Task-Optimized Convolutional Recurrent Networks Align with Tactile Processing in the Rodent Brain

Trinity Chung[*,1], **Yuchen Shen**[*,2], **Nathan C. L. Kong**[4], **and Aran Nayebi**[2, 3, 1]

[1]Robotics Institute, Carnegie Mellon University; Pittsburgh, PA 15213
[2]Machine Learning Department, Carnegie Mellon University; Pittsburgh, PA 15213
[3]Neuroscience Institute, Carnegie Mellon University; Pittsburgh, PA 15213
[4]Department of Psychology, University of Pennsylvania; Philadelphia, PA 19104
[*] Equal contribution.
{trinityc, yuchens3, anayebi}@cs.cmu.edu; nclkong@sas.upenn.edu

Abstract

Tactile sensing remains far less understood in neuroscience and less effective in artificial systems compared to more mature modalities such as vision and language. We bridge these gaps by introducing an Encoder-Attender-Decoder (EAD) framework to systematically explore the space of task-optimized temporal neural networks trained on realistic tactile input sequences from a customized rodent whisker-array simulator. We identify convolutional recurrent neural networks (ConvRNNs) as superior encoders to purely feedforward and state-space architectures for tactile categorization. Crucially, these ConvRNN-encoder-based EAD models achieve neural representations closely matching rodent somatosensory cortex, saturating the explainable neural variability and revealing a clear linear relationship between supervised categorization performance and neural alignment. Furthermore, contrastive self-supervised ConvRNN-encoder-based EADs, trained with tactile-specific augmentations, match supervised neural fits, serving as an ethologically-relevant, label-free proxy.

For neuroscience, our findings highlight nonlinear recurrent processing as important for general-purpose tactile representations in somatosensory cortex, providing the first quantitative characterization of the underlying inductive biases in this system. For embodied AI, our results emphasize the importance of recurrent EAD architectures to handle realistic tactile inputs, along with tailored self-supervised learning methods for achieving robust tactile perception with the same type of sensors animals use to sense in unstructured environments.

1 Introduction

Tactile perception plays an essential role in the manipulation and interpretation of complex environments through active sensing [Lederman and Klatzky, 2009]. Animals effectively leverage tactile sensors, such as whiskers, to precisely navigate, forage, and identify objects even in noisy and unstructured conditions [Grant, 2025]; specifically, contact whiskers deliver critical sensory input for navigation, foraging, and hunting in low-light environments [Ahl, 1986], while water-flow whiskers enable seals to discriminate prey movements from general water currents in similar conditions [Dehnhardt et al., 2001], and specialized insect hairs respond to wind stimuli, providing information essential for flight stability and agility [Sterbing-D'Angelo et al., 2011].

Rodents are a common model organism for studying tactile perception in the brain due to the fine experimental control they offer and their ability to discriminate object location, shape, and texture

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

using only their whiskers [Hires et al., 2015, Cheung et al., 2019]. These whiskers, or *vibrissae*, transmit rich mechanical signals to mechanoreceptors at their base, enabling nuanced environmental understanding without direct sensory receptors along their lengths. In fact, rodent whisking behavior is known to parallel how humans touch objects with their fingertips [Staiger and Petersen, 2021], such as in the Lateral Motion Exploratory Procedure [Lederman and Klatzky, 2009]. Despite significant interest in translating such biological capabilities into robots, artificial systems still struggle to match the tactile perceptual prowess of animals, limiting their functionality in realistic, unstructured environments [Navarro-Guerrero et al., 2023].

There are two primary reasons for this gap: one from the robotics hardware side, and the other from the neuroscience side. On the hardware side, current bio-inspired whisker sensors for robots face several critical limitations, including substantial hardware complexity when scaling sensor arrays beyond approximately 18-20 whiskers, each requiring individual transducers, significantly increasing wiring and computational demands [Pearson et al., 2011, Assaf et al., 2016]. These sensors also struggle to accurately discriminate multiple simultaneous stimuli such as airflow, direct contact, and inertia, unlike biological whiskers [Simon et al., 2023]. Additionally, mechanical discrepancies from biological whiskers–such as reduced sensitivity, limited flexibility, and constrained bending angles–impede precise tactile sensing in dynamic environments [Kent et al., 2023, Kent and Bergbreiter, 2024].

