Journal of Machine Learning Research 20 (2019) 1-56 Submitted 3/18; Revised 4/19; Published 5/19

A Particle-Based Variational Approach to Bayesian Non-negative Matrix Factorization

Muhammad A Masood MASOOD@G.HARVARD.EDU
Harvard John A. Paulson
School of Engineering and Applied Science
Cambridge, MA 02138, USA

Finale Doshi-Velez FINALE@SEAS.HARVARD.EDU
Harvard John A. Paulson
School of Engineering and Applied Science
Cambridge, MA 02138, USA

Editor: Francois Caron

Abstract

Bayesian Non-negative Matrix Factorization (BNMF) is a promising approach for understanding uncertainty and structure in matrix data. However, a large volume of applied work optimizes traditional non-Bayesian NMF objectives that fail to provide a principled understanding of the non-identifiability inherent in NMF—an issue ideally addressed by a Bayesian approach. Despite their suitability, current BNMF approaches have failed to gain popularity in an applied setting; they sacrifice flexibility in modeling for tractable computation, tend to get stuck in local modes, and can require many thousands of samples for meaningful uncertainty estimates. We address these issues through a particle-based variational approach to BNMF that only requires the joint likelihood to be differentiable for computational tractability, uses a novel transfer-based initialization technique to identify multiple modes in the posterior, and thus allows domain experts to inspect a small set of factorizations that faithfully represent the posterior. On several real datasets, we obtain better particle approximations to the BNMF posterior in less time than baselines and demonstrate the significant role that multimodality plays in NMF-related tasks.

Keywords: Bayesian, Non-negative Matrix Factorization, Stein discrepancy, Non-identifiability, Transfer Learning

1. Introduction

The goal of non-negative matrix factorization (NMF) is to find a rank-R_{NMF} factorization for a non-negative data matrix X (D dimensions by N observations) into two non-negative factor matrices A and W. Typically, the rank R_{NMF} is much smaller than the dimensions and observations ($R_{\text{NMF}} \ll D, N$).

$$X \approx AW \quad | \qquad X \in \mathbb{R}_+^{D\times N}, \quad A \in \mathbb{R}_+^{D\times R_{\text{NMF}}}, \quad W \in \mathbb{R}_+^{R_{\text{NMF}}\times N}$$

The linear, additive structure of these non-negative factor matrices makes NMF a popular unsupervised learning framework for discovering and interpreting latent structure in data. Each observation in the data X is approximated by an additive combination of the R_{NMF} columns of A with the combination weights given by the column of W corresponding to that observation. In this way, the basis matrix A provides a part-based representation of the data and the weights matrix W provides an R_{NMF}-dimensional latent representation of the data under this part-based representation.

The ability to easily interpret NMF solutions in this way has made them appealing in many applied areas. A few applications of NMF include understanding protein-protein interactions (Greene et al., 2008), topic modeling (Roberts et al., 2016), hyperspectral unmixing (Bioucas-Dias et al., 2012), polyphonic music transcription (Smaragdis and Brown, 2003), discovering molecular pathways from genomic samples (Brunet et al., 2004), and summarizing activations of a neural network for greater interpretability (Olah et al., 2018).

However, the analysis and interpretation of latent structure in a dataset via NMF is affected by the possibility that several non-trivially different pairs of A, W may reconstruct the data X equally well. This non-identifiability of the NMF solution space has been studied in detail in the theoretical literature (Pan and Doshi-Velez, 2016; Donoho and Stodden, 2003; Arora et al., 2012; Ge and Zou, 2015b; Bhattacharyya et al., 2016), and domain experts using NMF as a tool have noticed this issue as well. Greene et al. (2008) use ensembles of NMF solutions to model chemical interactions, while Roberts et al. (2016) conduct a detailed empirical study of multiple optima in the context of extracting topics from large corpora.

Bayesian approaches to NMF promise to characterize this parameter uncertainty in a principled manner by solving for the posterior $p(A, W|X)$ given prior $p(A, W)$ and likelihood $p(X|A, W)$ e.g. Schmidt et al. (2009); Moussaoui et al. (2006). Having a representation of uncertainty in the parameters of the factorizations can assist with the proper interpretation of the factors, allowing us to place low or high confidence on parameters of the factorization. However, computational tractability of inference limits the application of the Bayesian approach. Uncertainty estimates obtained from current Bayesian methods are often of limited use: variational approaches (e.g. Paisley et al. (2015); Hoffman and Blei (2015)) typically underestimate uncertainty and fit to a single mode; sampling-based approaches (e.g. Schmidt et al. (2009); Moussaoui et al. (2006)) also rarely switch between multiple modes and often require many thousands of samples for meaningful uncertainty estimates.

As a result of the limitations of current Bayesian approaches, domain experts tend to rely on non-Bayesian approaches to characterize uncertainty in NMF parameters. For example, Greene et al. (2008); Roberts et al. (2016); Brunet et al. (2004) all use random restarts to find multiple solutions.[1] Random restarts have no Bayesian interpretation (as they depend on the basins of attraction of each mode), but they do often find multiple optima in the objective that can be used to understand and interpret the data.

Contributions In this work, we present a transfer-learning approach that remains faithful to a principled Bayesian framework and can efficiently identify multiple, disconnected modes for any differentiable prior and likelihood model. Our transfer-learning based approach provides high-quality and diverse NMF initializations to seed a particle-based approximation to the Bayesian NMF (BNMF) posterior. We demonstrate our inference approach on two different BNMF models: first, the common exponential-Gaussian model; second, a novel

[1] Random restarts involve repeating an optimization procedure with different starting points that are independently sampled.

model that corresponds more closely with the desires of domain experts. Through our experiments, we demonstrate that:

- On a large number of real-world datasets, our particle-based posterior approximations consistently outperform baselines in terms of both posterior quality and computational running time.
- Our approach allows us to produce relatively small (less than 100 NMFs) sets of particles that belong to multiple modes of the posterior landscape, have distinct interpretations, and exhibit variability in performance on downstream tasks—all of which may be essential for a domain expert to inspect and understand the full solution space.
- Our novel practitioner-friendly BNMF model involves a new scale-fixing prior that removes many uninteresting multiple optima and captures the kinds of loss-insensitive regions that are important in many applications. Inference in this non-conjugate model is significantly more challenging than with more standard BNMF models, but our approach handles this case with ease.

2. Inference Setting

The general process of Bayesian modeling consists of three main parts. First, we must select a model (a likelihood and prior). Next, we perform inference on the model given data (under some objective). Finally, we evaluate the quality of the inference. The main innovation in this work is a novel transfer-based approach to the inference phase (Section 4). Along the way, we also introduce a novel model for BNMF that is more closely aligned to what domain experts desire from NMFs (Section 5.2).

When performing inference, we must choose how we will approximate the true posterior $p(A, W|X)$. For notational simplicity, let θ represent NMF parameters (A, W). We approximate the full BNMF posterior $p(\theta|X)$ with a discrete variational distribution $q(\theta|\theta_{1:M}, w_{1:M})$ that has M different point-masses θ_m. Each θ_m represents a different NMF solution's full set of parameters: $\theta_m = \text{vec}[A_m^T, W_m]$, and is assigned probability mass w_m. The functional form of the variational distribution is given by:

$$\begin{gathered} p(\theta|X) \approx q(\theta|\theta_{1:M}, w_{1:M}) = \sum_{m=1}^{M} w_m \delta(\theta - \theta_m) \\ \text{s.t } w_{1:M} \in \Delta^{M-1}, \quad \text{where} \quad \theta_m = \text{vec}[A_m^T, W_m] \end{gathered} \tag{1}$$

where δ is the Dirac delta distribution and Δ^{M-1} is the probability simplex in $\mathbb{R}^M$. Particle-based approximations are attractive to domain experts because each sample represents something that they can inspect and understand.

While there exist many methods for particle-based approximations (Monte Carlo, Sequential Monte Carlo, Markov Chain Monte Carlo), these techniques often only enjoy theoretical guarantees in the limit of infinite or very large samples. Recent work in Stein discrepancy evaluation (Liu and Feng (2016); Chwialkowski et al. (2016); Gretton et al. (2006); Liu et al. (2016); Gorham and Mackey (2015); Ranganath et al. (2016); details in

Section 3) now enables us to measure the quality of an *arbitrary* finite collection as a posterior approximation.[2] As such, it opens the door to entirely new classes of particle-generation techniques, where traditional conditions, such as detailed balance, are now replaced with minimizing the Stein discrepancy $\mathbb{S}_p(q)$[3] between the true BNMF posterior $p(\theta|X)$ and the discrete approximation $q(\theta|\theta_{1:M}, w_{1:M})$:

$$q^*(\theta|\theta_{1:M}, w_{1:M}) = \text{argmin}_{q(\theta|\theta_{1:M}, w_{1:M})} \mathbb{S}_{p(\theta|X)}(q(\theta|\theta_{1:M}, w_{1:M})) \text{ s.t. } w_{1:M} \in \Delta^{M-1} \quad (2)$$

As with all variational inference problems, the problem of posterior inference is now reduced to the problem of optimizing the objective above; we are free to explore any method for producing settings $\{\theta_{1:M}, w_{1:M}\}$ to minimize the Stein discrepancy to the true posterior.

In the following, we observe that the task of minimizing this Stein discrepancy often depends on producing high-quality, diverse factorization collections $\theta_{1:M}$ and determining their associated weights $w_{1:M}$. In Section 4, we introduce a transfer-learning based approach to efficiently suggest a diverse collection of particles and optimize their associated weights. We describe traditional BNMF as well as a novel threshold-based NMF model, and discuss their merits in the context of our approach in Section 5. Experimental details including parameter choices for our approach as well as description of baselines and evaluation metrics is provided in Section 6. In Section 7, we compare our approach to more traditional particle-based approaches (MCMC), more naive ways of generating candidate particle collections, as well as directly attempting to optimize the Stein objective above. We evaluate the quality of different posterior approximations both based on their Stein discrepancies, likelihoods and reconstruction on held-out data.

3. Background

Bayesian Non-negative Matrix Factorization In BNMF, we define a prior $p(A, W)$ and a likelihood $p(X|A, W)$ and seek to characterize the posterior $p(A, W|X)$. These are related by Bayes' rule:

$$p(A, W|X) = \frac{p(X|A, W)p(A, W)}{p(X)}$$

There exist many options for the choice of prior and likelihood (e.g., exponential-Gaussian, (Paisley et al., 2015; Schmidt et al., 2009), Gamma Markov chain priors (Dikmen and Cemgil, 2009) and volume-based priors (Arngren et al., 2011)). The likelihood and prior choices are often chosen to have good computational properties (e.g. the resulting partial conjugacy of the exponential-Gaussian model). One advantage of our work is that we do not require the computationally convenient priors for inference.

Transfer learning The field of transfer learning aims to leverage models and inference applied to one problem to assist in solving related problems. It is of practical value because

[2] While the popular Kullback-Leibler divergence requires comparing the ratio of probability densities or probability masses, the Stein discrepancy can be used to compare a particle-based collection defined by probability masses with a continuous target distribution.

[3] This notation refers to the Stein discrepancy (a variational objective) between two distributions p and q. For a precise definition, see Section 3

there may be an abundance of data and computational resources for one problem but not another (see Pan and Yang (2010) for a survey). In this work, we shall use the solutions to BNMF from small, synthetic problems to help solve much larger NMF problems.

Stein discrepancy The Stein discrepancy $\mathbb{S}_p(q)$ is a divergence from distributions $q(\theta)$ to $p(\theta)$ that only requires sampling from the variational distribution $q(\theta)$ and evaluating the score function of the target distribution $p(\theta)$. The Stein discrepancy is computed over some class of test functions $f \in \mathcal{F}$ and satisfies the closeness property for operator variational inference (Ranganath et al., 2016): it is non-negative in general and zero only for some equivalence class of distributions $q \in \mathcal{Q}_0$. For a rich enough function class, the only distribution for which the Stein discrepancy is zero is the distribution p itself. For discrete distributions like our $q(\theta|\theta_{1:M}, w_{1:M})$ from Section 2, the approximation quality to the posterior distribution $p(\theta|X)$ of interest can be analytically computed using recent advances in Stein discrepancy evaluation with kernels (Liu and Feng, 2016; Chwialkowski et al., 2016; Gretton et al., 2006; Liu et al., 2016; Gorham and Mackey, 2015). The Stein discrepancy is related to the maximum mean discrepancy (MMD): a discrepancy that measures the worst-case deviation between expectations of functions $h \in \mathcal{H}$ under p and q (Gretton et al., 2006).

$$\mathrm{MMD}(\mathcal{H}, q, p) := \sup_{h \in \mathcal{H}} \mathbb{E}_{\theta \sim q}[h(\theta)] - \mathbb{E}_{\theta' \sim p}[h(\theta')]$$

The Stein operator $\mathcal{T}_p$ corresponding to the distribution p is given by

$$(\mathcal{T}_p h)(x) := \frac{1}{p(x)} \langle \nabla, p(x)h(x) \rangle$$

and under its application, the function space $\mathcal{H}$ is transformed into another function space $\mathcal{T}_p(\mathcal{H}) = \mathcal{F}$. The advantage of applying this operator to the MMD equation is that expectations under p of any $f \in \mathcal{F}$ are zero, i.e. $\mathbb{E}_{\theta' \sim p} f(\theta')) = 0$ (Barbour and Brown, 1992). The Stein discrepancy is given by:

$$\mathbb{S}_p(\mathcal{H}, q) := \sup_{f \in \mathcal{T}_p(\mathcal{H})} (\mathbb{E}_{\theta \sim q} f(\theta))^2$$

Computing the Stein discrepancy is of particular interest when the distribution p is intractable. Evaluating the Stein discrepancy does not require expectations over p and the Stein operator $\mathcal{T}_p$ only depends on the unnormalized distribution via the score function $\nabla_\theta \log p(\theta)$.

In this work, we use a kernelized form of the Stein discrepancy. For every positive definite kernel $k(\theta, \theta')$, a unique Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ is defined. Chwialkowski et al. (2016) showed that the Stein operator applied to an RKHS defines a modified positive definite kernel $\mathcal{K}_p$ given by:

$$\begin{aligned}\mathcal{K}_p(\theta, \theta') &= \nabla_\theta \log p(\theta)^T \nabla_{\theta'} \log p(\theta') k(\theta, \theta') \\ &\quad + \nabla_{\theta'} \log p(\theta')^T \nabla_\theta k(\theta, \theta') \\ &\quad + \nabla_\theta \log p(\theta)^T \nabla_{\theta'} k(\theta, \theta') \\ &\quad + \sum_{i=1}^{d} \frac{\partial^2 k(\theta, \theta')}{\partial \theta_i \partial \theta'_i}\end{aligned} \tag{3}$$

Finally, the Stein discrepancy is simply the expectation of the modified kernel $\mathcal{K}_p$ under the joint distribution of two independent variables θ, $\theta' \sim q$.

$$\mathbb{S}_p(q) = \mathbb{E}_{\theta,\theta' \sim q}[\mathcal{K}_p(\theta, \theta')]$$

For a discrete distribution over $\theta_{1:M}$ with probability masses $w_{1:M}$ (of the form in equation 1), this can be evaluated exactly (Liu and Lee, 2016) as:

$$\begin{aligned} \mathbb{S}_p(q) &= \sum_{i,j=1}^{M} w_i w_j \mathcal{K}_p(\theta_i, \theta_j) \\ &= \mathbf{w}^T \mathbf{K} \mathbf{w} \end{aligned} \tag{4}$$

The (pure) quadratic form $\mathbf{w}^T\mathbf{K}\mathbf{w}$ is a reformulation where $\mathbf{K} \in \mathbb{R}^{M \times M}$ is the (positive definite) pairwise kernel matrix with entries $\mathbf{K}_{ij} = \mathcal{K}_p(\theta_i, \theta_j)$ and the probability masses $w_{1:M}$ are embedded into a vector $\mathbf{w} \in \mathbb{R}^{M \times 1}$. Our particle-based variational objective (equation 2) simplifies to the form in equation 4. In Section 4, we will provide a method for estimating $\theta_{1:M}$ and $w_{1:M}$ for the BNMF problem.

4. Approach

In this Section, we describe our transfer-based inference. As noted in Section 2, creating a particle-based posterior involves two distinct parts: creating a collection of candidate NMFs $\theta_{1:M}$, and then optimizing their weights $w_{1:M}$. We introduce a novel transfer-based approach that uses state-of-the-art (non-Bayesian) algorithms to efficiently generate the candidate NMFs $\theta_{1:M}$ (Section 4.1). Given $\theta_{1:M}$, we optimize the weights $w_{1:M}$ via standard convex optimization tools to minimize Stein discrepancy. (See Algorithm 1 for the full algorithm.) In Section 7, we compare our approach for generating candidate NMFs and weights to other baselines, including those that use traditional methods for particle generation (MCMC), other ways of creating candidate NMFs (and then again using a convex optimization on the weights), and gradient-based optimization of the objective.

4.1. Learning factorization parameters $\theta_{1:M}$ via Transfer Learning

A natural approach to finding the factorization parameters $\theta_{1:M}$ is to optimize for them directly via the variational objective (equation 2), however, as we shall see in Section 7, the direct approach tends to get stuck in poor local optima and is computationally expensive. Since the quality of the variational approximation is determined solely by the value of the variational objective under a given set of parameters $\theta_{1:M}$, $w_{1:M}$, we are free to employ any technique that produces a suitable collection $\theta_{1:M}$.

We observe that to minimize the Stein discrepancy, we will need solutions that are both high-quality and diverse. Random restarts have been previously used to find multiple solutions in general (Gendreau and Potvin, 2010) and for NMF in particular (Greene et al., 2008; Roberts et al., 2016; Brunet et al., 2004). These restarts can take advantage of specialized (non-Bayesian) optimization algorithms for NMF (Lee and Seung, 2001; Lin, 2007; Hsieh and Dhillon, 2011) that are widely used in applied settings to produce single factorization parameters θ_m from some initialization; there also exist algorithms to speed up

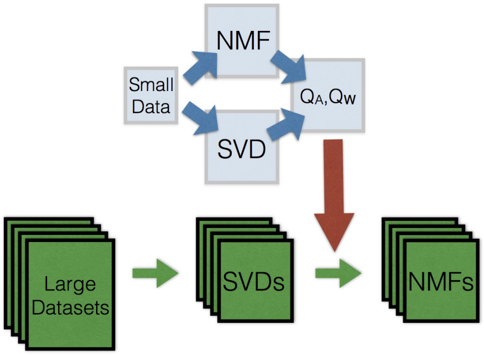


Figure 1: A schematic of the transfer learning procedure for NMF: A small dataset is used to learn transformation matrices Q_A, Q_W. We then apply these transformation matrices to multiple larger datasets (with any number of dimensions or observations) using its SVD to obtain a transfer-based initialization.

convergence of these methods (Salakhutdinov et al., 2002; Wild et al., 2004; Xue et al., 2008; Boutsidis and Gallopoulos, 2008). However, these random restarts do not take advantage of any structure of NMF; for each new NMF instance they propose random initializations from scratch. As such, many initializations may converge to the same mode—a waste of computational effort—while missing other modes (especially when the number of restarts is small).

In this Section, we introduce a transfer-based technique (which we will call Q-Transform) to speed-up, as compared to random restarts, the process of finding a diverse set of factorizations from high-density regions of the posterior. Our initializations are determined by identifying the low-rank subspace of the data (via singular value decomposition (SVD)) and then transforming it in specific ways. Figure 1 shows a schematic illustrating the idea: we generate subspace transformation matrices Q_A, Q_W from a number of small, synthetic datasets and then apply those transformations to the dataset of interest. These transformations serve as more intelligent initializations—compared to random restarts—from which to apply NMF algorithms to obtain a more diverse collection of high-quality NMFs. Because our initializations are almost always already decent NMFs, convergence is also computationally faster. To explain our Q-Transform procedure, we first define the subspace transformation matrices, then describe the method for generating transformation matrices Q_A, Q_W using synthetic data, and finally discuss how to apply them to real data sets (transfer learning).

