Improving Optimization in Models With Continuous Symmetry Breaking

Robert Bamler [1] **Stephan Mandt** [1]

Abstract

Many loss functions in representation learning are invariant under a continuous symmetry transformation. As an example, consider word embeddings (Mikolov et al., 2013b), where the loss remains unchanged if we simultaneously rotate all word and context embedding vectors. We show that representation learning models with a continuous symmetry and a quadratic Markovian time series prior possess so-called Goldstone modes. These are low cost deviations from the optimum which slow down convergence of gradient descent. We use tools from gauge theory in physics to design an optimization algorithm that solves the slow convergence problem. Our algorithm leads to a fast decay of Goldstone modes, to orders of magnitude faster convergence, and to more interpretable representations, as we show for dynamic extensions of matrix factorization and word embedding models. We present an example application, translating modern words into historic language using a shared representation space.

1. Introduction

Symmetries are coordinate transformations that leave a certain quantity invariant, such as the loss function of a machine learning model. *Continuous* symmetries, as opposed to discrete symmetries like mirror symmetries, are parameterized by real-valued numbers, e.g., rotation angles. In theoretical physics, continuous symmetries are often the starting point to formulate effective theories (Arnol'd, 2013). In particular, *gauge theories* describe the propagation of continuous symmetries across space, and explain many fundamental forces in nature (Peskin, 1995). This paper uses methods from gauge theory in the context of machine learning.

Symmetries may be spontaneously broken by interactions, i.e., weak couplings of system parameters. Many phases of matter, e.g., solids, magnets, or superfluids, emerge when such symmetry breaking occurs. The Goldstone theorem then guarantees the existence of low energy excitations, called Goldstone modes (Altland & Simons, 2010), which signalize shallow directions of the energy landscape.

In this paper, we show that Goldstone modes also appear in representation learning. Their low excitation energy translates to a small contribution of Goldstone modes to the loss function. This leads to an ill-conditioned optimization problem and to slow convergence of gradient descent. We present an algorithm that solves the problem of slow convergence by separating the small symmetry breaking contributions to the loss function from the symmetry obeying terms. The algorithm uses artificial gauge fields as a concise parameterization of the symmetry breaking terms, and minimizes over them efficiently using natural gradients.

The particular model class we consider are dynamic matrix factorizations and dynamic embedding models (Lu et al., 2009; Koren, 2010; Charlin et al., 2015; Bamler & Mandt, 2017; Rudolph & Blei, 2017). These are time series models that exhibit multiple copies of a representation learning problem coupled by a quadratic regularizer. The coupling penalizes sudden changes of model parameters along the time dimension, thus allowing the model to share statistical strength across time steps. Specifically, the coupling is a sum over squared differences between the model parameters of adjacent time steps. In a Bayesian setup, such a coupling arises from a Gaussian Markovian time series prior.

Since these models typically do not assign any predefined meanings to specific directions in the representation space, the loss function is often invariant under a collective rotation of the embedding vectors. For example, a simultaneous rotation of all word and context embedding vectors in word2vec (Mikolov et al., 2013b) does not change the loss. A similar rotational symmetry exists in matrix factorization models. The quadratic coupling between adjacent time steps breaks the symmetry. We show that, even if the coupling is strong, the symmetry breaking contributions to the loss function can be small, which leads to a small contribution to the gradient, and to slow convergence of gradient descent.

Our contributions are as follows:

- We identify a broad class of models which suffer from

[1]Disney Research, Glendale, California. Correspondence to: Robert Bamler <robert.bamler@disneyresearch.com>, Stephan Mandt <stephan.mandt@disneyresearch.com>.

Submitted to the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, 2018.

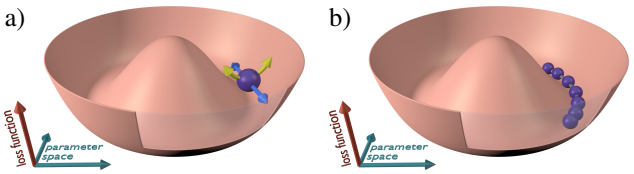

Figure 1. a) the loss ℓ of a rotationally symmetric model has a continuum of degenerate minima. b) Goldstone mode; despite its large amplitude, this configuration has small excess loss.

slow convergence due to Goldstone modes. We explain the effect of Goldstone modes on the speed of convergence both mathematically and pictorially.

- Using ideas from gauge theories, we propose Goldstone Gradient Descent (Goldstone-GD), an algorithm that speeds up convergence by separating the optimization in the subspace of symmetry transformations from the remaining coordinate directions.
- We evaluate the Goldstone-GD algorithm experimentally with dynamic matrix factorizations and Dynamic Word Embeddings. We find that Goldstone-GD converges orders of magnitude faster and finds more interpretable embedding vectors than standard gradient descent (GD) or GD with diagonal preconditioning.
- For Dynamic Word Embeddings (Bamler & Mandt, 2017), Goldstone-GD allows us to find historical synonyms of modern English words, such as "wagon" for "car". Without our advanced optimization algorithm, we could not obtain this result.

Our paper is structured as follows. In Section 2, we specify the model class under consideration, provide concrete example models, and introduce the slow convergence problem. Section 3 describes related work. In Section 4, we propose the Goldstone-GD algorithm that solves the slow convergence problem. We report experimental results in Section 5 and provide concluding remarks in Section 6.

2. Problem Setting

In this section, we discuss the problem of slow convergence in representation learning with a continuous symmetry and a time series prior. We first provide a geometric visualization of Goldstone modes (Section 2.1), describe the relevant the class of models (Section 2.2), and list concrete examples (Section 2.3). We finally show that Goldstone modes lead to slow convergence of gradient descent (GD) (Section 2.4).

2.1. Geometric Picture of Goldstone Modes

We give an intuitive picture of Goldstone modes in representation learning, deferring a more formal one to Section 2.4.

