Real-world examples of Machine Learning

UNIVERSITY OF OXFORD
Mathematical Institute

Lida Kanari

Mathematical Institute
University of Oxford

Introduction to Machine Learning, December 2025

Oxford Mathematics

Table of Contents

The Super Weight in Large Language Models

Mengxia Yu, De Wang, Qi Shan, Colorado J Reed, Alvin Wan (2025)

Link to the paper.

Summary

- This paper studies the importance of the model parameters for the performance of the LLM.
- The authors discover that a tiny set of extremely important scalar weights in LLMs (**super weights**) are important for the performance of the model. Pruning a single super weight can catastrophically harm generation quality.
- They provide a **data-free, single-forward-pass method** to identify super weights and release an index for common open LLMs.
- Super weights cause **super activations** that propagate through layers and are linked to skip connections.
- Practical implication: preserving super weights dramatically improves accuracy of the model.

1) Super Weights

- **Reminder:** For GPT3 we estimated about $175B$ parameters needed to be tunned.
- **Finding:** Models contain a tiny number of weights, "super weights", that are disproportionately important. Pruning one such weight can raise perplexity by orders of magnitude and collapse zero-shot accuracy.

Figure 1 — Qualitative effect of pruning a super weight

Figure: example model output before (left) and after pruning a super weight (right).

- Visual demonstration: pruning a single super weight can produce near-gibberish output.
- Use this to motivate why average metrics hide extreme sensitivities.
- Which downstream modules (Attention head, MLP) could amplify a local perturbation?

1) Super Weights

- **Reminder:** For GPT3 we estimated about $175B$ parameters needed to be tunned.
- **Finding:** Models contain a tiny number of weights, "super weights", that are disproportionately important. Pruning one such weight can raise perplexity by orders of magnitude and collapse zero-shot accuracy.
- **Where they appear:** Frequently in MLP matrices, typically in early layers; numbers are consistent across model family and size.

Quantitative importance (Table 1)

Llama-7B	Arc-c	Arc-e	Hella.	Lamb.	PIQA	SciQ	Wino.	AVG	C4	Wiki-2
Original	41.81	75.29	56.93	73.51	78.67	94.60	70.01	70.11	7.08	5.67
Prune SW	19.80	39.60	30.68	0.52	59.90	39.40	56.12	35.14	763.65	1211.11
Prune Non-SW	41.47	74.83	56.35	69.88	78.51	94.40	69.14	69.22	7.57	6.08
Prune SW, +SA	26.60	54.63	56.93	12.79	67.95	61.70	70.01	50.09	476.23	720.57

Table: performance metrics for original vs prune-SW vs prune-non-SW.

- Table shows that pruning **super weights** severely degrades performance, reducing accuracy.
- Pruning 7000 other weights only marginally affects quality.
- By preserving the activation (SA) but pruning **super weights** only partly improves performance.

2) Identifying Super Weights

- The authors propose a **data-free**, single-forward-pass identification algorithm to locate super weights (no calibration dataset required).
- They provide an index of coordinates for several open LLMs.

Figure — Identification / Index of Super Weights

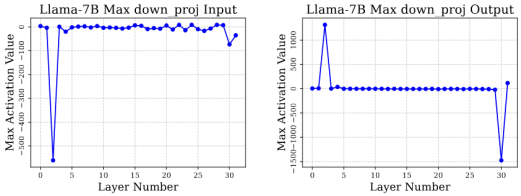


Figure: how to identify the super weights across model layers.

- Shows where super weights occur across layers and model sizes.
- The authors also publish coordinates for many open models — useful for replication and lab assignments.
- Does the distribution suggest a training-time origin or architecture-induced phenomenon?

3) Super Activation Mechanism

- **Observation:** Super weights induce **super activations** — massive activations at a fixed channel/position that persist across layers and prompts.
- **Mechanism:** The super weight amplifies an otherwise ordinary activation into an outlier.
- By pruning **super weights**, the **super activations** disappears. Therefore, SW can be identified by identifying SA in the network.

