Reduce and Conquer: Independent Component Analysis at linear sample complexity

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2025

Abstract

Feature learning at the scale of deep neural networks remains poorly understood due to the complexity of deep network dynamics. Independent component analysis (ICA) provides a simple unsupervised model for feature learning, as it learns filters that are similar to deep networks. ICA extracts these features from the higher-order correlations of the inputs, which is a computationally hard task in high dimensions with a sample complexity of at least $n \gtrsim D^2$ for D-dimensional inputs. In practice, this difficulty is overcome by running ICA in the d-dimensional subspace spanned by the leading principal components of the inputs, which is often taken to be $d = D/4$. However, there exist no theoretical guarantees for this procedure. Here, we first conduct systematic experiments on ImageNet to demonstrate that running FastICA in a *finite* subspace of $d \sim O_D(1)$ dimensions yields non-Gaussian directions in the D-dimensional image space. We then introduce a "subspace model" for synthetic data, and prove that FastICA does indeed recover the most non-Gaussian direction in a sample complexity that is linear in the input dimension. We finally show experimentally that deep convolutional networks trained on ImageNet exhibit behaviour consistent with FastICA: during training, they converge to the principal subspace of image patches before or when they find non-Gaussian directions. By providing quantitative, rigorous insights into the working of FastICA, our study thus unveils a plausible feature-learning mechanism in deep convolutional neural networks.

1. Introduction

Independent component analysis (ICA) is an unsupervised learning algorithm that separates mixed signals, like images or sound, into statistically independent components [5, 7, 18]. When applied to natural images, ICA yields localised, edge-like filters [4, 19, 25], also known as Gabor filters [12]. These filters resemble the early-layer features learnt by deep convolutional neural networks trained on image classification [14, 22, 24]. This similarity makes ICA a simple, yet insightful model for understanding feature learning in deep neural networks.

One of the puzzles of feature learning from images is that Gabor filters are learnt from the higher-order correlations between pixels, which is a computationally hard task. In the case of ICA, which seeks to identify projections of the inputs $s := w \cdot x$ that have a maximally non-Gaussian distribution $p(s)$, this hardness can be quantified: On the standard ICA data model, the fundamental limit for weakly recovering a non-Gaussian direction in D-dimensional inputs is $n \gtrsim D^2$ [1, 33] as $D \to \infty$. The situation is even worse if we consider the most popular algorithm, FastICA [18], which has a sample complexity of $n \gtrsim D^4$ [30]. While Ricci et al. [30] recently showed performing ICA by minimising a "smoothed" ([8]) loss using SGD reduces the sample complexity down to the

optimal $n \gtrsim D^2$, this is still expensive on large-dimensional inputs like images. It is also a poor model of the learning dynamics of neural networks, which are able to extract information from the non-Gaussian part of their inputs early during training, i.e. after having seen not many samples [28].

In practice, the computational difficulty of FastICA is overcome by running the algorithm in the subspace spanned by the leading principal components of the data, with a common prescription suggesting to use a subspace of dimension $d = D/4$ [19]. While this convention by itself does not alleviate the poor sample complexity, improving only the constants, it raises a couple of intriguing question: 1. Is it possible to reliably find non-Gaussian directions in high dimensions ($D \to \infty$) by running FastICA in a subspace of fixed dimension $d = O_D(1)$? If so, can we prove that it does, and what does it tell us about the structure of images? And finally, do deep neural networks apply a similar strategy to learn Gabor filters?

Main contributions Here, we answer all three questions in the affirmative. In particular, we ...

- ...demonstrate experimentally that running FastICA in a subspace of fixed dimension $d = O_D(1)$ yields non-Gaussian directions in the D-dimensional image space on ImageNet (Section 3);
- ...introduce a new "subspace" model for data, for which we prove that strong recovery of a non-Gaussian direction is possible at linear sample complexity when running ICA in a subspace (Section 4);
- ...show experimentally that different deep convolutional neural networks converge to the principal subspace of image patches before or when they find non-Gaussian directions, mirroring the ICA approach to feature learning (Section 5).

2. Background: ICA and the FastICA algorithm

Given a set of n inputs $\mathcal{D} = \{x^\mu\}_{\mu=1}^n$ drawn i.i.d. from a data distribution $\mathbb{P}$ with zero mean and identity covariance, ICA searches for a unit vector $w \in \mathbb{S}^{d-1}$ that yields projections $s = w \cdot x$ of the inputs that are maximally non-Gaussian:

$$w^* := \arg\max_{\|w\|=1} \mathbb{E}_{\mathcal{D}}\, G(w \cdot x), \tag{1}$$

where $G : \mathbb{R} \to \mathbb{R}$ is a suitable "contrast function" which measures non-Gaussianity, for example $G(s) := -e^{-s^2/2}$. As a preprocessing step, it is very common to reduce the dimension of the inputs by projecting them onto a d-dimensional subspace spanned by the leading principal components. This is an efficient way of exploiting the information contained in the covariance matrix of the inputs, before that information is removed by the whitening. Usually, the dimension of the original ambient space D is cut down to $d = D/4$ [17, 19].

The most popular algorithm to do the optimisation in Eq. (1) is FastICA [18], which finds extremal points of the population loss $\mathcal{L}(w) := \mathbb{E}_{\mathbb{P}}\, G(w \cdot x)$ via a second-order fixed-point iteration. Given an initial weight vector drawn at random on the unit sphere, $w_0 \sim \text{Unif}(\mathbb{S}^{d-1})$, we iterate the **FastICA updates** for $t \geq 1$ until convergence:

$$\begin{cases} \widetilde{w}_t &= \mathbb{E}_{\mathcal{D}}[x\, G'(w_{t-1} \cdot x)] - \mathbb{E}_{\mathcal{D}}[G''(w_{t-1} \cdot x)] w_{t-1}, \\ w_t &= \widetilde{w}_t / \|\widetilde{w}_t\|. \end{cases} \tag{2}$$

See Hyvärinen and Oja [18] for a detailed derivation and discussion.

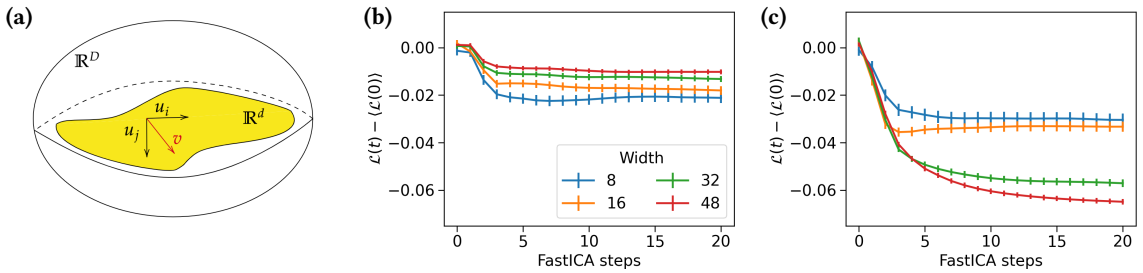

Figure 1: **Running ICA in the principal subspace of small dimension $d \sim O_D(1)$ yields non-Gaussian directions that generalise in large dimension D (a)** FastICA can be run in the ambient dimension D of the inputs or on the d-dimensional subspace spanned by the principal components u_i. **(b)** Change in loss with respect to initialisation when running FastICA on ImageNet patches at linear sample complexity $n = 2D$. As we increase the patch width, and hence D, the change in the loss tends towards zero. **(c)** Running FastICA in the subspace spanned by the leading $d = 32$ principal components yields a significantly non-Gaussian direction, as shown by the decay of the loss evaluated in D dimensions. Error bars indicate the error in the mean over runs. Full details in Appendix A.1.

