Hans van Gorp, Oisín Nolan, Yonina C. Eldar, and Ruud J. G. van Sloun

The Geometry of LISTA

The celebrated iterative soft-thresholding algorithm (ISTA) and its accelerated variant, fast ISTA (FISTA), are classical signal processing methods used to solve the least absolute shrinkage and selection operator (LASSO) problem [1]. This problem spans various applications, including medical imaging, direction-of-arrival estimation, astronomy, and sparse coding. Within the broader trend in signal processing of transitioning from classical model-based approaches to deep learning-based methods, the learned ISTA (LISTA) algorithm was proposed as a way to solve the LASSO problem using deep learning while preserving the original ISTA structure [2].

LISTA learns a fast approximation to the LASSO problem by casting the weight matrices of the ISTA algorithm as learnable parameters and making them unique for each iteration, a technique now known as deep unfolding [3]. LISTA achieves superior reconstruction results in fewer iterations compared to ISTA. This is due to two main factors. First, by learning its weights, LISTA addresses potential modeling mismatches, such as imperfect knowledge of noise behavior or the forward model. Second, even with an accurately known model, the sparsifying basis might be too complex to implement efficiently using classical methods. Here, we consider a geometric interpretation to gain insight into why LISTA performs well with significantly fewer iterations (or folds) compared to ISTA. To that end, we use the fact that both models are continuous piecewise linear (CPWL) functions. Our main contributions are as follows:

- We demonstrate that existing bounds on the complexity of the geometry for ISTA and LISTA are insufficient to explain their differences; their geometries must be assessed experimentally.
- We introduce the concepts of expected knot density and decision density as practical metrics to evaluate the geometry of these algorithms.
- We establish a lower bound on the maximum a posteriori (MAP) optimal decision density for sparse linear inverse problems.
- We demonstrate the effect of the loss function on the geometry of LISTA. Training LISTA with an L1 norm produces fewer larger regions and a lower knot density compared to an L2 loss, which creates many small regions and a higher knot density.
- We highlight that LISTA converges faster than ISTA and, when trained with L1, produces a simpler geometry with lower knot and decision densities closer to the optimal.

Dr. Roula Nassif acted as the associate editor for this article and recommended it for publication.

Digital Object Identifier 10.1109/MSP.2025.3638279
Date of current version: 7 August 2026

Linear inverse problems and (L)ISTA

We consider linear inverse problems of the form $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$, where $y \in \mathbb{R}^M$ is the measurement, $\mathbf{x} \in \mathbb{R}^N$ is the signal of interest that we want to reconstruct, $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the measurement matrix, and $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 I)$ is additive noise. Furthermore, $M \ll N$, which results in an underdetermined system with potentially infinite solutions. We assume the signal contains at most K nonzero entries, i.e., $\|\mathbf{x}\|_0 \leq K$. A solution to this problem may be found by solving the LASSO problem

$$\hat{\mathbf{x}} \in \operatorname*{argmin}_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \quad (1)$$

where λ is a weighting factor that determines the relative importance of the Laplacian sparsity prior on $\mathbf{x}$ versus the reconstruction likelihood with respect to $\mathbf{y}$. The ISTA algorithm solves this problem iteratively using a proximal approach, taking gradient steps to minimize the squared error between the observed and estimated measurements and applying the soft-thresholding operator $\mathcal{S}_\lambda$ to promote sparsity.

$$\mathbf{z}^{(t)} = (\mathbf{I} - \mu \mathbf{A}^\top \mathbf{A})\mathbf{x}^{(t-1)} + \mu \mathbf{A}^\top \mathbf{y} \quad (2)$$

$$\mathbf{x}^{(t)} = \mathcal{S}_\lambda(\mathbf{z}^{(t)}) \quad (3)$$

where μ controls the size of the gradient steps that minimize the reconstruction loss. The initial solution $\mathbf{x}^{(0)}$ could be arbitrarily chosen from $\mathbf{x}^{(0)} \in \mathbb{R}^N$; however, here, we follow the common practice of setting it to zero. The soft-thresholding operator solves the subproblem $\operatorname{argmin}_{\mathbf{x}} (1/2)\|\mathbf{x} - \mathbf{z}^{(t)}\|_2^2 + \lambda \|\mathbf{x}\|_1$.