Figure — impact of superweights on activations

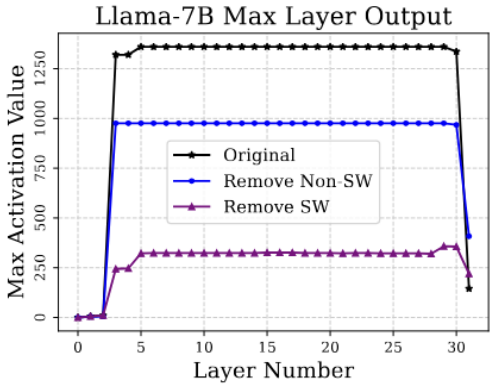


Figure: activations related to super weights.

- Demonstrates that preserving a tiny set of weights yields much better activation outcomes.

Figure — impact of superweights on output probabilities

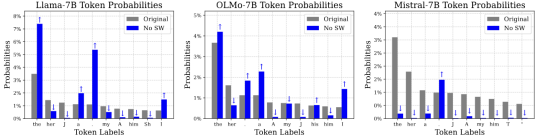


Figure: output tokens related to super weights.

- Super weights affect output token probability distributions.
- Pruning of super weights leads to non-sensical outputs.

Figure — impact of superweights on accuracy

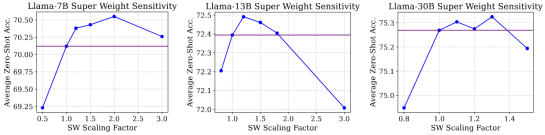


Figure: amplifying super-weights increases accuracy.

- How variations in the magnitude of super weights impact the model's quality?
- They multiply the super weights by a scaling factor ranging from 0.0 to 3.0
- Amplifying super weights can improve model accuracy to some extend.

Breakthroughs

- The paper pinpoints **extreme single-scalar sensitivity** in LLMs and provides a practical, data-free identification method.
- It links weight outliers to activation outliers (mechanistic explanation) and shows how preserving a tiny set of scalars can be more effective than expensive mixed-precision schemes.
- Research impact: invites more fine-grained robustness/sensitivity analyses.

Why this paper is interesting?

- Shows that tiny, unexpected model elements can have large scale effects. What does this tell us about interpretability and robustness in LLMS?
- What is the origin of super-weights?
- What could be the relation to training dynamics?
- Are there ways to train models without such single points of failure?

The art of using t-SNE for single-cell transcriptomics

Dmitry Kobak, Philipp Berens (2019)

Nature Communications

Link to the paper.

Background

- Single-cell transcriptomics yields ever growing data sets containing RNA expression levels for thousands of genes from up to millions of cells.
- Common data analysis pipelines include a dimensionality reduction step for visualising the data in two dimensions, most frequently performed using t-distributed stochastic neighbour embedding (t-SNE).
- Good for visualization, but naive applications often suffer from severe shortcomings, such as the global structure of the data is not represented accurately.
- The authors propose an alternative approach for dimensionality reduction that works well for RNA-seq data sets.

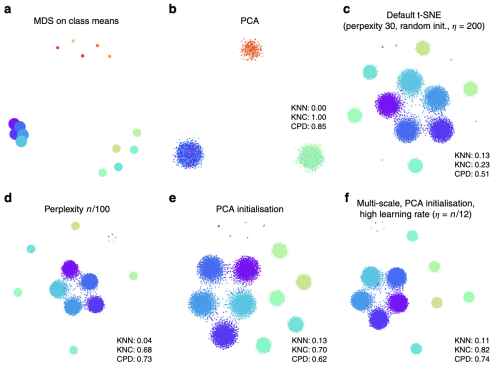
Highlights

- Provides a practical t-SNE pipeline for large single-cell transcriptomic data that better preserves global structure.
- Recommends three main modifications: PCA initialisation, multi-scale similarities, and a larger learning rate.
- Quantifies embedding faithfulness with three metrics (KNN, KNC, CPD) and demonstrates the pipeline on synthetic and several real scRNA-seq data sets.
- Compares to alternatives (e.g., UMAP) and discusses reproducibility, mapping new cells to references, and trade-offs.

1) PCA initialisation

- The authors use the first two principal components (after scaling) to initialise t-SNE rather than random initialisation.
- This approach injects information about global geometry early so t-SNE can focus on local fine structure while preserving large-scale relationships; improves reproducibility (reduces random seed dependence).
- This demonstrates how PCA captures macroscopic structure and how PCA init reduces arbitrary cluster placement on the projection space.