3. Running FastICA in the principal subspace of ImageNet patches consistently yields non-Gaussian directions

We first verified experimentally whether running FastICA in a subspace of *fixed* input dimension $d \sim O_D(1)$ yields a non-Gaussian direction in the ambient D-dimensional space when $D \to \infty$, see Fig. 1(a). To this end, we ran FastICA on a set of $n = 2D$ patches that were randomly sampled from ImageNet [11]. In Fig. 1(b), we show change in the loss $\mathcal{L}(w)$, averaged over a held-out test set, compared to the value of the test loss at initialisation $\langle \mathcal{L}(w_0) \rangle$, where the average $\langle \cdot \rangle$ is over different random initial conditions. We can see that not only are the changes in the loss small; as we increase the width of the patches, and hence their dimension D, the curves tend towards zero, suggesting that FastICA does not recover a non-Gaussian direction at linear sample complexity on ImageNet in the asymptotic limit of large inputs $D \to \infty$, in agreement with Ricci et al. [30].

The picture changes fundamentally if we run FastICA in the subspace spanned by the leading $d = 32$ principal components, keeping the dimension of the subspace *fixed* as we increase the size of the inputs, see Fig. 1(c). We now find that the FastICA consistently finds a direction that, after projecting back up into the D-dimensional space, yields non-Gaussian projections $s = w \cdot x$.

This experiment therefore suggests that running FastICA in a subspace of *constant*, rather than just smaller dimensions, yields non-Gaussian directions in image space even at linear sample complexity, circumventing the fundamental computational limitations of FastICA.

4. FastICA provably recovers a non-Gaussian direction at linear sample complexity on the "subspace model"

We now seek to confirm the picture that emerged from our experiments in Section 3 by proving recovery of a non-Gaussian direction with linear sample complexity by FastICA. In the standard input model for (noisy) ICA, the computational hardness of the task in D dimensions requires $n \gtrsim D^2$ which cannot be reduced by projecting down to lower dimensions [1, 33]. The standard ICA data model therefore cannot account for the experimental results on ImageNet in Fig. 1.

Definition of the subspace model Here, we therefore introduce the "subspace model", a new data model that accounts for the success of PCA + ICA on ImageNet. The key idea is to spike the higher-order correlations of the inputs with a spike v that is a linear combination of the leading principal components of the inputs, which we denote by $u_r \in \mathbb{S}^{D-1}$ and with are orthonormal. We consider n inputs in D dimensions $x^\mu = (x_i^\mu)$, $\mu = 1, \dots, n$, that are sampled according to

$$x^\mu = \sum_{\rho=1}^{r} \sqrt{\beta_2^\rho} g_\rho^\mu u^\rho + h^\mu v + z^\mu, \qquad v = \sum_{\rho}^{r} \alpha_\rho u^\rho, \tag{3}$$

where the Gaussian latent variables $g_\rho^\mu \sim_{\text{i.i.d.}} \mathcal{N}(0,1)$, while the latent variable h^μ is drawn from a non-Gaussian distribution; for example, we could take $h^\mu = \pm 1$ to be Rademacher. All latent variables g_ρ^μ, h^μ are independent of each other. The noise z^μ is drawn from a multivariate normal distribution with mean 0 and covariance $\mathbb{1} - vv^\top$, which ensures that the non-Gaussian direction v can *not* be recovered using principal component analysis (PCA), just like Gabor filters cannot be found by PCA. The coefficients α_ρ are normalised as $\sum_\rho^r \alpha_\rho^2 = 1$. To prove that FastICA can recover the non-Gaussian direction, we proceed in two steps.

Step 1: projecting down We first show that when projecting inputs from the subspace model (3) onto the lower dimensional subspace spanned by the top-r leading principal components, we obtain a "reduced" input model that follows the ICA model. More precisely, we find that the reduced inputs $y^\mu = (y_i^\mu)$, $\mu = 1, \dots, n$, follow a noisy ICA model [30] whose "non-Gaussian" spike is simply, up to normalisation, the projection of the original spike v:

$$y^\mu = \sqrt{\gamma}\, h^\mu \widetilde{v} + \hat{z}^\mu \in \mathbb{R}^r, \tag{4}$$

where $\widetilde{v} = \text{proj}(Sv)/\|\text{proj}(Sv)\|$, for suitable γ and $\hat{z}^\mu$. This is proved in Theorem 2. In the proof, we assume that PCA yielded the exact principal subspace $\{u^r\}$. However, in practice, as known from standard results for PCA in high dimensions [2], PCA will return the subspace up to some small error. However, this only reduces the signal-to-noise ratio of the noisy ICA model. More precisely, we prove in Lemma 3 that, if n is such that $n, D \to \infty$ and n/D that diverges arbitrarily slowly, with high probability we still recover the reduced inputs (4), up to vanishing error terms.

Step 2: Gradient flow We now analyse the performance of ICA on inputs drawn from the reduced model (4), at a linear sample complexity. Theorem 4, informally stated below, proves that gradient descent (GD) on the ICA loss $\mathcal{L}(w)$ converges to the non-Gaussian spike of the reduced model.

Theorem 1 *(Informal) Consider the low dimensional inputs drawn from the reduced model* (4)*. Since the population loss for ICA $\mathcal{L}(w)$ exhibits monotonicity and the dynamics of GD can be approximated via gradient flow, with high probability GD converges to the non-Gaussian spike $\widetilde{v}$ in the limit $n \to \infty$.*

5. Deep convolutional neural networks recover the principal subspace of their inputs before or when they find non-Gaussian directions

Deep convolutional neural networks (CNNs) learn similar filters as ICA in their first convolutional layer. Motivated by our results on ICA, we investigated whether deep CNNs also identify the principal subspace first, and then find non-Gaussian Gabor filters. To this end, we trained three deep CNNs (AlexNet, Resnet18, and DenseNet121) on ImageNet using standard training recipes, see