$$\mathcal{S}_\lambda(\mathbf{z}) = \operatorname{sign}(\mathbf{z}) \max(\operatorname{abs}(\mathbf{z}) - \lambda, 0). \quad (4)$$

ISTA requires many iterations to converge, and its accuracy depends on precise knowledge of the forward model $\mathbf{A}$. These challenges led to the development of LISTA, which replaces the linear transformations $\mathbf{I}-\mu\mathbf{A}^\top\mathbf{A}$ and $\mu\mathbf{A}^\top$ with learned weight matrices $\mathbf{W}_1^{(t)}$ and $\mathbf{W}_2^{(t)}$ that can vary with each iteration t, yielding

$$\mathbf{x}^{(t)} = \mathcal{S}_\lambda(\mathbf{W}_1^{(t)}\mathbf{x}^{(t-1)} + \mathbf{W}_2^{(t)}\mathbf{y}). \quad (5)$$

While random initialization of the trainable parameters is possible, we kick-start the training process by initializing the parameters of LISTA using those of the ISTA algorithm, $\mathbf{W}_1^{(t)} = \mathbf{I}-\mathbf{A}^\top\mathbf{A}$ and $\mathbf{W}_2^{(t)} = \mathbf{A}^\top$, for all iterations t. The learnable parameters for LISTA are then optimized by minimizing a loss function comparing the target vector $\mathbf{x}$ to the predicted vectors $\mathbf{x}^{(T)}$. We will test both mean squared error (L2) and mean absolute error (L1) between $\mathbf{x}$ and $\mathbf{x}^{(T)}$ as the loss function for LISTA.

Structure and properties of CPWL functions

The geometry of both ISTA and LISTA can be described using the concept of CPWL functions. Following [4], a CPWL function is defined as shown in Definition 1.

Definition 1

A function $f:\mathbb{R}^M \to \mathbb{R}^N$ is CPWL if it is continuous, and there exists a set $\{f^q \in \{1,\ldots,Q\}\}$ of affine functions and closed subsets $(\Omega_q)_{q=1}^Q$ of $\mathbb{R}^M$ with nonempty and pairwise disjoint interiors such that $\cup_{q=1}^Q \Omega_q = \mathbb{R}^M$ and $f_{|\Omega_q} = f^q$ on Ω_q. The Ω_q are called the projection regions.

Note that f is single-valued and continuous across $\mathbb{R}^M$ as each input lies in exactly one region with a uniquely defined affine mapping. Additionally, ISTA and LISTA are both CPWL functions as they comprise a composition of CPWL functions (linear transformations and the soft-thresholding operator) [4]. An example of the projection regions created by a single iteration of ISTA is shown in Figure 1. Here, an ISTA algorithm operating on $f:\mathbb{R}^2 \to \mathbb{R}^3$ splits the input space into eight linear projection regions. By splitting each projection region into its convex components, we obtain a set of convex regions, $\Pi = (P_1,\ldots,P_L)$. The minimum number of convex regions L that is obtained is termed the number of convex regions. In Figure 1, the 8 projection regions are split into 19 convex regions.

The number of regions is a useful parameter in the geometrical analysis of nonlinear models, indicating the model's complexity and ability to have different input–output relationships. The number of regions is only one aspect; the distribution and location of the regions are also significant. For instance, in classification, regions tend to concentrate around the decision boundaries [5]. While CPWL analysis is often discussed in the context of neural networks, like LISTA, these tools are equally applicable to ISTA.

Geometric analysis of ISTA and LISTA

Without considering the effects of parameter selection, the geometric complexity of (L)ISTA can already be bounded from above by rewriting the upper bound provided by Goujon et al. [4].

$$\text{Number of projection regions} \le (2^N)^T \quad (6)$$

$$\text{Number of convex regions} \le (3^N)^T. \quad (7)$$

These bounds represent the theoretical maximum number of regions but do not necessarily reflect those obtained in a realized model, where 'realized' refers to LISTA after training and ISTA after hyperparameter tuning. For instance, in Figure 1, we observe 19 convex regions for a single ISTA iteration, while (7) gives a maximum of 27. Another way to upper bound the number of projection regions is to view the linear inverse problem as a classification task. Once the support S of the reconstruction is known, the LASSO problem can be bypassed, and the optimal MAP solution is

$$\hat{\mathbf{x}}_S = \underset{\mathbf{x}_S}{\operatorname{argmin}} \|\mathbf{A}_S\mathbf{x}_S - \mathbf{y}\|_2^2 \quad (8)$$

