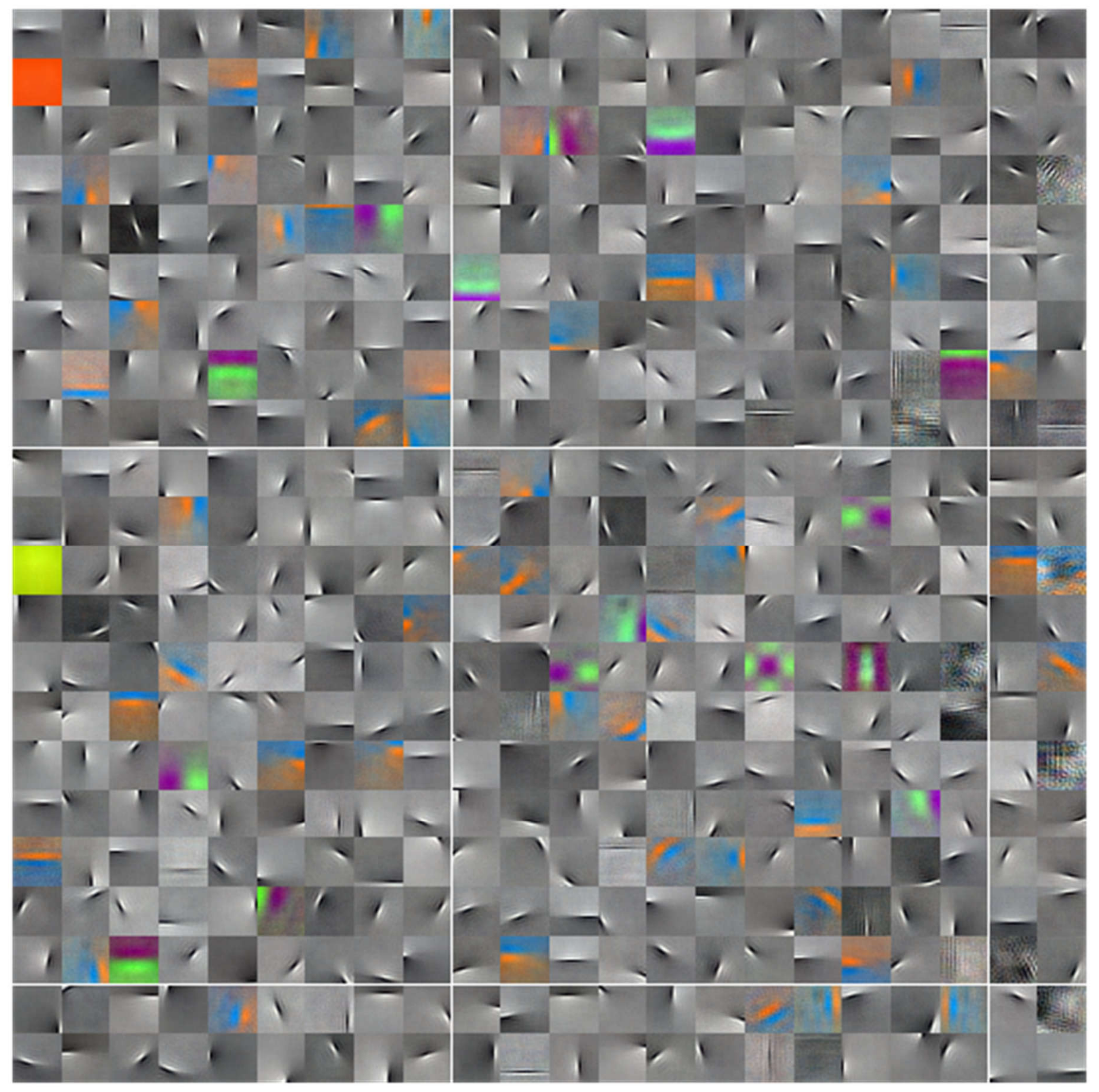
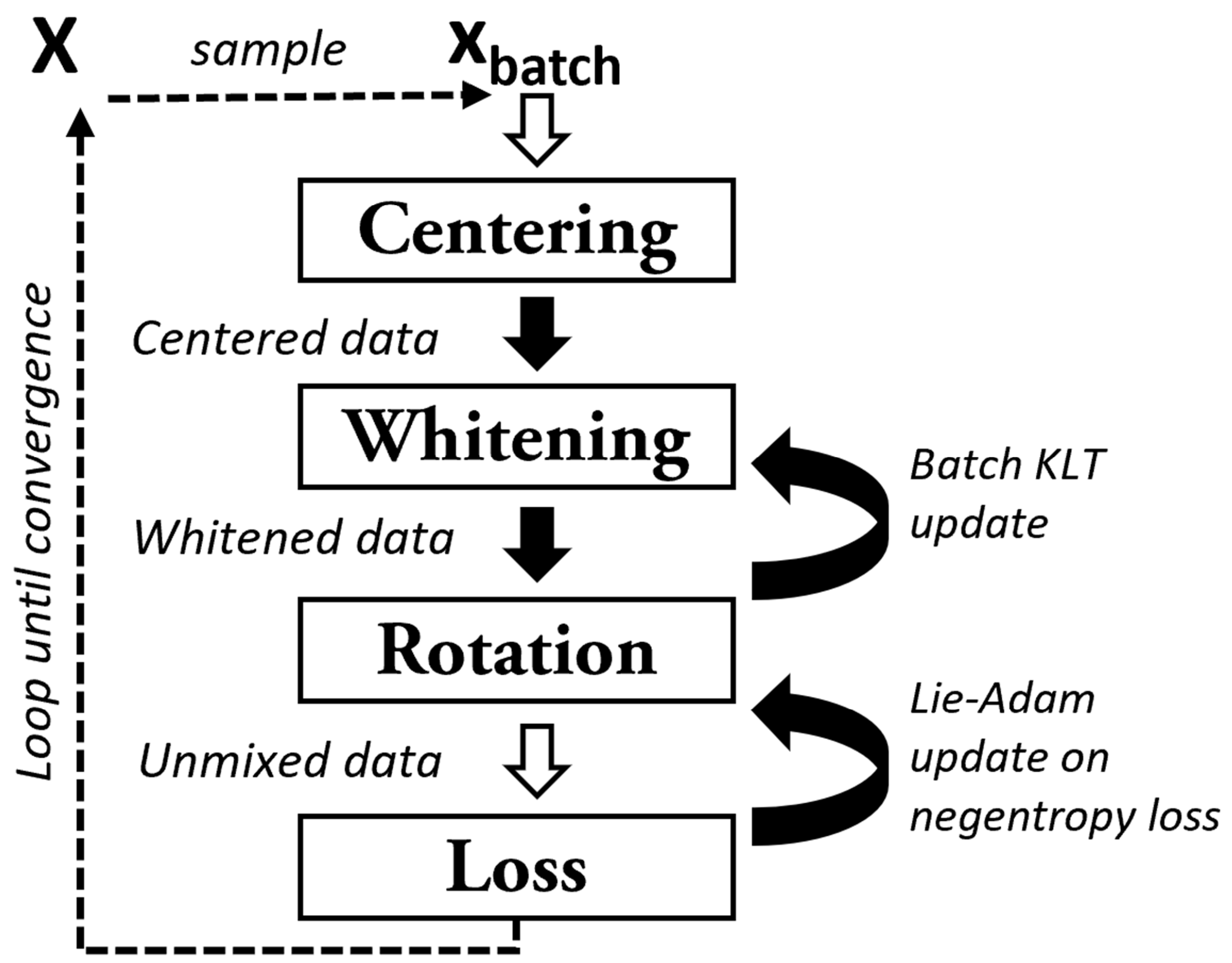
LARGE-SCALE INDEPENDENT COMPONENT ANALYSIS BY SPEEDING UP LIE GROUP TECHNIQUES