Subspace transformations Q_A, Q_W relating SVD and NMF A low dimensional approximation for the data X can be obtained via the top R_{SVD} vectors of the SVD $A_{\text{SVD}}, W_{\text{SVD}}$. An NMF A, W of rank R_T (which may be different from R_{SVD}) also leads to an approximation of the data. The NMF factors are interpretable due to the non-negativity constraint whereas the SVD factors typically violate non-negativity. However, both approaches describe low dimensional subspaces that can be used to understand and approximate the data. These subspaces are the same when $R_{\text{SVD}} = R_T$ and the NMF is exact (i.e. $X = AW$; corresponds to Type I non-identifiability in Pan and Doshi-Velez (2016). Under

Algorithm 1 Particle-based Variational Inference for BNMF using Q-Transform

```
Input: Data $\{X\}$, Rank $\{R_{\text{NMF}}\}$, # Factorizations $M$
Step 1: Perform $M$ repetitions of Algorithm 2 to get matrices $\{Q_A^m, Q_W^m\}_{m=1}^M$ or re-use them if previously constructed
Step 2: Apply $Q$-Transform (Algorithm 3) to get Initializations $\{A_0^m, W_0^m\}_{m=1}^M$
Step 3: Apply NMF algorithm to get Factorizations $\{A^m, W^m\}_{m=1}^M$
Step 4: Apply Algorithm 5 using a given BNMF model to get weights $\{w^m\}_{m=1}^M$ for approximate posterior
Output: Discrete NMF Posterior $\{w^m, A^m, W^m\}_{m=1}^M$
```

these conditions, there exist transformation matrices Q_A, Q_W to obtain the non-negative basis and weights exactly in terms of the singular value decomposition matrices:

$$\begin{aligned} \text{If} \quad X = A_{\text{SVD}} W_{\text{SVD}} = AW \quad \text{then} \\ A = A_{\text{SVD}} Q_A \quad W = Q_W W_{\text{SVD}} \end{aligned}$$

When the data X is not an exact NMF but rather a perturbation of it (i.e. $X = AW + \epsilon$), the singular subspace of the matrix is bounded by Wedin's theorem (Wedin, 1972). We therefore still expect that there exist transformation matrices $Q_A \in \mathbb{R}^{R_{\text{SVD}} \times R_T}, Q_W \in \mathbb{R}^{R_T \times R_{\text{SVD}}}$ to yield approximations of the NMF factorizations that can be expressed in terms of the singular value decomposition matrices.

$$A_Q = A_{\text{SVD}} Q_A \approx A \quad W_Q = Q_W W_{\text{SVD}} \approx W$$

Our transfer-based strategy will involve identifying candidate matrices $A_Q \in \mathbb{R}^{D \times R_T}, W_Q \in \mathbb{R}^{R_T \times N}$ such that $A_{\text{SVD}} Q_A$ and $Q_W W_{\text{SVD}}$ are likely to be good initializations for an NMF of the data X. (Note that we assume that computing the SVD to obtain A_{SVD} and W_{SVD} from the data X is straight-forward.) We will describe the details for using these initializations below, but first we describe how we might create a collection of candidate transformation matrices Q_A, Q_W.

Generating transformations Q_A, Q_W for NMF initialization. To generate candidate transformations, we note that if we have already computed an NMF A, W for a dataset X, the appropriate transforms Q_A, Q_W can be computed by relating the SVD factors $A_{\text{SVD}}, W_{\text{SVD}}$ to A, W (e.g. via linear least squares). We propose to generate candidate transforms by using random restarts on small, synthetic datasets X_s that follow some generative model for NMF, where we can run (non-Bayesian) NMF algorithms quickly and solve for Q_A, Q_W (Algorithm 2). Multiple pairs of transformation matrices can be obtained by repeating Algorithm 2 with different random initializations to compute NMF of the synthetic data X_s, as well as by generating multiple synthetic datasets (see Section 8.1 for experiments and discussion of alternate generation procedures). Since the transformations Q_A, Q_W act on the inner dimensions (columns of A_{SVD} and rows of W_{SVD}), we emphasize that they can be applied to new datasets with any number of dimensions D and number of observations N.

Algorithm 2 Generate Q-Transform Matrices

```
Input: Synthetic Data $\{X_s\}$, SVD Dimension $\{R_{\text{SVD}}\}$, Transfer Dimension $\{R_T\}$
$A_{\text{SVD}}, W_{\text{SVD}} \leftarrow$ Compute top $R_{\text{SVD}}$ SVD of $X_s$
$A_{\text{NMF}}, W_{\text{NMF}} \leftarrow$ Compute rank-$R_T$ NMF of $X_s$ using random initialization
$Q_A = \arg\min_Q \|A_{\text{NMF}} - A_{\text{SVD}}Q\|_F$ via linear least squares
$Q_W = \arg\min_Q \|W_{\text{NMF}} - QW_{\text{SVD}}\|_F$ via linear least squares
Output: $Q_A$, $Q_W$
```

Algorithm 3 Apply Q-Transform

```
Input: Real Data $\{X\}$, SVD Rank $\{R_{\text{SVD}}\}$, NMF Rank $\{R_{\text{NMF}}\}$, Transformation Matrices $\{Q_A, Q_W\}$
$A_{\text{SVD}}, W_{\text{SVD}} \leftarrow$ Compute top $R_{\text{SVD}}$ SVD of $X$
$\tilde{A_0} = A_{\text{SVD}}Q_A, \quad \tilde{W_0} = Q_W W_{\text{SVD}}$
$A_0, W_0 \leftarrow$ Apply non-negativity and fix dimensions: Algorithm 4 $(\tilde{A_0}, \tilde{W_0}, R_{\text{NMF}})$
Output: $A_0$, $W_0$
```

Creating initializations for a new dataset. Given the top SVD factors of a new dataset $A_{\text{SVD}}, W_{\text{SVD}}$, we apply the Q-Transform (Algorithm 3) which multiplies SVD factors by the Q_A, Q_W matrices and adjusts entries of $A_{\text{SVD}}Q_A$ and $Q_W W_{\text{SVD}}$ to ensure non-negativity and correct dimensions using Algorithm 4 to obtain initializations A_0, W_0 that can be used as input for any standard (non-Bayesian) NMF algorithm (e.g. Cichocki and Phan (2009), Févotte and Idier (2011)). Algorithm 4 ensures that all values in the initialization A_0, W_0 are non-negative as well as provides a way to pad the initialization if the size R_T of the transforms Q_A, Q_W are smaller than the desired NMF rank R_{NMF}. The latter is an important point: in Section 8.3 we find that it is often the first few dimensions of the transformation that contain transferable information, and the rest provide little benefit. This observation also allows us to use transforms of some rank R_T on problems with a range of desired NMF ranks R_{NMF}. Finally, running the algorithm gives us a set of factorization parameters $\theta_m = \text{vec}[A_m^T, W_m]$ that we may (or may not) ultimately decide to keep in our approximation of the true posterior.

In the experiments in Section 7, we find that knowledge from these transformations Q_A, Q_W can be transferred to real datasets by re-using them to relate the top SVD factors of other datasets to high quality, approximately non-negative factorizations.[4]

4.2. Learning weights $w_{1:M}$ given parameters $\theta_{1:M}$

To infer the weights corresponding to a given factorization collection $\theta_{1:M}$, we minimize the Stein discrepancy (Algorithm 5) subject to the simplex constraint on the weights. This process involves first computing the pairwise kernel matrix[5] $\mathbf{K}$ using the kernel $\mathcal{K}_p$ in equation 3. The objective function is convex and can be solved using standard convex optimization solvers. Given point-masses $\theta_{1:M}$, this framework can be employed to infer weights

[4] Code and demonstrations at https://github.com/dtak/Q-Transfer-Demo-public-/

[5] As the kernel $\mathcal{K}_p$ is positive definite, $\mathbf{K}$ is also positive definite.

Algorithm 4 Initialization Adjustment

```
Input: Approximation matrices $\{A_Q, W_Q\}$, NMF Rank $\{R_{\text{NMF}}\}$
$\tilde{A}_0 \leftarrow$ Absolute Value($A_Q$)
$\tilde{W}_0 \leftarrow$ Absolute Value($W_Q$)
Transfer Rank $R_T = \#$ Columns of $A_Q$
if NMF Rank $R_{\text{NMF}} >$ Transfer Rank $R_T$ then
    $r = R_{\text{NMF}} - R_T$
    Pad $\tilde{A}_0, \tilde{W}_0$ with matrices $M_{D\times r}$ and $M_{N\times r}$ having small random entries so that initializations are the correct dimensions and matrices $M_{D\times r}, M_{N\times r}$ have little effect of the product $\tilde{A}_0\tilde{W}_0$.
    $A_0 \leftarrow [\tilde{A}_0, M_{D\times k}]$
    $W_0 \leftarrow [\tilde{W}_0^T, M_{N\times k}]^T$
else if NMF Rank $R_{\text{NMF}} <$ Transfer Rank $R_T$ then
    Pick the top $R_{\text{NMF}}$ columns of $\tilde{A}_0$ and rows of $\tilde{W}_0$
    $A_0 \leftarrow \tilde{A}_0[:, 0 : R_{\text{NMF}}]$
    $W_0 \leftarrow \tilde{W}_0[0 : R_{\text{NMF}}, :]$
end if
```

Algorithm 5 Kernelized Stein inference for discrete approximations to posterior

```
Input: Particles $\theta_{1:M}$, Score $\nabla_\theta \log p(\theta)$, RKHS $\mathcal{H}$ defined by kernel $k$
Step 1: Compute pairwise kernel matrix $\mathbf{K_{i,j}} = \mathcal{K}_p(\theta_i, \theta_j)$ (from equation 3)
Step 2: Find probability masses that minimize the Stein discrepancy for the given point-masses: $\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbf{w}^T\mathbf{K}\mathbf{w} \quad \text{s.t.} \quad \mathbf{w} \in \Delta^{M-1}$ via standard convex optimization.
Output: Probability masses $\mathbf{w}^*$
```

for discrete approximations to any posterior for which the score function $\nabla_\theta \log p(\theta)$ can be computed.

5. BNMF Models

Section 4 outlined a general procedure for producing a particle-based approximation to the BNMF posterior using transfer learning. In Section 7, we compare our approach to other particle-based approaches for BNMF. However, before going to the results, we first describe the two BNMF models (below) as well as our experimental procedure (Section 6).

The first model we shall use in our experiments is the commonly-used exponential-Gaussian model. This model is computationally convenient to use (e.g. Schmidt et al. (2009) derive a Gibbs sampler for this model) but the scale-flexible prior allows for multiple optima that are essentially the same factorization, and the Gaussian likelihood severely penalizes solutions of differing quality even when all solutions may be far from perfect reconstructions. These properties make this popular model less desirable from the perspective of a domain expert seeking to understand their data. The second model is a novel threshold-based model with a scale-fixing prior that at once removes scale ambiguities and allows for the kinds of likelihood ambiguities that practitioners expect—in particular, when the NMF is already an approximation of the data, solutions with different absolute likelihoods but whose relative differences are small compared to the magnitude of the likelihood may be considered similar by a practitioner (Roberts et al., 2016). Our threshold-based likelihood model allows the practitioner to choose what levels of error are effectively the same for their purposes.

Before continuing, we emphasize again that our transfer-based inference approach can be applied to *any* BNMF model; in this paper we demonstrate our approach on the following two models because together they include a standard model often-used in the machine learning community and a novel model of interest to the practitioner community. Importantly, because our inference approach decouples the process of model choice, particle generation, and particle weighting, we use the same particle generation process (non-Bayesian optimization algorithms using the Frobenius objective) for both models. In Section 7, we demonstrate empirically that this particle generation process is robust enough; that is, we do not require processes tuned to each model.

5.1. Exponential-Gaussian Model for BNMF

The commonly used exponential-Gaussian BNMF model uses a Gaussian likelihood and exponential priors for the basis and weights matrices:

$$p_{\mathcal{N}}(X|A,W) = \prod_{d,n} \mathcal{N}(X_{d,n}, (AW)_{d,n}, \sigma_X^2)$$

$$p(A) = \prod_{d=1}^{D}\prod_{r=1}^{R} p(A_{d,r}), \quad A_{d,r} \sim \mathrm{Exp}(\lambda_{d,r})$$

$$p(W) = \prod_{n=1}^{N}\prod_{r=1}^{R} p(W_{r,n}), \quad W_{r,n} \sim \mathrm{Exp}(\lambda_{r,n})$$

As derived in Schmidt et al. (2009), the combination of exponential priors and Gaussian likelihoods results in element-wise conjugate parameter updates; in general, this model enjoys relatively straightforward inference approaches.

That said, as noted above, the exponential-Gaussian has several drawbacks from the perspective of a domain expert seeking to interpret their data via NMF. First, especially in settings where the model is misspecified (which will almost always be the case), the reconstruction error of even the best factorization may be relatively large. Even so, the Gaussian likelihood will tend to make the posterior highly peaked around the MAP solution—and exclude factorizations of only slightly worse (relative approximation) quality with respect to the overall error. However, domain experts may have found those factorizations interesting, as they have about the same relative error. Second, the exponential prior allows for some amount of uncertainty simply due to scale, which is typically uninteresting for domain experts. In the following, we introduce a model that addresses both of these shortcomings; because our transfer-based inference approach does not require conjugacy, we will be able to efficiently compute approximate posteriors for such more complex models.

5.2. Threshold-based, Scale-Fixing Model for BNMF

The procedure described in Algorithm 1 for finding a discrete approximation to the BNMF posterior does not depend on any special properties (such as conjugacy) and only requires the joint density $p(X, W, A)$ to be differentiable in order to make inference tractable. Such flexibility is important as different applied domains use different notions of factorization quality: squared Euclidean distance is commonly used in hyperspectral unmixing (Bioucas-Dias et al., 2012), Kullback-Leibler divergence in image analysis (Lee and Seung, 2001) and Itakura-Saito divergence in music analysis (Févotte et al., 2009).

A common theme in many applied domains is that small differences in factorization quality may not be important if all factorizations have some large level of approximation error. In such cases, domain experts may be interested in all of these solutions (Roberts et al., 2016). At the same time, solutions that are different only in scale are likely uninteresting. Below, we present a novel prior and likelihood that reflect these application-specific preferences of practitioners in a Bayesian framework. In particular, our model class allows domain experts to take any application-specific notion of a high-quality factorization—conjugate or not—and put it into a Bayesian context.

Likelihood: Soft Insensitive Loss Function (SILF) over NMF objectives We define a likelihood that is maximum (and flat) in the region of high quality factorizations and decays as factorization quality decreases. To do so, we use the soft insensitive loss function (SILF) (Chu et al., 2004): a loss function defined over the real numbers $\mathbb{R}$, where the loss is negligible in some region around zero defined by the insensitivity threshold ϵ, and grows linearly outside that region (see figure 2). A quadratic term depending on the smoothness parameter β, makes the transition between the two main regions smooth. This transition region has length 2β, making smaller values of β correspond to sharper transitions between the flat and linear loss regions. We adapt the SILF from (Chu et al., 2004) to only be defined over the non-negative numbers $\mathbb{R}_+$ (as is typical with NMF objectives) and define it as:

$$\mathrm{SILF}_{\epsilon,\beta}(y) = \begin{cases} 0 & 0 \leq y \leq (1-\beta)\epsilon \\ \frac{(y-(1-\beta))^2}{4\beta\epsilon} & (1-\beta)\epsilon \leq y \leq (1+\beta)\epsilon \\ y - \epsilon & y \geq (1+\beta)\epsilon \end{cases}$$

To form the likelihood, we apply the SILF loss to an NMF objective $f_X(A, W)$ to give:

$$P(X|W, A) = \frac{1}{Z} e^{(-C \times \mathrm{SILF}_{\epsilon,\beta}(f_X(A,W)))} \tag{5}$$

We emphasize that the SILF-based likelihood allows the domain expert to use an NMF objective $f_X(A, W)$ that is best suited to their task and can specify a threshold under that objective for identifying high-quality factorizations. Once an NMF objective is chosen, the domain expert can easily choose appropriate parameters for the SILF-based likelihood since the parameters (insensitivity factor ϵ and smooth transition factor β) are interpretable and the likelihood can be visually inspected (as a one-dimensional function of a chosen NMF objective) to validate parameter choices.

Prior: uniform over basis and unambiguous in factorization scaling Often, domain experts do not have specific notions of what the prior over factorizations should be. However, prior distributions can have a large effect. These effects are undesirable if the prior was chosen for computational convenience rather than based on some true knowledge about the problem. Another concern is that NMF has a scaling and permutation ambiguity

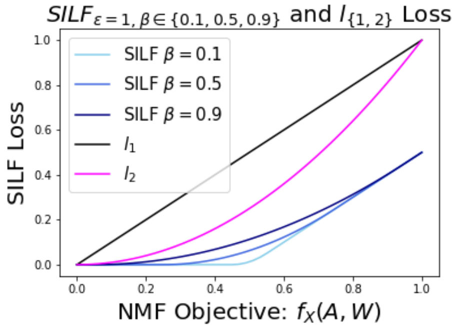


Figure 2: A comparison of SILF loss and commonly used l_1, l_2 loss functions. The SILF insensitivity parameter ϵ is set to 0.5, and the smooth transition factor β is varied. Small values of β lead to sharp transition in the SILF loss profile whereas the transition is less abrupt for large values of β. In contrast, other popular loss functions such as l_1 or l_2 do not have insensitive regions, and in the case of NMF, treat the objective function as the sole guide for factorization quality.

that is uninteresting in practice:

$$AW = \underbrace{ASP}_{\widetilde{A}} \underbrace{(SP)^{-1}W}_{\widetilde{W}} \quad \text{where S is a positive diagonal matrix, P is a permutation matrix} \tag{6}$$

Depending on the priors chosen, this ambiguity can add redundancy to the posterior distribution.

To facilitate exploration of the space of distinct high-quality factorizations, we propose an NMF prior that eliminates redundancy due to scale and is also uniform over the space of factorizations. Specifically, we let each column of the basis matrix A_r be generated by a symmetric Dirichlet distribution with parameter $\alpha = 1$. This prior determines a unique scale of the factorization and is uniform over the basis matrix A for that scaling. For W, we use a prior where each entry $W_{r,n}$ is i.i.d from an exponential distribution with parameter $\lambda_{r,n}$. The exponential distribution has support over all $\mathbb{R}_+$ ensuring that any weights matrix W corresponding to a column-stochastic basis matrix A is a valid parameter setting under our model, and that the posterior is proper.

$$p(A) = \prod_{r=1}^{R} p(A_r), \quad A_r \sim \text{Dir}(\mathbf{1}_D)$$

$$p(W) = \prod_{n=1}^{N} \prod_{r=1}^{R} p(W_{r,n}), \quad W_{r,n} \sim \text{Exp}(\lambda_{r,n})$$

6. Experimental Setup

In this Section, we provide details of our experimental settings and parameter choices, and describe our baseline algorithms and datasets. Our experiments are performed on a wide array of benchmark NMF datasets as well as on Electronic Health Records (EHR) data of patients with Autism Spectrum Disorder (ASD) that is of interest to the medical community (see quantitative and qualitative results in Section 7).

6.1. Model, Evaluation, and Inference Settings

Model: exponential-Gaussian model parameters: We set the standard deviation σ_X to be equal to the empirical standard deviation of a reference NMF. The exponential parameter was set to one for each entry in the basis and weights matrices ($\lambda_{d,r} = \lambda_{r,n} = 1$).

Model: SILF model parameters: While any objective can be put into the SILF likelihood, in the following, we used the squared Frobenius objective $f_X(A, W) = \|X - AW\|_F^2$. To set the threshold parameter ϵ for each dataset, we use an empirical approach where we find a collection of 50 high-quality factorizations under default settings of scikit-learn (Pedregosa et al., 2011). The objective function is evaluated for each of them $\{f_i\}_{i=1}^{50}$ and $\epsilon = 1.2 \max_i f_i$. We set the remaining SILF likelihood sensitivity parameters $\beta = 0.1$, $C = 2$. For the prior, we identically set the exponential parameter for each entry: $\lambda_{r,n} = 1$.

Inference: Generating Q-transform matrices for transfer: For the Q-Transform initializations, we set the transfer rank and SVD rank $R_T = R_{\text{SVD}} = 3$. We generated twenty sets of synthetic data $X_s \in \mathbb{R}_+^{12\times 12}$ using non-negative matrices of rank R_T with truncated Gaussian noise. For each synthetic dataset, we find five pairs of transformation matrices through random restarts. In *all* our experiments, the *same* set of $M_{\max} = 100$ pairs of transformation matrices $\{Q_A^m, Q_W^m\}_{m=1}^{100}$ are applied to each of the real datasets.

Inference: Solver for inferring weights $w_{1:M}$: The optimization for the weights $w_{1:M}$ (Step 2 in Algorithm 5) is carried out using the Splitting Conic Solver (SCS) in the convex optimization package CVXPY (Diamond and Boyd, 2016).