$$\hat{\mathbf{x}}_S = (\mathbf{A}_S^\top\mathbf{A}_S)^{-1}\mathbf{A}_S^\top\mathbf{y} = \mathbf{A}_S^+\mathbf{y} \quad (9)$$

where $\mathbf{x}_S$ contains only the nonzero elements of $\mathbf{x}$, and $\mathbf{A}_S$ is the submatrix of $\mathbf{A}$ corresponding to indices in S. $\mathbf{A}_S^+$ denotes the Moore-Penrose pseudoinverse. Because each of the possible supports has one associated optimal linear

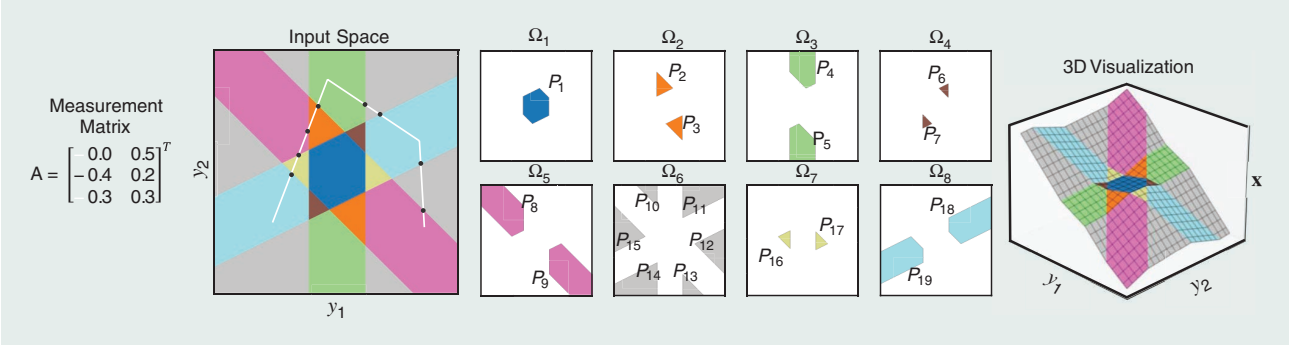

FIGURE 1. An example of the linear regions produced by a single iteration of ISTA on the input space $\mathbf{y}$ (with dimensions M = 2 and N = 3). 8 linear projection regions (Ω_q) and 19 convex piecewise linear regions (P_l) are produced. A random path (white) across the input space is also shown in the overall plot, along which several knots (black) are observed. The rightmost visualization shows the regions in 3D to visualize the output $\mathbf{x} \in \mathcal{R}^3$ we take the sum of the individual components.

projection, an optimal model should not assign more regions than the total number of possible supports.[1]

$$\text{Optimal number of projection regions} \leq \sum_{k=0}^{K} \binom{N}{k}. \tag{10}$$

This leads to two key observations. First, the bounds in (6) and (7) are identical for ISTA and LISTA given the same number of iterations. However, LISTA typically yields a solution in far fewer iterations than ISTA, making these bounds insufficient to explain their differences. Second, the bound in (6) is often much larger than the optimal number in (10), even for $T=1$, since $K \ll N$. Yet, both ISTA and LISTA usually require multiple iterations to reach a good solution. We hypothesize that having enough regions is not sufficient; correct placement and boundary shaping are also critical. Therefore, methods to assess the geometry of realized models are needed.

One such method is to finely sample the input space and compute the Jacobian at each point; samples with identical Jacobians belong to the same projection region. Mathematically, this can be expressed as

$$\text{Number of projection regions} \approx \left|\{\text{Jacobian}(f(\mathbf{y}))\,|\,\mathbf{y}\in\mathcal{G}\}\right| \tag{11}$$

where $\mathcal{G}\subset\mathbb{R}^M$ is a grid of sampling points, $\text{Jacobian}(f(\mathbf{y}))$ denotes the Jacobian of f at location $\mathbf{y}$. This approach was used to generate Figure 1. However, it relies on dense grid sampling, which becomes impractical in high dimensions and cannot be visualized beyond two dimensions. To overcome this issue, we slice the input space along a 2D plane defined by three anchor points, called $\mathbf{a_0}$, $\mathbf{a_1}$, $\mathbf{a_2}$ (e.g., dataset samples). This consists of sampling a grid of points in 2D, $\mathcal{H}\subset\mathbb{R}^2$, and projecting them into $\mathbf{y}$ space