Figure 1 — Synthetic dataset: preserving global geometry



- Default t-SNE places clusters arbitrarily
- Use PCA initialization, multi-scale and higher learning rate recovers meaningful large-scale arrangement.
- Use KNN/KNC/CPD shown in the paper to quantify improvement.
- How does initialisation influence final layout?

2) Multi-scale similarities

- Combine multiple perplexity values (e.g., default 30 and a large perplexity like $n/100$) to capture both local and more global neighbourhoods simultaneously.
- Multi-scale blends both and yields embeddings that balance micro/meso/macro structure.
- Small perplexity favors local detail, large perplexity favors global structure.

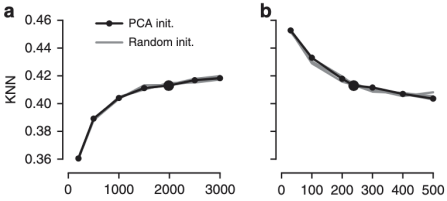
3) Increased learning rate and exaggeration for large data

- They use a larger learning rate (rule of thumb $\eta = n/12$ whenever this exceeds 200) to improve convergence for large n; for very large data use early exaggeration and downsampling-based initialisation.
- The default $\eta = 200$ is often too small for large data sets and may lead to poor optimization or suboptimal local minima.

4) Embedding quality metrics

- **KNN:** fraction of original k-nearest neighbours preserved — measures local/microscopic fidelity.
- **KNC:** fraction of nearest class means preserved — measures mesoscopic structure (cluster relationships).
- **CPD:** Spearman correlation between pairwise distances in high dimensions vs embedding — measures global/macroscopic geometry.

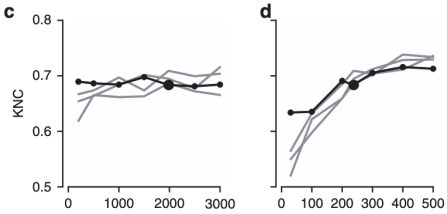
Figure 3 — Metrics and very large dataset strategy



KNN comparisons and example for extremely large datasets.

- Shows numeric comparison of embedding quality across parameter choices and comparisons with UMAP.
- When to prefer t-SNE pipeline vs UMAP (consider tradeoffs: global structure vs local tightness, runtime, reproducibility).

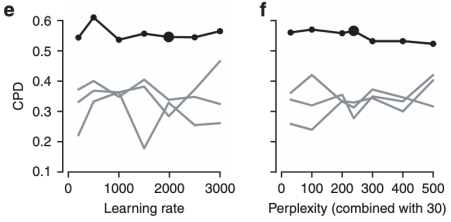
Figure 3 — Metrics and very large dataset strategy



KNC comparisons and example for extremely large datasets.

- Shows numeric comparison of embedding quality across parameter choices and comparisons with UMAP.
- When to prefer t-SNE pipeline vs UMAP (consider tradeoffs: global structure vs local tightness, runtime, reproducibility).

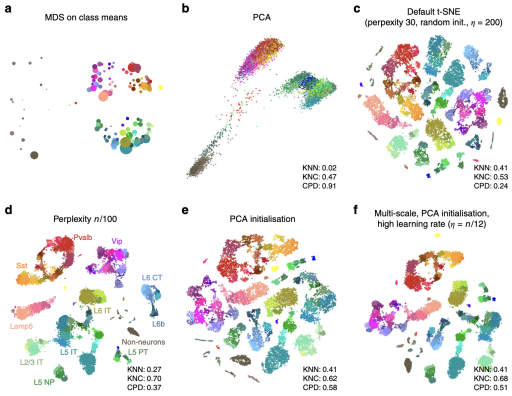
Figure 3 — Metrics and very large dataset strategy



CPD comparisons and example for extremely large datasets.