Inference and evaluation: Stein discrepancy base RKHS and parameters: The Stein discrepancy for our variational objective requires a function space to optimize over. This optimization over the function space has an analytical solution when a Reproducing Kernel Hilbert Space (RKHS) is used. Gorham and Mackey (2017) show that the Inverse Multiquadric (IMQ) kernel is a suitable kernel choice for Stein discrepancy calculations as it detects non-convergence to posterior[6] for $c > 0$ and $b \in (-1, 0)$.

$$k_{\text{IMQ}}(\theta_i, \theta_j) = (\|\theta_i - \theta_j\|^2 + c^2)^b$$

Since the length scales of the basis and weights matrix differ, we define a kernel via a linear combination of two IMQ kernels defined separately over the basis A and weights W.

$$k([A_1, W_1], [A_2, W_2]) = \frac{1}{2\gamma_A}(\|A_1 - A_2\|^2 + c_A^2)^{b_A} + \frac{1}{2\gamma_W}(\|W_1 - W_2\|^2 + c_W^2)^{b_W} \qquad (7)$$

Here $\gamma_A = (c_A^2)^{b_A}$ and similarly $\gamma_W = (c_W^2)^{b_W}$ are scaling factors that ensure the kernel takes values between 0 and 1. In general, across our datasets, the Dirichlet prior on the

[6] Gorham and Mackey (2017) also prove that popular Gaussian and Matern kernels fail to detect non-convergence when the dimensionality of its inputs is greater than 3.

basis matrix induces a small length scale for A and a larger length scale for the weights W. We uniformly set $c_A = 1 \times 10^{-2}$, $c_W = 1 \times 10^3$ and $b_A = b_W = -0.5$ across all our datasets.

We note that choosing sensible values for these parameters—and validating them—is important. Kernel parameters that induce length scales that are too small or too large give rise to a similarity measure that either considers all factorizations completely dissimilar or completely similar respectively. In our experiments, our kernel choice gives rise to a similarity measure that distinguishes across collections of factorizations obtained from different algorithms. Our kernel similarity analysis shows agreement with difference between factorizations as measured by the Frobenius distances between basis and weights matrices (see figures 24, 25, 26 in Appendix). The range in kernel similarity values and its agreement with alternative measures indicates that our parameter choices for the kernel are reasonable and fairly robust.[7]

Evaluation: Measuring computational time In experiments, we keep track of the time taken (initialization and optimization) to produce each of the $M_{\text{max}} = 100$ factorizations. We sample collections of size $M = \{5, 25, 50\}$ from these factorizations and report the total time taken to produce the factorizations in the collection alongside reporting the Stein Discrepancies for the approximate BNMF posteriors.

For the baselines below, the reported runtimes correspond to time taken to generate NMFs $\{\theta_m\}_{m=1}^M$ in the approximate posterior. For initialization approaches this corresponds to the time taken to generate the initialization and subsequent optimization time. To allow for a transparent comparison of the performance of these initialization approaches with MCMC and gradient-based algorithms, we report runtimes at various points in the duration of the MCMC chain and for the gradient-based algorithms. For more details on measuring computational time, see Appendix E in supplementary materials.

6.2. Baselines

In the previous Section, we described the implementation details for our transfer-based inference approach. In this Section, we describe implementation details for three classes of baselines for our experiments: MCMC, which represents standard practice for generating particle-based posteriors; gradient-based approaches which directly minimize the Stein variational objective, which represent our main competitors; and alternate initialization approaches, which represent simpler ablations on our approach.

Markov Chain Monte Carlo baselines MCMC approaches involve sampling from a Markov Chain whose stationary distribution is the posterior of interest, and are often considered the gold-standard for approximating posterior distributions (as opposed to variational methods). That said, for a finite sample size, MCMC will still be approximate—and thus we must still evaluate its quality with respect to the Stein objective. In this work, we consider two different MCMC baselines:

- **Hamiltonian Monte Carlo (HMC)** Our HMC was initialized with an NMF obtained using the default settings of scikit-learn (Pedregosa et al., 2011) (warm start), and adaptively selects the step-size using the procedure outlined in Neal et al. (2011).

[7] Factorizations across our different datasets have different scales but the kernel parameters were fixed across all datasets.

We run the chain for a total of 10000 samples and at various intermediate points thin it to $M = \{5, 25, 50\}$ factorizations and compute the Stein discrepancy using Algorithm 5. We repeat this experiment three times to capture variability in the performance of the HMC.

For our scale-fixing prior in Section 5.2, we needed to simulate Hamiltonian dynamics as defined on the manifold of the simplex. To do this, we incorporate a reparametrization trick (Betancourt, 2012; Altmann et al., 2014) to sample under the column-stochastic (simplex) constraints of the basis matrix A, and a mirroring trick (Patterson and Teh, 2013) for sampling from the positive orthant for the weights matrix W.

- **Gibbs Sampling**. Only the exponential-Gaussian model admits a conjugate form for straight-forward Gibbs sampling. For experiments using the exponential-Gaussian, we use the same number of samples and thinning factor as with HMC for a Gibbs sampler. Similarly to the HMC baseline, the Gibbs sampler was also initialized with an NMF obtained using the default settings of scikit-learn (Pedregosa et al., 2011) (warm start).

Gradient-based baselines Gradient-based baselines optimize the collection of factorizations directly via gradient descent on the Stein variational objective. They represent the class of inference approaches most similar to ours. Gradient-based approaches typically require fixing the size of the collection. In our experiments, we set the size of this collection to be equal to $M = 5$. Due to the large memory requirement of running this algorithm with automatic differentiation using autograd (Maclaurin et al., 2015), we were unable to run these algorithms for larger M. We impose scaling and non-negativity constraints after every gradient step (for a total of 2000 steps) and keep track of the Stein discrepancy in relation to the algorithm's runtime. The experiment is repeated three times to capture variability in its performance over multiple iterations. We use the following three algorithms:

- **SVGD**: Stein Variational Gradient Descent is a functional gradient descent algorithm (Liu and Wang, 2016) that optimizes a collection of particles (factorizations) to approximate the posterior. We replace the RBF kernel from the original work with the more principled IMQ-based kernel defined in equation 7.
- **SVGD-Q** is a variant were we initialize SVGD with the Q-Transform.
- **DSGD**: Direct Stein Gradient Descent is a variant where we replace the functional gradient descent of SVGD with the gradient of the Stein discrepancy (using automatic differentiation (Baydin et al., 2015; Maclaurin et al., 2015)).

Initialization-based baselines Our Q-transform approach can be thought of as an initialization approach: we provide a way of creating a collection of particles that we believe are likely to be representative of the posterior. Our main algorithm can be run with any process for creating the collection (step 2 of Algorithm 1). Our final set of baselines considers other alternatives to creating the collection.

- **Random restarts** Our random restart initializations for NMF in scikit-learn (Pedregosa et al., 2011) set each entry of the factors A, W as independent, coming from a truncated standard normal distribution. These entries are all scaled by $\eta = \sqrt{\frac{1}{R_{\text{NMF}}} \sum_{D,N} X_{d,n}}$ and are given by: $A^0_{d,k}, W^0_{k,n} \sim \eta|\mathcal{N}(0,1)|$.

- **NNDSVDar** NNDSVDar is a variant of a popular initialization technique called Nonnegative Double Singular Value Decomposition (NNDSVD) which was introduced by Boutsidis and Gallopoulos (2008). It is based on approximating the SVD expansion with non-negative matrices. Since the NNDSVD algorithm is deterministic, this only gives a single initialization. The NNDSVDar variant of this initialization replaces the zeros in the NNDSVD initialization with small random values. We use the scikit-learn initialization for NNDSVDar which uses a randomized SVD algorithm (Halko et al., 2011), and note that it introduces some additional variability in the initializations.

6.3. Datasets

Our datasets cover a range of different types and can be divided into three main categories (count data, grayscale face images and hyperspectral images). The ranks for hyperspectral data are chosen according to ground truth values. In the 20-Newsgroups data, we select articles from 16 newsgroups (hence the rank 16) and for other datasets we pick a rank that corresponds to explaining at least 70 percent of the variance in the data (as measured by the SVD). Table 1 provides a description of each dataset as well as the rank used and a citation. The Autism dataset is of interest to the medical community for understanding disease subtypes in the Autism spectrum and is not publicly available. The remaining datasets are public and are considered standard benchmark datasets for NMF. In our experiments, we hold out ten percent of the observations and report performance on both provided and held-out observations.

Table 1: Datasets for NMF

Dataset	Dimension	Observations	Rank	Description
20-Newsgroups	1000	8926	16	Newspaper articles (20NG, 2013)
Autism	2862	5848	20	Patient visits (Doshi-Velez et al., 2014)
LFW	1850	1288	10	Grayscale Faces Images (LFW, 2017)
Olivetti Faces	4096	400	10	Grayscale Faces Images (Samaria, 1994)
Faces CBCL	361	2429	10	Grayscale Faces Images (CBCL, 2000)
Faces BIO	6816	1514	10	Grayscale Faces Images (Jesorsky et al., 2001)
Hubble	100	2046	8	Hyperspectral Image (Nicolas Gillis, 1987)
Salinas A	204	7138	6	Hyperspectral Image (SalinasA, 2015)
Urban	162	10404	6	Hyperspectral Image (Zhu et al., 2014)

7. Results

In this Section, we compare computational time and Stein discrepancy values for variational posteriors obtained through different algorithms. For the exponential-Gaussian model, our approach using Q-Transform is either the most-competitive or second in performance to the Gibbs sampler for this model. Under the SILF model, we find that our approach for BNMF posterior approximation using transfer learning (Q-Transform) consistently produces the highest quality posterior approximations in the shortest amount of time (see Section 6.1 for details on runtime calculation). Inspection of factorization parameters from Q-Transform reveals that the parameter uncertainty captured by the BNMF posterior approximation has meaningful consequences for interpreting and utilizing these factorizations.

In the supplement, we provide an in-depth look at our results. We report on quality metrics for both the training data (figures 18, 19, 15 and 16) as well as held-out data (figures 20 and 17); we report on multiple metrics for measuring diversity of factorizations obtained from different algorithms (figures 24, 25, 26, 21, 22 and 23). Overall, these results support the notion that the Stein discrepancy is lowest for algorithms with the most diverse collection of high-quality factorizations.

7.1. Exponential-Gaussian Model Results

In figure 3, we show the performance of our algorithm and other competing baselines across our various datasets. Overall, we note that the best approximate posteriors are produced in the shortest time either by our Q-Transform algorithm or the Gibbs sampler for this model. Using random restarts for initialization yields approximate posteriors with similar Stein discrepancies to our approach but typically takes more time. The gradient-based approaches (even Q-SVGD which is initialized with Q-Transform) rarely do well, often plateauing at much higher discrepancies.

While the likelihood term in this model is invariant to (redundant) scalings[8], a limitation is that the prior (chosen for computational convenience) is dependent on the scaling. We find that this is an undesirable feature because the posterior landscape includes infinite redundant scalings and therefore requires greater effort from the inference procedure to find appropriate scalings of factorizations. Another concern is that the likelihood model is not directly expressible in terms of whatever properties might be of interest to a practitioner. To address our concerns regarding the exponential-Gaussian model, we focus for the remainder of this work on the threshold-based model with scale-fixing prior.

7.2. SILF-based Model Results

In figure 4, we show the performance of our algorithm and other competing baselines across our various datasets. Recall that the Stein discrepancy variational objective involves terms that consider both the quality of the factorizations (as given by the score function $\nabla_\theta \log p(\theta)$) and their similarity (as given by the base RKHS kernel $k(\theta_i, \theta_j)$). The NNDSVDar initializations and thinned HMC samples lead to factorizations that are high-quality but often not diverse (see diversity analysis in supplementary material: figures 24, 25, 26). The SVGD and the DSGD are generally the worst performing algorithms. These methods are often unable to find factorization parameters that meet the quality criteria of the SILF likelihood (see quality analysis in supplementary material: figures 18 and 19). This is understandable because even using simple gradient-based approaches to find a single high-quality NMF turns out to be difficult, hence the existence of a literature on specialized algorithms for performing NMF. Our Q-transform algorithm and random restarts are able to find sam-

[8] Basis and weights matrices can be multiplied by any positive diagonal matrix and its inverse (respectively) to yield ‘new’ factors that identically reconstruct the data but differ in scale

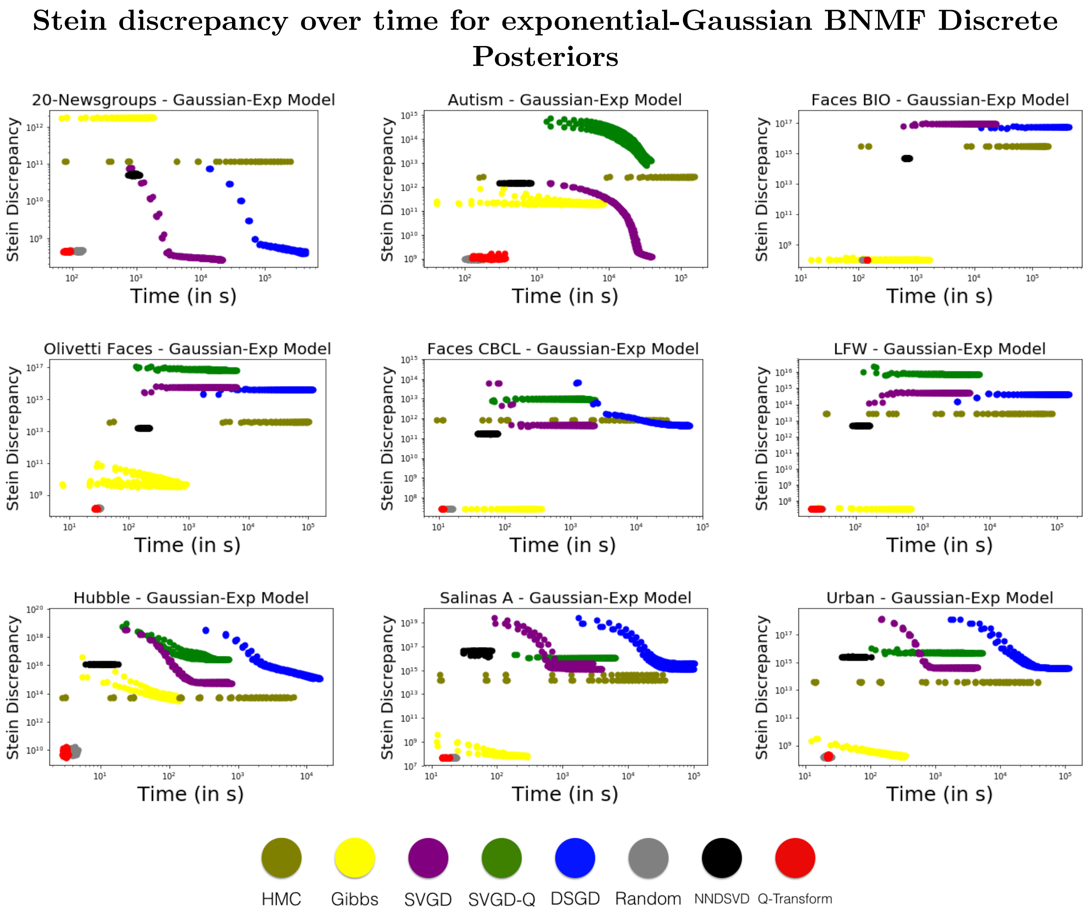
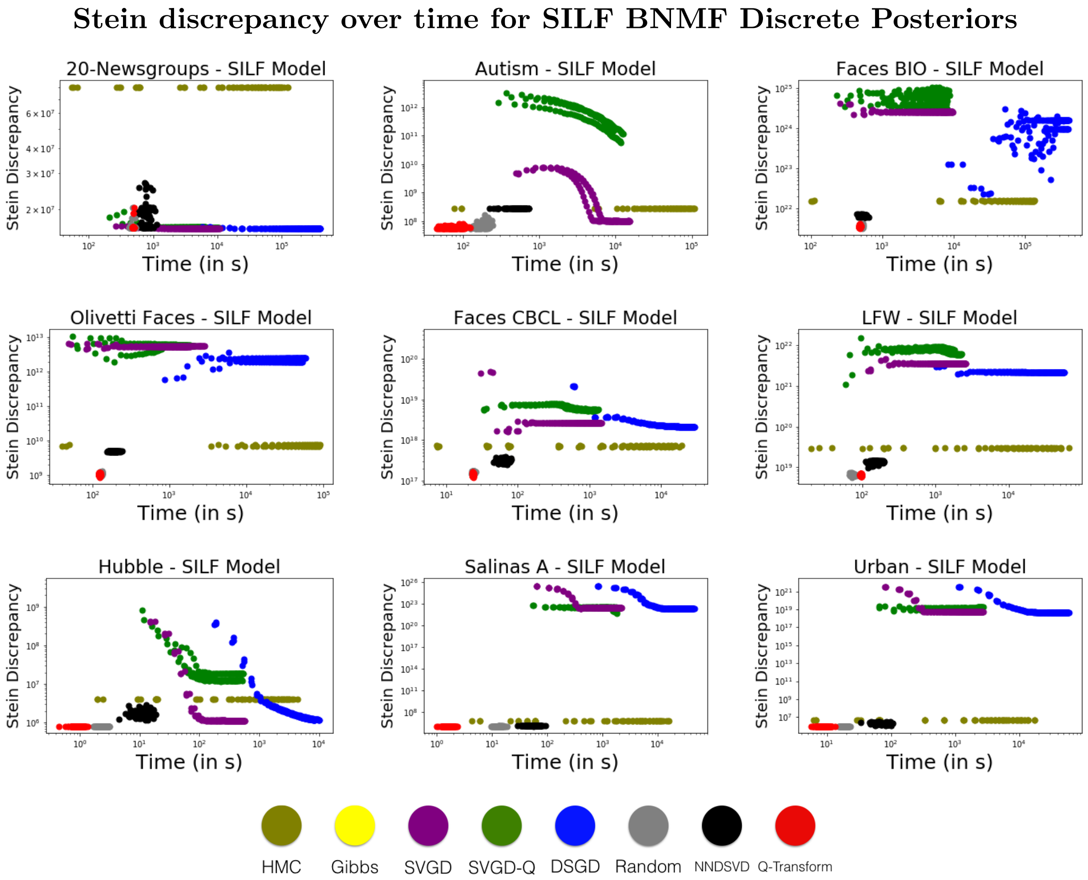


Figure 3: For each dataset we show the quality of the BNMF approximate posterior ($M = 5$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to the exponential-Gaussian BNMF (lowest Stein discrepancy) are produced either using the Q-Transform initializations (in red) for the Gibbs sampler (in yellow).

ples that are both high-quality and diverse, thus achieving the lowest Stein discrepancies; however, our Q-transform algorithm does so in the shortest time.

Figures 31 and 32 in the Appendix show results for $M = \{25, 50\}$ where Q-Transform continues to have a runtime advantage over other baselines. Additionally, for some datasets (Olivetti Faces, LFW and Faces BIO) Q-Transform also produces higher quality of the posterior approximations. Variational posteriors constructed using thinned samples from HMC significantly lack diversity as the Stein discrepancies for collections of size 5, 25 and 50 are comparable. This indicates that the HMC chain only explores a small region of the posterior distribution and can be confirmed through the diversity analysis in the Appendix (figures 24, 25, 26). Sminchisescu et al. (2007) notes that in high dimensional spaces, we expect there to be many ridges of probability as there are likely to be some directions in which the posterior density decays sharply. Alternatively, there may be several isolated modes with no connecting regions of high probability making it particularly challenging for the HMC chain to avoid getting stuck in a local mode of the BNMF posterior.



Stein discrepancy over time for SILF BNMF Discrete Posteriors

20-Newsgroups - SILF Model

Autism - SILF Model

Faces BIO - SILF Model

Olivetti Faces - SILF Model

Faces CBCL - SILF Model

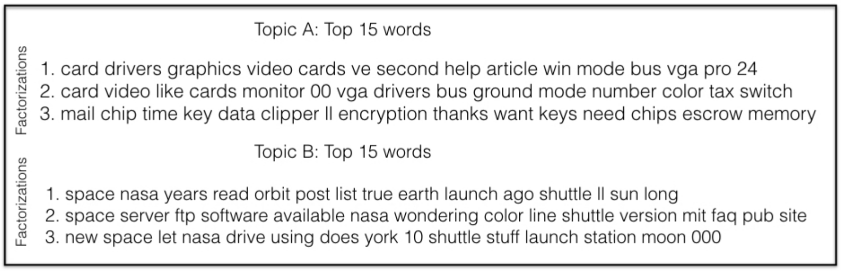
LFW - SILF Model

Hubble - SILF Model

Salinas A - SILF Model

Urban - SILF Model

Figure 4: For each dataset we show the quality of the BNMF approximate posterior ($M = 5$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).



Topic A: Top 15 words

Factorizations

1. card drivers graphics video cards ve second help article win mode bus vga pro 24
2. card video like cards monitor 00 vga drivers bus ground mode number color tax switch
3. mail chip time key data clipper ll encryption thanks want keys need chips escrow memory

Topic B: Top 15 words

Factorizations

1. space nasa years read orbit post list true earth launch ago shuttle ll sun long
2. space server ftp software available nasa wondering color line shuttle version mit faq pub site
3. new space let nasa drive using does york 10 shuttle stuff launch station moon 000

Figure 5: The top 15 words for topic A (computers/electronics) and topic B (space) shows that different factorizations provide an emphasis on different terms. In topic A, the top word from factorization 1 and 2 is 'card', but it does not appear in the top 15 words of factorizations 3. Instead a similar term 'chip' is emphasized in Factorization 3. In topic B, the terms 'space' and 'nasa' appear in all three factorizations but factorization 2 is the only one with digital terms like 'ftp', 'server','site' and 'faq'. In contrast factorization 1 and 3 both contain more physical terms like 'sun', 'moon','launch'.