$$\text{Number of projection regions along a slice} \approx \left|\{\text{Jacobian}(f(\mathbf{Bh}+\mathbf{b}))\,|\,\mathbf{h}\in\mathcal{H}\}\right| \tag{12}$$

where $\mathbf{B}\in\mathbb{R}^{M\times 2}$ and $\mathbf{b}\in\mathbb{R}^M$ define the 2D slice. To align with the anchor points, these are set as

$$\mathbf{b}=\mathbf{a}_0,\quad \mathbf{B}=[\mathbf{a}_1-\mathbf{a}_0\,,\mathbf{a}_2-\mathbf{a}_0]. \tag{13}$$

The number of regions on such a slice can also be computed using SplineCam [5], which constructs and back-projects nonlinearity boundaries as graphs. Note that SplineCam operates only on 2D slices.

A more scalable alternative is to estimate the expected knot density, as proposed by Goujon et al. [4]. A "knot" refers to a nonlinear point of an activation function. For example, a rectified linear unit has a single knot at zero, while the soft-thresholding operator has two at $\pm\lambda$. The knot density measures the number of knots encountered along a random path γ in the input space, normalized by the path length (see Figure 1). It is inversely related to the size (and thus the number) of convex linear regions produced by a CPWL function [4]. While Goujon et al. provided bounds for randomly initialized models, these do not distinguish between ISTA and LISTA. We therefore compute the expected knot density for realized models under a chosen path-generating distribution that can be implemented in code. This provides a tractable and model-agnostic proxy for geometric complexity that scales to high-dimensional settings, where direct region counting is infeasible. It also enables a fair comparison between ISTA and LISTA under realistic parameterizations. The expected knot density is defined as

$$\rho_{kt}=\mathbb{E}_{\gamma\sim p_\gamma}\left[\frac{kt_f^\gamma}{\int_\gamma dl}\right] \tag{14}$$

where ρ_{kt} is the expected knot density, kt_f^γ is the number of knots encountered by function f along path γ, and $\int_\gamma dl$ is the path length. In our case, f is implemented by ISTA or LISTA. We estimate this quantity using Monte Carlo (MC) sampling, as shown in Box 1.

To compare with the optimal solution, which depends on support recovery, we extend our analysis to include the recovered support. Instead of using the full Jacobian, we binarize it to visualize support cardinality. This leads to the definition of decision density

$$\rho_d=\mathbb{E}_{\gamma\sim p_\gamma}\left[\frac{d_f^\gamma}{\int_\gamma dl}\right] \tag{15}$$

where d_f^γ counts the number of support changes (decision boundaries) along γ.

By definition, $\rho_d\leq\rho_{kt}$. A large gap between the two indicates that many CPWL regions correspond to a single decision region, while a small gap suggests a more efficient mapping. As shown earlier, the optimal MAP solution ideally uses one CPWL region per decision region, making a small gap desirable. We can lower bound the optimal decision density by noting that two randomly sampled signals $\mathbf{x}$ will almost surely have different supports when $K\ll N$. A straight path between them should therefore cross at least one decision boundary. Thus, the optimal decision density in input space is bounded.

$$\text{Optimal } \rho_d \geq \mathbb{E}_{\mathbf{y}_1,\mathbf{y}_2\sim p_D}\left[\frac{1}{\|\mathbf{y}_1-\mathbf{y}_2\|_2}\right]. \tag{16}$$

Box 1: Pseudocode for knot density calculation

Input: (Discrete) Path-generating distribution p_γ, (L)ISTA model $f(.)$, number of MC trials L
Output: ρ_{kt}

```
kt = 0,  λ = 0
for l = 1 to L do // Expectation through MC
    γ ~ p_γ   // Sample a path *
    λ = λ + length(γ)
    J = Jacobian(f(γ_1))  // Initial Jacobian
    for i = 2 to length(γ) do
        J* = Jacobian(f(γ_i))  // Other Jacobian
        if J* ≠ J then // Jacobian change detected
            kt = kt + 1
        end if
        J = J*  // Set new initial Jacobian
    end for
end for
ρ_kt = kt/λ  // Knot density
```

*Note that in practice, a path γ is represented as a discrete sequence of points sampled from a continuous trajectory defined by the distribution p_γ. We denote the ith point along this sequence as γ_i.