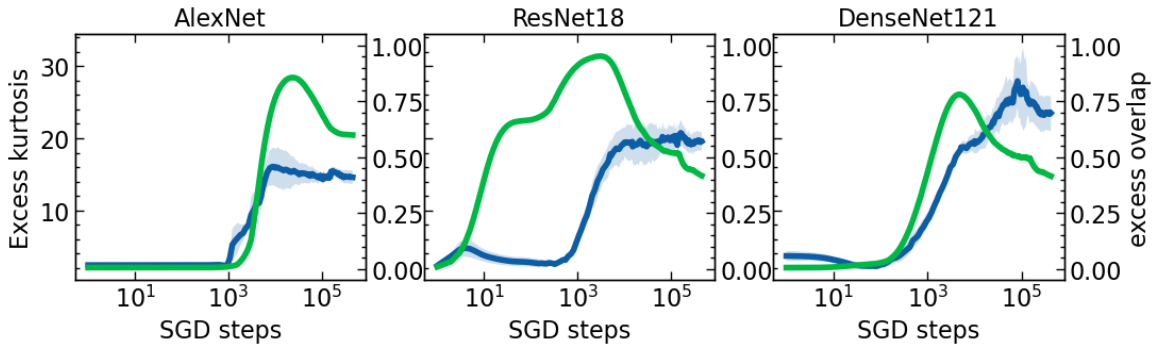

Figure 2: **Deep neural networks converge to the principal subspace of image patches before or when they find non-Gaussian directions.** For three deep convolutional neural networks trained on ImageNet, we plot the overlap with the principal subspace of image patches, (Eq. (A.1), blue), obtained via PCA, and the excess kurtosis of the first-layer representations, which measures their non-Gaussianity (Eq. (A.2), green). Details in Appendix A.2.

Appendix A.2 for more details. Throughout training, we monitored whether the CNNs found non-Gaussian directions, comparable to ICA, by tracking the excess kurtosis, Eq. (A.2), of projections $s_\mu^k = w^k \cdot x^\mu$ of ImageNet patches x^μ along the first-layer filters $\{w^k\}_{k=1,\dots,K}$ of the CNNs. We show the excess kurtosis, averaged over patches and neurons, in blue in Fig. 2. After an initial phase of low excess kurtosis close to initialisation, the non-Gaussianity of projections increases sharply between 10^3 and 10^4 steps in all three networks, before plateauing.

Interestingly, this increase in non-Gaussianity, which is when the receptive fields form [20, 30], is preceded by or coincides with the recovery of the principal subspace of the inputs by the CNNs. We quantify this overlap by measuring the excess overlap of the filters with the subspace spanned by the top-k principal components, which measures how much variability in the weights is captured by the subspace compared to a random basis. An excess overlap of 0 indicates that weights are not concentrated in the principal subspace beyond what one would expect by chance, where as an excess overlap of 1 indicates that the weights are in the span of the subspace. In all three networks, we found that the deep CNNs recover the principal subspace before or precisely when they discover the non-Gaussian filters, which is consistent with the picture that emerged from our analysis of ICA.

6. Concluding perspectives

We have shown that ICA can provably recover a non-Gaussian direction v in linear sample complexity if 1. v lives in the span of the principal components of the data, and 2. we perform PCA as a preprocessing step. Intriguingly, we found that deep CNNs likewise recover the principal subspace of their inputs before or precisely when they learn the non-Gaussian first-layer filters.

Our experimental observation reflects previous work that showed that neural networks learn distributions of increasing complexity, focusing on pair-wise correlations first before going to higher-order correlations, both on images [3, 20, 21, 28], in natural language processing [6, 29] and on a mixture classification task [3].

Our results also echo earlier work showing that gradient descent occurs in a tiny subspace of input space [13, 23] and that the Hessian of over-parametrised neural networks has a low-rank structure [26, 31, 32]. How these two perspectives – on the geometry of the loss landscape and on the data structure – relate to each other is an intriguing avenue for further work.

References

[1] Arnab Auddy and Ming Yuan. Large dimensional independent component analysis: Statistical optimality and computational tractability. *to appear in Annals of Statistics*, 2024.

[2] Afonso Bandeira, Amit Singer, and Thomas Strohmer. Mathematics of data science. *Book draft available at https://people. math. ethz. ch/abandeira/BandeiraSingerStrohmer-MDS-draft. pdf*, 4, 2020.

[3] Lorenzo Bardone and Sebastian Goldt. Sliding down the stairs: How correlated latent variables accelerate learning with neural networks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 3024–3045. PMLR, 2024.

[4] Anthony Bell and Terrence J Sejnowski. Edges are the 'independent components'of natural scenes. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996.

[5] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[6] Nora Belrose, Quintin Pope, Lucia Quirke, Alex Mallen, and Xiaoli Fern. Neural networks learn statistics of increasing complexity. *arXiv:2402.04362*, 2024.

[7] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3): 287–314, 1994.

[8] Alex Damian, Eshaan Nichani, Rong Ge, and Jason D. Lee. Smoothing the landscape boosts the signal for SGD: Optimal sample complexity for learning single index models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=73XPopmbXH.

[9] Alex Damian, Loucas Pillaud-Vivien, Jason D Lee, and Joan Bruna. The computational complexity of learning gaussian single-index models. *arXiv preprint arXiv:2403.05529*, 2024.

[10] Yatin Dandi, Emanuele Troiani, Luca Arnaboldi, Luca Pesce, Lenka Zdeborová, and Florent Krzakala. The benefits of reusing batches for gradient descent in two-layer networks: Breaking the curse of information and leap exponents. *arXiv preprint arXiv:2402.03220*, 2024.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, 93(26): 429–441, 1946.

[13] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

[14] Florentin Guth and Brice Ménard. On the universality of neural encodings in cnns. In *ICLR 2024 Workshop on Representational Alignment*, 2024.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

[16] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.243.

[17] Aapo Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999.

[18] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[19] Aapo Hyvärinen, Jarmo Hurri, and Patrick O Hoyer. *Natural image statistics: A probabilistic approach to early computational vision.*, volume 39. Springer Science & Business Media, 2009.

[20] Alessandro Ingrosso and Sebastian Goldt. Data-driven emergence of convolutional structure in neural networks. *Proceedings of the National Academy of Sciences*, 119(40):e2201854119, 2022.

[21] Kevin Kögler, Aleksandr Shevchenko, Hamed Hassani, and Marco Mondelli. Compression of structured data with autoencoders: Provable benefit of nonlinearities and depth. In *International Conference on Machine Learning*, pages 24964–25015. PMLR, 2024.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[23] Brett W Larsen, Stanislav Fort, Nic Becker, and Surya Ganguli. How many degrees of freedom do we need to train deep networks: a loss landscape perspective. *arXiv preprint arXiv:2107.05802*, 2021.

[24] Jack Lindsey, Samuel A. Ocko, Surya Ganguli, and Stephane Deny. The effects of neural resource constraints on early visual representations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1xq3oR5tQ.

[25] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[26] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020.

[27] Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, and Paola Gervasio. *Matematica Numerica*, volume 77 of *UNITEXT*. Springer Milan, 2014. doi: 10.1007/978-88-470-5644-2. URL http://link.springer.com/10.1007/978-88-470-5644-2.

[28] Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. Neural networks trained with sgd learn distributions of increasing complexity. In *International Conference on Machine Learning*, pages 28843–28863. PMLR, 2023.

[29] Riccardo Rende, Federica Gerace, Alessandro Laio, and Sebastian Goldt. A distributional simplicity bias in the learning dynamics of transformers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GgV6UczIWM.

[30] Fabiola Ricci, Lorenzo Bardone, and Sebastian Goldt. Feature learning from non-gaussian inputs: the case of independent component analysis in high dimensions. *arXiv preprint arXiv:2503.23896*, 2025.