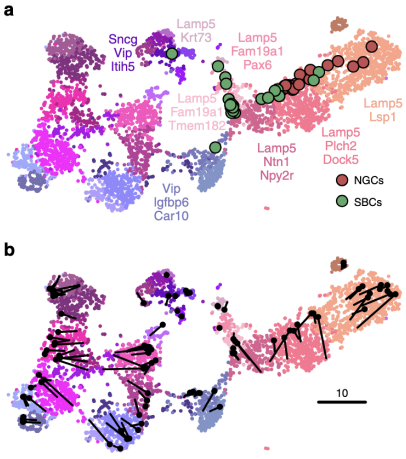
- Shows numeric comparison of embedding quality across parameter choices and comparisons with UMAP.
- When to prefer t-SNE pipeline vs UMAP (consider tradeoffs: global structure vs local tightness, runtime, reproducibility).

Figure 2 — Faithful t-SNE on mouse cortex data



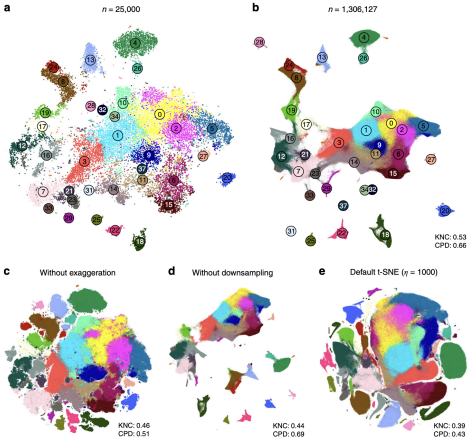
- Improved preservation of hierarchical relationships among 133 clusters when using the recommended pipeline.
- Emphasize reproducibility: PCA init removes random seed variability.
- t-SNE visualisations of the Tasic et al. dataset (23,822 cells). Compare default vs pipeline.

Figure 5 — Improved t-SNE on large scRNA-seq dataset



- Visualization of a large single-cell RNA-sequencing data set using the t-SNE pipeline.
- Pipeline includes PCA initialization, multi-scale similarity kernels, high learning rate.
- Compared to default t-SNE, the embedding better preserves global geometry — hierarchical relationships between major cell classes become interpretable (not random "islands").

Figure 7 — Mapping new cells onto existing t-SNE atlas



- Projection of new single-cell data onto a precomputed reference t-SNE embedding (atlas), enabling consistent cross-experiment visualization.
- Maintains locality and global relations: new cells integrate appropriately into the overall structure without disrupting existing clusters.
- Useful for comparing related datasets or tracking changes over time.

Summary

- The paper provides a practical, reproducible recipe for producing t-SNE visualisations that better reflect global and mesoscopic structure in scRNA-seq data.
- Core recommendations: PCA initialisation, multi-scale similarities, higher learning rate; plus exaggeration/downsampling for very large datasets.
- Quantitative evaluation (KNN/KNC/CPD) supports that the pipeline balances local detail and global fidelity better than naive defaults.

Why this is important?

- **Practical impact:** Simple changes to the t-SNE pipeline yield more faithful visualisations and more reproducible results for scRNA-seq — important when plotting atlases or comparing experiments.
- **Methodological rigor:** Using quantitative metrics rather than relying solely on subjective visual assessment.

Sparse Reduced-Rank Regression for Exploratory Visualisation of Paired Multivariate Data

Dmitry Kobak, Yves Bernaerts, Marissa A. Weis, Federico Scala, Andreas S. Tolias, Philipp Berens (2021)

Link to the paper.

Background

- In genomics, transcriptomics, and related biological fields (collectively known as omics), the combination of experimental techniques can yield multiple sets of features for the same set of biological samples.
- One example is Patch-seq, a method combining single-cell RNA sequencing with electrophysiological recordings from the same cells.
- The authors present a framework based on sparse reduced-rank regression (RRR) for obtaining an interpretable visualisation of the relationship between the transcriptomic and the electrophysiological data.
- Sparse RRR can provides a valuable tool for the exploration and visualisation of paired multivariate datasets.

Highlights

- They propose a sparse reduced-rank regression (sRRR) framework for exploratory visualisation of paired multivariate data (e.g., Patch-seq: gene expression and electrophysiology).
- Use elastic-net regularisation to produce sparse, interpretable low-rank mappings.
- Introduce the **bibiplot** (biplot analog) to visualise relationships.
- Demonstrates that sRRR yields compact, interpretable 2D visualisations that link features in view X (genes) to view Y (electrophysiology), and includes cross-validation to select sparsity/rank.