[1]A proof of which can be found on the associated GitHub page: https://github.com/OisinNolan/geometry_of_lista/blob/main/proof.pdf.

Experimental setup and evaluation criteria

To assess the differences between ISTA and LISTA in realized models, we use an experimental approach.[2] We expect different geometrical behavior between ISTA and the two LISTA models (L1 or L2 loss), both quantitatively and qualitatively. Quantitative differences will be assessed in terms of knot density, decision density, and reconstruction performance on the hold-out test set. A model is considered better if it has a small gap between knot and decision density, a decision density close to the optimal, and a lower loss. We hypothesize that both LISTA models will achieve superior reconstruction results compared to ISTA with few iterations. However, when run for several hundred iterations, ISTA is expected to catch up in reconstruction performance, assuming exact knowledge of the forward model. Additionally, we expect LISTA to approach the optimal knot density at earlier iterations.

To test qualitative differences, we will use a 2D slice on the input anchored on three one-sparse orthogonal datapoints. We expect LISTA to center its regions around the datapoints, while ISTA regions should expand infinitely in all directions as it is not trained on data. Additionally, LISTA trained with L1 should display more accurate sparsity regions compared to LISTA trained with L2 loss as the gradient of L2 loss quickly goes to zero before reconstruction becomes sparse.

The experimental settings are as follows: dimensions $M = 16$, $N = 64$, and $K = 8$. The noise follows a zero-mean independent identically distributed (i.i.d.) Gaussian distribution with $\sigma_n = 0.01$. The measurement matrix $\mathbf{A}$ entries are sampled from a zero-mean i.i.d. Gaussian distribution with unit variance and the entries are normalized by the matrix's largest singular value. Data samples $\mathbf{x}$ have sparse values equal to zero and nonsparse values uniformly in [1, 2] or [−2, −1], with the number of nonsparse values uniformly sampled between $k = 0$ and $k = K$.

To train LISTA, we use a dataset of 10,000 samples. Each model is trained for 100 epochs with a batch size of 64 using the ADAM optimizer with default hyperparameters and either an L1 or L2 loss. LISTA uses 10 iterations, while ISTA uses 1,024 iterations. ISTA parameters λ and μ are set by grid search on the training set for optimal reconstruction performance (L2). To calculate knot and decision densities, we generate random paths by sampling points from the training dataset and linearly interpolating between them. Specifically, we construct a path by selecting a sequence of training samples and stepping from one point toward the next using a fixed step size ϵ, generating intermediate points until the target is reached. The process then continues from that point toward the following sample, and so on, until the desired path length is reached. The expectation is computed by MC sampling of 10 paths, each with 2^{16} (65,536) points and a step size of 0.001. Additionally, we estimate the lower bound on the optimal decision density from (16) to be ≈ 0.5 using MC sampling.

Quantitative comparison of ISTA and LISTA

The quantitative results are shown in Figure 2. Several interesting findings emerge. First, at 10 iterations, both LISTA models outperform ISTA in reconstruction performance. While ISTA eventually catches up after 1,024 iterations, this comes at a significantly

[2]The code used in the experiments is available on GitHub: https://github.com/OisinNolan/geometry_of_lista.