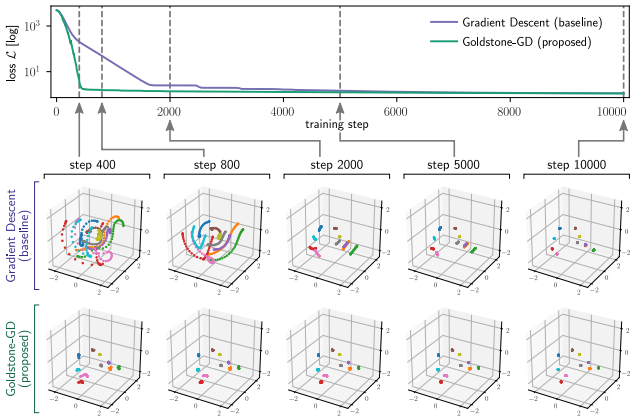

Figure 2. Experimental evidence for Goldstone modes in a dynamic matrix factorization with embedding dimension $d = 3$ (see Section 5.1), and slow convergence of gradient descent.

We consider a representation learning model whose loss is invariant under rotations of all embedding vectors in the representation space. For example, consider factorizing a large matrix X into the product $U^\top V$ of two smaller matrices U and V by minimizing the loss $\ell(U, V) = ||X - U^\top V||_2^2$. We refer to the columns of U and V as embedding vectors. Rotating all embedding vectors by the same orthogonal matrix R such that $U \leftarrow RU$ and $V \leftarrow RV$ does not change the loss since $(RU)^\top RV = U^\top (R^\top R) V = U^\top V$.

For the sake of illustration, we consider a toy model with a two-dimensional representation space and a single embedding vector. The red surface in Figure 1a) shows the loss function ℓ. The loss is rotationally symmetric, which leads to a degenerate minimum. The purple sphere depicts the single embedding vector and sits at one exemplary minimum. We can force the model to prefer one minimum over all others by adding a small term to the loss that does not obey the rotational symmetry. Even if this symmetry breaking term is tiny it can change the position of the selected minimum by a large distance in the representation space.

Such a small symmetry breaking term arises, e.g., in time series models like dynamic matrix factorizations (Lu et al., 2009; Koren, 2010; Sun et al., 2012; Charlin et al., 2015). These models combine several instances of a rotationally symmetric model, coupling them with a quadratic regularizer that penalizes differences between model parameters of adjacent instances. Figure 1b) illustrates a time series model with $T = 7$ time steps. Each purple sphere depicts the embedding vector for one time step. The embedding vectors are connected by a quadratic coupling, which we can think of as springs between neighboring spheres (not drawn). The fact that the chain is not yet contracted to a single point reflects a small deviation from the minimum of the total loss, called a Goldstone mode.

Goldstone modes appear in practice: the $3d$ plots in Figure 2 show snapshots of the embedding space in a small scale Gaussian dynamic matrix factorization with $T = 30$ and a $3d$ representation space (details in Section 5.1). Points with the same color show the evolution of a given embedding vector along the time dimension of the model. In this toy example, the local loss ℓ is identical for each time step. Thus, in the optimum, the chains of points should, again, contract to a single point. The upper row of $3d$ plots shows that Goldstone modes decay only slowly under gradient descent (the contraction of the chains happens slowly). In contrast, our proposed Goldstone-GD algorithm eliminates Goldstone modes quickly (bottom row). Goldstone modes contribute only little to the loss, as can be seen in the upper panel in Figure 2 after step $\sim$ 2000. However, the small difference in the loss can manifest itself in a large difference of the fitted embedding vectors, as we discuss in Section 2.4.

2.2. Model Class Specification

More formally, the slow convergence problem arises in the following class of models. We consider data $\mathbf{X} \equiv \{X_t\}_{t=1:T}$ that are associated with additional metadata t, such as a time stamp. For each t, the task is to learn a low dimensional representation Z_t by minimizing a local loss function $\ell(X_t; Z_t)$. We add a quadratic regularizer $\psi(\mathbf{Z})$ that couples the representations $\mathbf{Z} \equiv \{Z_t\}_{t=1:T}$ along the t-dimension. We refer to ψ as the prior, adopting language of a Bayesian setup. Thus, the overall loss function is

$$\mathcal{L}(\mathbf{Z}) = \sum_{t=1}^{T} \ell(X_t; Z_t) + \psi(\mathbf{Z}). \tag{1}$$

For each task t, the representation Z_t is a matrix whose columns are low dimensional embedding vectors. We assume that ℓ is invariant under a collective rotation of Z_t: let R be an arbitrary orthogonal rotation matrix of the same dimension as the embedding dimension, then

$$\ell(X_t; RZ_t) = \ell(X_t; Z_t). \tag{2}$$

Finally, we consider a special form of prior which is quadratic in $\mathbf{Z}$, and which is defined in terms of a sparse symmetric coupling matrix $\mathbf{L} \in \mathbb{R}^{T\times T}$:

$$\psi(\mathbf{Z}) = \tfrac{1}{2}\mathrm{Tr}(\mathbf{Z}^\top \mathbf{L}\mathbf{Z}). \tag{3}$$

Here, the matrix-vector multiplications are carried out in t-space, and the trace runs over the remaining dimensions. As we show in Section 2.3, Gaussian Markovian time series priors fall into this class, where $\mathbf{L}$ is a tridiagonal matrix. More generally, $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the Laplacian matrix of a sparse weighted graph (Poignard et al., 2018). Here, $\mathbf{A}$ is the adjacency matrix, whose entries are the coupling strengths between tasks, and the degree matrix $\mathbf{D}$ is diagonal and defined so that the entries of each row of $\mathbf{L}$ sum up to zero.

Equations 1, 2, and 3 specify the problem class of interest in this paper. We introduce the specific models we use for our experiments in Section 2.3. In Section 2.4, we show that this model class suffers from a slow convergence problem.

2.3. Exemplary Models

In this section, we introduce three particular instances of the model class presented in Section 2.2. These are also the models that we investigate in our experiments in Section 5.