1) Sparse Reduced-Rank Regression (sRRR)

- **Model:** Multivariate regression $Y = XB + E$ with coefficient matrix B constrained to low rank ($\mathrm{rank}(B) = r$) and sparsity on the factors.

$$\mathcal{L}_{\mathrm{OLS}} = \|\mathbf{Y} - \mathbf{XB}\|^2$$

1) Sparse Reduced-Rank Regression (sRRR)

- **Model:** Multivariate regression $Y = XB + E$ with coefficient matrix B constrained to low rank ($\mathrm{rank}(B) = r$) and sparsity on the factors.

$$\mathcal{L}_{\mathrm{RRR}} = \|\mathbf{Y} - \mathbf{X}\mathbf{W}\mathbf{V}^{\top}\|^2,$$

The product WV^T forms the matrix of regression coefficients that has rank r.

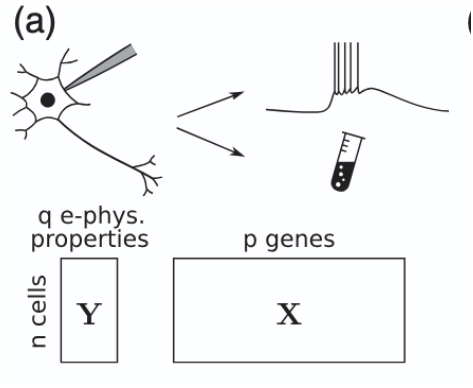
1) Sparse Reduced-Rank Regression (sRRR)

- **Model:** Multivariate regression $Y = XB + E$ with coefficient matrix B constrained to low rank ($\mathrm{rank}(B) = r$) and sparsity on the factors.
- **Why low rank?** Low-rank B captures a small number of latent factors to explain the relation between the two variables.
- **Why sparse?** Elastic-net penalties yield sparsity in the mapping so only a small subset of X features (genes) drive each latent dimension. This approach improves interpretability.

2) Elastic net and model selection

- There are over 20 thousand genes in a mouse genome. The authors use elastic net regularisation, which combines ℓ_1 (lasso) and ℓ_2 (ridge)
- **Elastic net:** Combines ℓ_1 and ℓ_2 penalties to select correlated predictors (genes) while controlling shrinkage.
- This stabilises the selection as opposed to pure Lasso.
- **Selecting rank and sparsity:** Authors use cross-validation to choose rank r and the regularisation parameters, and they demonstrate sensible defaults and diagnostic plots.

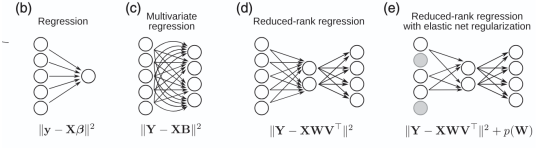
Figure 1 — Method schematic



Schematic of Patch-seq.

- Experimental setup
- Acquiring gene information.
- Acquiring electrical information.
- Combined inputs for each cell.

Figure 1 — Method schematic



Schematic of sRRR (factorisation $B = UV^\top$).

- Inputs for combined features X, y.
- Fit sparse low-rank mapping from X to Y.
- Elastic net regularization.
- Obtain 2D scores for visualization.

3) The bibiplot: interpretable 2-D visualisation

- The authors introduce a **bibiplot** which displays both sample scores (projected rows of X) and sparse loadings (selected gene / e-phys variables) on the same 2D plane.
- Visually links which genes drive which electrophysiological properties and which cells occupy particular regions in the joint space.

Figure 6 — The bibiplot

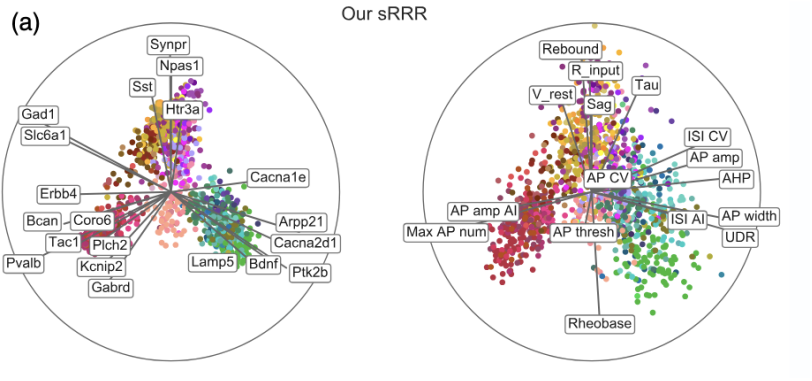


Figure: Bibiplot linking two different features.