7.2.1. Interpretation and Utilization of Posterior Estimates

BNMF posteriors can provide insight into the non-identifiability present within a particular dataset. Different factorizations may explain the data as a whole equally well, but do it through dictionary elements that have different interpretations, or can be used to understand specific parts of the data better than other factorizations. We show visual examples of diversity in the top words of the 20 Newsgroups BNMF posteriors and examples of how performance in downstream tasks for the 20 Newsgroups and Autism dataset is dependent on the posterior samples. Our analysis yields meaningful insights that could not be gained through a single factorization.

20-Newsgroups Our BNMF of 20-Newsgroups was a rank 16 decomposition of posts from 4 categories. In figure 6, we show the held-out AUC of a classifier trained to predict those categories based on the weights matrix W from each factorization in our variational posterior. Even though all of these factorizations have essentially equivalent reconstruction (see figure 19 in supplementary material), there exists a significant variation in the performance of these NMFs on the prediction tasks. The best performing NMF for one category is generally not the best (or even one of the top performing) NMFs for other categories. This observation may be valuable to a practitioner intending to use the NMF for some downstream task: different samples explain different patterns in the data. In figure 5, we see that this is indeed true: even after alignment,[9] distinct NMF factorizations have top words that indicate different emphasis across topics.

Autism Spectrum Disorder (ASD) In addition to core autism symptoms, Doshi-Velez et al. (2014) describe three major subtypes in autism spectrum disorder: those with higher rates of neurological disorder, those with higher rates of autoimmune disorders, and those with higher rates of psychiatric disorders. In figure 7, we show the number of topics that

[9] We compare topics after finding the permutation of columns that best aligns them by solving the bipartite graph matching problem. We minimize the cost given by the angle between topics.

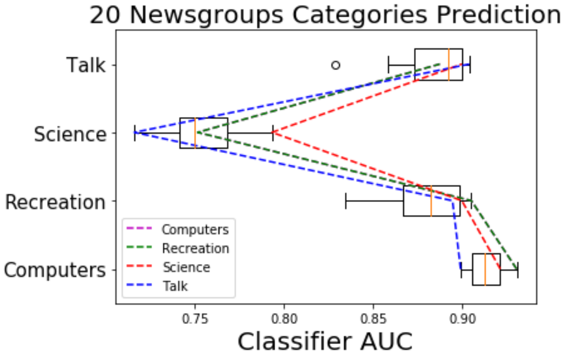


Figure 6: Classifiers trained on feature vectors from different factorizations yield variability in prediction performance (as measured by AUC). The dotted lines show the factorization that produces the best performing classifier for each category. The factorization (blue dotted line) that predicts the 'Talk' category best is actually one of the worst performing factorizations for the 'Science' category. This variability in performance demonstrates that no single factorization gives the best latent representation for the overall prediction task.

contain key terms corresponding to these areas (expressive language disorder, epilepsy, asthma, and attention deficit disorder) across different factorizations in the variational posterior obtained via Q-Transform. The large variation suggests that different factorizations in the particle-based posterior are spending different amount of modeling effort across these known factors; knowing that such uncertainty exists is essential for clinicians who may be trying to interpret topics to understand patterns in autism spectrum disorder.

On the same set of patients, we can also ask whether we can predict the onset of certain medical issues in the subsequent patient trajectory. We train a classifier on the weights of the NMFs to predict the onset of these medical issues. Similar to the category prediction results in 20-Newsgroups, figure 8 shows that there is a large variability (around 0.1 in AUC) in the performance of classifiers trained on the weights matrices of different factorizations on the prediction task. No single factorization has the best performance across the different prediction tasks.

7.3. Extension: BNMF in the presence of missing data

In the presence of missing data, there is perhaps an even greater need to understand the uncertainty in factorization parameters for NMF. The factorization space of a fully observed dataset forms a subset of the factorization space in the presence of missing data. Our particle-based approach to BNMF posterior approximation can be applied to the missing data setting by making some minor adjustments to the experimental settings.

The multiplicative update algorithm for NMF (Lee and Seung, 2001) can be adjusted so that the update equations for factorization parameters only consider the observed data. We use an implementation of this modification to the multiplicative update algorithm[10] to find a completion of the data X, compute the SVD subspace and then apply our Q-Transform initializations. Figure 9 demonstrates that our approach to BNMF can be extended to the

[10] https://github.com/scikit-learn/scikit-learn/pull/8474/commits/a838f94c8c832aaf57140f23bd8c8a14daec2626

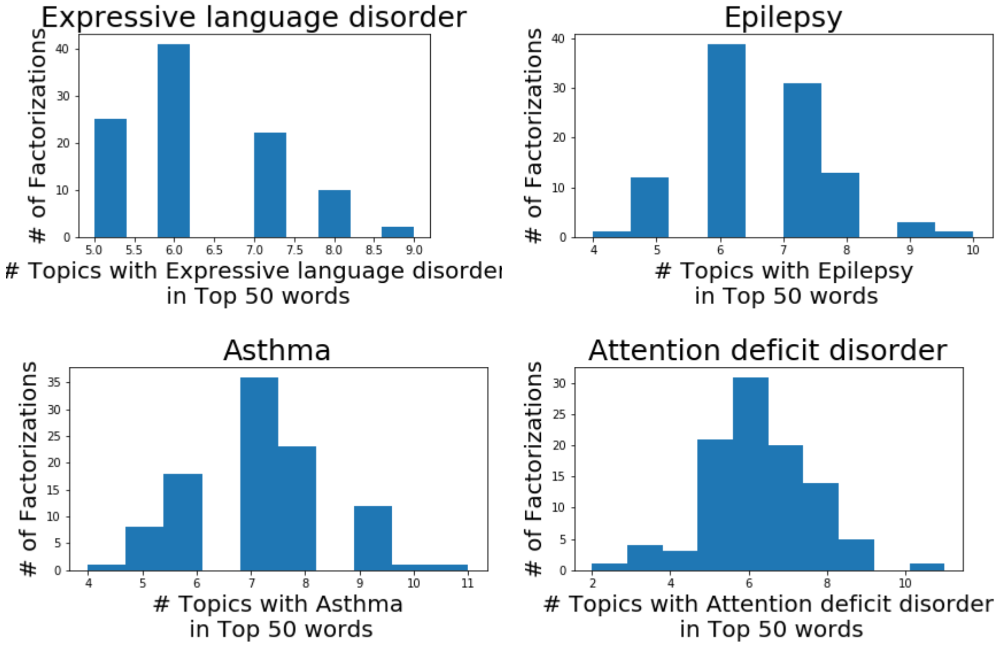


Figure 7: We explore top words in the topics relating to key terms of interest to clinicians and discover that different NMFs place varying amount of emphasis on different terms. Such variability is of interest to clinicians who may be trying to interpret topics to understand patterns in ASD.

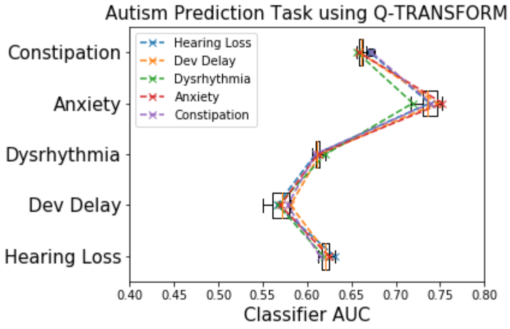


Figure 8: Classifiers trained on weights matrix of $M_{\max} = 100$ different factorizations to predict the presence of certain medical codes in a patient's trajectory exhibit significant variability in prediction on a test set (as measured by AUC). Different factorizations lead to top predictors for the onset of different medical issues.

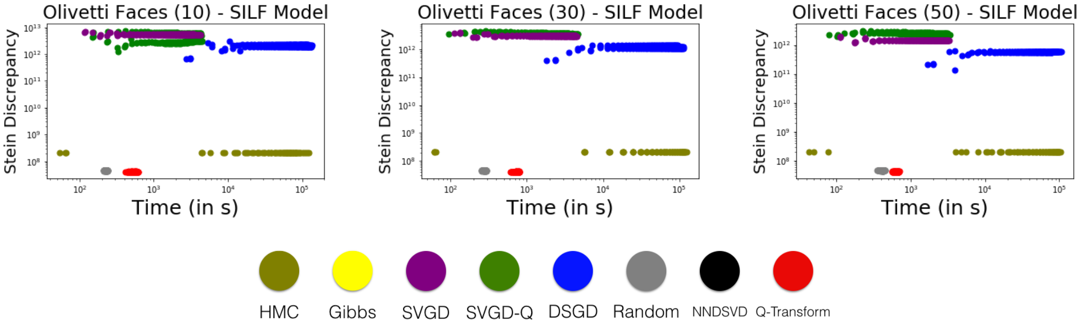


Figure 9: Under different percentages of missingness in the Olivetti Faces dataset (10%, 30%, 50%), the quality of the BNMF approximate posterior and the corresponding runtime of Q-Transform and the other baselines is shown. The best discrete posterior approximations to BNMF are produced using the Q-Transform initializations (in red).

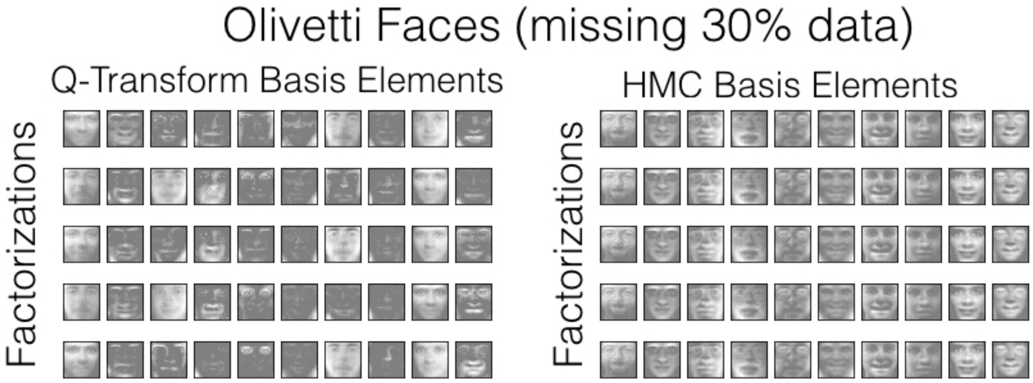


Figure 10: Sample factorizations from the variational posterior using Q-Transforms show that a diverse range of basis elements can be use to approximate the data. However, HMC samples seem to be identical indicating that HMC was only exploring a very small region of the posterior space.

case where the data matrix X is partially observed. For the Olivetti Faces dataset with varying degrees of missingness, the Q-Transform approach to BNMF consistently finds posterior approximations that are significantly better (as measured by Stein discrepancy) than other baselines whereas for a given M, the runtime is second-lowest.

Figure 10 shows sample factorizations from the variational posterior using Q-Transform and HMC samples. To allow for comparison, we have aligned the positions of the basis (dictionary) elements to a reference factorization using the bipartite matching algorithm. It is clear from looking at the Q-Transform factorizations that a diverse range of dictionaries can be used to approximate the data well whereas the HMC chain only explores one set of dictionary elements. Interestingly, the diversity of solutions obtained using Q-Transform have visually interpretable differences, i.e. these are not simply perturbations of some ground truth basis elements. Some of the basis elements look like faces and some of them look like different shadow or lighting configurations. In contrast, the factorization samples from HMC have basis elements that look identical. This indicates that the HMC has explored a limited region of the posterior space.

8. Discussion: When is Q-Transform successful?

Our ability to extract *transferable* low-rank transformation matrices from an SVD and an instance of NMF indicates that there exist similarities across different NMF problems. In this Section we seek to develop a better intuition behind the success of the Q-Transform initializations at exploiting these similarities. In this Section, we provide discussion and smaller-scale experiments to shed light on when, why, and how our Q-transform approach is successful.

8.1. Q-Transform Generating Process

In our approach, we generated candidate Q-Transform matrices (Algorithm 2) by applying random restarts to small, synthetic data sets. We focused on this approach because small datasets are much faster to train, and with synthetic data sets, we can know at least one ground truth NMF and level of noise. However, there are obviously a large number of choices for the data used to generate candidate Q-Transform matrices.

In figure 11, we present results with a variety of different methods for generating candidates. In all cases, the source data was of small dimension ($X_S \in \mathbb{R}^{15\times 15}$), and the target data was larger ($X_T \in \mathbb{R}^{500\times 500}$). The target data had a true non-negative rank of 10 and factors were generated with i.i.d. entries from a standard normal. In all these experiments we set the transfer rank to be $R_T = R_{\text{SVD}} = 3$. We explored six ways of generating candidates from the source data:

- Uniform data: Generating a dataset X_S where each entry is i.i.d. with a uniform distribution in [0,1]; then apply random restarts to find candidate transforms.
- Simple sub-sample data: Generating dataset X_S by uniformly selecting 15 rows and columns of the target data X_T; then apply random restarts to find candidate transforms
- Column-projection data: Generating dataset X_S by sub-sampling 15 columns of X_T and applying a random projection into $\mathbb{R}^{15}$ for each each column; then apply random restarts to find candidate transforms.
- Dirichlet factors: Generating factors A, W with each column of A, W from a Dirichlet distribution (with concentration parameter α set to 1); let $X_S = AW$+Gaussian Noise; then apply random restarts to find candidate transforms.
- Uniform factors: Generating factors A, W with each entry i.i.d. from a uniform distribution in [0,1]; let $X_S = AW +$ Gaussian Noise; then apply random restarts to find candidate transforms.
- Gaussian factors: Generating factors A, W with each entry i.i.d. from a standard normal distribution; let $X_S = AW +$ Gaussian Noise; then apply random restarts to find candidate transforms.

The methods that produced the source data from some true NMF factors produced candidate transformations that resulted in the highest quality initializations on the target data (figure 11). In settings where a practitioner deals with a collection of similar NMF datasets (e.g. music analysis, hyper spectral images), there may be more clever ways in which the NMF solution spaces corresponding to a real dataset may yield more appropriate Q-Transforms specific to that type of data. Finally, we find in figure 12 that the performance of Gaussian factors method does not vary with the rank of the synthetic data (the transfer rank is still held fixed).

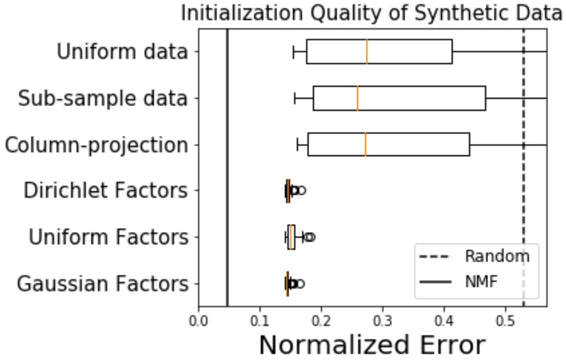


Figure 11: For different synthetic data X_S generating procedures, we show the initialization quality obtained via the Q-Transform matrices on a target data X_T. Dirichlet, Uniform, Gaussian have significantly superior performance compared to Sub-sample, Column-projection and Uniform data. For comparison, we show the quality of NMF solutions (solid line) and random initializations (dashed line).

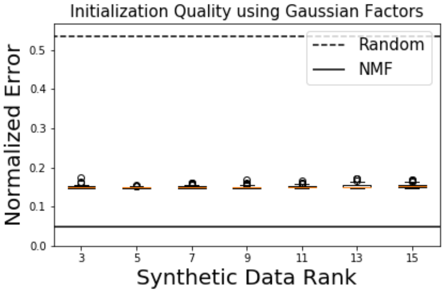


Figure 12: Using Gaussian factors for the synthetic data generation process with different ranks does not appear to change the quality of the Q-Transform initialization quality on the target data X_T. This indicates that this generating procedure is not sensitive to the rank in order to produce high quality (close to true NMF solution) initializations using Q-Transform. For comparison, we show the quality of NMF solutions (solid line) and random initializations (dashed line).

8.2. The Q-Transform Initialization versus Noise

In Section 4, we sought high-quality initializations because they generally require less time to converge. On synthetic target data $X_T = AW + \epsilon N_o$ (D = N = 500, R = 20) we explore the effect of increasing noise (ϵ) on the quality of our transfer-based NMF initializations and the time taken to converge. Specifically: are there noise regimes in which the Q-transform method works better, and noise regimes in which it does not?

We normalize the norm of the noise matrix to be equal to the norm of the data $\|N_o\| = \|AW\|$ so that the contribution of signal AW and noise ϵN_o to the data is equal when $\epsilon = 1$. We continue to use the same 100 pairs of Q_A, Q_W matrices. We compare the performance of Q-Transform over random restarts in terms of initialization quality (ratio of the reconstruction error from Q-Transform to the reconstruction error from random restart)

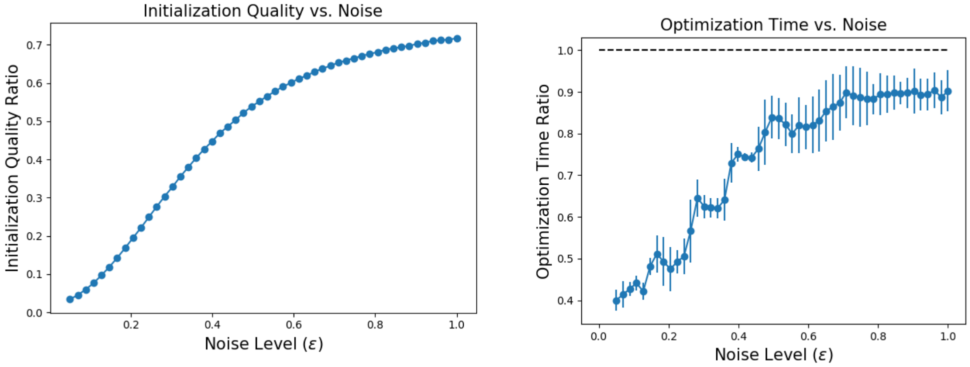


Figure 13: In the low-noise regime, the reconstruction error of Q-Transform initializations is significantly less than random restart initializations. This relative advantage gets smaller as the noise level increases. Similarly, the time taken to converge is significantly shorter than the random restart approach under the low noise scenario and continues to increase with noise. As expected, at high noise levels there exists no additional advantage to the Q-Transform approach (the optimization time ratio approaches 1).

and time to convergence (ratio of time taken using Q-Transform initialization to time taken using random restart). In both metrics, the Q-Transform has an advantage over random restarts for values of the noise ϵ smaller than 1, and the advantage is greatest for smallest noise. Figure 13 shows that the advantage of Q-Transform initializations is highest in a low noise regime and decreases as the noise increases. This behavior makes sense because as noise increases, the data is no longer truly low rank.

8.3. Selecting ranks

We emphasize that there are two distinct ranks that need to be chosen when applying our technique. The first is the rank of the factorization R_{NMF}. There exist multiple approaches for choosing this rank, e.g. Tan and Févotte (2009); Alquier and Guedj (2017), and they can be applied to our approach (as well as any other NMF algorithm).

The second is choosing the transfer rank R_T. The transformation dimensions R_T and R_{SVD} determine the dimensions of transformation matrices Q_A, Q_W which map basis vectors defining the top SVD subspace of dimension R_{SVD} to a set of R_T non-negative basis vectors that approximate the same subspace. The full initialization for NMF is obtained by either padding the initialization with small entries ($R_T < R_{\text{NMF}}$) or removing extra columns and rows of the factor matrices ($R_T > R_{\text{NMF}}$). (For simplicity, we consider the case where the transfer rank and SVD rank are equal $R_T = R_{\text{SVD}}$ and the resulting transformation matrices Q_A, Q_W are square.)