[31] Levent Sagun, Leon Bottou, and Yann LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2016.

[32] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017.

[33] Eszter Szekely, Lorenzo Bardone, Federica Gerace, and Sebastian Goldt. Learning from higher-order correlations, efficiently: hypothesis tests, random features, and neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=uHml6eyoVF.

Appendix A. Experimental details

A.1. Figure 1

(a) We run FastICA on a set of $n = 2D$ patches randomly sampled from ImageNet of width w, with $D = w^2$. We use the logcosh activation function, and use the full dataset to compute the weight update at each step. We preprocess image patches by centering them and whitening them. We compute the ICA loss averaged over a held-out test set of 10 000 patches, and subtract the loss computed for a random direction over the same test set, averaged over 20 random directions.

(b) Same as in (a), except that after whitening, we only keep the leading $d = 32$ principal components of each image and run FastICA in this subspace. We plot the loss in the original space of images of dimension d by projecting the ICA weight vector back into the d-dimensional space using the eigenvectors of the patch-patch covariance matrix.

Note that while Ricci et al. [30] consider an idealised setting where a fresh set of data is used to compute the gradient, in these experiments we reused the same batch of inputs at every iteration of FastICA. It is known that this actually simplifies the task of recovering single directions in certain cases [9, 10]; however, in the case of FastICA the task remains hard: the FastICA loss with standard activation functions is only sensitive to even cumulants, and due to whitening, the first non-trivial cumulant is the fourth, yielding a computational lower bound for recovering a non-Gaussian direction of $n \asymp D^2$ [33].

A.2. Figure 2

Training We trained three deep convolutional neural networks, AlexNet [22], DenseNet121 [16] and Resnet18 [15] on the ImageNet 1000 class classification task [11]. We used standard training recipes for all three networks, training with SGD for 90 epochs. We employ a cosine learning rate schedule with initial learning rate of 0.1 for Resnet18 and DenseNet, and 0.01 for AlexNet. We further chose momentum of 0.9, weight decay of 10^{-4}, and mini-batch size of 256. During training, we randomly cropped images to 224×224 pixels, and randomly flipped them horizontally. For all three architectures, we made three runs with different random number seeds, and took 50 logarithmically spaced snapshots of the weights during each training run.

Subspace overlap We first verified whether the first-layer convolutional filters at the end of training were predominantly located in a subspace spanned by the leading principal components. To this end, we first computed the principal components of patches sampled at random from ImageNet, which look roughly like Fourier components in two dimensions see the left plot in Fig. A.1. We then determined the excess overlap of the final weights at the end of training w^* with the principal subspace. To this end, we vectorised the convolutional filters w^*d, which have dimension d, and computed the excess overlap with a matrix of the leading k eigenvectors $U \in \mathbb{R}^{k \times d}$ as

$$m = 1 - \frac{\left\| w - U^\top U w \right\|_2}{\|w\|_2} - k/d, \tag{A.1}$$

where the factor of k/d is the overlap of the filters with a *random* basis of rank k. The excess overlap m thus captures how much of the vector w is captured by the basis U in excess of what

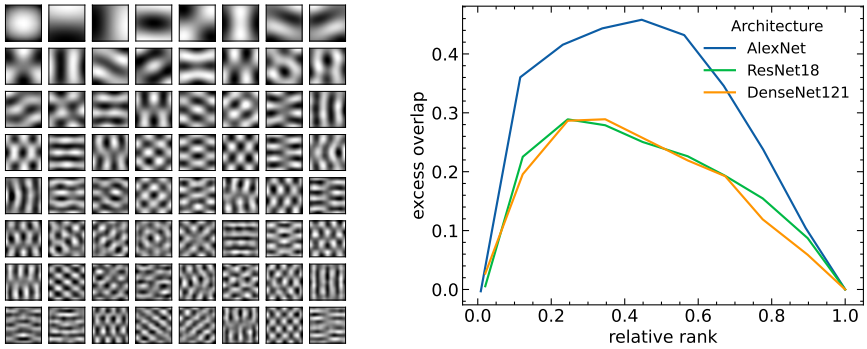

Figure A.1: **The final filters of deep neural networks are concentrated in the subspace spanned by the principal components.** *(Left)* The first 64 principal components of patches of 11×11 pixels (the size of AlexNet filters) sampled randomly from ImageNet images. Filters are shown here for the red channel and shown in greyscale; filters for the other channels look the same. *(Right)* For three deep convolutional neural networks trained on ImageNet, we plot the excess overlap, Eq. (A.1), of the final first-layer filters with the principal subspace of increasing rank, normalised by input dimension since filters in the three architectures have different sizes. The excess overlap captures how much of the filters is contained by the leading principal components compared to a random basis of the same rank.

would be captured by a random basis of the same size. We plot the excess overlap m against the relative rank k/d for the filters of the three networks in Fig. A.1 There is a peak around $k = .4d$. We then plotted the excess overlap of the filters with the fixed subspace of the leading $k = .4d$ principal components for each snapshot of the networks, yielding the green curves in Fig. 2.

Excess kurtosis For each weight snapshot, we extracted the weights for each first-layer convolution and computed the dot-products $s_\mu^k = w^k \cdot x^\mu$ between the weight of the kth convolution for randomly sampled patches from ImageNet x^μ. We normalised the s_μ^k for each neuron (i.e. fixed k) to have variance 1, and computed the excess kurtosis of the resulting s_μ^k,

$$\kappa_{\text{ex}}(s) = \mathbb{E}s^4 - 3. \tag{A.2}$$

We then averaged over all the neurons to obtain the curves shown in Fig. 2(b).

Appendix B. Theoretical analysis

B.1. Notation

We used the standard conventions $o, O, \Theta, \Omega, \omega$ for asymptotics. We recall them here for sequences, Let $\{x_k\}_{k\in\mathbb{N}}$, $\{y_k\}_{k\in\mathbb{N}}$ be real valued sequences. Then:

$$\begin{aligned}
x_k \in o(y_k) &\iff \lim_{k\to\infty} \frac{x_k}{y_k} = 0 \\
x_k \in O(y_k) &\iff \exists\, C > 0 \in \mathbb{R} \quad \forall k > k_0 \quad |x_k| \leq C|y_k| \\
x_k \in \Theta(y_k) &\iff \exists\, C_1, C_2 > 0 \quad \forall k > k_0 \quad C_1 y_k \leq x_k \leq C_2 y_k \\
x_k \in \Omega(y_k) &\iff \exists\, C > 0 \in \mathbb{R} \quad \forall k > k_0 \quad |x_k| \geq C|y_k| \\
x_k \in \omega(y_k) &\iff \lim_{k\to\infty} \frac{|x_k|}{|y_k|} = \infty
\end{aligned}$$

We occasionally used the shorthands $x_k \ll y_k$ for $x_k = o(y_k)$, $x_k \lesssim y_k$ for $x_k = O(y_k)$ and $x_k \asymp y_n$ for $x_k = \Theta(y_k)$. When it is not clear what is the index of the sequence, we write O_k, e.g. $d = O_D(1)$.