4) Application to Patch-seq datasets

- **Main application:** Patch-seq datasets (single cells with both transcriptomic and electrophysiological measurements).
- **sRRR** reveals biologically meaningful gene–ephys associations.
- **Other uses:** Any paired multivariate data (e.g., imaging features versus behaviour, multi-omics).

Figure 2 — Cross-validation and variable selection

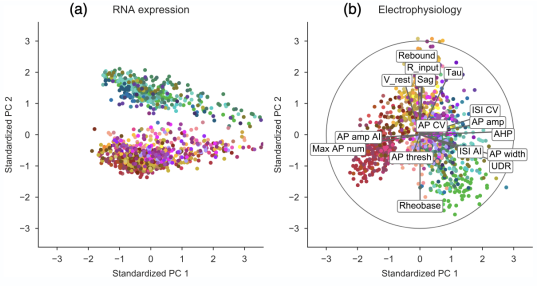


Figure: CV curves and example sparsity paths / chosen genes.

- (a) Principal component analysis (PCA) of the transcriptomic data in the M1 dataset.
- (b) PCA biplot of the electrophysiological data in the same dataset.
- Grey lines show correlations of individual electrophysiological features with PC1 and PC2.

Figure 3 — Patch-seq example: interpretable associations

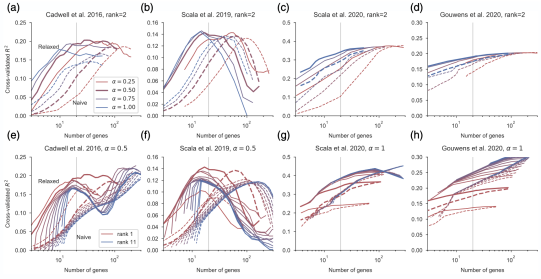
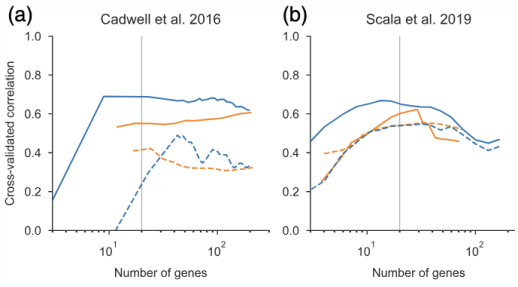


Figure: bibiplot from Patch-seq data showing cell clusters and driving genes.

- Cross- validation performance of sparse RRR with $r = 2$.
- Horizontal axis shows the average number of selected genes obtained for each λ.
- They obtained a strong improvement in predictive performance if, after RRR with elastic net penalty with coefficients λ and α, they take the genes with non-zero coefficients and run RRR again using $\alpha = 0$ (i.e. pure ridge).

Figure 5 — Sparse RRR biplot on Patch-seq data



- Cross-validation estimates of correlations between the transcriptomic and the electrophyiological reduced-rank regression (RRR) components depending on λ.
- Horizontal axis shows the average number of selected genes obtained for each λ.
- Solid blue line: RRR component 1. Dashed blue line: RRR component 2. Orange lines: Witten et al. (2009).

Summary

- sRRR extends reduced-rank regression by adding sparsity to produce interpretable low-rank mappings between paired multivariate views.
- The bibiplot is an interpretable visual tool linking sample projections and selected variables from both views.
- Cross-validation and elastic-net regularisation are central to achieve a practical balance between prediction and interpretability.

Why this is important?

- **Interpretability and visualization:** sRRR directly produces visualizations that are interpretable in terms of a small set of original features — crucial for biological discovery.
- **Practical for multimodal data:** method designed for paired high-dimensional datasets.
- **Bridges prediction and exploration:** sRRR is between purely predictive reduced-rank models but also uses descriptive biplots, offering both predictive power and clarity.