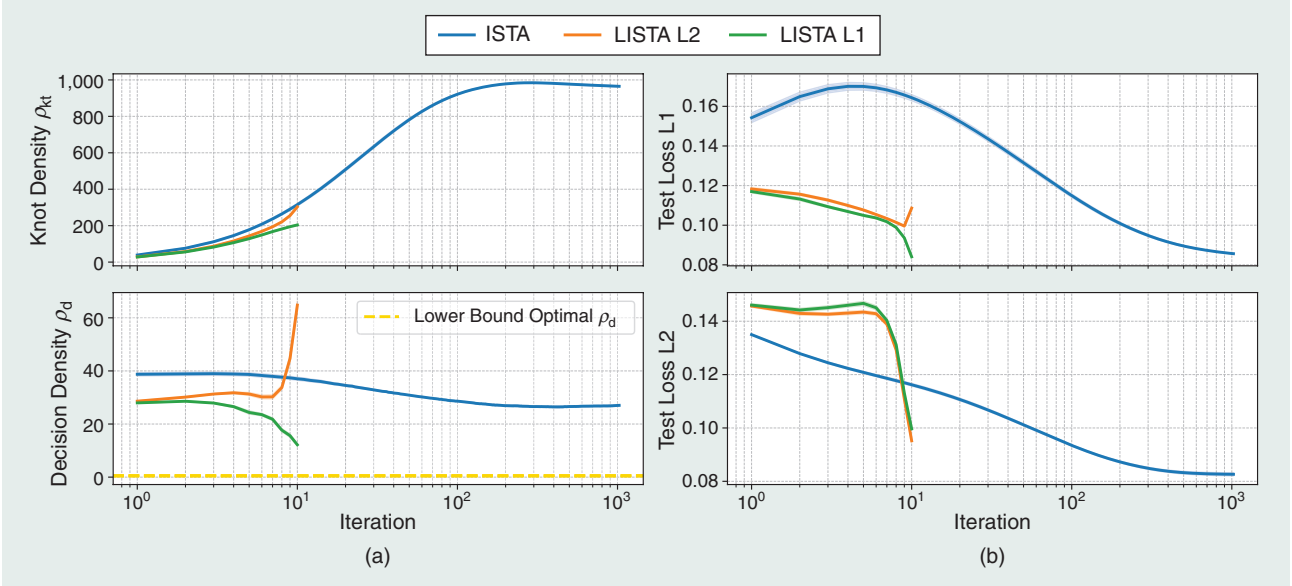

FIGURE 2. (a) and (b) A comparison of knot densities ρ_{kt} (14), decision densities ρ_d (15), and test set reconstruction losses. The knot density for ISTA approaches its numerical maximum under the current sampling resolution (1,000) and is likely even higher in practice. Both LISTA models reach effective solutions in significantly fewer iterations than ISTA. All three models exhibit similar trends in knot density over iterations, with LISTA L1 showing a slightly lower overall density. In contrast, decision densities differ more substantially: LISTA L1 achieves the lowest decision density, while LISTA L2 shows a sharp increase in the final iterations, suggesting that LISTA L1 converges to a more optimal number of decision regions.

higher computational cost. On our machine (NVIDIA GeForce RTX 2080 Ti), LISTA required approximately 4 μs per inference, compared to 0.3 ms for ISTA. This gain comes at the expense of memory; LISTA requires 211 MB to store its weights, whereas ISTA uses only 5 MB. Additionally, because the LISTA models are trained only on their final reconstructions, the intermediate outputs display high loss that decreases sharply toward the final iteration. Third, there is a large gap between knot density and decision density for all models, showing that adjacent linear regions are used to reconstruct the same sparsity, creating arbitrary differences between them. Finally, the decision density of all three models is much larger than the optimal suggested by (16). However, LISTA L1 comes closest to this optimal, while training with an L2 loss seems to have the opposite effect.

Qualitative analysis of geometric behavior

To qualitatively analyze the produced geometry, we show the linear regions along a 2D slice in the input space in Figure 3. The plane is constructed from three anchor points using the procedure detailed in (12) and (13), with the anchor points selected from the training data. Specifically, we use three distinct examples with sparsity level 1. The same slice is used for all models.

The top plots in Figure 3(a) reveal that ISTA initially increases the number of regions ($T = 10$) and then collapses them around the data ($T = 1{,}024$). However, outside the direct paths between the three anchors, ISTA displays a large number of regions. LISTA trained with an L1 loss produces few large regions around the data at $T = 10$ iterations, in contrast to LISTA L2, which creates many small regions everywhere.

Figure 3(b) shows the support of the reconstruction along the 2D slice and the decision boundaries where the support changes. Many Jacobian regions fall within the same support, indicating similar Jacobians. We hypothesize that many regions are needed to create curved decision boundaries, formed by stacking straight boundaries from the soft-thresholding nonlinearity.

Furthermore, Figure 3(b) reveals important differences between the models. First, because ISTA does not use trainable parameters, its decision regions are less curvy compared to LISTA. The use of the same set of parameters for ISTA over all iterations means that to attain decision boundaries of complex geometries, it has to run for many iterations. The LISTA models, on the other hand, achieve much more complicated shapes in a much shorter time window. Second, the plot reveals the dramatic effect of the training loss on the geometry of LISTA. While the model trained using L2 creates a large number of decision regions of high support cardinality, the model trained using L1 creates few decision regions of low support cardinality that coincide with the linear regions found in Figure 3(a). This indicates that LISTA L1 creates a geometry closer to the optimal from (10) and (16). That is, it groups data of the same support into the same, or at least a minimal amount of, linear region(s).