B.2. The subspace model turns out to be a spiked cumulant model after preprocessing

Before running ICA, the inputs are usually preprocessed in a canonical way. This standard procedure includes whitening to ensure that the covariance matrix is the identity, together with PCA, to reduce the dimensionality of the original data. It means that, if the inputs are in $\mathbb{R}^D$, they are usually projected in the space generated by the first fixed $r < D$ principal components, and then ICA is performed. We consider the inputs drawn according to the subspace model (3). Note that we are assuming that the spikes $\{u^1, \dots, u^r\}$ are the first r - leading principal components, and that the spike v is a linear combination of those. In the following theorem, we prove that the preprocessed inputs follow a spiked cumulant distribution, where the spike which corresponds to the non-Gaussian signal is simply the normalized projection on the suitable low dimensional subspace of the rotated (w.r.t. the whitening matrix) original non-Gaussian spike.

Theorem 2 (Reduced ICA model) *Consider the subspace model (3). If the common preprocessing steps are performed, i.e. the data are whitened and projected onto the low dimensional subspace spanned by the leading principal components $\{u^1, \dots, u^r\}$, the transformed inputs are drawn according to the following model:*

$$y^\mu = \sqrt{\gamma}\, h^\mu \widetilde{v} + \hat{z}^\mu \in \mathbb{R}^r,$$

where $\widetilde{v} = \mathrm{proj}(\mathrm{Sv})/\|\mathrm{proj}(\mathrm{Sv})\|$ is the normalised projection on the low dimensional subspace of the whitened original spike v. More precisely, γ is a suitable signal to noise ratio, $S \in \mathbb{R}^{D\times D}$ is the whitening matrix for our original inputs and $\hat{z}^\mu$ is a zero-mean noise vector with covariance matrix $\sum_{\rho=1}^r (1 - (Sv)_\rho^2)\, e_\rho\, e_\rho^\top$, with $\{e_1, \dots, e_r\}$ vectors of the canonical basis for $\mathbb{R}^r$.

Proof The subspace model (3) is build in such a way that the first r leading components are the spikes $\{u^1, \dots, u^r\}$, being the covariance matrix of the inputs $C = \sum_{\rho=1}^r \beta_2^\rho\, u^\rho (u^\rho)^\top + \mathbb{1}_d$. Importantly, the non-Gaussian spike to be recovered from (3) is a linear combination of the already mentioned spikes, which implies that all the signal is contained in the subspace generated by them.

We are going to show that when we whiten the data and project them onto the subspace generated by $\{u^1, \dots, u^r\}$, we still recover a spiked cumulant model in a fixed finite dimension r.

Whitening To whiten the data, if is sufficient to multiply them by the whitening matrix $S := \sqrt{C^{-1}}$, which in the case of the subspace model (3) is simply

$$S = U \begin{pmatrix} 1/\sqrt{1+\beta_2^1} & & & & & \\ & \ddots & & & & \\ & & 1/\sqrt{1+\beta_2^r} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} U^\top,$$

where $U := [u^1, \dots, u^r, e_{r+1}, \dots, e_D] \in \mathbb{R}^{D\times D}$ is an orthogonal matrix, being $\{e_{r+1}, \dots, e_D\}$ part of the canonical basis for $\mathbb{R}^D$. Moreover, we can from now on set $S_{ii} = 0$ for $i = r+1, \dots, D$, since in the next step we will only retain the first r principal components.

The new - whitened - inputs are now defined as $\widetilde{x}^\mu := Sx^\mu = Uy^\mu$, for any $\mu = 1, \dots, n$. Note that the entries of $y^\mu \in \mathbb{R}^D$ are just the components of the original input x^μ with respect to the basis $\{u^1, \dots, u^r, e_{r+1}, \dots, e_D\}$, after the rotation due to the whitening. More precisely, for $\rho = 1, \dots, r$, the entries of y^μ read

$$y_\rho^\mu := \frac{1}{\sqrt{1+\beta_2^\rho}}(x^\mu \cdot u^\rho) = \frac{1}{\sqrt{1+\beta_2^\rho}}\Big[\sqrt{\beta_2^\rho} g_\rho^\mu + h^\mu \alpha^\rho + (u^\rho \cdot z^\mu)\Big], \tag{B.1}$$

where we have used the fact that the non-Gaussian spike v is a linear combination of the pairwise orthogonal spikes $\{u^1, \dots, u^r\}$ and that α_ρ are the corresponding coefficients. Since the noise vector z^μ has mean zero and covariance matrix $\mathbb{1} - vv^\top$, it holds that $(u^\rho \cdot z^\mu) \sim \mathcal{N}(0, 1-\alpha_\rho^2)$. Hence,

$$\widetilde{z}_\rho^\mu := \frac{1}{\sqrt{1+\beta_2^\rho}}\Big[\sqrt{\beta}_2^\rho g_\rho^\mu + (u^\rho \cdot z^\mu)\Big] \sim \mathcal{N}\Big(0,\, 1 - \frac{\alpha_\rho^2}{1+\beta_2^\rho}\Big).$$

We conclude that the non trivial entries of each y^μ are given by

$$y_\rho^\mu = \frac{\alpha_\rho}{\sqrt{1+\beta_2^\rho}} h^\mu + \widetilde{z}_\rho^\mu. \tag{B.2}$$

On the other hand, recall that we have $y_i^\mu = 0$ for $i = r+1, \dots, D$.

Projection on the subspace spanned by the principal components We now project each whitened input $\widetilde{x}^\mu$ onto the subspace generated by the first r principal components $\{u^1, \dots, u^r\}$, which simply means that we take track only of the r non trivial components given by (B.2), i.e. the ones for $\rho = 1, \dots, r$. More formally, but with a slight abuse of notation, we keep calling the projected vectors y^μ, so that each projected input is now $y^\mu = [u^1, \dots, u^r]^\top \widetilde{x}^\mu \in \mathbb{R}^r$.

It is left to show that, after whitening and PCA, our data are still spiked and, more precisely, we have recovered nothing but a spiked cumulant model in $\mathbb{R}^r$. Indeed,

$$\begin{aligned} y^\mu &= \sum_{\rho=1}^r y_\rho^\mu\, e_\rho = \Big(\sum_{\rho=1}^r \frac{\alpha_\rho}{\sqrt{1+\beta_2^\rho}}\, e_\rho\Big)\sqrt{\beta_4} h^\mu + \sum_{\rho=1}^r \widetilde{z}_\rho^\mu\, e_\rho \\ &= h^\mu\, \mathrm{proj}(Sv) + \hat{z}^\mu = \|\mathrm{proj}(Sv)\| h^\mu \underbrace{\mathrm{proj}(Sv)/\|\mathrm{proj}(Sv)\|}_{\text{new spike } \widetilde{v}} + \hat{z}^\mu \\ &= \|\mathrm{proj}(Sv)\| h^\mu\, \widetilde{v} + \hat{z}^\mu, \end{aligned}$$

where $\text{proj}(Sv)$ is the projection of Sv onto the subspace generated by the spikes $\{u^1, \dots, u^r\}$, $\widetilde{v} := \text{proj}(Sv)/\|\text{proj}(Sv)\|$, and $\hat{z}^\mu$ is a Gaussian random vector of zero mean and covariance matrix $\sum_{\rho=1}^{r}(1-(Sv)_\rho^2)\, e_\rho\, e_\rho^\top$.
In conclusion, the new (projected and whitened) inputs are nothing but

$$y^\mu = \|\text{proj}(Sv)\|\, h^\mu \widetilde{v} + \hat{z}^\mu \in \mathbb{R}^r,$$

hence we found that the inputs are drawn according to a spiked cumulant model, where $\text{proj}(Sv)$ is the spike corresponding to the original non-Gaussian signal h^μ. If we set $\gamma = \|\text{proj}(Sv)\|^2$, the thesis follows. ■