The choice of the transfer rank R_T is specific to our algorithm, and in figure 14 we investigate how well our transfer learning performs for different choices of transfer rank R_T. In the experiment, we extract a set of 100 transformation matrices Q_A, Q_W for transfer dimensions $R_T = R_{\text{SVD}} = \{1, 2, \ldots, 10\}$ using synthetic source data ($D = N = 15$). Once constructed, we applied the transformation matrices to a 500×500 target dataset X_T of rank $R = 10$. We find that even though the dataset X_T has rank 10, the rank 10 transformation

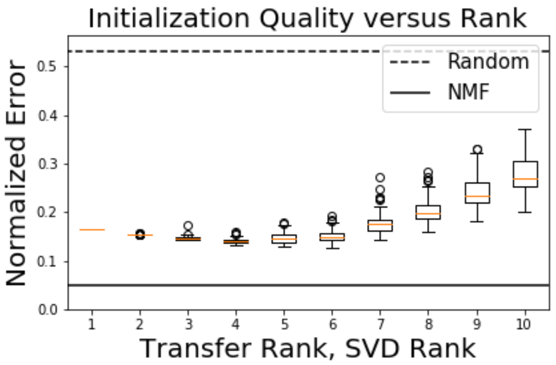


Figure 14: On a synthetic target dataset ($D = N = 500, K_{\text{NMF}} = 10$), we apply Q-Transform initializations using varying transfer ranks and SVD ranks $R_T = R_{\text{SVD}} = \{1, 2, \ldots, 10\}$. We see that for a range of low rank values, the Q-Transform initializations are high quality, but at larger values the quality of initializations gets worse. The dotted line shows the quality of random initializations and the solid line shows the quality NMF solutions. The reconstruction errors are normalized by the norm of the data.

matrices found using the 15×15 synthetic source dataset are unable to successfully transfer to this new dataset. We see that the error initially decreases, but then increases as the transfer rank increases. This result suggests that the top directions of variation hold the most transferable information across NMF problems.

8.4. Sign Convention for SVD

In considering when Q-Transform is successful, we note that there exists an intrinsic ambiguity in the sign of the singular vectors of X: changing the sign of any column of A_{SVD} and corresponding row of W_{SVD} gives a valid SVD. For Q-Transform to work, we must apply a consistent resolution of the sign ambiguity (e.g. from Bro et al. (2008)). This ensures that learned transformations Q_A, Q_W map in a consistent way to SVD decompositions of new datasets.

9. Related Work

There is a large body of work on inference for BNMF. Sampling-based approaches include Gibbs sampling (Schmidt et al., 2009), Hamiltonian Monte Carlo (Schmidt and Mohamed, 2009), and reversible jump variants (Schmidt and Mørup, 2010). All of these have trouble escaping local modes (Masood et al., 2016), and are often constrained to a limited class of tractable distributions. Variational approaches to BNMF have successfully yielded interpretable factorizations (Bertin et al., 2009; Cemgil, 2009; Paisley et al., 2014; Hinrich and Mørup, 2018) but also typically only capture one mode and rely on mean-field or other modeling assumptions to make inference tractable. We note that in many cases, priors of convenience—for example, exponential distributions—can induce a single dominant mode, even when that was not the intent of the practitioner.

Closer to the goals of our work, Gershman et al. (2012) develop a non-parametric approach to variational inference that provides flexibility in modeling the number of Gaussian components required to approximate a posterior. However, the isotropic covariance in the model makes it unsuitable for applying it to BNMF. With regard to the inference process, our Q-Transform approach to finding multiple optima is most similar to Ročková and George (2016) and Paatero and Tapper (1994), who use rotations to find solutions to a single matrix factorization problem that are sparse and non-negative respectively. In contrast, we use rotations to find multiple non-negative solutions, and also demonstrate how these rotations can be *re-used* for transfer learning.

More broadly, recent work on NMF has involved theoretical work on non-identifiability with new algorithms that can provably recover the NMF under certain assumptions (Li and Liang, 2017; Bhattacharya et al., 2016; Ge and Zou, 2015a). However, these assumptions are often difficult to check and may indeed be violated in practice; Bayesian methods typically provide more flexibility in modeling and assumptions.

All of the works above typically assume some desired factorization rank. There also exists work on models that automatically detect the rank—through automatic relevance determination for NMF (Tan and Févotte, 2009) or more recently, via a rank-adaptive prior Alquier and Guedj (2017). These works are complementary to ours, in that those techniques could be combined with our transfer-based approach of generating candidates of whatever rank those algorithms determine is appropriate.

The ability of Stein discrepancies to assess the quality of any collection of particles (Gorham and Mackey, 2015) has resulted in large recent interest in other ways to create collections of samples (Oates et al., 2017; Liu and Wang, 2016). Liu et al. (2016) and Chwialkowski et al. (2016) showed that kernelized Stein discrepancy could be computed analytically in Reproducing Kernel Hilbert Spaces (RKHS); Pu et al. (2017) and Feng et al. (2017) use neural networks instead. Ranganath et al. (2016) establish the Stein discrepancy as a valid variational objective. To our knowledge, Stein discrepancy-based posterior approximation has not been applied to NMF, and yet, we see that it allows us to leverage existing non-Bayesian approaches to characterize these multi-modal posteriors. In our work, the Dirichlet prior on the columns of the basis matrix A is important to ensure that we avoid a known saddle point of the zero factorization (from likelihood term) that yields a corresponding zero for the score function.

10. Conclusion

In this work, we presented a novel transfer learning-based approach to posterior estimation in BNMF. Simply creating collections of factorizations via random restarts on our Q-Transform initializations, and then weighing them, produces diverse collections that approximate the posterior well (the NNDSVDar-based methods fail to produce diverse collections for posterior estimation). In contrast, the functional gradient descent of SVGD and direct gradients of Stein discrepancy (DSGD) perform worse to the collection-based approaches, requiring more time and also limiting the user to specify in advance the number of factorizations. Hamiltonian Monte Carlo also suffers from difficulties in exploring the posterior space, something random initializations are well suited to. Our transfer learning approach consistently produces the highest quality posterior approximations.

Through Q-Transform, we introduce a way to speed-up the process of finding multiple diverse NMFs. The discovery that Q-Transform matrices can transfer from synthetic to multiple real datasets is exciting and also suggests interesting questions for further research. For example, what is the theoretical nature of the similarities between principal eigenspaces of different non-negative matrices and the relation between their SVD and NMF bases? And, how does the synthetic data generation process used to obtain Q-Transform matrices impact the initializations and the effectiveness of the Q-Transform algorithm in general?

More broadly, our qualitative results demonstrate that even relatively simple models, such as NMF, can have multiple optima that are comparable under the objective function but have large variation in how well they explain different portions of the data—or how they perform on different downstream tasks. Thus, it is important to be able to compute these posteriors efficiently.

Acknowledgments

We would like to thank anonymous reviewers for their insightful comments and suggestions. Additionally, we thank Andreas Krause, Mike Hughes, Melanie Pradier, Nick Foti, Omer Gottesman and other members of the Dtak lab at Harvard for many helpful conversations and feedback. MAM and FDV acknowledge support from AFOSR FA 9550-17-1-0155.

References

20NG. The 20 newsgroups text dataset scikit-learn 0.19.1 documentation. `http://scikit-learn.org/stable/datasets/twenty_newsgroups.html`, July 2013. (Accessed on 01/23/2018).

Pierre Alquier and Benjamin Guedj. An oracle inequality for quasi-Bayesian Nonnegative Matrix Factorization. *Mathematical Methods of Statistics*, 26(1):55–67, 2017.

Yoann Altmann, Nicolas Dobigeon, and Jean-Yves Tourneret. Unsupervised post-nonlinear unmixing of hyperspectral images using a Hamiltonian Monte Carlo algorithm. *IEEE Transactions on Image Processing*, 23(6):2663–2675, 2014.

Morten Arngren, Mikkel N Schmidt, and Jan Larsen. Unmixing of hyperspectral images using Bayesian Non-Negative Matrix Factorization with volume prior. *Journal of Signal Processing Systems*, 65(3):479–496, 2011.

Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a Non-Negative Matrix Factorization–provably. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 145–162. ACM, 2012.

Andrew D Barbour and Timothy C Brown. Stein's method and point process approximation. *Stochastic Processes and their Applications*, 43(1):9–31, 1992.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

Nancy Bertin, Roland Badeau, and Emmanuel Vincent. Fast Bayesian NMF algorithms enforcing harmonicity and temporal continuity in polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, pages 29–32. IEEE, 2009.

Michael Betancourt. Cruising the simplex: Hamiltonian Monte Carlo and the Dirichlet distribution. In *AIP Conference Proceedings 31st*, volume 1443, pages 157–164. AIP, 2012.

Chiranjib Bhattacharya, Navin Goyal, Ravindran Kannan, and Jagdeep Pani. Non-Negative Matrix Factorization under Heavy Noise. In *International Conference on Machine Learning*, pages 1426–1434, 2016.

Chiranjib Bhattacharyya, IISC ERNET, Navin Goyal, COM Ravindran Kannan, and COM Jagdeep Pani. Non-Negative Matrix Factorization under heavy noise. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1426–1434, 2016.

José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(2):354–379, 2012.

Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for Non-Negative Matrix Factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

Rasmus Bro, Evrim Acar, and Tamara G Kolda. Resolving the sign ambiguity in the Singular Value Decomposition. *Journal of Chemometrics*, 22(2):135–140, 2008.

Jean-Philippe Brunet, Pablo Tamayo, Todd R Golub, and Jill P Mesirov. Metagenes and molecular pattern discovery using Matrix Factorization. *PNAS*, 101(12):4164–4169, 2004.

CBCL. Home — poggio lab. `http://poggio-lab.mit.edu/`, 2000. (Accessed on 01/23/2018).

Ali Taylan Cemgil. Bayesian inference for Non-Negative Matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009, 2009.

Wei Chu, S Sathiya Keerthi, and Chong Jin Ong. Bayesian support vector regression using a unified loss function. *IEEE transactions on neural networks*, 15(1):29–44, 2004.

Kacper Chwialkowski, Heiko Strathmann, and Arthur Gretton. A kernel test of goodness of fit. *arXiv preprint arXiv:1602.02964*, 2016.

Andrzej Cichocki and Anh-Huy Phan. Fast local algorithms for large scale Non-Negative Matrix and Tensor Factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(3):708–721, 2009.

Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

Onur Dikmen and A Taylan Cemgil. Unsupervised single-channel source separation using Bayesian NMF. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA'09. IEEE Workshop on*, pages 93–96. IEEE, 2009.

David Donoho and Victoria Stodden. When does Non-Negative Matrix Factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, page None, 2003.

Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):e54–e63, 2014.

Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized Stein Variational Gradient Descent. *arXiv preprint arXiv:1707.06626*, 2017.

Cédric Févotte and Jérôme Idier. Algorithms for Non-Negative Matrix Factorization with the β-divergence. *Neural computation*, 23(9):2421–2456, 2011.

Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Non-Negative Matrix Factorization with the Itakura-Saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

Rong Ge and James Zou. Intersecting faces: Non-Negative Matrix Factorization with new guarantees. In *International Conference on Machine Learning*, pages 2295–2303, 2015a.

Rong Ge and James Zou. Intersecting faces: Non-Negative Matrix Factorization with new guarantees. In *International Conference on Machine Learning*, page X. ICML, 2015b.

Michel Gendreau and Jean-Yves Potvin. *Handbook of metaheuristics*, volume 2. Springer, 2010.

Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012.

Jackson Gorham and Lester Mackey. Measuring sample quality with Stein's method. In *Advances in Neural Information Processing Systems*, pages 226–234, 2015.

Jackson Gorham and Lester Mackey. Measuring sample quality with kernels. *arXiv preprint arXiv:1703.01717*, 2017.

Derek Greene, Gerard Cagney, Nevan Krogan, and Pádraig Cunningham. Ensemble Non-Negative Matrix Factorization methods for clustering protein–protein interactions. *Bioinformatics*, 2008.

Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2006.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Jesper Løve Hinrich and Morten Mørup. Probabilistic Sparse Non-negative Matrix Factorization. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 488–498. Springer, 2018.

Matthew D Hoffman and David M Blei. Structured stochastic Variational Inference. In *Artificial Intelligence and Statistics*, 2015.

Cho-Jui Hsieh and Inderjit S Dhillon. Fast coordinate descent methods with variable selection for Non-Negative Matrix Factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1064–1072. ACM, 2011.

Oliver Jesorsky, Klaus J Kirchberg, and Robert W Frischholz. Robust face detection using the hausdorff distance. In *International conference on audio-and video-based biometric person authentication*, pages 90–95. Springer, 2001.

Daniel D Lee and H Sebastian Seung. Algorithms for Non-Negative Matrix Factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

LFW. Lfw face database : Main. `http://vis-www.cs.umass.edu/lfw/`, May 2017. (Accessed on 01/23/2018).

Yuanzhi Li and Yingyu Liang. Provable Alternating Gradient Descent for Non-negative Matrix Factorization with Strong Correlations. *arXiv preprint arXiv:1706.04097*, 2017.

Chih-Jen Lin. Projected gradient methods for Non-Negative Matrix Factorization. *Neural computation*, 19(10):2756–2779, 2007.

Qiang Liu and Yihao Feng. Two Methods for wild Variational Inference. *arXiv preprint arXiv:1612.00081*, 2016.

Qiang Liu and Jason D Lee. Black-box importance sampling. *arXiv preprint arXiv:1610.05247*, 2016.

Qiang Liu and Dilin Wang. Stein Variational Gradient Descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, pages 2370–2378, 2016.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized Stein discrepancy for goodness-of-fit tests. In *International Conference on Machine Learning*, pages 276–284, 2016.

Dougal Maclaurin, David Duvenaud, and Ryan P Adams. Autograd: Reverse-mode differentiation of native python. In *ICML workshop on Automatic Machine Learning*, 2015.

Arjumand Masood, Weiwei Pan, and Finale Doshi-Velez. An empirical comparison of sampling quality metrics: A case study for Bayesian Non-Negative Matrix Factorization. *arXiv preprint arXiv:1606.06250*, 2016.

Saïd Moussaoui, David Brie, Ali Mohammad-Djafari, and Cédric Carteret. Separation of Non-Negative mixture of Non-Negative sources using a Bayesian approach and MCMC sampling. *Signal Processing, IEEE Transactions on*, 54(11):4133–4145, 2006.

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

Robert J. Plemmons Nicolas Gillis. Hubble telescope hyperspectral image data, 1987. URL `https://sites.google.com/site/nicolasgillis/code`.

Chris J Oates, Mark Girolami, and Nicolas Chopin. Control functionals for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3):695–718, 2017.

Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. https://distill.pub/2018/building-blocks.

Pentti Paatero and Unto Tapper. Positive Matrix Factorization: A Non-Negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126, 1994.

John Paisley, David M Blei, and Michael I Jordan. Bayesian Non-Negative Matrix Factorization with stochastic variational inference. *Handbook of Mixed Membership Models and Their Applications. Chapman and Hall/CRC*, 2015.

John William Paisley, David M Blei, and Michael I Jordan. Bayesian Non-Negative Matrix Factorization with Stochastic Variational Inference., 2014.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Weiwei Pan and Finale Doshi-Velez. A characterization of the non-uniqueness of Non-Negative Matrix Factorizations. *arXiv preprint arXiv:1604.00653*, 2016.

Sam Patterson and Yee Whye Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, pages 3102–3110, 2013.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Yunchen Pu, Zhe Gan, Ricardo Henao, Chunyuan Li, Shaobo Han, and Lawrence Carin. Stein Variational Autoencoder. *arXiv preprint arXiv:1704.05155*, 2017.

Rajesh Ranganath, Dustin Tran, Jaan Altosaar, and David Blei. Operator Variational Inference. In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

Margaret E Roberts, Brandon M Stewart, and Dustin Tingley. Navigating the local modes of big data. *Computational Social Science*, page 51, 2016.

Veronika Ročková and Edward I George. Fast Bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 111(516):1608–1622, 2016.

Ruslan Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. On the convergence of bound optimization algorithms. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 509–516. Morgan Kaufmann Publishers Inc., 2002.

SalinasA. Multispec — home. `https://engineering.purdue.edu/~biehl/MultiSpec/`, June 2015. (Accessed on 01/23/2018).

Ferdinando S. Samaria. The database of faces (olivetti), 1994. URL `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

Mikkel N Schmidt and Shakir Mohamed. Probabilistic Non-Negative Tensor Factorization using Markov Chain Monte Carlo. In *Signal Processing Conference, 2009 17th European*, pages 1918–1922. IEEE, 2009.

Mikkel N Schmidt and Morten Mørup. Reversible jump MCMC for Bayesian NMF. In *Proc. NIPS Workshop on Monte Carlo Methods for Modern Applications*, 2010.

Mikkel N Schmidt, Ole Winther, and Lars Kai Hansen. Bayesian Non-Negative Matrix Factorization. In *Independent Component Analysis and Signal Separation*, pages 540–547. Springer, 2009.

Paris Smaragdis and Judith C Brown. Non-Negative Matrix Factorization for polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177–180. IEEE, 2003.

Cristian Sminchisescu, Max Welling, and G Hinton. Generalized darting Monte Carlo. In *AISTATS*, pages 516–523. Citeseer, 2007.

Vincent YF Tan and Cédric Févotte. Automatic relevance determination in Nonnegative Matrix Factorization. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

Per-Åke Wedin. Perturbation bounds in connection with Singular Value Decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Stefan Wild, James Curry, and Anne Dougherty. Improving Non-Negative Matrix Factorizations through structured initialization. *Pattern recognition*, 37(11):2217–2232, 2004.

Yun Xue, Chong Sze Tong, Ying Chen, and Wen-Sheng Chen. Clustering-based initialization for Non-Negative Matrix Factorization. *Applied Mathematics and Computation*, 205(2): 525–536, 2008.

Feiyun Zhu, Ying Wang, Shiming Xiang, Bin Fan, and Chunhong Pan. Structured sparse method for hyperspectral unmixing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:101–118, 2014.

Appendix A: Quality of factorizations

We measure the quality of factorizations in terms of the log of the joint likelihood (figures 18 and 15) as well the Frobenius NMF objective (figures 19 and 16). We see that both quality measures are in agreement with each other. We also assess the quality of factorizations via the reconstruction error of heldout data (figures 20 and 17).

Overall, Q-Transform, Random, NNDSVDar, HMC and (for the exponential-Gaussian model) Gibbs produce high quality factorizations. The initialization-based approaches all work well as they use specialized NMF algorithms that are designed to find high quality factorizations. HMC was given a warm start with a high likelihood initialization and the chain continues to stay in high likelihood regions. The remaining gradient-based approaches for optimizing a collection of particles (SVGD and DSGD) fail to produce high quality factorizations. This is indicative of the need for specialized NMF algorithms designed to work with the constraints and structure of the NMF problem and highlight how difficult it is to apply a naive gradient descent approach for finding NMFs.

Gaussian Exponential BNMF Log of Joint Likelihood

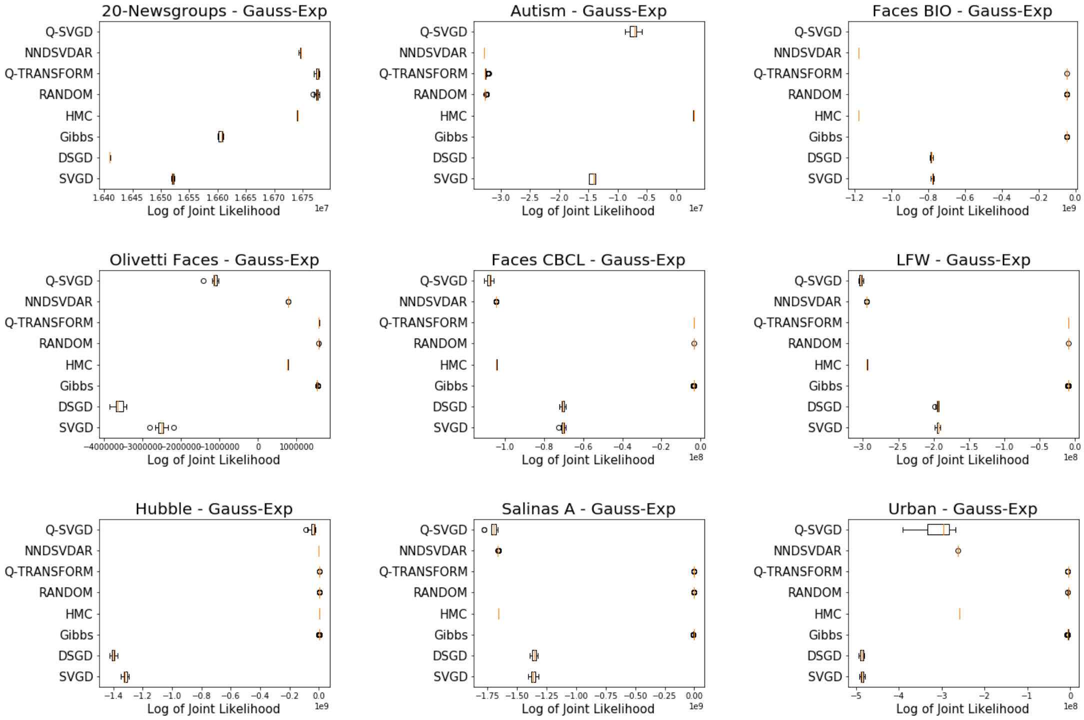


Figure 15: The joint likelihood of factorizations shows that SVGD and DSGD generally produce the worst quality factorizations. The remaining algorithms produce higher quality factorizations.