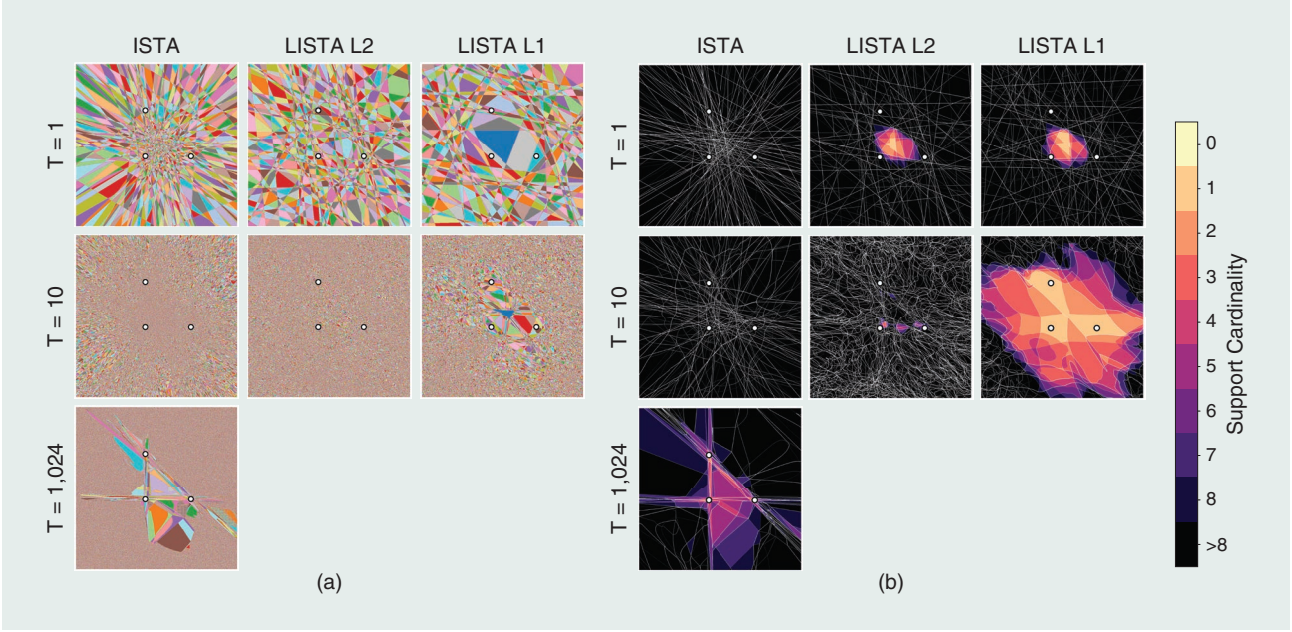

FIGURE 3. The visualization of (a) projection regions and (b) reconstruction sparsity for the three methods along a 2D input slice. The slice is defined by three points in the input space $\mathbb{R}^M$, selected from the training data, which map to orthogonal 1-sparse vectors in the output space $\mathbb{R}^N$. These anchor points are indicated by circular white markers. In (b), color denotes the cardinality of the recovered support (which, under the data distribution, ranges up to 8), while white lines indicate decision boundaries where the support changes. This figure is best viewed digitally and with zoom. ISTA produces a geometry aligned with the directions between the three anchor points. In contrast, LISTA L2 forms many small decision regions, while LISTA L1 yields a simpler structure with larger regions centered around the data.

Conclusion

We investigated the geometric properties of ISTA and LISTA, demonstrating both as CPWL functions and introducing the expected knot and decision densities as practical metrics. Contrary to intuition, our findings indicated that the highest possible number of knots is not optimal. Instead, the optimal MAP solution for sparse reconstruction depends on the dimensions, particularly N and K, and involves one linear region per support. This theoretical optimum leads to a lower bound on the optimal decision density and to the fact that the knot and decision densities should be equal. In other words, a MAP optimal model has no gap between its knot and decision densities, and these densities are equal to (16).