Dense dynamic matrix factorization. Consider the task of factorizing a large matrix X_t into a product $U_t^\top V_t$ of two smaller matrices. The latent representation is the concatenation of the two embedding matrices,

$$Z_t \equiv (U_t, V_t). \tag{4}$$

In a Gaussian matrix factorization, the local loss function is

$$\ell(X_t; Z_t) = -\log \mathcal{N}(X_t; U_t^\top V_t, I) \tag{5}$$

In dynamic matrix factorization models, the data $\mathbf{X}$ are observed sequentially at discrete time steps t, and the representations $\mathbf{Z}$ capture the temporal evolution of latent embedding vectors. We use a Markovian Gaussian time series prior with a coupling strength λ,

$$\psi(\mathbf{Z}) = \frac{\lambda}{2}\sum_{i=1}^{N}\sum_{t=1}^{T-1} ||z_{t+1,i} - z_{t,i}||_2^2. \tag{6}$$

Here, the vector $z_{t,i}$ is the i^{th} column in the matrix Z_t, i.e., the i^{th} embedding vector, and N is the number of columns. The prior allows the model to share statistical strength across time. By multiplying out the square, we find that ψ has the form of Eq. 3, and its Laplacian matrix $\mathbf{L}$ is tridiagonal.

Sparse dynamic matrix factorization. In a sparse matrix factorization, the local loss ℓ involves only few components of the matrix $U_t^\top V_t$, where the latent representation is again $Z_t \equiv (U_t, V_t)$. We consider a model for movie ratings where each user rates only few movies. When user i rates movie j in time step t, we model the likelihood to obtain the binary rating $x \in \{\pm 1\}$ with a logistic regression,

$$p(x|u_{t,i}, v_{t,j}) = \sigma(x\, u_{t,i}^\top v_{t,j}) \tag{7}$$

with the sigmoid function $\sigma(\xi) = 1/(1 + e^{-\xi})$. The full likelihood $p(X_t|Z_t)$ for time step t is the product of the likelihoods of all observed ratings at time step t. The local loss function is

$$\ell(X_t; Z_t) = -\log p(X_t|Z_t) + \frac{\gamma}{2}||Z_t||_2^2. \tag{8}$$

Here, $||\cdot||_2$ is the Frobenius norm, and we add a quadratic regularizer with strength γ since data for some users or movies may be scarce. We distinguish this local regularizer from the time series prior ψ, given again in Eq. 6, as the local regularizer does not break the rotational symmetry.

Dynamic Word Embeddings. Word embeddings map words from a large vocabulary to a low dimensional representation space such that neighboring words are semantically similar, and differences between word embedding vectors capture syntactic and semantic relations. We consider the Dynamic Word Embeddings model (Bamler & Mandt, 2017), which uses a probabilistic interpretation of the Skip-Gram model with negative sampling, also known as word2vec (Mikolov et al., 2013b; Barkan, 2017), and combines it with a time series prior. The model is trained on T text sources with time stamps t, and it assigns two time dependent embedding vectors $u_{t,i}$ and $v_{t,i}$ to each word i from a fixed vocabulary. The embedding vectors are obtained by simultaneously factorizing two matrices, which contain so-called positive and negative counts of word-context pairs. Therefore, the representation $Z_t \equiv (U_t, V_t)$ for each time step is invariant under orthogonal transformations. The prior is a discretized Ornstein-Uhlenbeck (OU) process, i.e., it combines a random diffusion process with a quadratic regularizer. Analogously to the movie recommendations model, we absorb the quadratic regularizer in the per task loss ℓ.

2.4. The Slow Convergence Problem

In this section, we identify the slow convergence problem in time series models with continuous symmetries as an ill-conditioning of the Hessian of the loss $\mathcal{L}$ at its minimum.

We consider a model of the form of Eqs. 1-3. Due to the continuous rotational symmetry, each local loss function ℓ has a manifold of degenerate minima, and the Hessian of ℓ vanishes within this manifold. Consider, e.g., the two-dimensional rotationally symetric loss ℓ in Figure 1a). It is given by $\ell(z) = -\frac{1}{2}||z||_2^2 + \frac{1}{4}||z||_2^4$ where $z \in \mathbb{R}^2$. Setting $\nabla_z \ell$ to zero, we find that the manifold of degenerate minima is the circle with radius $||z||_2 = 1$ around the origin. The Hessian at any point z on this circle is $H^{(\ell)} = 2zz^\top$. It has a zero eigenvalue for the direction perpendicular to z, i.e., the direction of a small rotation (blue arrows in Figure 1a)).

Thus, within the subspace of small symmetry transformations, only the Hessian of the prior remains, which is

$$H^{(\psi)}_{tki,t'lj} \equiv \frac{\partial^2 \psi(\mathbf{Z})}{\partial Z_{tki}\, \partial Z_{t'lj}} = \delta_{kl}\delta_{ij}\mathbf{L}_{tt'}. \tag{9}$$

Here, δ denotes the Kronecker delta, and $k, l \in \{1, \dots, d\}$ run over the rows and $i, j \in \{1, \dots, N\}$ over the columns of the matrices Z_t and $Z_{t'}$. The Hessian of ψ therefore has the same eigenvalues as the Laplacian matrix $\mathbf{L}$ of the coupling graph, each with a multiplicity of Nd.

Every Laplacian matrix has a zero eigenvalue because its rows add up to zero. The corresponding eigenvector $w_0 \propto (1, \dots, 1)^\top \in \mathbb{R}^T$ describes a global rotation of the representations for all tasks t by the same amount, which does not concern us here. A global rotation leaves the total loss $\mathcal{L}$ invariant, implying that the minimum of $\mathcal{L}$ is degenerate. Convergence within the valley of degenerate minima is not required since any minimum is a valid solution.