Matthias Hermann, Georg Umlauf, Matthias O. Franz

IEEE ICASSP 2022

We were interested in computing a mini-batch-capable end-to end algorithm to identify statistically independent components (ICA) in large scale and high-dimensional datasets. Current algorithms typically rely on pre-whitened data and do not integrate the two procedures of whitening and ICA estimation. Our online approach estimates a whitening and a rotation matrix with stochastic gradient descent on centered or uncentered data. We show that this can be done efficiently by combining Batch Karhunen-Löwe-Transformation with Lie group techniques. Our algorithm is recursion-free and can be organized as feed-forward neural network which makes the use of GPU acceleration straight-forward. Because of the very fast convergence of Batch KLT, the gradient descent in the Lie group of orthogonal matrices stabilizes quickly. The optimization is further enhanced by integrating ADAM, an improved stochastic gradient descent (SGD) technique from the field of deep learning. We test the scaling capabilities by computing the independent components of the well-known ImageNet challenge. Due to its robustness with respect to batch and step size, our approach can be used as a drop-in replacement for standard ICA algorithms where memory is a limiting factor.





X

sample

X_{batch}

Centering

Centered data

Whitening

Whitened data

Rotation

Unmixed data

Loss

Loop until convergence

Batch KLT update

Lie-Adam update on negentropy loss

Examples of the first 484 independent components estimated from the ImageNet dataset (1:2 · 10^6 examples) (left). Every tile represents a single column of the mixing matrix which is reshaped to 3 × 200 × 200 for illustration purpose. In Lie-ADAM, we used a learning rate of 0.01 and a batch size of 484. The model was trained with three runs through the dataset which took 3h on standard hardware with a single GPU. A schematic overview of the algorithm is shown on the right.

Objective is finding unmixing matrix $\mathbf{A^{-1}}$, with independent components $\mathbf{S} = \mathbf{A^{-1}X}$

$$\mathbf{A^{-1}} = \mathbf{RW_{white}}$$

(Trivialization, whitening matrix and rotation matrix)

(1) Mutual information approximation using negentropy with $G(\cdot) = \log\cosh(\cdot)$

$$I(s_{1:k}) \propto -\sum J(s_i) \propto \sum (E[G(s_i)] - E[G(\boldsymbol{z})])^{\mathbf{2}}$$

(2) Computing the gradient of negentropy w.r.t the Lie algebra:

$$\nabla_{\mathbf{\Theta}} I = (\nabla_{\mathbf{R}} I)^T \mathbf{R} - \mathbf{R}^T (\nabla_{\mathbf{R}} I)$$

(3) Geodesic flow [4] using Plücker parametrization $\mathbf{r}$ and ADAM-optimization [2] :

$$\mathbf{R}_{i+1}^T = \exp(-\eta \mathbf{\Theta}_{\mathbf{r}_i}) \mathbf{R}_i^T$$

Key benefits of the algorithm

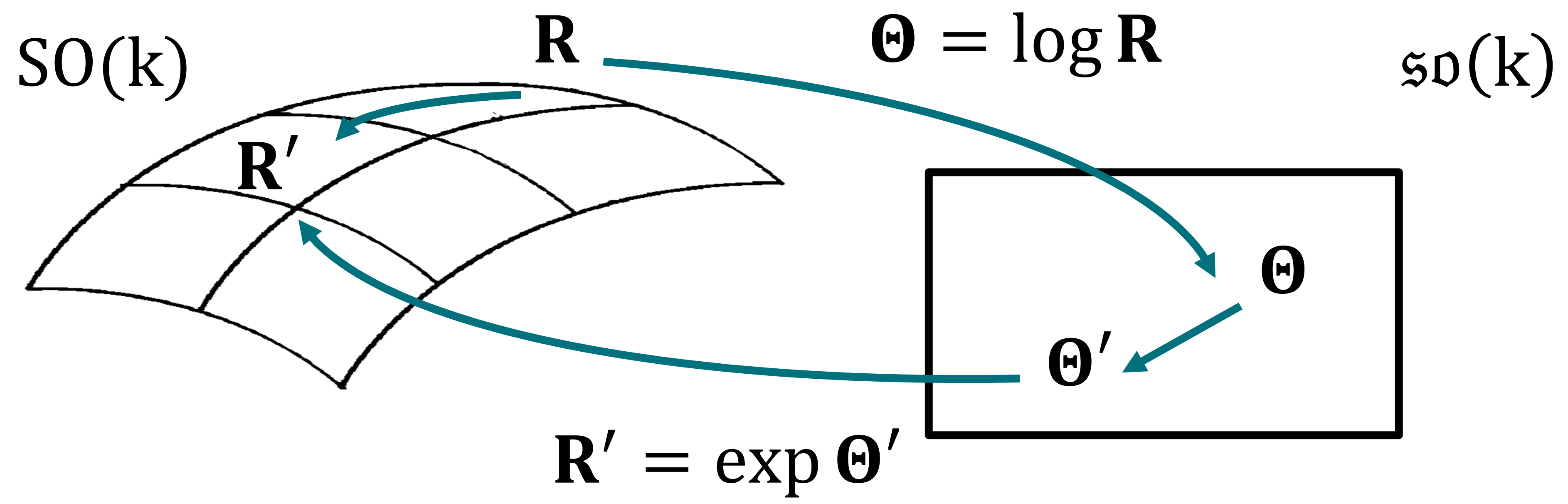
The proposed Lie-Adam approach offers two computation modes:

a) In offline mode, it works very similar to L-BFGS-based algorithms.

a) in stochastic mode, it offers fast convergence rates while maintaining highly accurate solutions.