Thanks to Theorem 2, in case of inputs drawn from the subspace model and subjected to standard preprocessing steps, ICA is supposed to recover one single low dimensional spike belonging to the subspace spanned by the leading principal components $\{u^1, \dots, u^r\} \subset \mathbb{R}^r$. If the signal-to-noise ratios $\{\beta_2^1, \dots, \beta_2^r\}$ of the principal components are equal, up to normalisation and rescaling, the said spike is nothing but the vector $\alpha := (\alpha_1, \dots, \alpha_r)^\top$, whose components are the ones of the original spike v corresponding to the non-Gaussian latent variable h^μ in the subspace model. In Section B.3 we discuss what happens to the reduced model when we do not have access to the exact value of the leading principal components, but we estimate it through PCA. Relying on the fact that the population loss for ICA on the reduced model exhibits convenient monotonicity properties, in Section B.4 we prove the convergence, in a linear sample complexity, of gradient descent (GD), therefore concluding that this kind of finite optimisation is indeed effective the context of the subspace model.

B.3. Errors in estimation of PCs at linear sample complexity in high dimensions

Recall that we are interested in analysing the performance of ICA at a linear sample complexity, namely when the number of samples scales as the dimension of the ambient space D, up to logarithmic factors. So, we start by using an infinite amount of samples to estimate the first principal components of our D-dimensional inputs. Then, we project down in low dimensions. This means that, in practice, the principal components have to be estimated through PCA, and this may have an impact on the distribution of the reduced model. Indeed, PCA at linear sample complexity only yields an estimate $\hat{u}^\rho$ of the leading principal components, whose overlap with u^ρ can be computed exactly: in case of a finite signal-to-noise ratio and $n = \Theta(D)$ it is strictly smaller than 1, see for example Eq. 3.37 of Bandeira et al. [2] for the rank-1 case. However, a number of samples n such that $n/D \to \infty$, even if the divergence is arbitrary slow, is sufficient to perfectly recover with high probability the eigenvectors of the covariance matrix of the inputs. Hence, in practice, one considers the change of basis provided by the matrix U is made by $\widehat{U} := [\hat{u}_1, \dots, \hat{u}_r, e_{r+1}, \dots, e_D] \in \mathbb{R}^{D\times D}$, where $\{\hat{u}^1, \dots, \hat{u}^r\}$ is the set of the estimators provided by PCA at a linear sample complexity of the first r-leading components. Fix $\varphi \in [0,1]$, which is the overlap between the estimator for each principal component and the principal component itself, and define $\varepsilon = \sqrt{1-\varphi^2}$. When $n/D \to \infty$ arbitrarily slow, e.g. it is sufficient that $n = \Theta(D \log D)$, we have that $\varepsilon = \varepsilon(D) \to 0$, and then clearly $\varphi = \varphi(D) \to 1$. For the derivation of the (perturbed) reduced model, we assume that the estimated PCs are

$$\hat{u}^\rho = \varphi\, u^\rho + \varepsilon \widetilde{u}^\rho, \tag{B.3}$$

for $\rho = 1, \dots, r$ and $\widetilde{u}^\rho \perp u^\rho$, $\|\widetilde{u}^\rho\| = 1$. The following lemma shows that we can obtain the reduced model even if the eigenvectors of the covariance matrix of the inputs are approximated

by PCA. The model is simply the one found in Theorem 2, up to some errors introduced with the perturbed eigenvectors, which however vanish for $D \to \infty$.

Lemma 3 *Consider the subspace model (3) and assume we perform the usual preprocessing steps for ICA without having access to the exact value of the covariance spikes, but only to $\hat{u}^\rho$ defined in Eq. B.3. Consider a number of samples n such that n/D diverges arbitrary slow, e.g. $n = \Theta(D \log D)$. Then, with high probability, the transformed inputs are nothing but the reduced model introduced in Theorem 2, up to vanishing errors.*

Proof We need to compute, for $\mu = 1, \dots, D$, the whitened inputs $\widetilde{x}^\mu = Sx^\mu = \widehat{U} y^\mu$, where S is the whitening matrix. After that, we will project the inputs onto the principal subspace. Note that for any $\rho, \rho' = 1, \dots, r$, we obtain

$$u^\rho \cdot \hat{u}^{\rho'} = (\varphi u^\rho + \varepsilon \widetilde{u}^\rho) \cdot u^{\rho'} = \varphi\, \delta_{\rho\rho'} + \varepsilon\, (\widetilde{u}^\rho \cdot u^{\rho'}).$$

Hence, with respect to the usual pairwise orthonormality of the spike generating the principal subspace, there is an extra "error term" provided by the fact that we cannot precisely estimate the principal components with a linear number of samples. Taking into account these perturbations, we compute for $\mu = 1, \dots, n$, $\rho = 1, \dots, r$ the components of y^μ, i.e.

$$\begin{aligned}
y^\mu_\rho =& \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sum_{\rho'=1}^{r} \sqrt{\beta_2^{\rho'}}\, g^\mu_{\rho'} (\hat{u}^\rho \cdot u^{\rho'}) + \sqrt{\beta_4} h^\mu \sum_{\rho'}^{r} \alpha_{\rho'} (\hat{u}^\rho \cdot u^{\rho'}) + z^\mu \cdot \hat{u}^\rho \Big] \\
=& \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sqrt{\beta_2^\rho} g^\mu_\rho \varphi + \sum_{\rho'=1}^{r} \varepsilon \sqrt{\beta_2^{\rho'}}\, g^\mu_{\rho'} (\tilde{u}^\rho \cdot u^{\rho'}) + \underbrace{h^\mu \alpha_\rho \varphi + \sum_{\rho'}^{r} \alpha_{\rho'} \varepsilon h^\mu (\widetilde{u}^\rho \cdot u^{\rho'})}_{\text{non-Gaussian signal}} + z^\mu \cdot \hat{u}^\rho \Big] \\
=& \sqrt{\frac{1}{1+\beta_2^\rho}} h^\mu \Big(\alpha_\rho \varphi + \sum_{\rho'=1}^{r} \alpha_{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'}) \varepsilon \Big) + \widetilde{z}^\mu_\rho,
\end{aligned}$$

where the Gaussian noise is

$$\widetilde{z}^\mu_\rho = \frac{1}{\sqrt{1+\beta_2^\rho}} \Big[\sqrt{\beta_2^\rho} g^\mu_\rho \varphi + \sum_{\rho'=1}^{r} \sqrt{\beta_2^{\rho'}}\, g^\mu_{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'}) \varepsilon + \varphi (z^\mu \cdot u^\rho) + \varepsilon (z^\mu \cdot \widetilde{u}^\rho) \Big].$$