These limitations also apply to anthropomorphic robot hands, which remain in a relatively early stage of development. For example, besides not being able to sense temperature, good mechanical skin has remained an open challenge in haptics for approximately four decades, primarily due to difficulties in creating artificial skins that maintain realistic deformation during object contact, with current pneumatic approaches proving inadequate to retain shape [Shimonomura, 2019, Dahiya et al., 2009]. While visuotactile camera-based sensors such as GelSight [Yuan et al., 2017] offer a limited haptic solution by providing high-resolution localized surface deformations, they fundamentally differ from human tactile sensing, which actively integrates diverse mechanoreceptor inputs over larger surfaces and multiple contacts [Ward-Cherrier et al., 2018]. Other issues persist with magnetic [Bhirangi et al., 2021], [Bhirangi et al., 2024a], which have low spatial resolution and is prone to error from electromagnetic interference [Bhirangi et al., 2024b]. These hardware limitations thus make it difficult at the moment to identify robust algorithms for tactile processing that operate on realistic sensory inputs.

On the neuroscience side, despite extensive experimental characterization of rodent somatosensory pathways (e.g. [Armstrong-James et al., 1992, Kerr et al., 2007]), the neural computations underlying precise tactile perception remain poorly understood, primarily due to a scarcity of computational models of this system. Rodent whisker sensing involves hierarchical processing, analogous to vision, where sensory signals propagate from primary neurons in the trigeminal ganglion through parallel thalamic pathways, eventually reaching the primary and secondary somatosensory cortices (S1 and S2) [Bosman et al., 2011, Moore et al., 2015]. However, relatively few computational approaches have explicitly modeled these neural transformations [Zhuang et al., 2017], and none have assessed if any of these models accurately match neural population responses in somatosensory cortex.

To bridge these gaps, we build and systematically evaluate temporal neural networks explicitly trained on biomechanically-realistic force and torque tactile sequences that mice receive, customized from the first complete 3D simulation of the rat whisker array by Zweifel et al. [2021]. We identify convolutional recurrent neural networks (ConvRNNs), particularly IntersectionRNNs, as superior in tactile categorization and neural alignment compared to feedforward (ResNet) and attention-based state-space models (S4 and Mamba). Leveraging supervised and contrastive self-supervised learning adapted specifically for tactile data, we demonstrate a direct linear relationship between tactile categorization performance and neural fit, saturating the currently explainable neural variability and surpassing inter-animal consistency benchmarks, with contrastive self-supervised learning serving as an equally neurally predictive, label-free proxy. Thus, we provide the *first* quantitative characterization of inductive biases required for tactile algorithms to match brain processing.

2 Related Work

Task-optimized neural network models have emerged as currently the most quantitatively accurate framework for understanding brain function. Such goal-driven modeling first successfully explained

neural responses in primate visual areas across hierarchical stages [Yamins et al., Khaligh-Razavi and Kriegeskorte, 2014], and subsequently, auditory [Kell* et al., 2018, Feather et al., 2019], motor [Sussillo et al., Michaels et al.], memory [Nayebi et al., 2021], and language [Schrimpf et al., 2021] brain areas. While many of these results are in humans and non-human primates, goal-driven modeling approaches have more recently been extended to mouse visual cortex, demonstrating that shallow architectures trained via contrastive self-supervised learning best capture mouse visual cortical representations [Nayebi* et al., 2023, Bakhtiari et al., 2021].

Despite these advances, goal-driven computational modeling has not yet been extensively applied to tactile processing, even though tactile sensory systems share hierarchical and recurrent processing features with vision [Felleman and Van Essen, 1991]. Zhuang et al. [2017] provided a foundational goal-driven modeling approach for the rodent whisker-trigeminal system, yet their exploration was limited to relatively simple recurrent architectures and solely supervised categorization loss functions, and they did not compare their models to brain data. Here, we significantly broaden the architectural exploration via Encoder-Attender-Decoder parameterization. We incorporate sophisticated Convolutional RNN ("ConvRNN") architectures previously developed by Nayebi* et al. [2018], which were shown to match primate visual cortex dynamics. Additionally, we explore advanced temporal models such as State Space Models (SSMs) (e.g. S4 [Gu et al., 2021] and Mamba [Gu and Dao, 2023]), and Transformers [Haldar et al., 2024]. We also employ self-supervised loss functions validated in mouse [Nayebi* et al., 2023, Bakhtiari et al., 2021] and primate visual cortex [Zhuang et al., 2021], adapted to specifically deal with force and torque inputs. Finally, we provide the first direct neural comparison to rodent somatosensory cortical responses across all 64 models.

3 Methods

All our code for whisker simulation, model training, and neural data analysis is available on GitHub: `https://github.com/neuroagents-lab/2025-tactile-whisking`

High-variation tactile dataset generation with a biomechanical whisker simulator. Given the ongoing development of tactile sensor hardware to match the flexibility of biological tactile sensors such as fingertips and whiskers, we trained our models on synthetic data, focusing on object categorization and self-supervised learning using a high-variation dataset like ShapeNet [Chang et al., 2015]. This involves the simulated objects interacting with a biomechanically realistic rodent whisker model, WHISKiT Physics [Zweifel et al., 2021], the first 3D simulation of the rodent's complete whisker system, to provide high-dimensional force and torque inputs. Each of the 30 33 whiskers of an average whisker array [Belli et al., 2017, 2018] is modeled as a chain of rigid conical segments interconnected by torsional springs and actuated according to established equations of motion [Knutsen et al., 2008].

