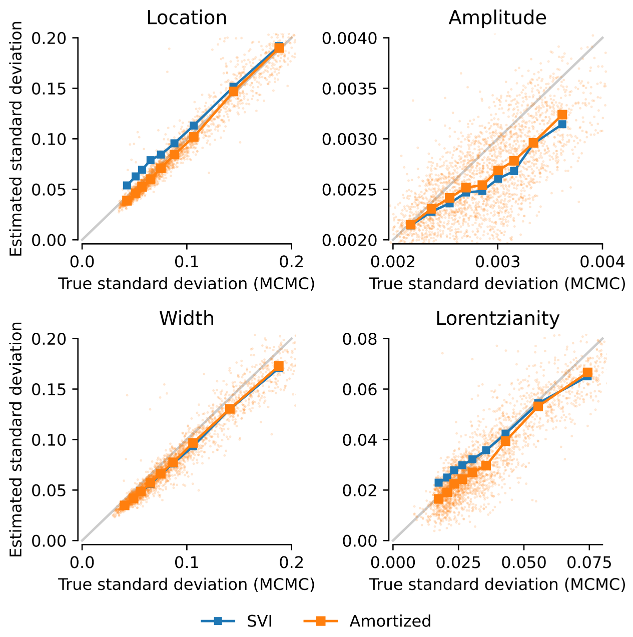


Fig. 7. Estimated versus true standard deviation as computed by MCMC. Lines with square markers show the mean of the estimated standard deviations computed in decile bins, and orange dots show the individual standard deviations for the amortized method.

5. REFERENCES

[1] H.W. Siesler, "Vibrational Spectroscopy," *Reference Module in Materials Science and Materials Engineering*, 2016.

[2] Tommy Sonne Alstrøm, Mikkel Nørgaard Schmidt, Tomas Rindzevicius, Anja Boisen, and Jan Larsen, "A pseudo-Voigt component model for high-resolution recovery of constituent spectra in Raman spectroscopy," *Proceedings of the 42nd Ieee International Conference on Acoustics, Speech and Signal Processing*, pp. 2317–2321, 2017.

[3] David Frich Hansen, Tommy Sonne Alstrøm, and Mikkel N. Schmidt, "Probabilistic signal estimation for vibrational spectroscopy with a flexible non-stationary gaussian process baseline model," *Preprint available at https://ssrn.com/abstract=4375926*, 2023.

[4] Roma Tauler, Bruce Kowalski, and Sydney Fleming, "Multivariate curve resolution applied to spectral data from multiple runs of an industrial process," *Analytical Chemistry*, vol. 65, no. 15, pp. 2040–2047, 8 1993.

[5] Seong-Heum Park, Hyeongseon Park, Hyunbok Lee, and Heung-Sik Kim, "Iterative peak-fitting of frequency-domain data via deep convolution neural networks," *Journal of the Korean Physical Society*, vol. 79, no. 12, pp. 1199–1208, Dec. 2021.

[6] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio, "Flow network based generative models for non-iterative diverse candidate generation," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, Eds. 2021, vol. 34, pp. 27381–27394, Curran Associates, Inc.

[7] Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio, "Trajectory balance: Improved credit assignment in GFlownets," in *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, Eds., 2022.

[8] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 40, pp. 1303–1347, 2013.

[9] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih, "Monte carlo gradient estimation in machine learning," *Journal of Machine Learning Research*, vol. 21, no. 132, pp. 1–62, 2020.

[10] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, , no. Ml, pp. 1–14, 2014.

[11] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," in *International Conference on Learning Representations*, 2017.