Mathematical Institute

A synaptic learning rule for exploiting nonlinear dendritic computation

Brendan A. Bicknell, Michael Hausser (2021)

Link to the paper.

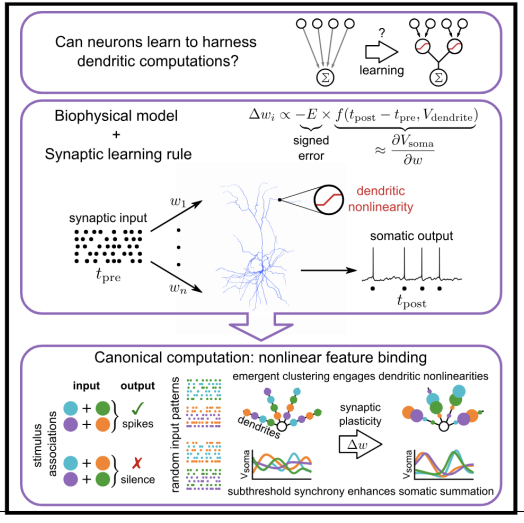
Background

- Information processing in the brain depends on the integration of synaptic input distributed throughout neuronal dendrites.
- Dendritic integration is a hierarchical process, proposed to be equivalent to integration by a multilayer network, potentially endowing single neurons with substantial computational power.
- They develop a learning rule from dendritic cable theory and use it to investigate the processing capacity of a detailed pyramidal neuron model.

Highlights

- Derive a synaptic learning rule based on dendritic cable theory to exploit nonlinear dendritic integration.
- Show a single detailed pyramidal neuron can learn complex spatio-temporal feature-binding tasks.
- Demonstrate synergy between spatial placement of synapses and temporal bursting, thereby enabling nonlinear computations in one neuron.
- Single neurons can learn network-level computations simply by tuning synaptic weights.

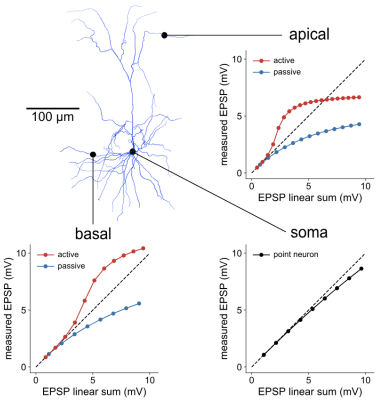
Graphical Abstract



1) Learning rule from dendritic cable theory

- Derive a synaptic update rule that explicitly accounts for dendritic cable properties (spatial attenuation, branch interactions, local nonlinearities).
- Unlike point-neuron rules, this rule lets the neuron tune synapses with awareness of their dendritic location and the local integrative environment.
- Which biological signals (local dendritic voltage, calcium) could plausibly implement the required local information for this rule?

Figure 1 — Model morphology and input patterns

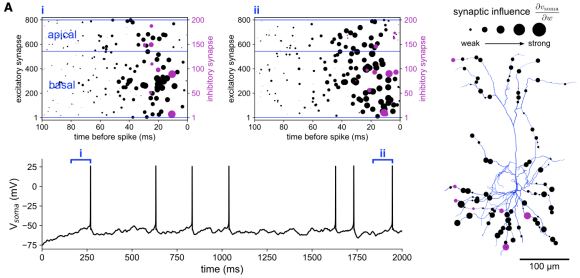


- Detailed neuron morphology and locations of synaptic inputs.
- The simulated response to increasing numbers of excitatory synaptic inputs at the indicated locations, compared with the peak of the linear sum of the same number of unitary EPSPs.
- Captures spatial attenuation, branch isolation, and local nonlinearities.

2) Single neuron learns nonlinear feature binding

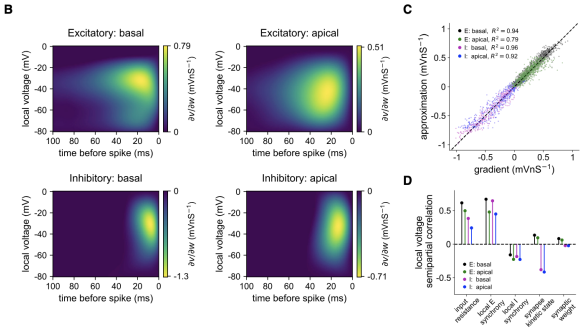
- In a morphologically detailed neuron model, the learning rule produces synaptic configurations that make the neuron selective to specific spatio-temporal input conjunctions.
- The neuron only spikes when inputs occur at particular dendritic regions *and* in particular temporal burst patterns.
- How is this different from a point-neuron trained on the same task?