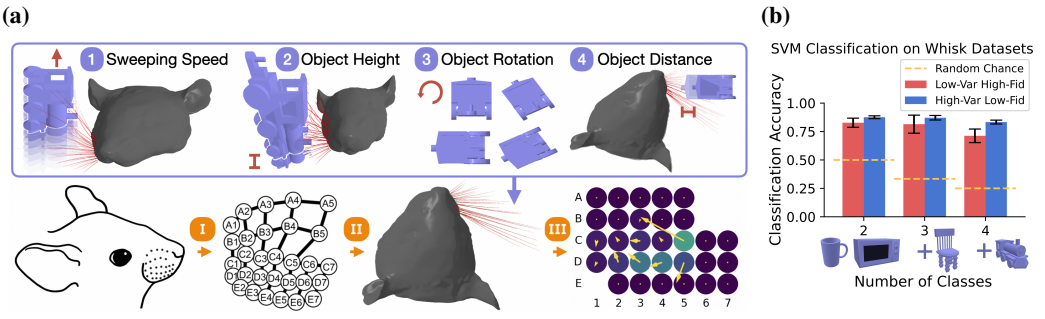

Figure 1: **ShapeNet Whisking Dataset. (a)** (I) With average mouse whisker array measurements from Bresee et al. [2023], (II) objects are whisked in simulation using WHISKiT [Zweifel et al., 2021] resulting in (III) force and torque data for sweeping 9981 ShapeNet objects of 117 categories with various sweep augmentations. The augmentations vary the (1) speed, (2) height, (3), rotation, and (4) distance of the objects relative to the whisker array. We constructed two datasets: a large, low-fidelity set with more sweep augmentations, and a small, high-fidelity set with fewer augmentations (see appendix). **(b)** An SVM classification on up to 4 different classes of objects (cups, microwaves, chairs, and trains) for the 2 datasets show that the classes are distinguishable (above chance).

Since our objective is to compare models with currently available mouse somatosensory data [Rodgers, 2022], we adapted this array to the 30 whiskers of the mouse [Bresee et al., 2023], arranged as a 5×7 array with zero padding. Furthermore, adult mice can exhibit maximal bite forces of approximately 8-10 Newtons [Freeman and Lemen, 2008, Table 1], and since facial tolerance to applied forces would realistically be a fraction of this maximal bite force, our chosen simulation clipping range of ± 1000 milli-Newtons (± 1 N) remains comfortably within biologically plausible limits.

We created a whisking dataset for the shape categorization task, generated by passively brushing objects against the whisker array. Our primary consideration here is to generate a high-variation dataset by which to strongly constrain the representations that are learned, allowing us to more effectively simulate evolutionary pressures, in line with the "Contravariance Principle" [Cao and Yamins, 2024] of goal-driven modeling. On each of the 9981 different objects from ShapeNet, we apply several combinations of sweep augmentations, including adjusting the object sweeping speed, height, rotation, and distance relative to the whisker array (Fig. 1). These augmentations on our passive whisker sweeps can be considered to produce data that is isomorphic to performing active whisking in a systematic way. The 117 object categories are selected based on the work of Zhuang et al. [2017] to ensure reasonably distinguishable classes. We constructed two datasets: one large dataset replicating the Zhuang et al. [2017] data with 288 randomized sweep augmentations and a physics simulation step rate of 110 Hz (High-Variation/Low-Fidelity), and another with 16 sweep augmentations and a higher simulation step rate of 1000 Hz (Low-Variation/High-Fidelity). See the Appendix A1 for detailed sweep augmentation procedures. Although higher temporal resolution is available from the simulation, we extract 22 timesteps for both datasets, corresponding to the average whisking frequency of 20 Hz in rodents [Sofroniew and Svoboda, 2015].

Encoder-Attender-Decoder (EAD) temporal model search parameterization. Tactile recognition is performed by the somatosensory cortex, which, though less understood than vision, exhibits hierarchical processing and long-range feedback connections [Lederman and Klatzky, 2009, Navarro-Guerrero et al., 2023]. These basic anatomical insights motivate our exploration of hierarchical, temporal neural network (TNN) models such as ConvRNNs shown previously to match primate visual cortex dynamics [Nayebi* et al., 2018, Nayebi et al., 2022], SSMs [Gu et al., 2021], Transformers [Haldar et al., 2024], and ResNets [He et al., 2016] as a feedforward control. We provide our library `PyTorchTNN` which enables large-scale exploration of integrating multiple TNN modules that are also combined by long-range feedback with feedforward connections.