The second smallest eigenvalue of a Laplacian matrix is called algebraic connectivity (de Abreu, 2007), and it is small in sparse graphs. In the Markovian time series prior in Eq. 6, the coupling graph is a one-dimensional chain, with algebraic connectivity $2\lambda(1 - \cos(\pi/T))$ (de Abreu, 2007), which vanishes as $O(1/T^2)$ for large T. Thus, even if the coupling strength λ is strong, the lowest nonzero eigenvalue of $H^{(\psi)}$ can be small for large T. In our experiments in Section 5, T is 30, 100, and 188, respectively. The small eigenvalue of the Hessian leads to an ill-conditioned optimization problem with slow convergence. We present our proposed solution to speed up convergence in Section 4.

3. Related Work

Symmetries in the input space of machine learning models have been exploited to reduce the number of independent model parameters. Convolutional neural networks (CNNs) (LeCun et al., 1998) use discrete translational symmetry to tie weights in a neural network layer. This idea was generalized to arbitrary discrete symmetries (Gens & Domingos, 2014) and to the (discrete) permutation symmetry of sets (Zaheer et al., 2017). Discrete symmetries are also exploited in inference with graphical models to reduce the size of the effective latent space (Bui et al., 2012; Noessner et al., 2013). Discrete symmetries do not lead to a small gradient in gradient descent because they do not give rise to configurations with arbitrarily small deviations from the optimum.

In this work, we consider models with continuous symmetries. These arise in the latent representation space, e.g., in deep neural networks (Badrinarayanan et al., 2015), matrix factorization (Mnih & Salakhutdinov, 2008; Gopalan et al., 2015), linear factor models (Murphy, 2012), and word embeddings (Mikolov et al., 2013a;b; Pennington et al., 2014; Barkan, 2017). Dynamic matrix factorizations (Lu et al., 2009; Koren, 2010; Sun et al., 2012; Charlin et al., 2015) and dynamic word embeddings (Bamler & Mandt, 2017; Rudolph & Blei, 2017) combine such models with a time series prior, which does not obey the symmetry. These are the models whose optimization we address in this paper.

The slow convergence in these models is caused by shallow directions of the loss function. Popular methods to escape a shallow valley of the loss in deep learning models (Duchi et al., 2011; Zeiler, 2012; Kingma & Ba, 2014) rely on diagonal preconditioning. As confirmed by our experiments, diagonal preconditioning does not speed up convergence in the models addressed in this paper since the shallow directions correspond to collective rotations of many model parameters, which are not aligned with the coordinate axes.

Natural gradients (Amari, 1998; Martens, 2014) are a more general form of preconditioning that has been applied to deep learning (Pascanu & Bengio, 2013) and to variational inference (Hoffman et al., 2013). Natural gradients take the information geometry of the parameter space into account. They use a Riemannian metric to map the gradient, which lives in the tangent space, to an update step in the cotangent space (Ollivier, 2015a;b). In general, natural gradients are expensive to obtain. We show that using an appropriate parameterization of the symmetry transformations, natural gradients in the symmetry subspace are cheap.

4. Goldstone Gradient Descent

In this section, we present our solution to the slow convergence problem that we identified in Section 2.4. Algorithm 1 summarizes the proposed Goldstone Gradient Descent (Goldstone-GD) algorithm. We lay out details in Section 4.1, and discuss hyperparameters in Section 4.2.

The algorithm minimizes a loss function $\mathcal{L}$ of the form of Eqs. 1-3. It alternates between standard gradient steps in the full parameter space (lines 4-5), and natural gradient steps in the subspace of small symmetry transformations ('symmetry subspace' for short; lines 7-8). Switching between the two spaces involves an overhead due to coordinate transformations (lines 6 and 9). We therefore always execute several consecutive gradient steps before switching between spaces (hyperparameters k_1 and k_2). For simplicity, the update step in line 5 is formulated here with a single constant learning rate. In our experiments, we also use adaptive learning rates and minibatch sampling, see Section 5.

4.1. Optimization in the Symmetry Subspace

We now describe the optimization in the symmetry subspace (lines 7-8 in Algorithm 1), and the coordinate transformations to and from this subspace (lines 6 and 9, respectively).

For given initial model parameters $\mathbf{Z}$, we minimize the loss $\mathcal{L}$ by only applying symmetry transformations. Let $\mathbf{R} \equiv \{R_t\}_{t=1:T}$ denote T orthogonal matrices. The task is to minimize the following auxiliary loss function over $\mathbf{R}$,

$$\mathcal{L}'(\mathbf{Z};\mathbf{R}) \equiv \mathcal{L}(R_1 Z_1, \ldots, R_T Z_T) - \mathcal{L}(Z_1, \ldots, Z_T) \tag{10}$$

with the nonlinear constraint $R_t^\top R_t = I \ \forall t$. If $\mathbf{R}^*$ minimizes $\mathcal{L}'$, then replacing $Z_t \leftarrow R_t^* Z_t$ decreases the loss $\mathcal{L}$ by eliminating all Goldstone modes. The second term on the right-hand side of Eq. 10 does not influence the minimization as it is independent of $\mathbf{R}$. Subtracting this term makes $\mathcal{L}'$ independent of the local loss functions ℓ: by using Eqs. 1-2, we can express $\mathcal{L}'$ in terms of only the prior ψ,

$$\mathcal{L}'(\mathbf{Z};\mathbf{R}) = \psi(R_1 Z_1, \ldots, R_T Z_T) - \psi(Z_1, \ldots, Z_T). \tag{11}$$

Algorithm 1: Goldstone Gradient Descent (Goldstone-GD)

Input: Loss function $\mathcal{L}$ of the form of Eqs. 1-3; learning rate ρ; integer numbers k_1 and k_2 of consecutive learning steps in full parameter space and in symmetry subspace, respectively.
Output: Local minimum of $\mathcal{L}$.