Figure 2 — Output selectivity across input combinations



- Spike / somatic voltage responses for different spatio-temporal input patterns.
- Example simulation of the active model stimulated with Poisson input into excitatory (black) and inhibitory (magenta) synapses.

Figure 2 — Output selectivity across input combinations

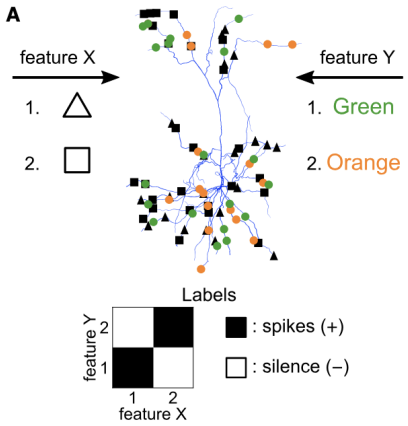


- Spike / somatic voltage responses for different spatio-temporal input patterns.
- The neuron responds only to specific conjunctions — evidence of learned feature binding.

3) Synergy of spatial placement and input timing

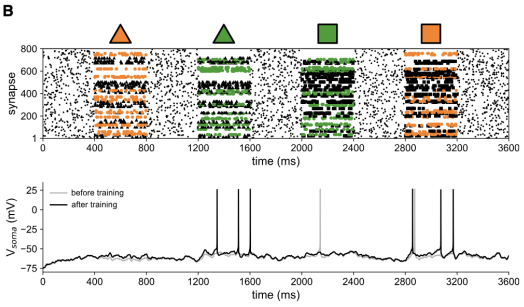
- Spatial distribution of synapses across branches interacts with temporal burst patterns to create nonlinear gating of outputs.
- The neuron implements logical-like conjunctions (e.g., A at branch X **and** B with timing pattern T) rather than simple linear sums.

Figure 3 — Mechanistic view: local dendritic events cause global output



- Local nonlinear events (NMDA spikes / plateaus) amplify selected patterns and propagate to the soma.
- (A) Nonlinear feature-binding problem. Synapses representing different stimulus features were randomly distributed throughout basal and apical dendrites.

Figure 3 — Mechanistic view: local dendritic events cause global output



- (A) Nonlinear feature-binding problem. Synapses representing different stimulus features were randomly distributed throughout basal and apical dendrites.
- (B) Example simulations of a model before (gray) and after (black) training on the task defined in (A)

4) Single neuron as a small multilayer processor

- Because dendritic subunits integrate nonlinearly and pass signals up the tree, a single neuron can implement hierarchical computations similar to a small deep network.
- Some functions attributed to networks of neurons might instead be implemented within single neurons.

Summary

- The authors derive a dendrite-aware synaptic learning rule and show in detailed neuron models that a neuron can learn complex spatio-temporal feature bindings.
- Results demonstrate spatial placement and timing synergy and that single cells can implement nontrivial nonlinear computations.
- Broader claim: biological neurons, thanks to dendrites, are computationally richer than standard point-neuron abstractions.

Why this is important?

- **Rethinking the unit of computation:** single neurons may be much more powerful processors than typically assumed.
- **Biology inspired algorithms:** Dendrite-inspired learning and architectural motifs could inspire new ML primitives or neuromorphic designs that leverage local subunit nonlinearities.
- **Practical:** This example is bridging the properties of computational neuroscience and ML and highlights the cost of oversimplification.

Summary

- Dendritic structure and local nonlinearities show that single neurons can solve complex tasks when guided by appropriate learning rules.
- Critical thinking about modeling choices: when is the point-neuron adequate and when is it insufficient?

Please provide feedback for the course!

Mathematical Institute

MMSC Machine Learning Special Topic Feedback Form

Thank you! Questions?