These considerations produce a rather large search space of model architectures. Therefore, to be able to effectively search the space of TNN models systematically, we came up with an Encoder-Attender-Decoder (EAD) parameterization, schematized in Fig. 2a. Convolutional recurrent and state-space layers (such as ConvRNNs and S4) are particularly effective for encoding smooth and compressible temporal signals, given their recurrent smoothing properties and linear-time complexity. In contrast, transformers and other attention-like architectures (e.g., Mamba) excel at processing temporally sparse or irregularly informative inputs, as they independently weight each timestep. Given that tactile inputs from force and torque sensors provide temporally smooth signals, encoder **(E)** layers closer to these inputs naturally benefit from convolutional and recurrent mechanisms that integrate information locally over time, effectively reducing redundancy and noise. In contrast, higher-level temporal aggregation **(A)**, which must selectively integrate meaningful signals across longer and potentially irregular timescales, is better served by attention-based architectures like Transformers or Mamba, which can dynamically weigh distinct timesteps based on their informational content. Finally, the decoder **(D)** is either the classification layer for supervised learning, or similarly producing self-supervised features for contrastive learning or autoencoding. The EAD paradigm flexibly combines these complementary modules, enabling an effective and structured exploration of the temporal model space for tactile processing.

Specifically, we explore the following EAD combinations (visualized in Fig. 2a): 1) Encoder: Zhuang's recurrent model [Zhuang et al., 2017], ResNet [He et al., 2016], S4 [Gu et al., 2021]; 2) Attender: Transformer [Haldar et al., 2024] (e.g., GPT), Mamba [Gu and Dao, 2023], None (i.e., attender-free); 3) Decoder: MLP. We further investigate different variants of Zhuang's model by replacing the recurrent layers in the ConvRNNs with UGRNN [Collins et al., 2017], Intersection-RNN [Collins et al., 2017], LSTM [Hochreiter and Schmidhuber, 1997], and GRU [Cho et al., 2014], and we present the update rules for the variants in Appendix A3. In the `PyTorchTNN` library, each time step in ConvRNNs corresponds to a single feedforward layer processing its input and passing it to the next layer, in contrast to treating each entire feedforward pass from input to output as a single

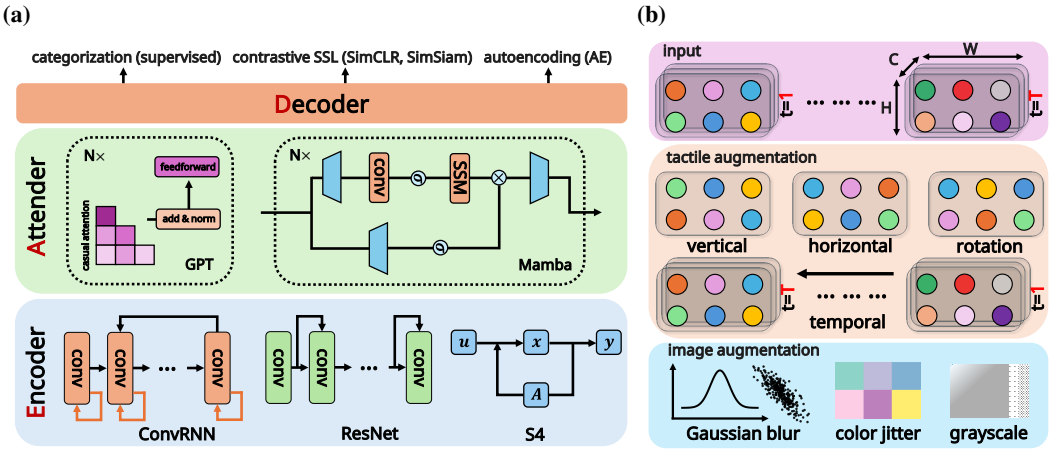

Figure 2: **(a)** **E**ncoder-**A**ttender-**D**ecoder (**EAD**) architecture, with task objectives being supervised categorization, self-supervised learning (SimCLR, SimSiam, autoencoding). The ConvRNN encoder includes **self-recurrence** at each layer where we vary different RNNs.
(b) Types of data augmentations applied to SSL models. Given a temporal **tactile input** over time T, our **tactile augmentation** vertically, horizontally, temporally flips, and rotates the features, while traditional **image augmentation** introduces Gaussian noise, color jitter, and grayscale.

integral time step, as is normally done with RNNs [Spoerer et al., 2017]. This implementation of temporal unrolling parallels biological systems, where stimuli are sequentially processed from one cortical layer to the next.