1 Initialize model parameters $\mathbf{Z}$ randomly
2 Initialize gauge fields $\tilde{\mathbf{\Gamma}} \leftarrow \mathbf{0}$
3 **repeat**
4 **repeat** k_1 **times**
5 Set $\mathbf{Z} \leftarrow \mathbf{Z} - \rho \nabla_{\mathbf{Z}} \mathcal{L}(\mathbf{Z})$
 ▷ *gradient step in full parameter space*
  end
6 Obtain $\mathbf{M}$ and ρ' from Eqs. 15 and 17
 ▷ *transformation to symmetry subspace*
7 **repeat** k_2 **times**
8 Set $\tilde{\mathbf{\Gamma}} \leftarrow \tilde{\mathbf{\Gamma}} - \rho' \mathbf{L}^+ \nabla_{\tilde{\mathbf{\Gamma}}} \mathcal{L}''(\tilde{\mathbf{\Gamma}};\mathbf{M})$
 ▷ *natural gradient step in symmetry subspace*
  end
9 Set $Z_{tki} \leftarrow Z_{tki} + \sum_{l=1}^{d} (\tilde{\Gamma}_{tkl} - \tilde{\Gamma}_{tlk}) Z_{tli} \quad \forall t,k,i$
 ▷ *transformation back to full parameter space*
until convergence

Artificial gauge fields. We turn the constrained minimization into an unconstrained one using a result from the theory of Lie groups (Hall, 2015). First, note that only special orthogonal transformations with $\det(R_t) = 1$ contribute to the slow convergence problem. Mirror transformations with $\det(R_t) = -1$ do not lead to Goldstone modes. Every special orthogonal matrix R_t is the matrix exponential of a skew symmetric $d \times d$ matrix Γ_t. Here, skew symmetry means that $\Gamma_t^\top = -\Gamma_t$, and the matrix exponential function $\exp(\cdot)$ is defined by its series expansion,

$$R_t = \exp(\Gamma_t) \equiv I + \Gamma_t + \frac{1}{2!}\Gamma_t^2 + \frac{1}{3!}\Gamma_t^3 + \ldots \tag{12}$$

which is not to be confused with the componentwise exponential of Γ_t, (the term Γ_t^2 in Eq. 12 is the matrix product of Γ_t with itself, not the componentwise square). Eq. 12 follows from the Lie group–Lie algebra correspondence for the Lie group $SO(d)$ (Hall, 2015). Note that setting all components of Γ_t to zero yields the identity I. For small Γ_t, R_t is a small rotation close to the identity. We enforce skew symmetry of Γ_t by parameterizing it via the skew symmetric part of an unconstrained $d \times d$ matrix $\tilde{\Gamma}_t$, i.e.,

$$\Gamma_t = \tilde{\Gamma}_t - \tilde{\Gamma}_t^\top. \tag{13}$$

We call the components of $\tilde{\mathbf{\Gamma}} \equiv \{\tilde{\Gamma}_t\}_{t=1:T}$ the gauge fields, invoking an analogy to gauge theory in physics.

Taylor expansion in the gauge fields. Eqs. 12-13 turn the constrained minimization of $\mathcal{L}'$ over $\mathbf{R}$ into an un-

constrained minimization over $\tilde{\mathbf{\Gamma}}$. However, the matrix-exponential function in Eq. 12 is numerically expensive, and its derivative is complicated because the group $SO(d)$ is non-abelian. We simplify the problem by introducing an approximateion. As the model parameters $\mathbf{Z}$ approach the minimum of $\mathcal{L}$, the optimal rotations R_t^* that minimize $\mathcal{L}'$ converge to the identity, and thus the gauge fields converge to zero. In this limit, the approximation becomes exact.

We approximate the auxiliary loss function $\mathcal{L}'$ by a second order Taylor expansion $\mathcal{L}''$. In detail, we truncate Eq. 12 after the term quadratic in Γ_t and insert the truncated series into Eq. 11. We multiply out the quadratic form in the prior ψ, Eq. 3, and neglect again all terms of higher than quadratic order in $\mathbf{\Gamma}$. Using the skew symmetry of Γ_t and the symmetry of the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$, we find

$$\mathcal{L}''(\tilde{\mathbf{\Gamma}}; \mathbf{M}) = \sum_{t,t'} A_{tt'} \operatorname{Tr}\left[\left(\Gamma_{t'} + \frac{1}{2}(\Gamma_{t'} - \Gamma_t)\Gamma_t\right) M_{tt'}\right] \tag{14}$$

where the trace runs over the embedding space, and for each $t, t' \in \{1, \dots, T\}$, we define the matrix $M_{tt'} \in \mathbb{R}^{d\times d}$,

$$M_{tt'} \equiv \sum_{i=1}^{N} z_{t,i} z_{t',i}^\top. \tag{15}$$

We evaluate the matrices $M_{tt'}$ when we switch from the optimization in the full parameter space to the optimization in the symmetry subspace, see line 6 in Algorithm 1. Note that the adjacency matrix $\mathbf{A}$ is sparse, and that we only need to obtain those matrices $M_{tt'}$ for which $A_{tt'}$ is nonzero.

Once we obtain gauge fields $\tilde{\mathbf{\Gamma}}^*$ that minimize $\mathcal{L}''$, the optimal update step for the model parameters would be $Z_t \leftarrow \exp(\tilde{\Gamma}_t^* - \tilde{\Gamma}_t^{*\top}) Z_t$. We avoid again a full evaluation of the expensive matrix exponential function and approximate it for small gauge fields by truncating after the linear term, resulting in the update step in line 9 of Algorithm 1.

Natural gradients. Lines 7-8 in Algorithm 1 minimize $\mathcal{L}''$ over the gauge fields $\tilde{\mathbf{\Gamma}}$ using GD. We speed up convergence using the fact that $\mathcal{L}''$ depends only on the prior ψ and not on ℓ. Since we know the Hessian of ψ, we can use natural gradients (Amari, 1998), resulting in the update step

$$\tilde{\mathbf{\Gamma}} \leftarrow \tilde{\mathbf{\Gamma}} - \rho' \mathbf{L}^+ \nabla_{\tilde{\mathbf{\Gamma}}} \mathcal{L}''(\tilde{\mathbf{\Gamma}}; \mathbf{M}) \tag{16}$$

with a constant learning rate ρ', discussed below. Here, we precondition with the psuedoinverse $\mathbf{L}^+$ of the Laplacian matrix. We obtain $\mathbf{L}^+$ by taking the eigendecomposition of $\mathbf{L}$ and inverting the eigenvalues, except for a single zero eigenvalue, which we leave at zero. This has to be done only once before entering the training loop. The zero eigenvalue corresponds to a global rotation of all Z_t by the same orthogonal matrix, which does not reduce the loss.