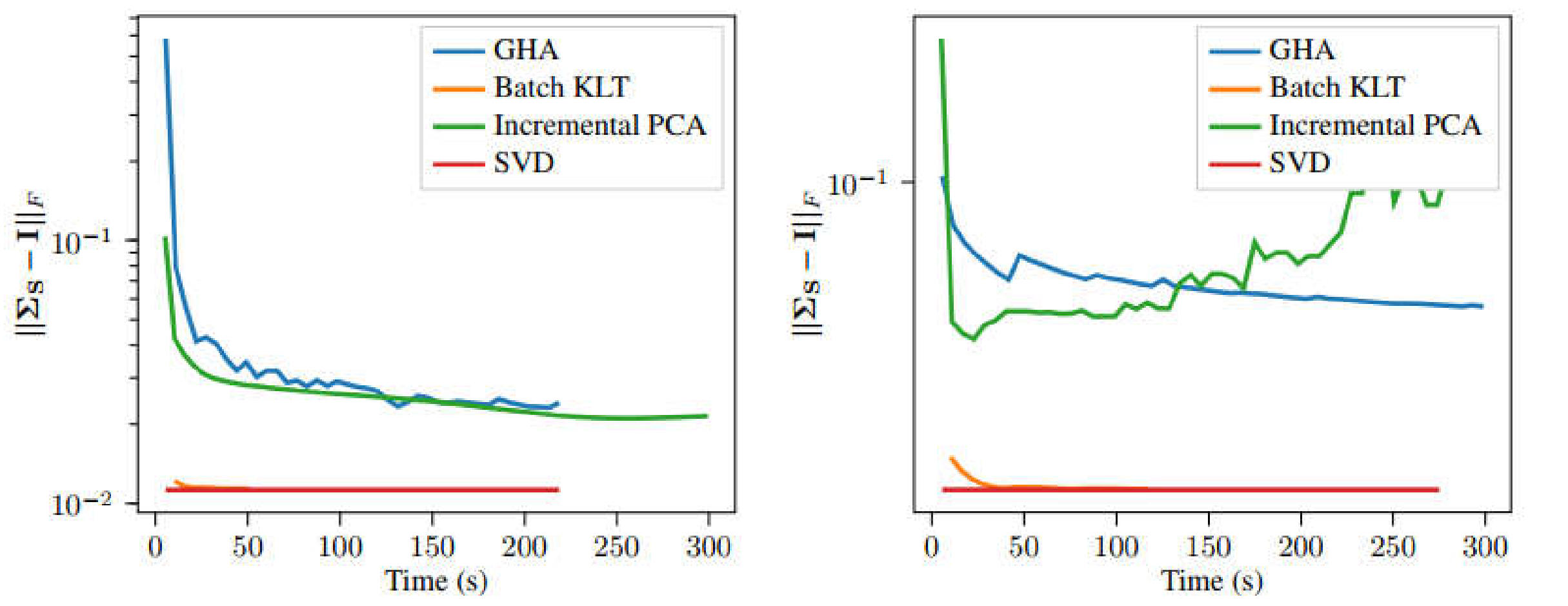
Our approach allows us to formulate the ICA update equations as a standard neural network and by using b-sized mini-batches the space complexity of the entire pipeline for d-dimensional inputs and k components is limited to O(d(k+b)).