Model Training and Tactile Augmentations. Besides the supervised categorization objective, we also consider self-supervised learning (SSL) losses: SimCLR [Chen et al., 2020], which learns robust representations via distinguishing the embeddings of augmentations of one image from other images, SimSiam [Chen and He, 2021], which maximizes the similarity between the embeddings of two augmentations of the same image, and autoencoding (AE) [Olshausen and Field, 1996], where we use a 3-layer deconvolution network to reconstruct an image from its sparse latent representation.

We use a train/validation/test split of 80/10/10%, and report the top-5 test accuracy. For supervised learning, we train models for 100 epochs and test on the checkpoint saved with the highest validation accuracy. For SSL objectives, we train for 100 epochs and save the model with the lowest validation loss, then continue training for another 100 epochs with the checkpoint frozen and a learnable linear layer on top of it. To stabilize training, we also consider adding layer norm [Ba et al., 2016] to the ConvRNNs as an alternative when training was unstable. All experiments are conducted on NVIDIA A6000 GPUs, and one set of model search experiment takes 8$\sim$16 hours.

In addition to the various *sweep augmentations* used to generate the initial dataset, we apply tactile augmentations at training time as a cheap way to generate more training data. We illustrate the tactile input, our tactile augmentations, and traditional image augmentations for SSL in Fig. 2b. Given a temporal tactile input over time T, the proposed augmentation either vertically flips, horizontally flips, rotates, or reverses the input over time. Although the variable whisker lengths mean that these operations are not completely physically accurate, our tactile augmentations still meaningfully mimic flipping/rotating the whisker array or reversing the direction of motion. We considered temporal masking as an additional augmentation, but found that this did not significantly improve task performance (Appendix A5, Tab. 7). Applying traditional image augmentations like color jitter or gray scale cause the models to fail to train, which provides evidence that the choice of augmentations should represent operations relevant to the given modality.

Neural Data Comparison. We use the neural dataset from Rodgers [2022], which recorded neural population activity across superficial (L2/3) to deep (L6) layers of the barrel cortex as the mice used their whiskers to perform a simple 2-object (convex/concave) classification task. Note that this low number of distinct stimuli is a critical limitation which we address in the Discussion section. After filtering for valid trials, the neural data we used contains a total of 999 neural units across 11 mice. The time window of each trial is clipped to the time of first whisker contact with the object until the time the mouse makes its decision (about 1-2 seconds). The neural response is binned into intervals

of approximately 45–50 ms, corresponding neatly with the 15–20 Hz whisking frequency typical of rodents (50 ms per cycle at 20 Hz [Sofroniew and Svoboda, 2015]), and aligning specifically with our model's integration scheme of five sub-timesteps (~9 ms each) per timestep. This choice provides a temporal resolution optimal for capturing detailed neural activity during whisker-object contact periods.

We evaluate neural alignment using Representational Similarity Analysis (RSA), a parameter-free approach that compares the pairwise dissimilarity structure of model and neural population responses across the same set of stimuli. RSA compares the pairwise dissimilarity structure of neural responses and model activations to a common set of stimuli, allowing direct comparison of representational geometry without requiring additional parameter fitting [Kriegeskorte et al., 2008].

To ensure only reliable neurons are included, we compute split-half internal consistency of the neural responses across trials and retain only neurons with a Spearman-Brown–corrected reliability greater than 0.5. All RSA evaluations are thus conducted on this filtered subset of self-consistent neurons, and corrected by this internal consistency. Further details about this procedure are included in Appendix A4.

The noise-corrected RSA Pearson's r score is also computed between one mouse to every other mouse, and then averaged this score across all mice to obtain the mean animal score which will serve as the baseline for model-brain evaluation, to account for inherent animal-to-animal variability (which we denote as "a2a" in the barplots). Thus, we want our models to match the brain, at least as well as animals do to each other.

Next, we replicated the experimental setup of the barrel cortex dataset Rodgers [2022] *in silico*. Fig. 4a shows the 6 different stimuli used in the experiment, which are concave/convex objects at three different distances (near, medium, far) from the whisker array. With 3D model reproduction of the concave/convex objects in the experiment, active whisking is simulated under the 6 stimuli using a real recorded whisking trajectory [Bresee et al., 2023] to generate the model input.

We then obtain the neural alignment score for models by computing the maximum, across all layers, of the median RSA score between each layer's representation and the neural responses, averaged across mice. The standard error is calculated across the RSA Pearson's r value from the model to each mouse.

4 Results

Validation of Sensor and Dataset via SVM Decoding. We first verified the basic discriminability of the whisker array dataset using support vector machine (SVM) classifiers on the High-Variation/Low-Fidelity and Low-Variation/High-Fidelity datasets (Fig. 1b). Both showed strong decoding performance, exceeding chance and remaining robust even with increased task difficulty from more object categories. The comparable results across sampling rates confirm that the smaller Low-Var./High-Fid. dataset retains sufficient discriminative tactile information, allowing for efficient model training without compromising performance.