Table 1. Computational complexities of operations in Goldstone-GD (L=line in Algorithm 1; #=frequency of the operation).

L	OPERATION	COMPLEXITY	#
5	gradient step in full param. space	model dependent	$\times k_1$
6	transformation to symmetry space	$O(TNd^2)$	$\times 1$
8	nat. grad. step in symmetry space	$O(Td^3 + T^2d^2)$	$\times k_2$
9	transformation to full param. space	$O(TNd^2)$	$\times 1$

We do not reset the gauge fields to zero when switching back to the full parameter space. Thus, when we return to the minimization of $\mathcal{L}''$ after interleaving k_1 standard gradient steps, we preinitialize $\tilde{\mathbf{\Gamma}}$ with the result from the previous minimization. This turned out to speed up convergence in our experiments. We explain the speedup with the observation that, by remembering the previous update step, the gauge fields act like a momentum in the symmetry subspace.

Learning rate. We find that we can automatically set ρ' in Eq. 16 to a value that leads to fast convergence,

$$\rho' = \frac{1}{TN\langle Z^2\rangle} \quad \text{with} \quad \langle Z^2\rangle \equiv \frac{1}{TNd}\sum_{t,k,i} Z_{tki}^2. \tag{17}$$

We arrive at this choice of learning rate due to the following considerations. First, consider the easier task of minimizing $\psi(\mathbf{Z})$ over the full parameter space. Here, the same preconditioning as in Eq. 16 with $\mathbf{L}^+$ leads to the update

$$\mathbf{Z} \leftarrow \mathbf{Z} - \rho\, \mathbf{L}^+ \nabla_{\mathbf{Z}} \psi(\mathbf{Z}) = \mathbf{Z} - \rho\, \mathbf{L}^+ \mathbf{L} \mathbf{Z} \tag{18}$$

where $\mathbf{L}^+\mathbf{L}$ is a projection that only removes the (irrelevant) nullspace of $\mathbf{L}$. The minimization would thus find the exact minimum $\mathbf{Z} = \mathbf{0}$ of ψ with a single update step with $\rho = 1$.

Of course, the objective for the minimization in the full parameter space is not ψ but the total loss $\mathcal{L}$. In the symmetry subspace, however, the objective is indeed (a reparameterization of) ψ, see Eq. 11. The reparameterization in terms of $\tilde{\mathbf{\Gamma}}$ leads to the matrices $M_{tt'}$ in Eq. 14, which are quadratic in the components of $\mathbf{Z}$ and linear in N. This suggests a learning rate $\rho' \propto 1/(N\langle Z^2\rangle)$. We find empirically for large models that the t-dependency of $M_{tt'}$ leads to a small mismatch between $\mathbf{L}^+$ and the Hessian of $\mathcal{L}''$. The more conservative choice of learning rate in Eq. 17 leads to fast convergence of the gauge fields in all our experiments.

4.2. Hyperparameters

Goldstone-GD has two integer hyperparameters, k_1 and k_2, which control the frequency of execution of each operation. Table 1 lists the computational complexity of each operation, assuming that the adjacency matrix $\mathbf{A}$ has $O(T)$ nonzero entries. Note that representation learning usually involves a dimensionality reduction, i.e., d is often orders of magnitude

smaller than N. Therefore, update steps in the symmetry subspace are cheap. In our experiments, we always set k_1 and k_2 such that the runtime increases by less than 10% compared to standard GD with the same number of update steps in the full parameter space.

5. Experiments

We evaluate the proposed Goldstone-GD optimization algorithm on the three example models introduced in Section 2.3. We compare Goldstone-GD to standard GD, to AdaGrad (Duchi et al., 2011), and to Adam (Kingma & Ba, 2014). Goldstone-GD converges orders of magnitude faster and fits more interpretable word embeddings.

5.1. Visualizing Goldstone Modes With Artificial Data

Model and data preparation. We fit the dynamic Gaussian matrix factorization model defined in Eqs. 4-6 in Section 2.3 to small scale artificial data. In order to visualize Goldstone modes in the embedding space we choose an embedding dimension of $d = 3$ and, for this experiment only, we fit the model to time independent-data. This allows us to monitor convergence since we know that the matrices U_t^* and V_t^* that minimize the loss are also time-independent. We generate artificial data for the matrix $X \in \mathbb{R}^{10\times 10}$ by drawing the components of two matrices $\bar{U}, \bar{V} \in \mathbb{R}^{3\times 10}$ from a standard normal distribution, forming $\bar{U}^\top \bar{V}$, and adding uncorrelated Gaussian noise with variance 10^{-3}. We use $T = 30$ time steps and a coupling strength of $\lambda = 10$.

Hyperparameters. We train the model with standard GD (baseline) and with Goldstone-GD with $k_1 = 50$ and $k_2 = 10$. We find fastest convergence for the baseline method if we clip the gradients to an interval $[-\bar{g}, \bar{g}]$ and use a decreasing learning rate $\rho_s = \rho_0(\bar{s}/(s+\bar{s}))^{0.7}$ despite the noise-free gradient. Here, s is the training step. We optimize the hyperparameters for fastest convergence in the baseline and find $\bar{g} = 0.01$, $\rho_0 = 1$, and $\bar{s} = 100$.

Results. Figure 2 compares convergence in the two algorithms. We discussed the figure at the end of Section 2.1. In summary, Goldstone-GD converges an order of magnitude faster even in this small scale setup in which the skew symmetric gauge fields Γ_t have only $d(d-1)/2 = 3$ independent parameters, i.e., there are only three types of Goldstone modes. Once the minimization finds minima of the local losses ℓ, differences in the total loss $\mathcal{L}$ between the two algorithms are small since Goldstone modes contribute only little to $\mathcal{L}$ (this is why they decay slowly in GD).