Clearly, for any $\rho = 1, \dots, r$, $\widetilde{z}^\mu_\rho$ is a centred Gaussian random variable with variance

$$c = \frac{1}{1+\beta_2^\rho} \Big(\varphi^2 \beta_2^\rho + \varepsilon^2 \sum_{\rho'=1}^{r} \beta_2^{\rho'} (\widetilde{u}^\rho \cdot u^{\rho'})^2 + 1 - \alpha_\rho^2 \Big) \in \mathbb{R}.$$

The other components y^μ_ρ for $\rho = r+1, \dots, D$ vanish. In the previous formulas, it has been emphasised the dependence of several terms on the overlap φ between any exact spike and its estimation provided by PCA. Indeed, we can assume now that all the other terms, depending on ε, vanish with high probability when $D \to \infty$, since e.g. $n = \Theta(D \log D)$. Hence, we get

$$y^\mu_\rho = \varphi \left(\sqrt{\frac{1}{1+\beta_2^\rho}} h^\mu \alpha_\rho + \sqrt{\frac{\beta_2^\rho}{1+\beta_2^\rho}} g^\mu_\rho + (z^\mu \cdot u^\rho) \right) + o_D(1), \tag{B.4}$$

where $\varphi \to 1$ with high probability. We can see that the components of the inputs in (B.4) are the ones previously obtained in Theorem 2 multiplied by the overlap φ. Now, the last preprocessing

step consists in projecting $y^\mu \in \mathbb{R}^D$ in the right low dimensional subspace. As in Theorem 2, with a slight abuse of notation, we adopt the notation y^μ to refer simply to the vector in $\mathbb{R}^r$ whose components are y^μ_ρ, for $\rho = 1, \dots, r$. Hence, the reduced inputs turn out to be $y^\mu + Ry^\mu$, where R is an "error" matrix whose entries are $r_{ij} = \hat{u}^i \cdot \hat{u}^j$ if $i \neq j$ and zero otherwise. Since $\hat{u}^i \cdot \hat{u}^j = \varphi\varepsilon(u^i \cdot \widetilde{u}^j) + \varphi\varepsilon(\widetilde{u}^i \cdot u^j) + \varepsilon^2(\widetilde{u}^i \cdot \widetilde{u}^j)$, these last error terms vanish with high probability when $D \to \infty$. ■

B.4. Optimization on the subspace

In this section we prove that it is possible to efficiently solve ICA on the subspace model (3) with $n = \Theta(D \log(D))$ (or even slower, it is sufficient that $n/D \to \infty$ to be sure that the errors introduced with the estimation of the principal components by PCA vanish for $D \to \infty$) samples after the projection on the principal r-dimensional subspace, with $r = O(1)$. The data model we refer to in this section is the reduced model (4), which is a simple spikes cumulant model (see e.g. [33]). In our case, the spike corresponding to the non-Gaussian latent variable, namely the one to be recovered, is $\widetilde{v}$ in (4). We choose to directly state and prove Theorem 4 for inputs distributed according to the reduced model because, when the number of samples is sufficiently large, the errors due to the estimation of the principal components have a limited impact on the distribution of the inputs, in the sense of Lemma 3.

On the algorithmic side, we consider a projected gradient method that uses the n samples in K batches of B samples each, with $KB = n$ and both diverging in the high dimensional limit, $K, B = \omega(1)$ (for example $K \approx B \approx \sqrt{n}$). So we denote with $\mathcal{B}_k = (x^\mu)_{\mu \in [B]}^k$ the k-th batch of i.i.d. samples from $\mathbb{P}$, the distribution of the already mentioned inputs in $\mathbb{R}^r$. The algorithm is:

$$\begin{cases} w_0 \sim \text{Unif}\left(\mathbb{S}^{r-1}\right), \\ \widetilde{w}_{k+1} = w_k + \eta \nabla_{\text{sph}} \frac{1}{B} \sum_{x \in \mathcal{B}_k} G(w \cdot x) \quad k > 1, \\ w_{k+1} = \frac{\widetilde{w}_{k+1}}{||\widetilde{w}_{k+1}||}, \end{cases} \tag{B.5}$$

where $\nabla_{\text{sph}} := (\mathbb{1} - w_k w_k^\top)\nabla$.

Theorem 4 (Convergence of gradient ascent) *Assume that the population loss $\mathcal{L}(w) = \mathbb{E}_{\mathbb{P}} G(w \cdot x)$ depends on w only through the the overlap $\alpha = w \cdot \tilde{v}$, and let G be such that:*

- $\mathcal{L} \in \mathcal{C}^2[-1, 1]$,
- $\mathcal{L}(\alpha)\alpha > 0$ *for all* $\alpha \neq 0$ *in* $[-1, 1]$.

We update w_k for $k \in \{0, \dots, K\}$ following the dynamic dictated by Eq. (B.5), where $\mathcal{B}_k = (x^\mu)_{\mu \in [B]}^k$ is the k-th batch of i.i.d. samples from $\mathbb{P}$, the spiked cumulant distribution with spike $\tilde{v} \in \mathbb{R}^r$ of (4). Then, w_K converges to the spike $\tilde{v}$ with high probability in the limit $n \to \infty$. More precisely, for any ε there exists n_ε, such that if $n > n_\varepsilon$, it is large enough so that we can choose B and K so that $BK \leq n$ and the event $|\alpha_K| = |w_K \cdot \tilde{v}| \geq 1 - \varepsilon$ happens with probability larger than $1 - \varepsilon$.

Proof The first part of the proof is devoted to estimate the error of replacing $\nabla_{\text{sph}} \frac{1}{B} \sum_{x \in \mathcal{B}_k} G(w \cdot x)$ with $\nabla_{\text{sph}} \mathcal{L}(\alpha)$. First note that G and $\mathbb{P}$ ($G \in \mathcal{C}^1$ and $\ell \in \mathcal{L}^2$) are regular enough so that we can differentiate under the integral sign and write: $\nabla \mathbb{E}[G(w \cdot x)] = \mathbb{E}[\nabla G(w \cdot x)] = \mathcal{L}'(\alpha) v$, where the

last equality can be shown with the orthogonality property of Hermite polynomials (see property 1 in [8]). We can now use the law of large numbers; if B is large enough, we can find an event with probability larger than $1-\varepsilon$ such that for all k:

$$\left|\mathbb{E}[\nabla G(w\cdot x)] - \frac{1}{B}\sum_{x\in\mathcal{B}_k}\nabla_{\text{sph}}G(w\cdot x)\right| \leq \eta^2.$$