ConvRNN Encoders Outperform Feedforward and Attention-based Architectures for Tactile Categorization. We next systematically evaluated the task performance of the EAD architectures trained on tactile force/torque sequences. Overall, we found that across EAD architectures, the

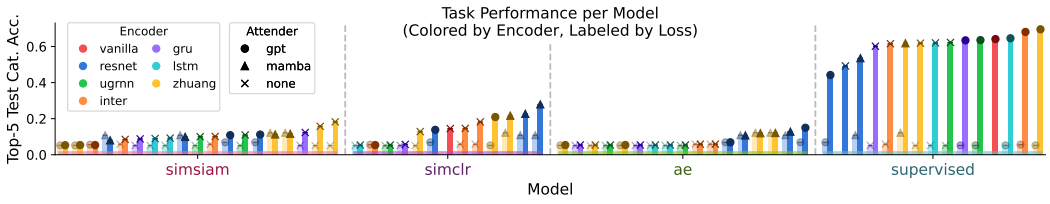

Figure 3: **Tactile Categorization Accuracy.** The lighter-colored left bar represents the randomly initialized version for every model. The best-performing model is Zhuang+GPT+Supervised (rightmost cyan bar). Models with the encoder being S4 are excluded as the training losses explode before the first epoch is finished.

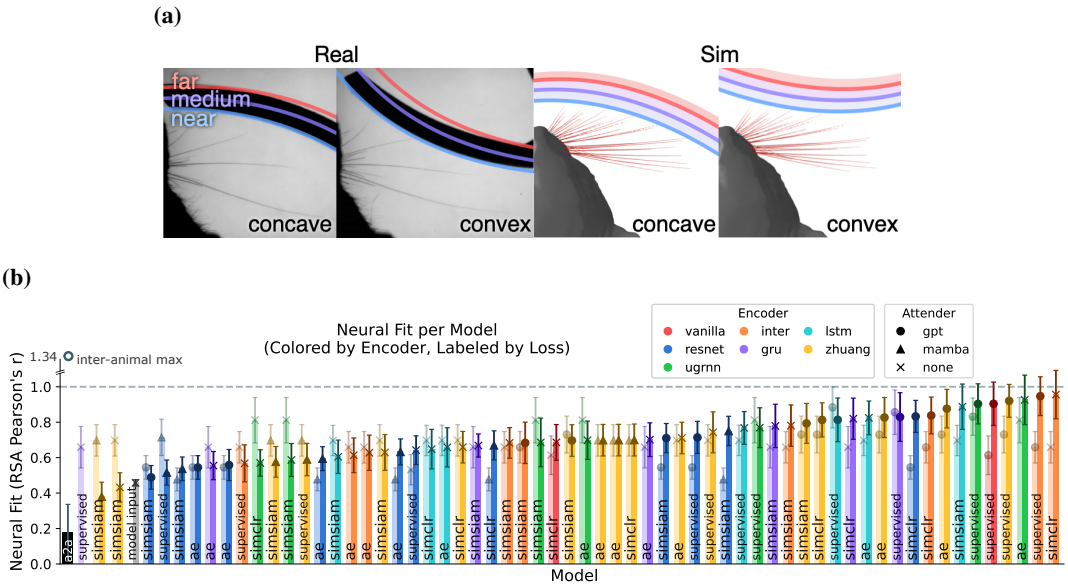

Figure 4: **Model Neural Evaluation.** **(a)** We use six different stimuli (concave/convex $\times$ near/medium/far) replicating the conditions in the mouse neural dataset in simulation. (Real images were taken from video recordings in neural dataset [Rodgers, 2022]). **(b)** Comparison of neural fit (noise-corrected RSA Pearson's r) across models. The mean animal-to-animal score is 0.18 and the maximum between all pairs of animals is 1.34. The leftmost "a2a" bar represents the *mean* animal-to-animal neural consistency score. The lighter-colored left bar represents the randomly initialized version for every model.

choice of encoder (E) was quite important for tactile recognition. Specifically, ConvRNN encoders, especially the IntersectionRNN [Collins et al., 2017], surpassed the purely feedforward ResNet18 and SSM (S4) encoders, in supervised tactile categorization tasks (Fig. 3). Additionally, models trained with our custom force-and-torque-specific contrastive self-supervised learning (SSL) augmentations (Fig. 2b) outperformed untrained networks of the same architecture, and those trained with standard image-based augmentations that involve Gaussian blur and color jittering [Chen et al., 2020] did not train with the best architecture despite hyperparameter tuning, demonstrating the importance of tailored tactile augmentations for enhancing task performance (Fig. 3).