Optimizing independence using Lie Group techniques



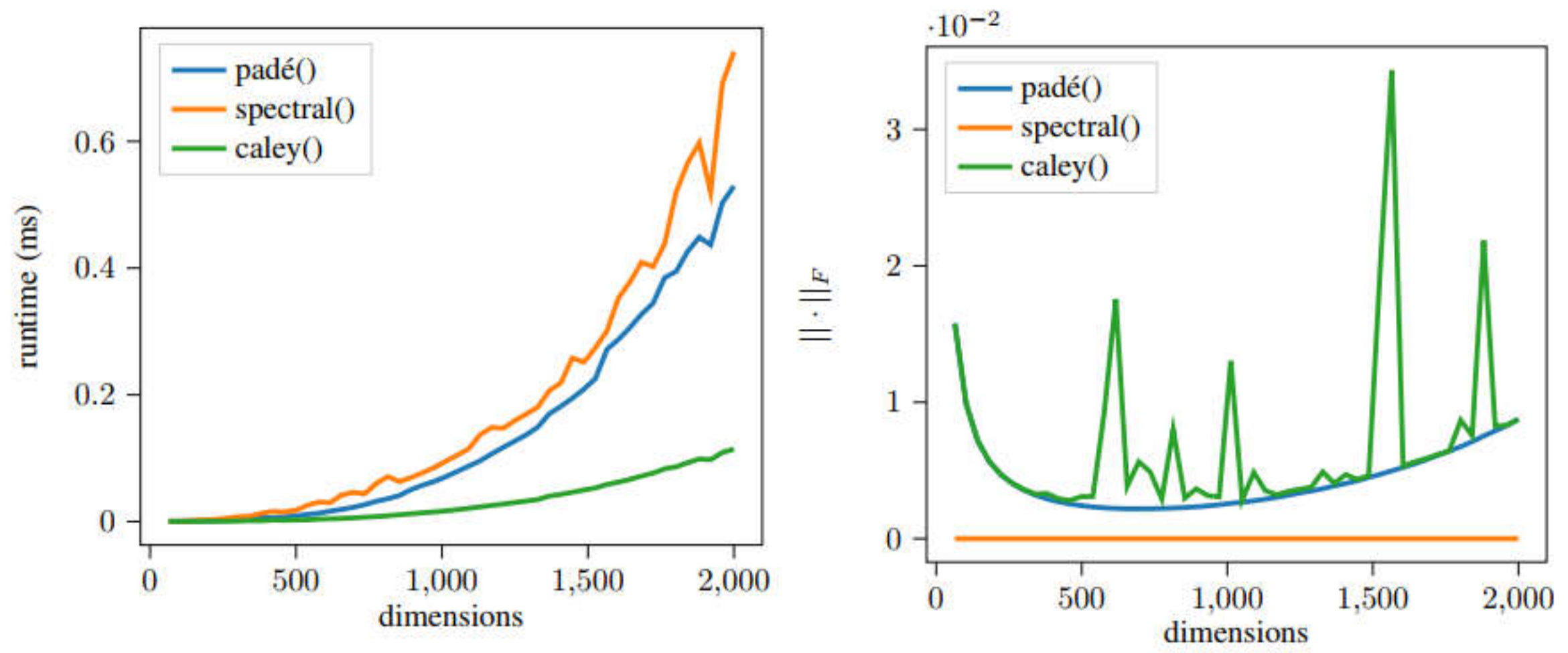
Every skew-symmetric matrix Θ can by related to an orthogonal matrix R computing the matrix exponential. SO(k) is a Lie group with a tangent space, called the Lie algebra 𝔰𝔬(k) that can be used for gradient updates. The so-called gradient flow makes use of this and computes ICA with orthogonality constraints using Lie group techniques.

High input dimensionality



Convergence of the 100 × 100 (left) and 1000 × 1000 (right) covariance matrix ΣS pertaining to the largest eigenvalues to the identity matrix as measured by the Frobenius norm on the CIFAR10 dataset (50.000 examples of size 3 × 32 × 32). As baseline, we show Singular Value Decomposition (SVD) which runs offline. Batch size was set to 100 and 1000, respectively.