However, ISTA and LISTA exhibit a large gap between knot and decision density, indicating that many CPWL regions are required to accurately describe the decision regions while also revealing suboptimal treatment of signals with the same support. Our experiments show that ISTA generates many superfluous regions, whereas LISTA trained with an L1 loss demonstrates a smaller gap between knot density and decision density as well as a decision density closer to the optimal. The choice of loss norm significantly impacts LISTA's geometry; L1 loss produces fewer and larger support regions, promoting sparsity, whereas L2 loss creates many small regions with high support cardinality. The underlying reasons for the observed gap remain an open question. Future work could explore how different loss functions influence the resulting geometry and how these insights extend to other unrolled architectures, such as alternating direction method of multipliers (ADMM) or half-quadratic splitting (HQS).

In conclusion, both LISTA and ISTA over-parameterize the problem in terms of their geometry, with knot and decision densities much larger than the theoretical least squares optimum. However, LISTA L1 shows the smallest gap between its realized densities and the optimal densities. Our study underscores the advantages of LISTA over ISTA and highlights the substantial influence of the choice of training loss.

Acknowledgment

Hans van Gorp and Oisín Nolan contributed equally to this article.

Authors

Hans van Gorp (h.v.gorp@tue.nl) received his M.Sc. degree in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2020. He is currently working toward his Ph.D. degree at the Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. The pursuit of this degree is done within the Biomedical Diagnostics (BM/d) Research Laboratory of the Eindhoven University of Technology in collaboration with Philips Sleep and Respiratory Care and the Kempenhaeghe Center for Sleep Medicine. His research interests include generative modeling, deep unfolding, and uncertainty analysis for sleep research with a special focus on interpretability and clinical relevance. He is a Graduate Student Member of IEEE.

Oisín Nolan (o.i.nolan@tue.nl) received his M.Sc. degree in artificial intelligence in 2022 from the University of Edinburgh, Edinburgh, United Kingdom. He is currently a Ph.D. candidate at the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. His research interests include ultrasound imaging, generative modeling, active inference, and inverse problems. He is a Graduate Student Member of IEEE.

Yonina C. Eldar (yonina.eldar@weizmann.ac.il) received her Ph.D. degree in electrical engineering and computer science from MIT. She is the Aoun Chair Professor of Electrical and Computer Engineering at Northeastern University, Boston, MA 02115 USA, and the Dorothy and Patrick Gorman Professorial Chair of Mathematics and Computer Science at the Weizmann Institute of Science, Rehovot 7610001, Israel, where she founded and heads the Signal Acquisition Modeling Processing and Learning Lab (SAMPL) and the Center for Biomedical Engineering. She is also a visiting professor at MIT and Princeton, a visiting scientist at the Broad Institute, and an adjunct professor at Duke University and was a visiting professor at Stanford. She is the editor-in-chief of *Foundations and Trends in Signal Processing*. She is a Fellow of IEEE and EURASIP and a member of the Israel Academy of Sciences and Humanities and of the Academia Europaea.

Ruud van Sloun (r.j.g.v.sloun@tue.nl) received his Ph.D. degree (cum laude) in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2018. He is an associate professor at the Department of Electrical Engineering at the Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands. From 2019 to 2020, he was a visiting professor with the Department of Mathematics and Computer Science at the Weizmann Institute of Science, Rehovot, Israel, and from 2020 to 2023, he was a Kickstart AI Fellow at Philips Research. He received an ERC starting grant, an NWO VIDI grant, an NWO Rubicon grant, and a Google Faculty Research Award. His research interests include closed-loop image formation, deep generative learning for signal processing and imaging, active signal acquisition, model-based deep learning, compressed sensing, ultrasound imaging, and probabilistic signal and image reconstruction. He is a Senior Member of IEEE.

References

[1] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004, doi: 10.1002/cpa.20042.

[2] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 399–406.

[3] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021, doi: 10.1109/MSP.2020.3016905.

[4] A. Goujon, A. Etemadi, and M. Unser, "On the number of regions of piecewise linear neural networks," *J. Comput. Appl. Math.*, vol. 441, May 2024, Art. no. 115667, doi: 10.1016/j.cam.2023.115667.

[5] A. I. Humayun, R. Balestriero, G. Balakrishnan, and R. Baraniuk, "SplineCam: Exact visualization and characterization of deep network geometry and decision boundaries," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 3789–3798, doi: 10.1109/CVPR52729.2023.00369.

SP