Importance of Architectural Inductive Bias. We note that the raw model input (the tactile force/torque sequences obtained from actively whisking on the 6 stimuli objects) achieves a correlation of 0.46 (Fig. 4b), and untrained randomly initialized models also achieve moderately positive correlation to the mouse neural data. Randomly initialized networks are highly non-random functions as they are architectures selected to perform the task well when trained. Therefore, architecture matters a lot, and this is a well-noted phenomenon in NeuroAI across brain regions [Yamins et al., Nayebi* et al., 2023, Schrimpf et al., 2020]. By doing this comparison, it allows us to isolate the contributions of the architecture vs. loss function for every pair of (architecture, loss function) tuples. The models that saturate our noise ceiling (bars on far right in Fig. 4b) are noticeably improved when trained.

ConvRNN Encoders Saturate Explainable Neural Variance in Rodent Somatosensory Cortex. To assess biological realism, we compared internal model representations to neural recordings from rodent whisker somatosensory cortex. All trained EAD models outperformed raw sensor inputs in neural alignment, underscoring the importance of nonlinear temporal processing in modeling brain-like tactile representations (Fig. 4b). In fact, the best models saturated *all* of the held-out explainable neural variance–*without* fitting any additional parameters–even when tested on entirely novel objects under substantially different experimental conditions (active whisker sensing rather than passive contact, as used in training). Remarkably, the neural predictivity exceeded the average inter-animal neural consistency (leftmost black "a2a" bar in Fig. 4b), thereby robustly passing the NeuroAI Turing Test on this dataset [Feather* et al., 2025]. Among these and consistent with the categorization results, EAD models with ConvRNN encoders consistently provided better neural fits compared to feedforward (ResNet) and SSM-based (S4) encoder architectures, underscoring the biological plausibility of recurrence in modeling tactile processing. In fact, we saw a strong

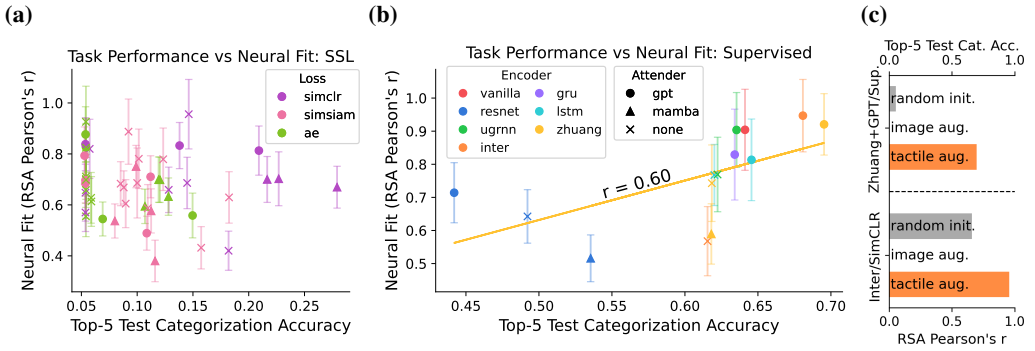

Figure 5: **Comparing Task Performance and Neural Fit. (a)** The task performance of SSL models are about one order of magnitude below the performance of supervised models, yet are able to achieve comparable neural fit. **(b)** For supervised models, we observe a trend of better task performance leading to increased neural correspondence. Plotting a best fit line, we find the correlation $r = 0.59$. **(c)** The tactile augmentations were effective in improving both the neural fit and task performance. The models were unable to be trained with image augmentations.

linear trend between tactile supervised categorization performance and neural fit ($r = 0.59$, Fig. 5b), with the model layers that best predict the tactile neural responses being closest to the decoder layer (Fig. A2). Although the number of task-optimized parameters matters for both categorization test set performance and neural fit generalization (Fig. A1), they are not the whole story, as the SimCLR-trained EAD with the IntersectionRNN encoder best matches the neural data with far fewer parameters ($\sim 3.80 \times 10^7$ parameters) than its supervised counterpart with a GPT-based attender ($\sim 6.38 \times 10^7$ parameters).

GPT-based Attenders Provide Modest Improvements in Task Performance and Neural Alignment. In addition to varying the encoder (E) layer of the EADs, we investigated different temporal aggregation ("Attender") schemas. We observed that GPT-based Attenders modestly outperformed Mamba-based and no-attention controls in both supervised task categorization and neural alignment (Fig. 5b). Although these improvements were subtle, the consistency of the result suggests that incorporating some form of attention downstream of the ConvRNN encoder is beneficial, particularly for our highest-performing neural models. This result suggests the prediction that attention-like mechanisms may be implemented in somatosensory cortex through selective modulation of hierarchical processing pathways, such as those from primary sensory neurons in the trigeminal ganglion through the thalamus and subsequently into primary and secondary somatosensory cortical areas (S1 and S2), which could be validated by experiments involving targeted perturbations or optogenetic manipulation of these specific pathways during tactile discrimination tasks.