So we can study the following dynamics

$$\begin{cases}\widetilde{w}_{k+1} = w_k + \eta\nabla_{\text{sph}}\mathcal{L}(\alpha_k) + O(\eta^2),\\ w_{k+1} = \frac{\widetilde{w}_{k+1}}{||\widetilde{w}_{k+1}||},\end{cases}$$

where $\nabla_{\text{sph}} := (\mathbb{1} - w_k w_k^\top)\nabla$ is the spherical gradient, that ensures that $\eta\nabla_{sph}\mathcal{L}(\alpha_k)$ is orthogonal to w_k, hence $||\widetilde{w}_{k+1}|| = \sqrt{1+\eta^2||\nabla_{\text{sph}}\mathcal{L}(\alpha_k)||^2}$. Since the step size $\eta\to 0$, we can apply the Taylor expansion $\frac{1}{\sqrt{1+x^2}} = 1 - x^2/2 + o(x^2)$. Since $\nabla\mathcal{L}$ is bounded thanks to our assumptions on G, this leads to

$$w_{k+1} = w_k + \eta\nabla_{\text{sph}}\mathcal{L}(\alpha_k) + O(\eta^2),$$

where the $O(\eta^2)$ is uniform in k. We take now the scalar product with the target vector $\tilde{v}$ to get the evolution of the overlap:

$$\begin{aligned}\alpha_{k+1} &= \alpha_k + \eta\tilde{v}\cdot(\mathbb{1} - w_k w_k^\top)\mathcal{L}'(\alpha_k)\tilde{v} + O(\eta^2)\\ &= \alpha_k + \eta\mathcal{L}'(\alpha_k)(1-\alpha_k^2) + O(\eta^2).\end{aligned} \tag{B.6}$$

Now it is helpful to consider the continuous time gradient flow $\alpha(t)$ that satisfies the equation:

$$\begin{cases}\dot{\alpha}(t) = \mathcal{L}'(\alpha(t))(1-\alpha^2(t)),\\ \alpha(0) = \alpha_0 > 0.\end{cases}$$

Note that we assumed $\alpha_0 > 0$ which happens exactly with probability 1/2, the case $\alpha_0 < 0$ is exactly analogous, we will only use that $\mathcal{L}'(\alpha) < 0$ for all $\alpha < 0$ to prove convergence to -1 instead of $+1$ (if the initialization is negative the algorithm will converge to $-\tilde{v}$ instead of $\tilde{v}$, identifying the same one dimensional subspace). Let $\varepsilon > 0$ and denote $0 < C_\varepsilon := \inf_{[\alpha_0, 1-\varepsilon]}\mathcal{L}'(\alpha_0)$. We have that for all $t > 0$, such that $\alpha(t) \leq 1-\varepsilon$, then $\alpha(t) > f(t)$ where f is the solution of the auxiliary Cauchy problem

$$\begin{cases}\dot{f}(t) = C(1-f^2(t)),\\ f(0) = \alpha_0 > 0,\end{cases}$$

which is solved by

$$f(t) = \frac{(\alpha_0+1)e^{2Ct} + \alpha_0 - 1}{(\alpha_0+1)e^{2Ct} - \alpha_0 + 1},$$

that converges monotonically to 1 as $t\to\infty$. This proves that $\lim_{t\to\infty}\alpha(t) = 1$. We can now transfer this information to the discrete dynamics using convergence of Euler's method (Theorem 10.2 in [27]). For any $\eta\in(0,\eta_0)$ and $k\in\mathbb{N}$:

$$\begin{aligned}|\alpha_k - \alpha(\eta k)| &\leq k\eta e^{\Lambda k\eta}O(\eta) + \sum_{i=1}^{k} O(\eta^2)\\ &\leq k\eta e^{\Lambda k\eta}O(\eta) + kO(\eta^2),\end{aligned}$$

where the first term comes from the convergence error and the second from the error in (B.6). It is possible to perform the second step since the $O(\eta^2)$ depends only on global property of $\mathcal{L}$, so it is uniform in k. Hence, the discrete dynamics approximate the gradient flow for all k such that $k\eta = O(1)$. To conclude the proof, we just need to pick η small enough and K large enough so that $t_K = K\eta$ is such that $\alpha(t) > 1 - \varepsilon/2$ and $|\alpha_K - \alpha(t_K)| \leq \varepsilon/2$. ■

The hypotheses required on G are verified for the most commonly used in practice contrast functions. For example, we can trivially deal with the case of the 4th Hermite polynomial $G(s) = s^4 - 3$ by using the Hermite orthogonality property (property 1 in [8]) and the likelihood ratio trick on the loss for ICA [30] whereas to justify the use of $G(s) = -e^{-s^2/2}$ we can explicitly compute the population loss.

Lemma 5 (Monotonicity of the population loss) *Consider $G(s) = -\exp\{-s^2/2\}$ and the reduced model (4). Then, the population loss for ICA, i.e. $\mathcal{L}(w) = \mathbb{E}_x[G(w \cdot x)]$, can be explicitly computed and depends only on the overlap $\alpha := w \cdot \tilde{v}$ in the following way:*

$$\mathcal{L}(\alpha) = -\sqrt{\frac{2}{4-\alpha^2}}\ \exp\left\{-\frac{\alpha^4 + 2\alpha^2}{(2+\alpha^2)(4-\alpha^2)}\right\}.$$

Proof The proof consists in the direct computation of the expectation of $G(x \cdot w)$ with respect to the distribution of the inputs. Precisely, for any $w \in \mathbb{R}^r$, we integrate with respect to the non Gaussian spike h and the noise vector z as follows:

$$\mathcal{L}(w) = \frac{1}{2\sqrt{(2\pi)^r}}\left[\left(\int G(w \cdot x)\, e^{-z^2/2} dz\right)\delta_{\{h=1\}} + \left(\int G(w \cdot x)\, e^{-z^2/2} dz\right)\delta_{\{h=-1\}}\right],$$

where each integrand, for $h = \pm 1$, is simply $e^{-\frac{1}{2}[(w\cdot x)^2 + \|z\|^2]}$. Therefore, we need to compute explicitly $(w \cdot x)^2$, which turns out to be

$$(w \cdot x)^2 = \frac{1}{\sqrt{2}} h\alpha + \frac{\alpha}{\sqrt{2}} + (w_{\perp\tilde{v}} \cdot z),$$

where $\alpha := w \cdot \tilde{v}$ and $w_{\perp\tilde{v}}$ is the portion of the weight vector orthogonal to the spike $\tilde{v}$. Now, we can decompose the noise in the component parallel to the spike, i.e. $z_{//\tilde{v}}$, the component paraller to $w_{\perp\tilde{v}}$, i.e. $z_{//w_{\perp\tilde{v}}}$, and everything else $z_{\perp}$, and integrate with respect these three contributions. ■

To summarize, we started from the high dimensional problem Eq. (3) in $\mathbb{R}^D$ with D large and a number of samples $n = \Theta(D\log(D))$ which would be too small to have recovery of v through a direct application of FastICA. However, the projection on the principal component subspace $\langle u_1, \dots, u_r \rangle$, which is possible to perform with arbitrarily small error (Lemma 3) leads to a spiked cumulant model in $\mathbb{R}^r$ (Theorem 2). Thanks to suitable properties of the population loss, verified in Lemma 5, we are able to recover the spiked direction $\tilde{v}$ with a linear sample complexity, as shown in Theorem 4.