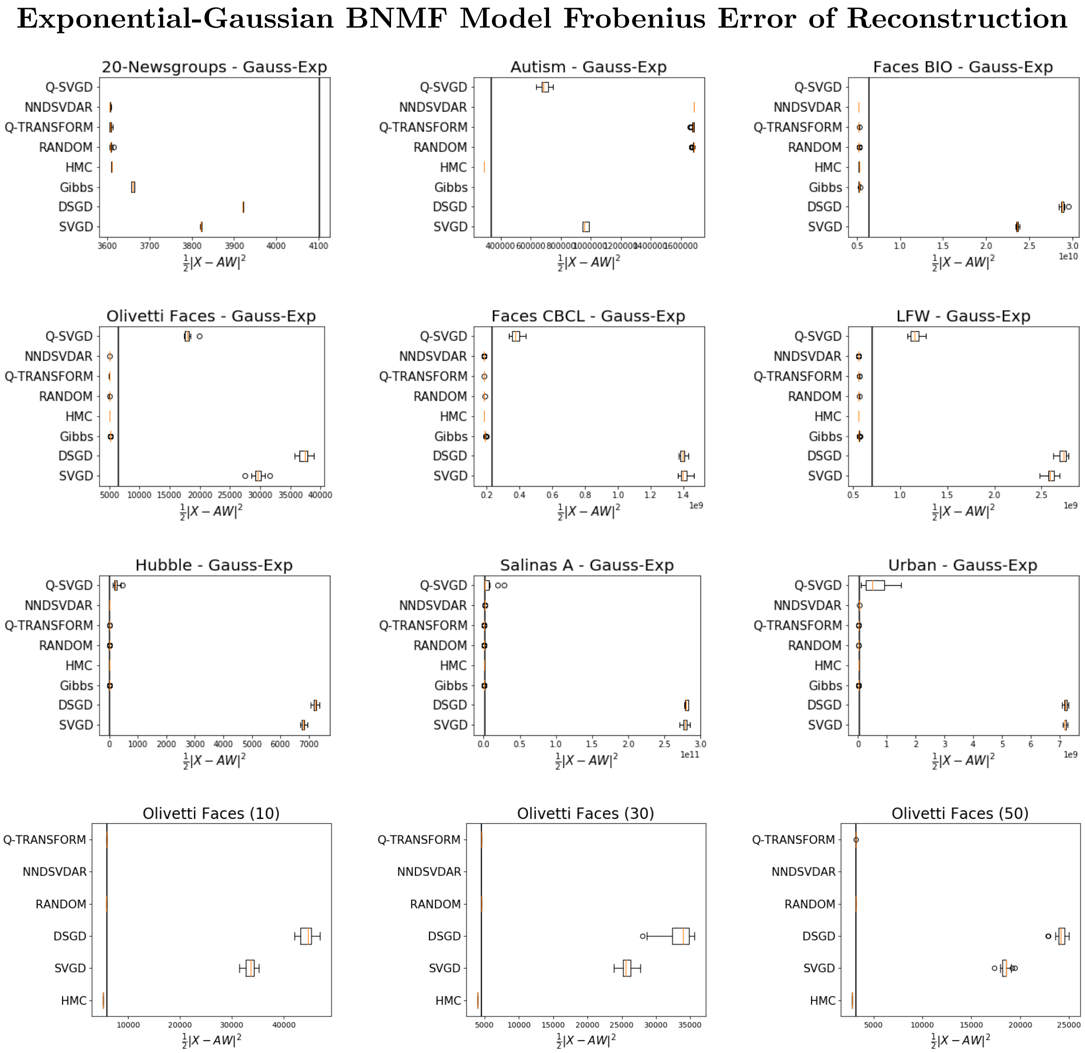


Figure 16: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that meet the threshold quality (black line) for useful factorizations. The other approaches consistently produce factorizations that meet this minimum quality requirement.

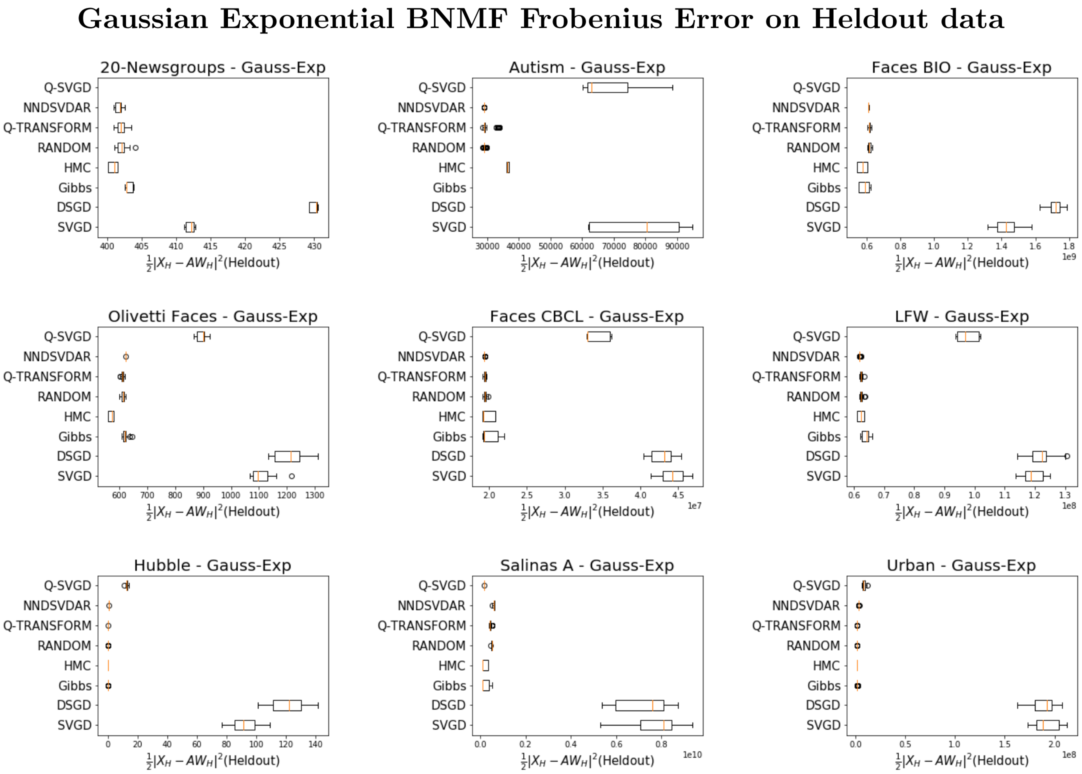


Figure 17: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that result in low error on heldout data. The other approaches consistently produce factorizations which generalize better and give low error on heldout data.

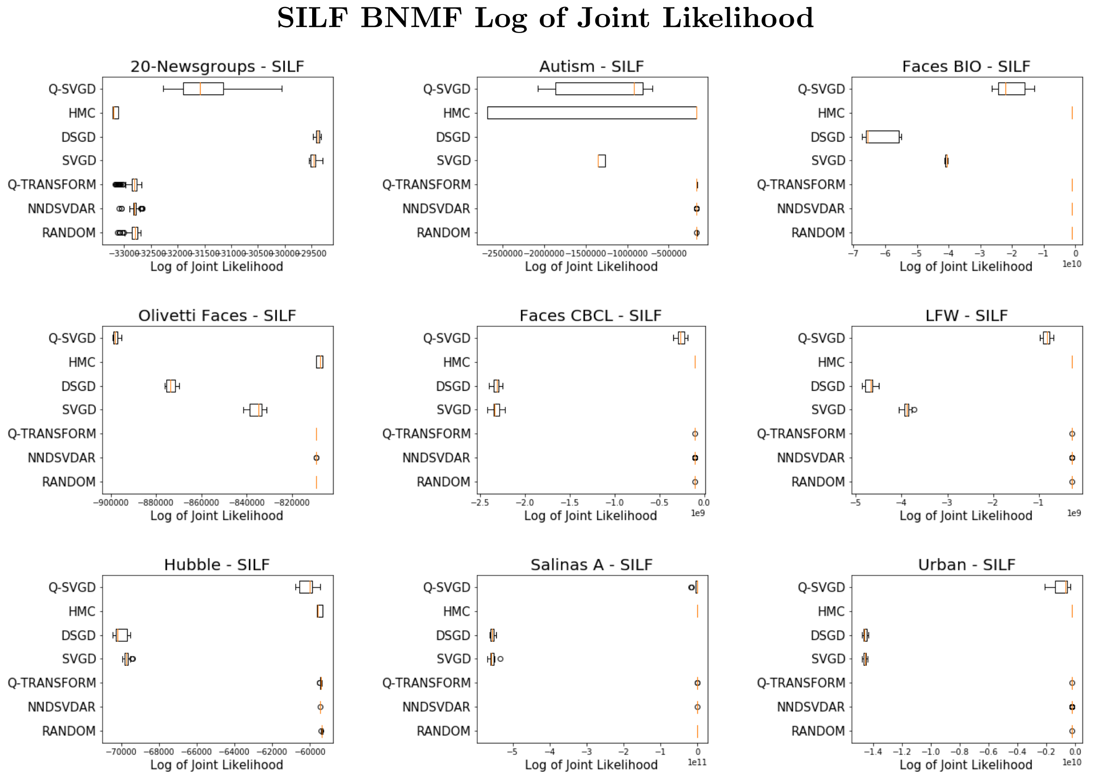


Figure 18: The joint likelihood of factorizations shows that SVGD and DSGD generally produce the worst quality factorizations. HMC, NNDSVDar, Random and Q-Transform produce high quality factorizations.

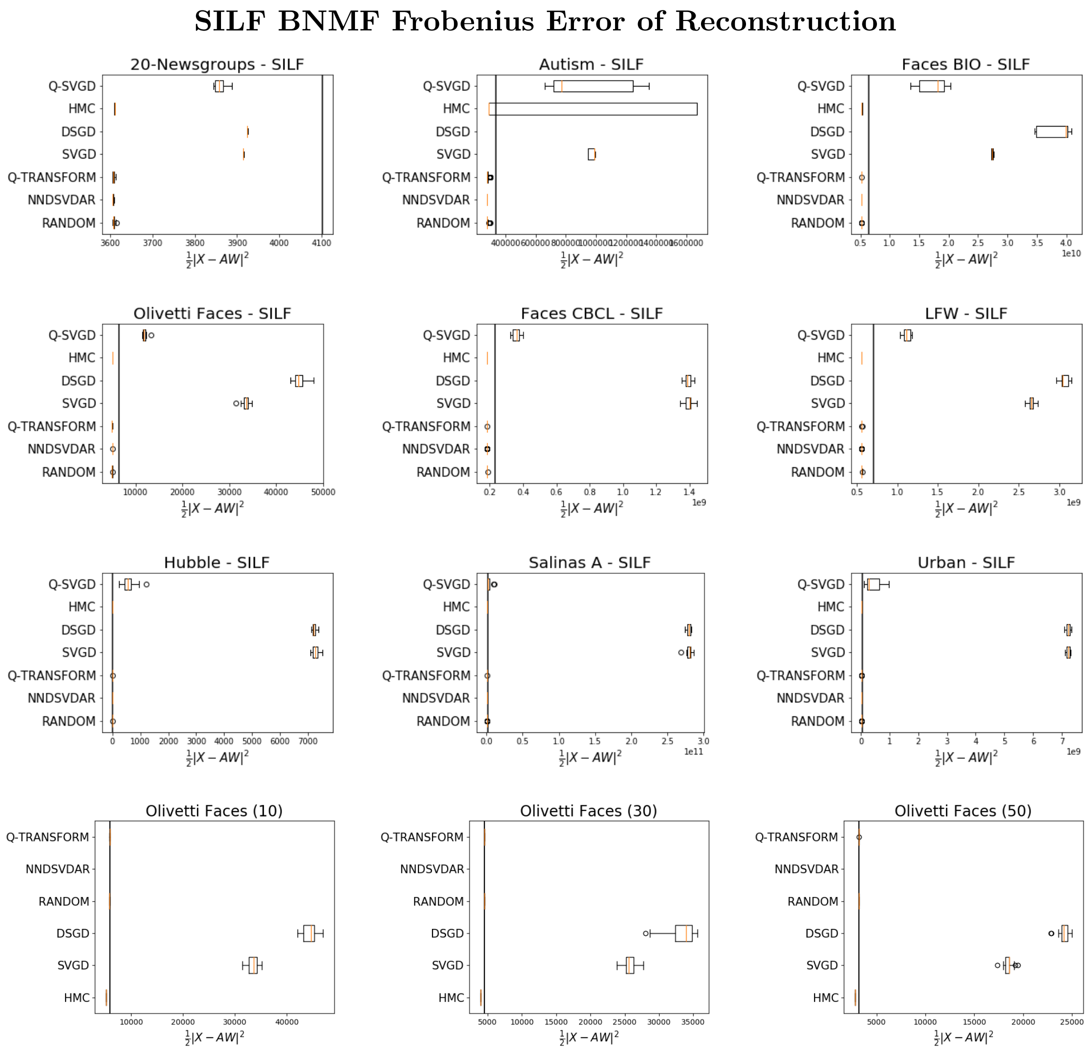


Figure 19: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that meet the threshold quality (black line) for useful factorizations. The other approaches consistently produce factorizations that meet this minimum quality requirement.

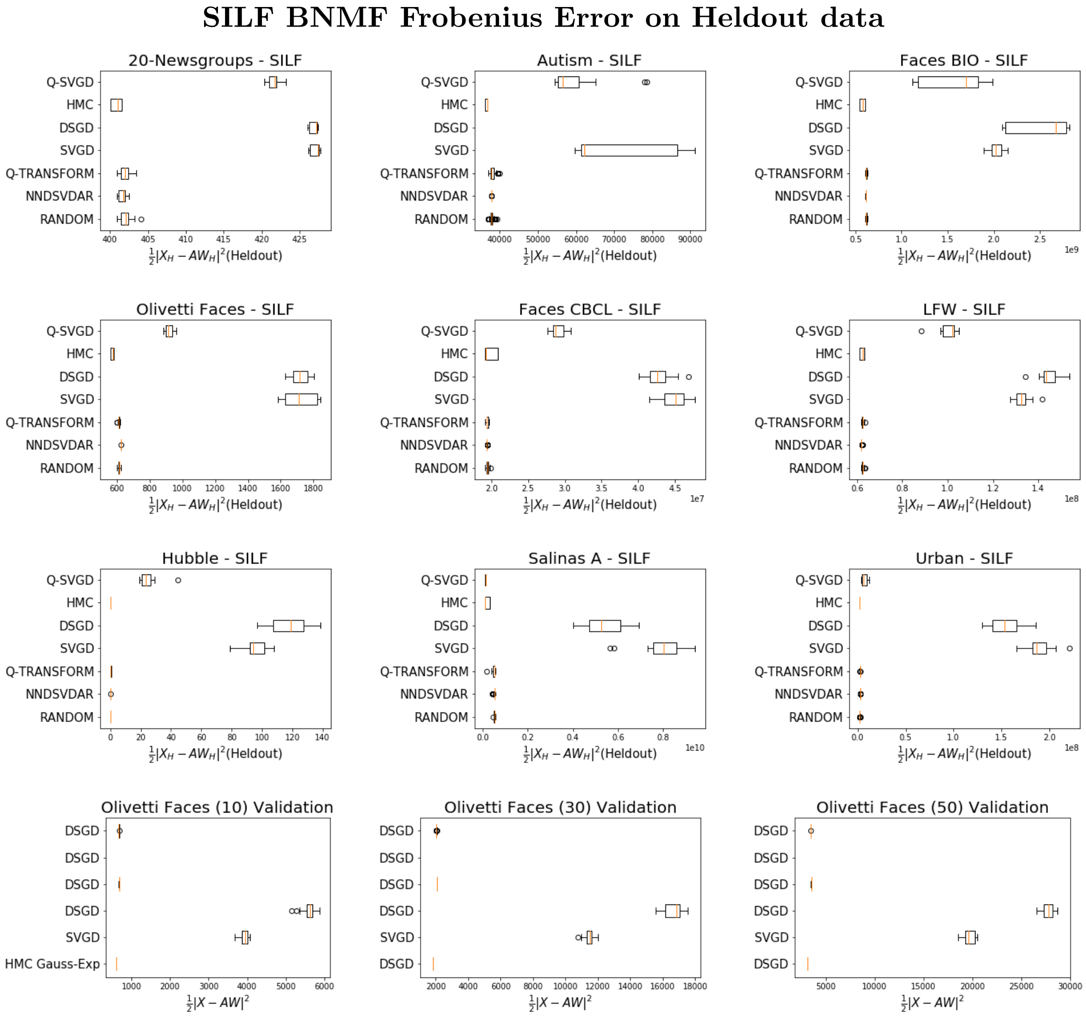


Figure 20: The reconstruction error of the factorizations shows that SVGD and DSGD are typically unable to find factorization parameters that result in low error on heldout data. The other approaches consistently produce factorizations which generalize better and give low error on heldout data. The matrix completion error on the missing data variant of the Olivetti faces dataset also gives similar results.

Appendix B: Diversity of factorizations

Similarity of factorizations is measured by the kernel (equation 7) for the base RKHS (figures 24 and 21) used in evaluating the Stein discrepancy and by pairwise distances between basis matrices (figures 25 and 22) and weights matrices (figures 26 and 23)[11]. Generally, the HMC chain exhibits the least exploration of the factorization space. Remaining algorithms, particularly Q-Transform exhibits higher amounts of diversity in the factorization space. The diversity metrics indicate that SVGD and DSGD give diverse factorizations but the quality metrics indicate that these factorizations are poor quality (do not correspond to high likelihood regions of the posterior) therefore such diversity is of little interest.

Exponential-Gaussian NMF Kernel similarity

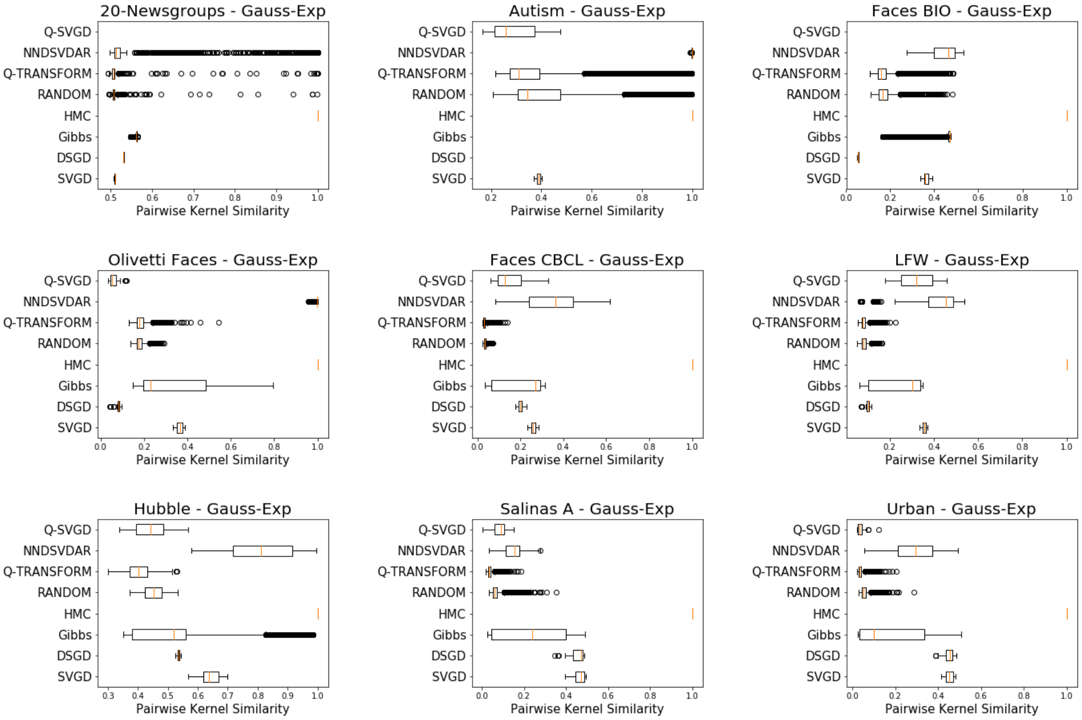


Figure 21: The kernel similarity indicates that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. Q-Transform and Random are the only algorithms that produce factorizations of high quality that are not similar.

[11] In the exponential-Gaussian model, we adjust the scalings of the factorizations so that we can meaningfully compare pairwise distances between basis and weights matrices.