High output dimensionality

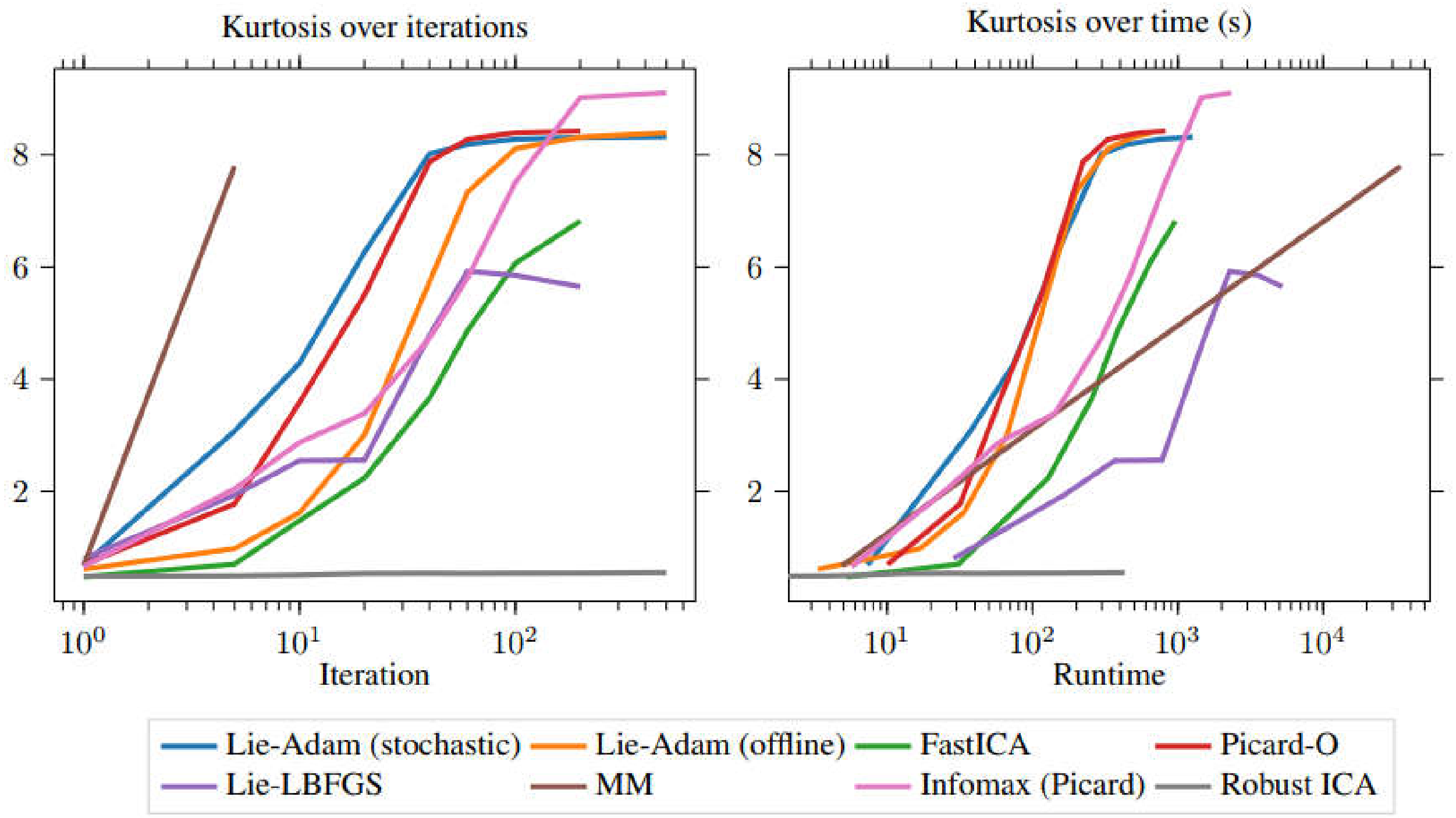


Runtime and precision of the matrix exponential methods comparison between spectral exp(M), cayley(M), padé(M) for M ~ N(0; 0:1) using PyTorch. Interestingly both the Cayley approximation

$$\mathrm{caley}(\mathbf{M}) = \left(\mathbf{I} - \frac{\mathbf{M}}{2}\right)^{-1} (\mathbf{I} - \frac{\mathbf{M}}{s})$$

and the Padé algorithm give similar accuracies (10^{-4}) up to 500 dimensions. However, for larger dimensions the Caley approximation shows large peaks.

Evaluation



The first plot shows the evolution of kurtosis of the computed ICs measured over iterations, the second plot over runtime for the STL10 dataset (10^5 examples of size 3 × 96 × 96). The learning rates are 0.01 in the offline scenario, and 0.001 for the stochastic scenario. The batch size and k is 484.

References

1. Levy and M. Lindenbaum, "Sequential karhunenloeve basis extraction and its application to images," in *Proceedings 1998 International Conference on Image Processing. ICIP98*, IEEE, vol. 2, 1998, pp. 456–460.
2. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014
3. J Hérault and B Ans, "Circuits neuronaux á synapses modifiables: Décodage de messages composites par apprentissage non supervisé [neuronal circuits with modifiable synapses: Decoding composite messages by unsupervised learning]," 1984.
4. Y. Nishimori, "Learning algorithm for independent component analysis by geodesic flows on orthogonal group," in *IJCNN'99. International Joint Conference on Neural Networks.* IEEE, vol. 2, 1999, pp. 933–938.
5. Plumbley, M. D. (2005). Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. Neurocomputing, 67, 161-197.