5.2. MovieLens Recommendations

Model and data set. We fit the sparse dynamic Bernoulli factorization model defined in Eqs. 6-8 in Section 2.3 to the Movielens 20M data set[1] (Harper & Konstan, 2016). We use embedding dimension $d = 30$, coupling strength $\lambda = 10$, and regularizer $\gamma = 1$. The data set consists of 20 million reviews of 27,000 movies by 138,000 users with time stamps from 1995 to 2015. We binarize the ratings by splitting at the median, discarding ratings at the median, and we slice the remaining 18 million data points into $T = 100$ time bins of equal duration. We split randomly across all bins into 50% training, 20% validation, and 30% test set.

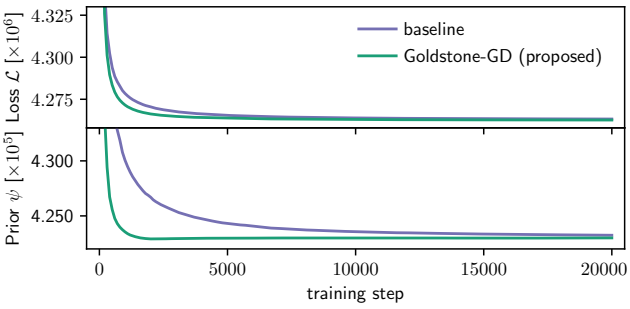

Figure 3. Training curves for MovieLens recommendations using sparse dynamic matrix factorization with Bernoulli likelihood.

Baseline and hyperparameters. We compare the proposed Goldstone-GD algorithm to GD with AdaGrad (Duchi et al., 2011) with a learning rate prefactor of 1 obtained from cross-validation. Similar to Goldstone-GD, AdaGrad is designed to escape shallow valleys of the loss, but it uses only diagonal preconditioning. We compare to Goldstone-GD with $k_1 = 100$ and $k_2 = 10$, using the same AdaGrad optimizer for update steps in the full parameter space.

Results. The additional operations in Goldstone-GD lead to a 1% increase in runtime per full update step. The upper panel in Figure 3 shows training curves for the loss $\mathcal{L}$ using the baseline (purple) and Goldstone-GD (green). The loss drops faster in Goldstone-GD, but differences in terms of the full loss $\mathcal{L}$ are small because the local loss functions ℓ are much larger than the prior ψ in this experiment. We show only the prior ψ in the lower panel of Figure 3. Both algorithms converge to the same value of ψ, but Goldstone-GD converges at least an order of magnitude faster. The difference in absolute values is small because Goldstone modes contribute little to ψ. They can, however, have a large influence on the parameter values, as we show next in experiments with Dynamic Word Embeddings.

5.3. Dynamic Word Embeddings

Model and data set. We perform variational inference (Ranganath et al., 2014) in Dynamic Word Embeddings

[1] https://grouplens.org/datasets/movielens/20m/

Table 2. Word aging: We translate modern words to the year 1800 using the shared representation space of Dynamic Word Embeddings.

QUERY	GOLDSTONE-GD	BASELINE
car	boat, saddle, canoe, wagon, box	shell, roof, ceiling, choir, central
computer	perspective, telescope, needle, mathematical, camera	organism, disturbing, sexual, rendering, bad
electricity	vapor, virus, friction, fluid, molecular	exercising, inherent, seeks, takes, protect
DNA	potassium, chemical, sodium, molecules, displacement	operates, differs, sharing, takes, keeps
tuberculosis	chronic, paralysis, irritation, disease, vomiting	trained, uniformly, extinguished, emerged, widely

(DWE), see Section 2.3. We fit the model to digitized books from the years 1800 to 2008 in the Google Books corpus[2] (Michel et al., 2011) (approximately 10^{10} words). We follow (Bamler & Mandt, 2017) for data preparation, resulting in a vocabulary size of 10,000, a training set of $T = 188$ time step, and a test set of 21 time steps. The paper proposes two inference algorithms: filtering and smoothing. We use the smoothing algorithm, which has better predictive performance than filtering but suffers from Goldstone modes. We set the embedding dimension to $d = 100$ due to hardware constraints and train for 10,000 steps using an Adam optimizer (Kingma & Ba, 2014) with a decaying prefactor of the adaptive learning rate, $\rho_s = \rho_0(\bar{s}/(s+\bar{s}))^{0.7}$, where s is the training step, $\rho_0 = 0.1$, and $\bar{s} = 1000$. We find that this leads to better convergence than a constant prefactor. All other hyperparameters are the same as in (Bamler & Mandt, 2017). We compare the baseline to Goldstone-GD using the same learning rate schedule and hyperparameters $k_1 = k_2 = 10$, which leads to an 8% increase in runtime.

Results. By eliminating Goldstone modes, Goldstone-GD makes word embeddings comparable across the time dimension of the model. We demonstrate this in Table 2, which shows the result of 'aging' modern words, i.e., translating them from modern English to the English of 1800. For each query word i, we report the five words i' whose embedding vectors $u_{i',1}$ at the first time step (year 1800) have largest overlap with the embedding vector $u_{i,T}$ of the query word at the last time step (year 2008). Overlap is measured in cosine distance (normalized scalar product) and we use the means of $u_{i,T}$ and $u_{i',1}$ under the variational distribution.