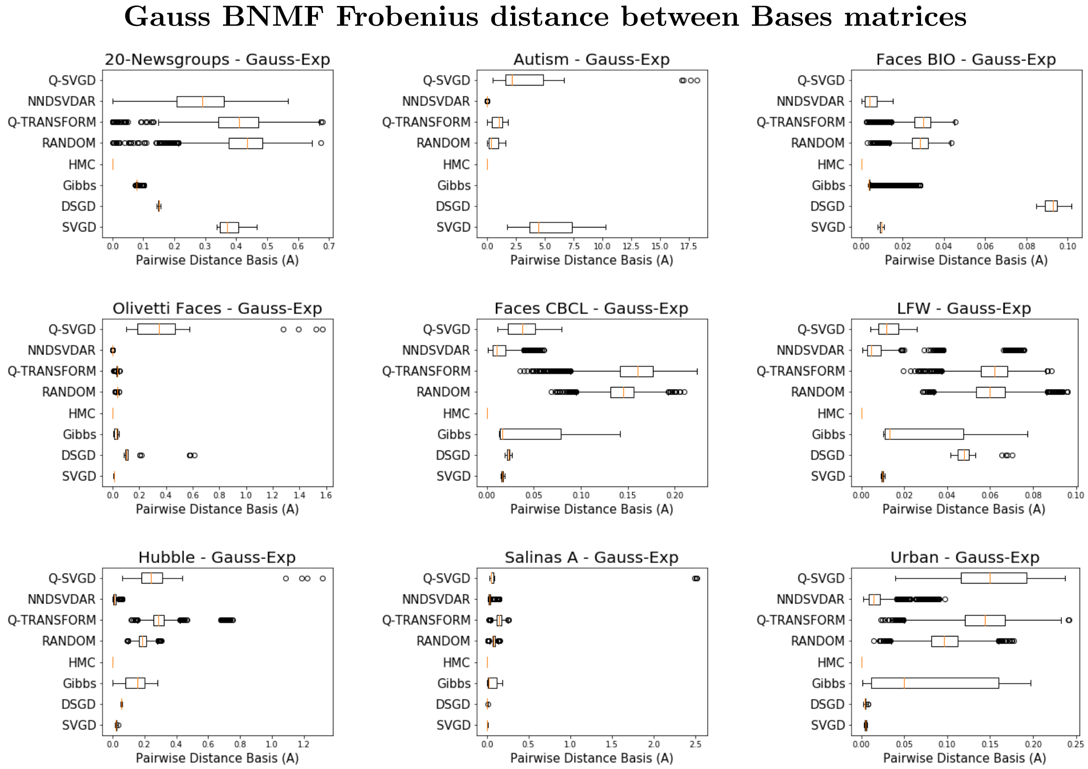


Figure 22: The pairwise distance between basis matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar, Q-SVGD, DSGD and SVGD factorizations are also very similar. Q-Transform and Random produce basis matrices that are more distinct than the Gibbs sampler.

Exponential-Gaussian BNMF Frobenius distance between weights matrices

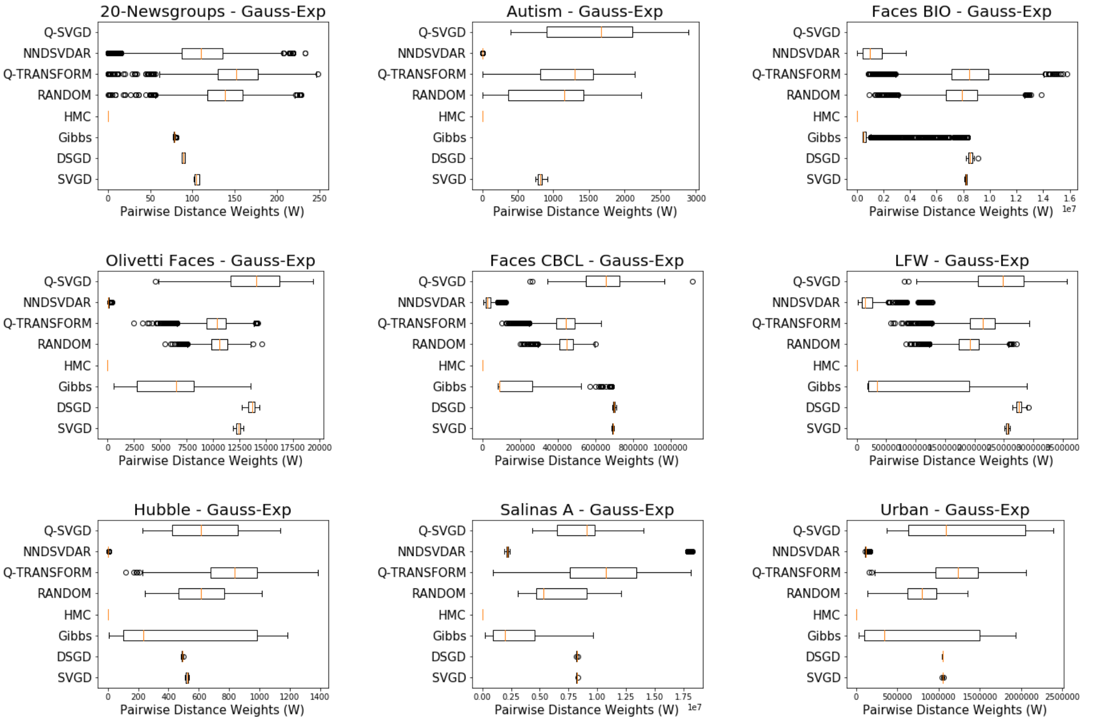


Figure 23: The pairwise distance between weights matrices shows that factorization collections obtained by DSGD, SVGD and Q-SVGD lead to diverse weights matrices. The remaining methods lead to similar levels of diversity within the factorization collections.

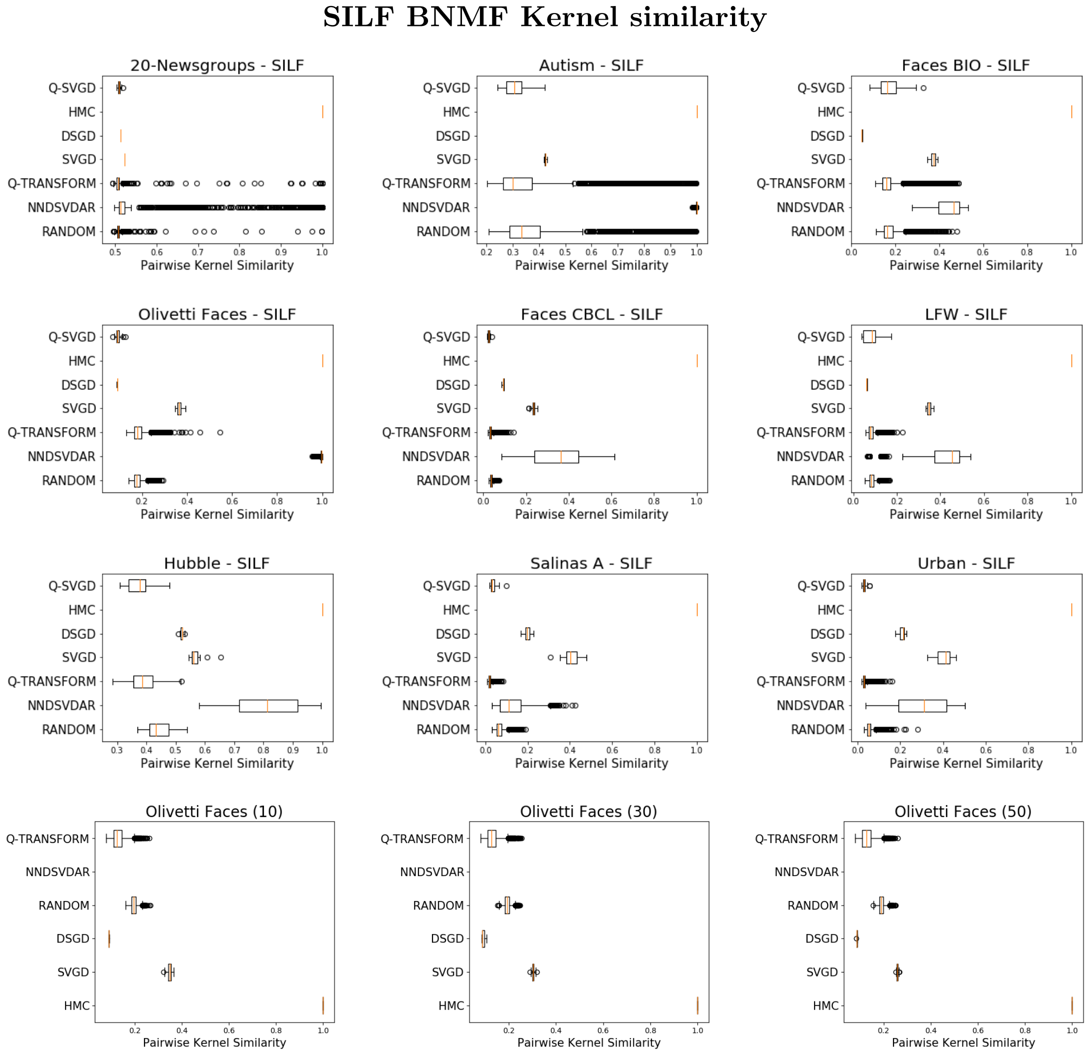


Figure 24: The kernel similarity indicates that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar.

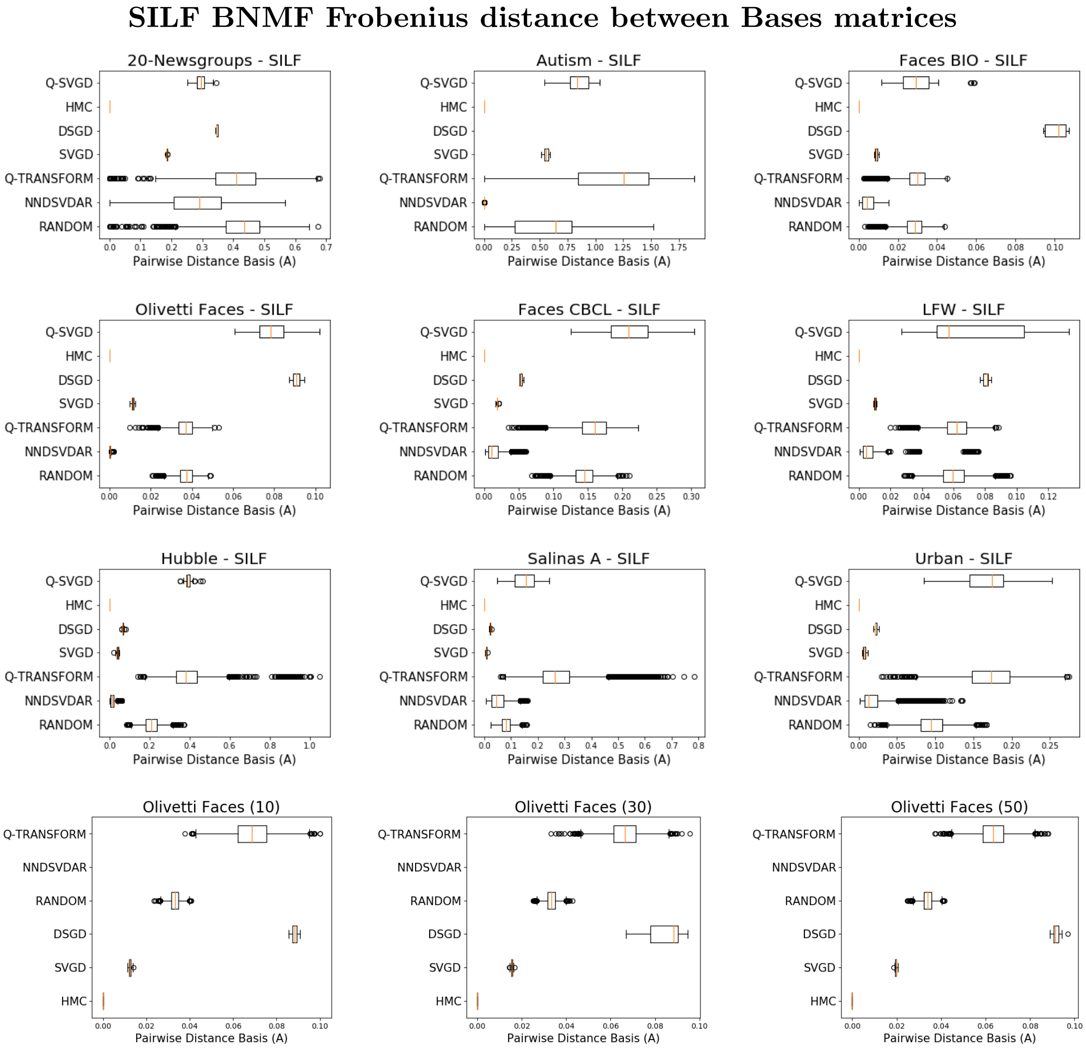


Figure 25: The pairwise distance between basis matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. *Q*-Transform and Random are the only algorithms that produce factorizations of high quality that are different.

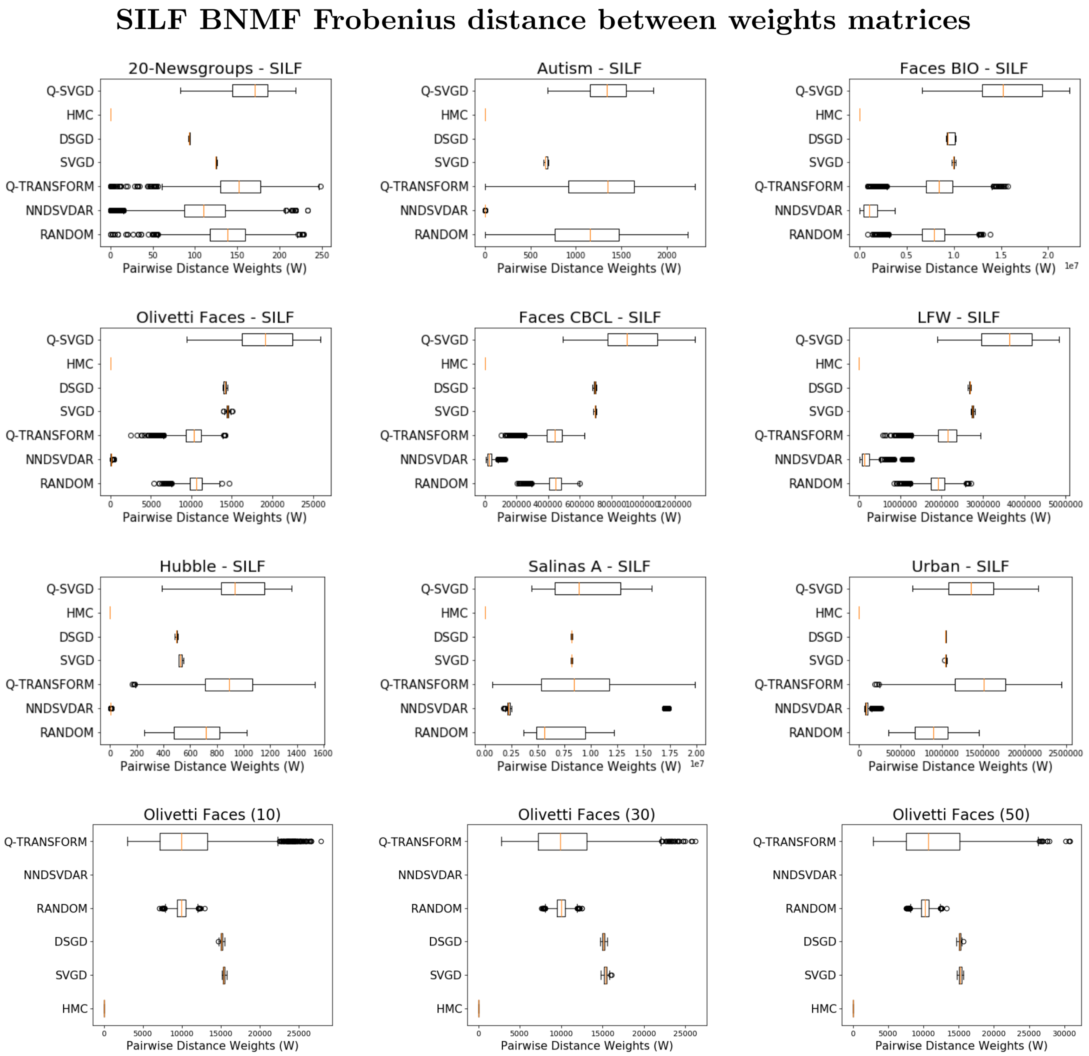


Figure 26: The pairwise distance between weights matrices shows that factorization collections obtained by HMC are most similar indicating that the HMC chain is only exploring a small region of the posterior. In many cases NNDSVDar factorizations are also very similar. Q-Transform and Random are the only algorithms that produce factorizations of high quality that are different.

Appendix C: Baseline Performance on Prediction Tasks

We show the performance of other baseline algorithms on the prediction tasks for the 20-Newsgroups and Autism datasets (see figures 6 and 8 in main text for reference)

Autism Prediction Task: Performance of baseline algorithms

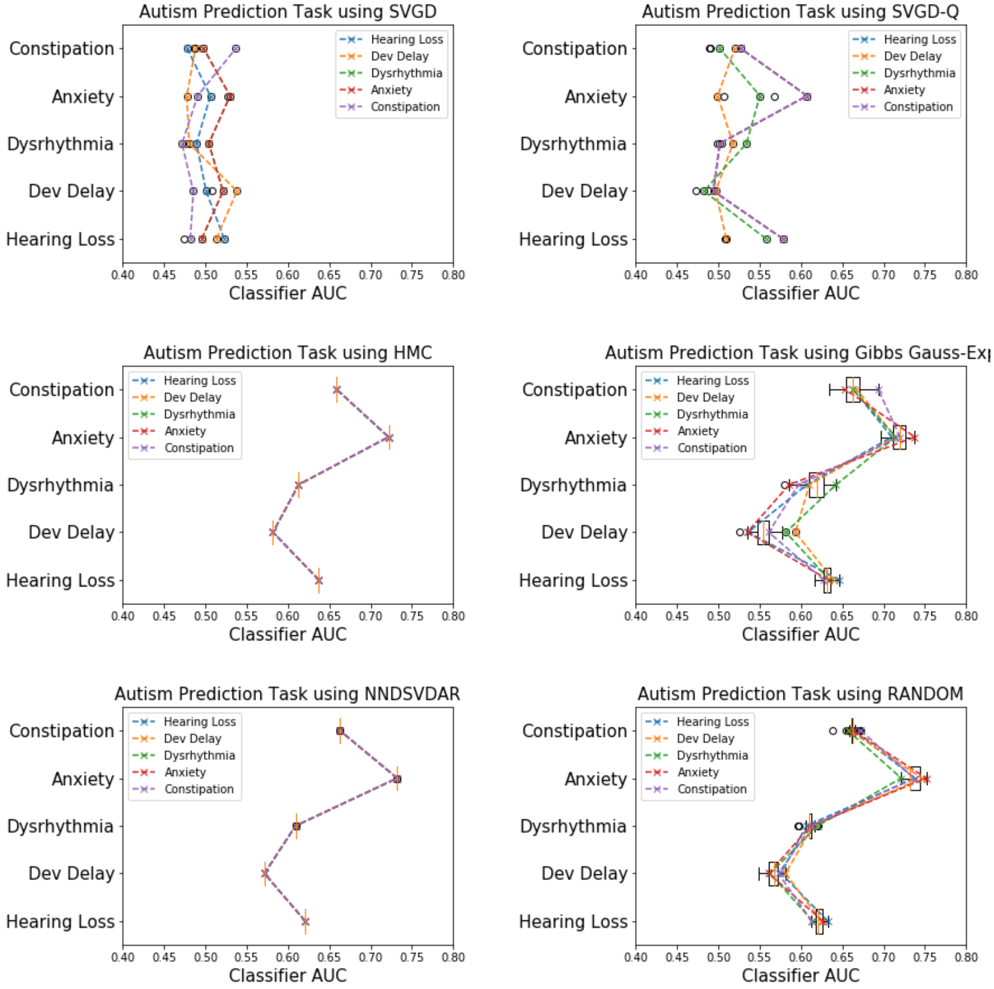


Figure 27: Variability in the prediction task on the Autism dataset shows that SVGD factorizations yield poor prediction (that are slightly improved upon by Q-SVGD), HMC and NNDSVDar factorizations predictions are high but not diverse, and Gibbs and random restarts yields performance similar to Q-Transform

20 Newsgroups Prediction Task: Performance of baseline algorithms

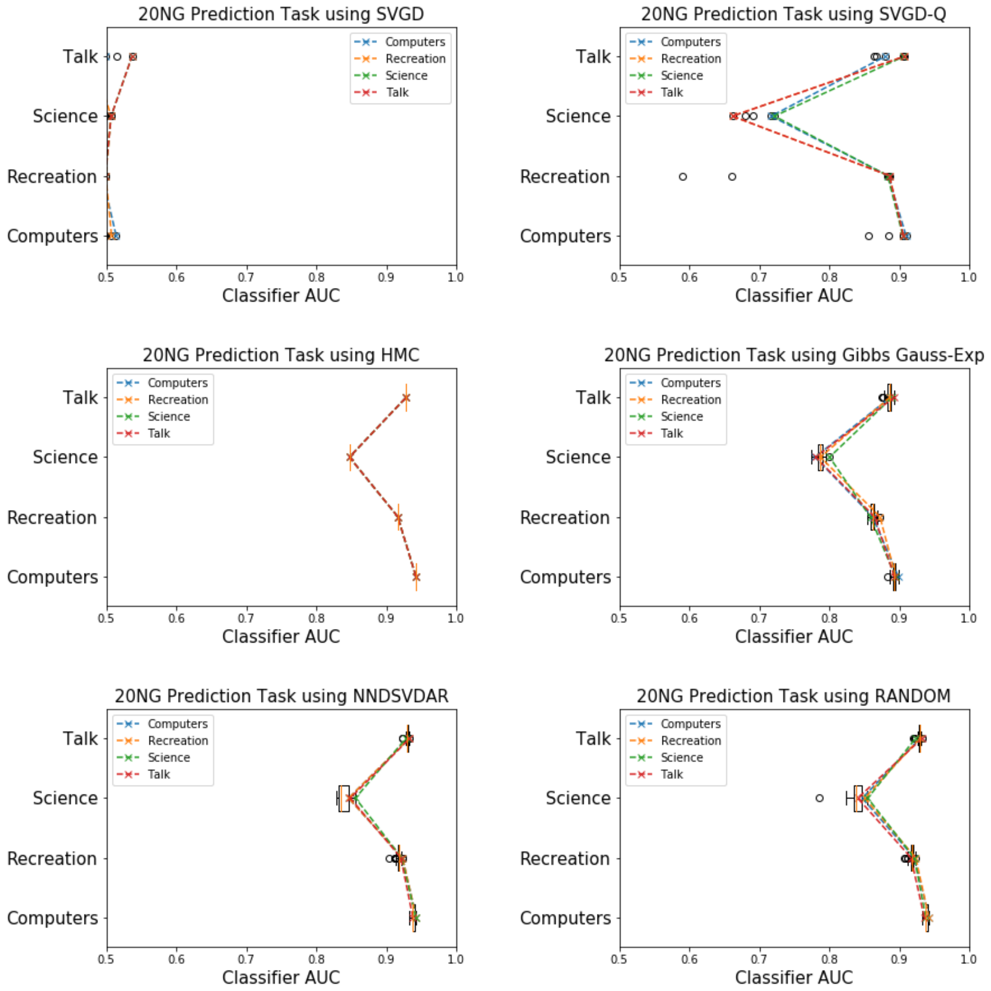


Figure 28: Variability in the prediction task on the 20 Newsgroups dataset shows that SVGD factorizations yield poor prediction (that is drastically improved upon by Q-SVGD), HMC factorizations are not diverse, NNDSVDar, Gibbs and random restarts yields some variability in prediction that is similar to Q-Transform

Appendix D: Discrete posteriors for $M = \{25, 50\}$

We provide results on the quality of discrete BNMF posteriors for the SILF and exponential-Gaussian models for $M = 25$ and $M = 50$.

Exponential-Gaussian BNMF Posterior Quality with $M = 25$

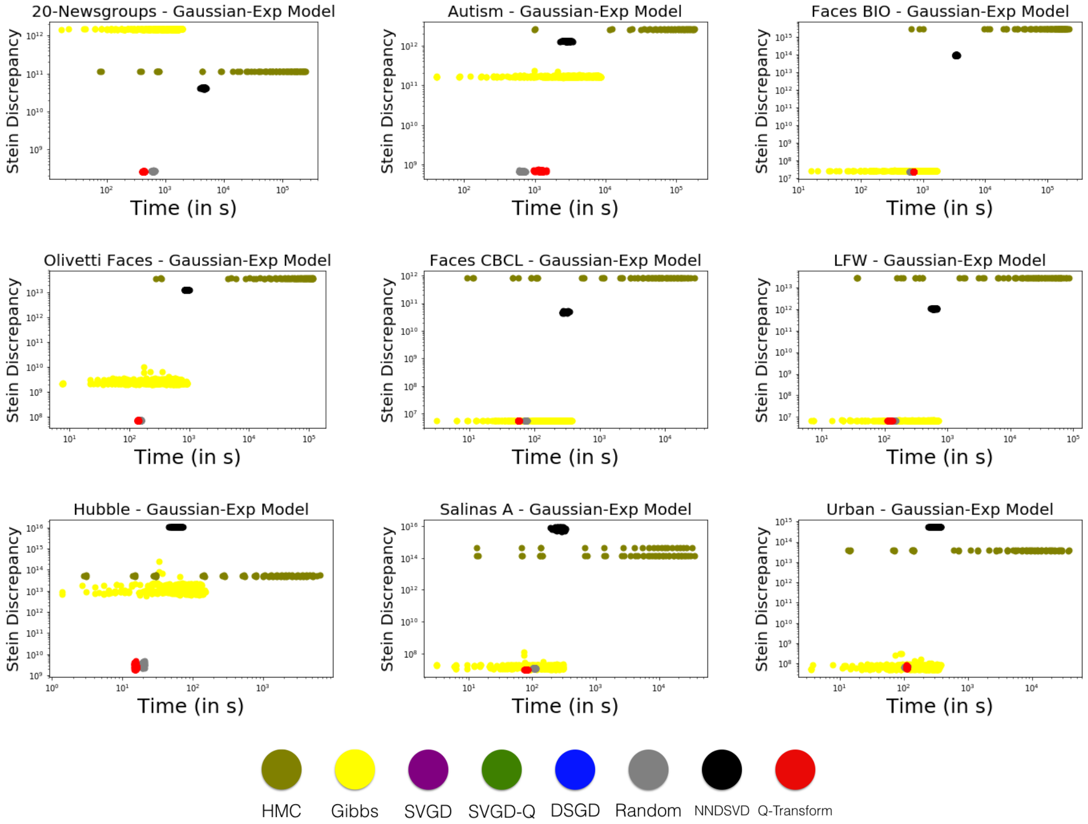


Figure 29: For each dataset we show the quality of the BNMF approximate posterior ($M = 25$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).

Exponential-Gaussian NMF Posterior Quality with $M = 50$

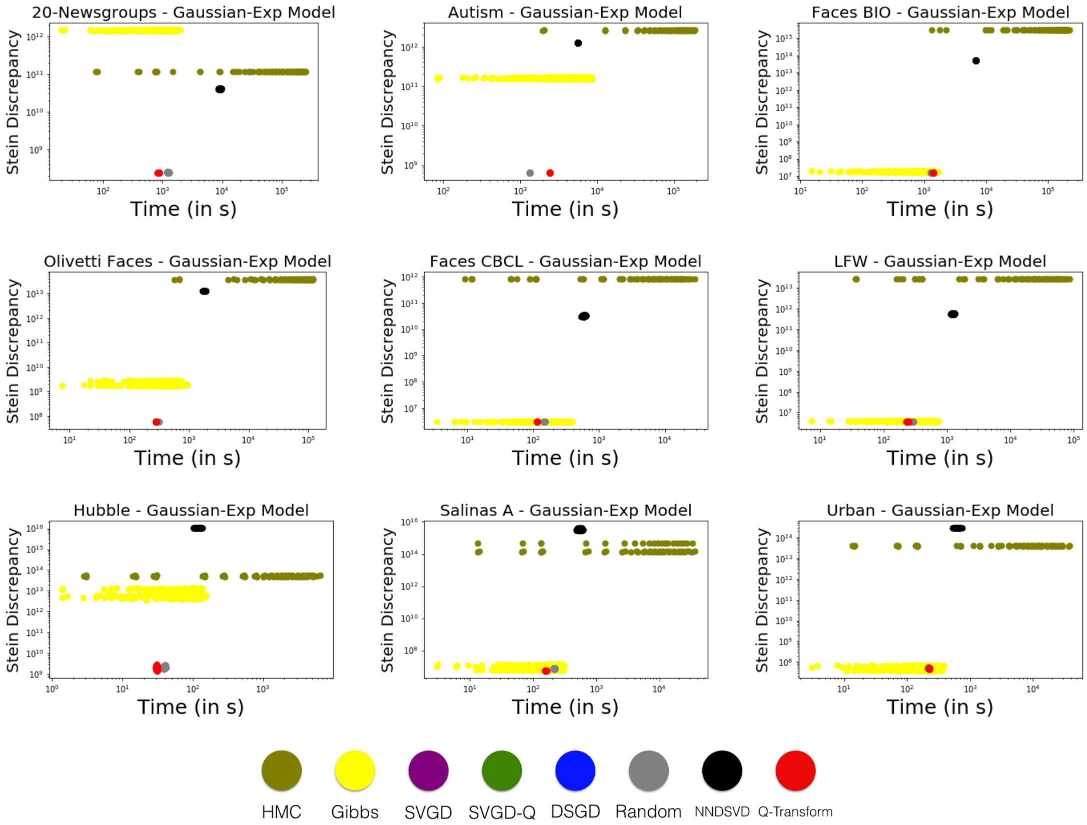


Figure 30: For each dataset we show the quality of the BNMF approximate posterior ($M = 50$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).

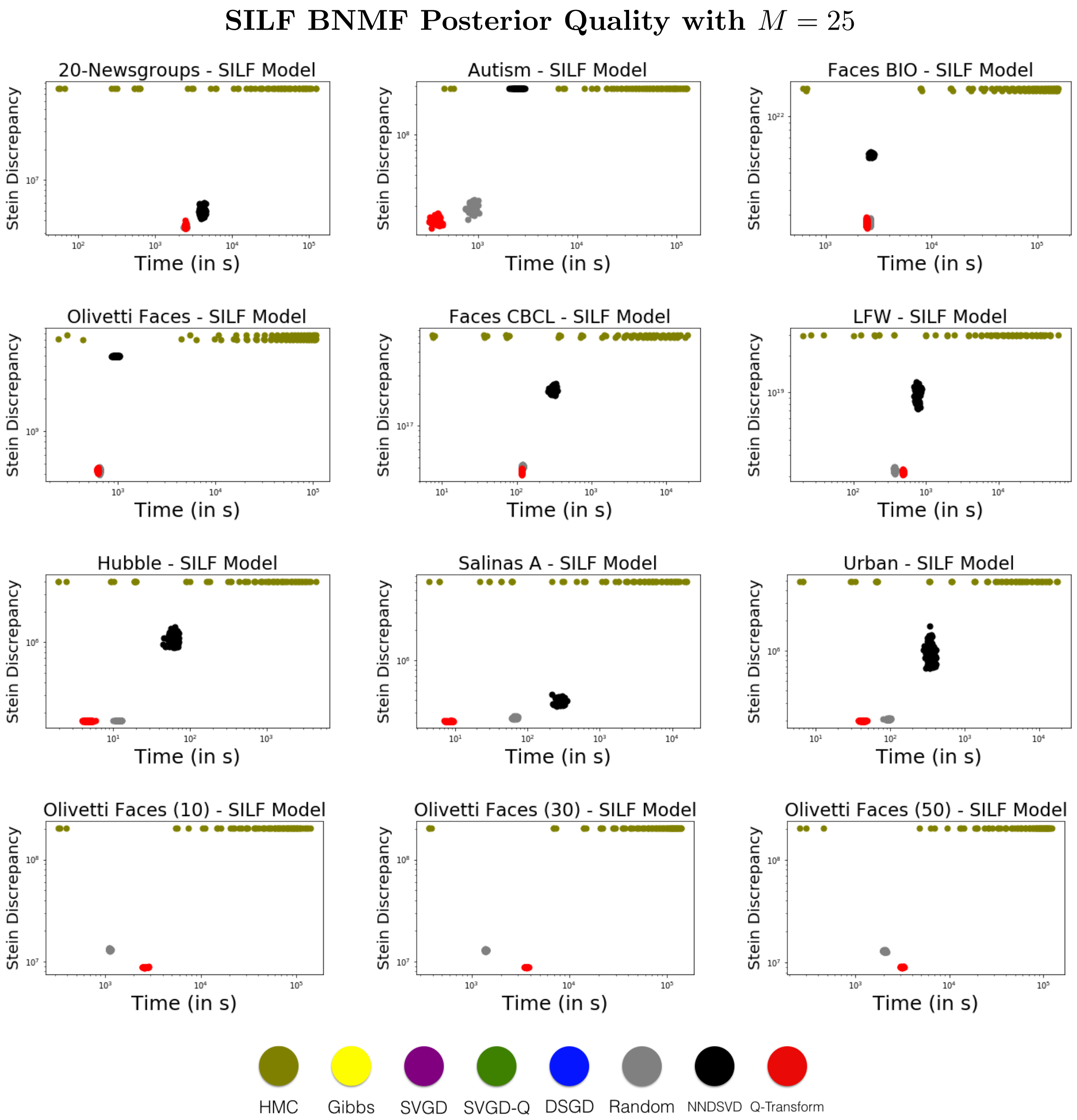


Figure 31: For each dataset we show the quality of the BNMF approximate posterior ($M = 25$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).

SILF BNMF Posterior Quality with $M = 50$

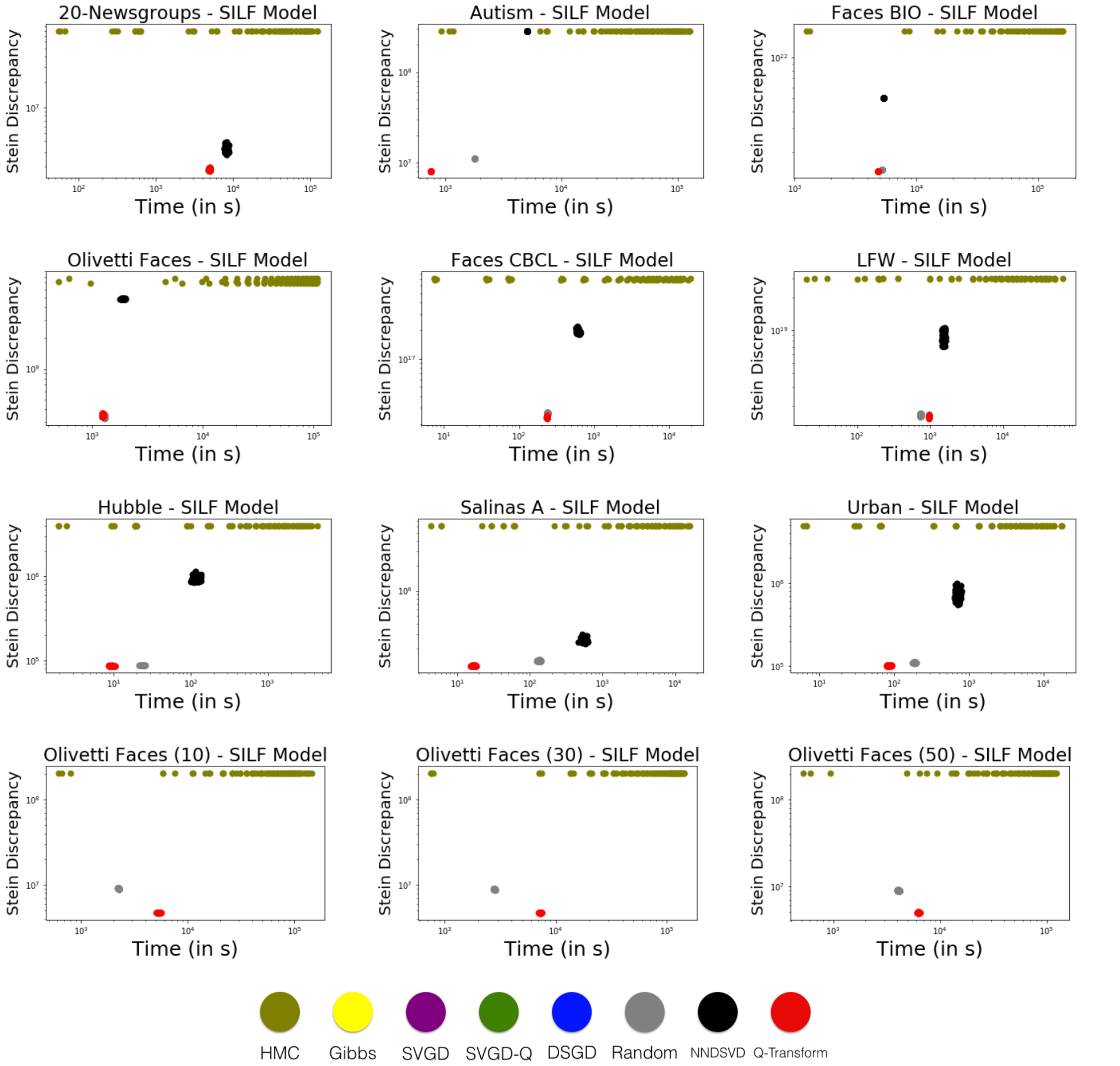


Figure 32: For each dataset we show the quality of the BNMF approximate posterior ($M = 50$) and the corresponding runtime of Q-Transform and the other baselines. Across multiple datasets, we see that the best discrete posteriors to BNMF (lowest Stein discrepancy) are produced in the least time using the Q-Transform initializations (in red).

Appendix E: Runtime Calculation

Here we provide full details on how our runtimes were calculated. Recall that both our transfer-based algorithm and the baselines all can be considered as having two phases: (1) Producing candidate particles $\{\theta_m\}_{m=1}^M$, and (2) Calculating the optimal weights $\{w_m\}_{m=1}^M$ to minimize the Stein discrepancy. Importantly, to give all the methods their best performance, we calculate optimal weights for *all* the methods, including those such as MCMC that produce unweighted collections.

Runtime for generating candidate particles Thus, all the methods only differ in how the candidate particles $\{\theta_m\}_{m=1}^M$ are found. To focus on the parts that differ, in our figures, we only report on the computational time required to produce these candidate particles. Below we describe the computations for obtaining the candidate particles:

- MCMC Methods: We keep track of the time from the initialization of the chain through each element added to the chain. At various points, we thin the current chain to the desired number of particles and record the time elapsed since initialization.
- Gradient-based Optimization: We keep track of the time taken from initialization through all gradient updates. At various points, we output the current particles and record the time elapsed since initialization.
- Initialization Approaches: For each factorization θ_m, we keep track of the time taken from initialization through optimization using NMF solvers. In the case of Q-Transform, we also include the time taken to find the transferable Q matrices (less than 0.5 seconds for all 100 pairs of Q-Transform matrices!), even though that cost is shared by all the datasets because we re-use the same matrices. The reported runtime is the sum of the time taken for the initialization procedure and for running the NMF solver.

Runtime for optimizing weights We do not report the time required to weight the particles because it is the same for all methods. In figure 33 we see the time to optimize the weights depends only on the number of particles and not their parameter values or quality.

Runtime for calculating weights: NMF with different parameters

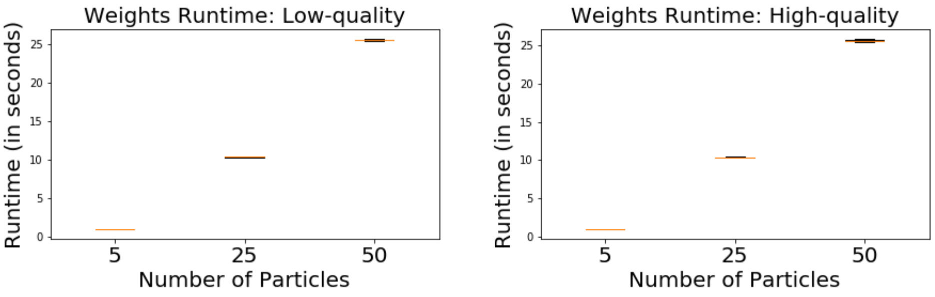


Figure 33: The runtime for calculating optimal weights (left: low-quality random parameters, right: “good” parameters obtained from NMF solvers) does not depend on the parameters; both have the same runtime costs and depend only on collection size.

Note for efficient batch weight optimization To produce the confidence intervals in our experiments, we had to run our approaches many times. To speed up these repetitions, we observed that the weight optimization step first requires computing the pairwise kernel matrix $\mathbf{K}_{ij} = \mathcal{K}_p(\theta_i, \theta_j)$ (from equation 3) and given $\mathbf{K}$, running a solver to find the weights $\{w_m\}_{m=1}^M$. When computing the weights for many subsets of a collection of factorizations (as is the case with the initialization based approaches), we can avoid recomputing terms in the kernel matrix multiple times by calculating the pairwise kernel matrix $\mathbf{K}_{\max}$ for a large collection of $M_{\max}$ factorizations. Subsequently, the matrix $\mathbf{K}_M$ for a given subset of size M (from this large collection) can be determined by simply choosing the relevant columns and rows of $\mathbf{K}_{\max}$. We use this approach to compute quickly compute weights for any M-sized collection sampled from a larger set of $M_{\max}$ factorizations. Note that this efficiency in experimental design trick does not affect our reported numbers, which include only the time to generate the candidate particles, but may help others achieve faster computational times if they are replicating our results.