Goldstone-GD finds words that are plausible for the year 1800 while still being related to the query (e.g., means of transportation in a query for 'car'). By contrast, the baseline method fails to find plausible results. Figure 4 provides more insight into the failure of the baseline method. It shows histograms of the cosine distance between word embeddings $u_{i,1}$ and $u_{i,T}$ for the same word i from the first to the last time step. In Goldstone-GD (green), most embeddings have a large overlap, reflecting that the usage of most words does not change drastically over time. In the baseline (purple),

[2] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

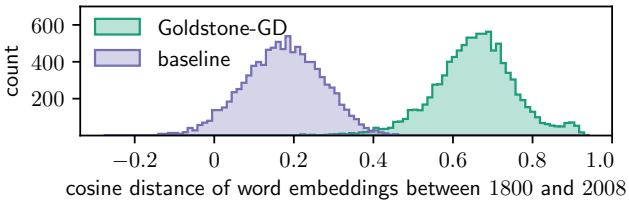

Figure 4. Cosine distance between word embeddings from the first and last year of the training data in Dynamic Word Embeddings.

no embeddings overlap by more then 60% between 1800 and 2008, and some embeddings even change their orientation (negative overlap). We explain this counterintuitive observation with the presence of Goldstone modes, i.e., the entire embedding spaces are rotated against each other.

For a quantitative comparison, we evaluate the predictive log-likelihood of the test set under the posterior mean, and find slightly better predictive performance with Goldstone-GD (-0.5317 vs. -0.5323 per test point). The improvement is small because the training set is so large that the influence of the prior in all but the symmetry directions is dwarfed by the likelihood. The main advantage of Goldstone-GD are the more interpretable embeddings, as demonstrated in Table 2.

6. Conclusions

We identified a slow convergence problem in representation learning models with a continuous symmetry and a Markovian time series prior, and we solved the problem with a new optimization algorithm, Goldstone-GD. The algorithm separates the minimization in the symmetry subspace from the remaining coordinate directions. Our experiments showed that Goldstone-GD converges orders of magnitude faster and fits more interpretable embedding vectors, which can be compared across the time dimension of a model. We believe that continuous symmetries are common in representation learning and can guide model and algorithm design.

Acknowledgements

We thank Ari Pakman for valuable and detailed feedback that greatly improved the manuscript.

References

Altland, Alexander and Simons, Ben D. *Condensed matter field theory*. Cambridge University Press, 2010.

Amari, Shun-Ichi. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Arnol'd, Vladimir Igorevich. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

Badrinarayanan, Vijay, Mishra, Bamdev, and Cipolla, Roberto. Understanding symmetries in deep networks. *arXiv preprint arXiv:1511.01029*, 2015.

Bamler, Robert and Mandt, Stephan. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 380–389, 2017.

Barkan, Oren. Bayesian Neural Word Embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Bui, Hung Hai, Huynh, Tuyen N, and Riedel, Sebastian. Automorphism groups of graphical models and lifted variational inference. *arXiv preprint arXiv:1207.4814*, 2012.

Charlin, Laurent, Ranganath, Rajesh, McInerney, James, and Blei, David M. Dynamic Poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pp. 155–162, 2015.

de Abreu, Nair Maria Maia. Old and new results on algebraic connectivity of graphs. *Linear Algebra and its Applications*, 423(1):53–73, 2007.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Gens, Robert and Domingos, Pedro M. Deep symmetry networks. In *Advances in Neural Information Processing Systems 27*, pp. 2537–2545. 2014.

Gopalan, Prem, Hofman, Jake M, and Blei, David M. Scalable recommendation with hierarchical Poisson factorization. In *UAI*, pp. 326–335, 2015.

Hall, Brian. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015.

Harper, F Maxwell and Konstan, Joseph A. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

Hoffman, Matthew D, Blei, David M, Wang, Chong, and Paisley, John William. Stochastic Variational Inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Kingma, Diederik and Ba, Jimmy. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, 2014.

Koren, Yehuda. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.

LeCun, Yann, Bottou, Léon, Bengio, Yoshua, and Haffner, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lu, Zhengdong, Agarwal, Deepak, and Dhillon, Inderjit S. A spatio–temporal approach to collaborative filtering. In *ACM Conference on Recommender Systems (RecSys)*, 2009.

Martens, James. New insights and perspectives on the natural gradient method. *arXiv preprint arXiv:1412.1193*, 2014.

Michel, Jean-Baptiste, Shen, Yuan Kui, Aiden, Aviva Presser, Veres, Adrian, Gray, Matthew K, Pickett, Joseph P, Hoiberg, Dale, Clancy, Dan, Norvig, Peter, Orwant, Jon, et al. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 331 (6014):176–182, 2011.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013a.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. 2013b.

Mnih, Andriy and Salakhutdinov, Ruslan R. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pp. 1257–1264, 2008.

Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

Noessner, Jan, Niepert, Mathias, and Stuckenschmidt, Heiner. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *AAAI Workshop: Statistical Relational Artificial Intelligence*, 2013.

Ollivier, Yann. Riemannian metrics for neural networks I: feedforward networks. *Information and Inference: A Journal of the IMA*, 4(2):108–153, 2015a.

Ollivier, Yann. Riemannian metrics for neural networks II: recurrent networks and learning symbolic data sequences. *Information and Inference: A Journal of the IMA*, 4(2): 154–193, 2015b.

Pascanu, Razvan and Bengio, Yoshua. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

Pennington, Jeffrey, Socher, Richard, and Manning, Christopher. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Peskin, Michael Edward. *An introduction to quantum field theory*. Westview press, 1995.

Poignard, Camille, Pereira, Tiago, and Pade, Jan Philipp. Spectra of laplacian matrices of weighted graphs: structural genericity properties. *SIAM Journal on Applied Mathematics*, 78(1):372–394, 2018.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David. Black box variational inference. In *Artificial Intelligence and Statistics*, pp. 814–822, 2014.

Rudolph, Maja and Blei, David. Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052*, 2017.

Sun, John Z, Varshney, Kush R, and Subbian, Karthik. Dynamic matrix factorization: A state space approach. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1897–1900, 2012.

Zaheer, Manzil, Kottur, Satwik, Ravanbakhsh, Siamak, Poczos, Barnabas, Salakhutdinov, Ruslan R, and Smola, Alexander J. Deep sets. In *Advances in Neural Information Processing Systems*, pp. 3394–3404, 2017.

Zeiler, Matthew D. ADADELTA: an Adaptive Learning Rate Method. *arXiv preprint arXiv:1212.5701*, 2012.