Self-Supervised SimCLR Training Matches Supervised Models and Serves as an Ethologically-Relevant Label-Free Proxy. We compared neural alignment achieved by supervised training against SSL methods, adapted with tactile-specific force-and-torque augmentations. This comparison is necessary because discriminating among 117 human-recognizable shape classes is not directly ethologically relevant for rodents. Nevertheless, ShapeNet's extensive diversity of 3D objects provides strong structural constraints for modeling whisker-trigeminal processing at biological scales, constraints which smaller, mouse-relevant object sets might fail to impose. Paralleling findings in vision [Yamins et al., Khaligh-Razavi and Kriegeskorte, 2014], where training on large-scale object categorization leads to generalizable representations beyond specific categories, we similarly suggest that ShapeNet's diversity provides meaningful constraints on network structure irrespective of exact object identity. Additionally, by demonstrating that self-supervised learning (SSL) methods yield neural representations comparable (if not marginally better) to supervised approaches, we directly address an open direction highlighted by Zhuang et al. [2017], who emphasized the need for developing more ethologically relevant yet practically scalable tasks for studying tactile processing.

In fact, we found specifically that *contrastive* SSL loss functions consistently reached neural alignment scores similar to the best supervised models, confirming its utility as an ethologically-relevant but label-free approach to tactile representation learning (Fig. 5a), and outperforming other *non-contrastive* self-supervised approaches such as autoencoding. The SimCLR-optimized Intersection-

RNN EAD achieved the best absolute neural alignment score across all models overall, and was comparable in its neural alignment to its supervised (and more parameter-dense) variant (rightmost green bars in Fig. 4b). This result mirrors findings in primate visual cortex [Zhuang et al., 2021], where contrastive SSL methods equally predicted neural response patterns compared to supervised alternatives, suggesting supervised learning serves as a proxy for this more ethologically-relevant loss function. Interestingly, the parity between contrastive SSL and supervised models observed here differs from mouse visual cortex, where contrastive SSL methods substantially exceeded supervised methods in neural predictivity [Nayebi* et al., 2023]. Notably, the SSL methods, which are best aligned with tactile neural representations overall, achieve only moderate linear probe categorization performance (Fig. 5a), implying that somatosensory cortex representations might prioritize broader, task-agnostic sensory encoding rather than purely specialized, categorization-driven features. Furthermore, just as we found for downstream task performance in Fig. 3, we also observed it was critical to have tactile-specific SSL augmentations for developing biologically accurate tactile representations, compared to both standard image-based SimCLR augmentations and randomly initialized architecture-fixed controls (Fig. 5c). We further provide evidence for the design of our tactile augmentation in Appendix A5.

5 Discussion

Implications for Somatosensory Cortical Processing. We developed a novel Encoder-Attender-Decoder (EAD) parameterization of the space of temporal neural network models (TNN) trained to perform tactile recognition on biomechanically realistic force and torque sequences, greatly extending and answering the open question posed by Zhuang et al. [2017] of characterizing the ethologically relevant constraints of rodent whisker-based tactile computations. Our results establish ConvRNN encoders, particularly the IntersectionRNN, as currently superior in tactile categorization performance and neural representational alignment compared to feedforward (ResNet) and state-space models (SSM), suggesting recurrent processing is more relevant overall to the rodent somatosensory system. Furthermore, we demonstrate that contrastive self-supervised learning (SimCLR), particularly when trained with tactile-specific augmentations, yields neural alignments comparable to supervised methods, highlighting supervised training as a proxy for a more ethologically relevant, label-free representation. The modest yet consistent benefits observed from GPT-based Attenders indicate potential attention-like mechanisms operating within hierarchical tactile processing pathways, suggesting a fruitful avenue for experimental validation.

Taken together, our findings indicate that nonlinear recurrent processing plays an essential role in the rodent somatosensory cortex, reflecting neural encoding strategies that prioritize broad, general-purpose tactile representations. This work provides the first quantitative characterization of the inductive biases required for tactile algorithms to match brain processing, opening new opportunities for deeper insights into sensory representation learning and somatosensory cortical dynamics.

Implications for Embodied AI and Robotics. Current artificial tactile sensors and models fall short of animal-like capabilities, limiting the functional use of robots in real-world, unstructured scenarios. Our results underscore the importance of biomechanically realistic inputs and temporally recurrent architectures for developing tactile perception models that perceive touch similarly as animals do. Our demonstration that tactile-specific (sequential force/torque array) SSL augmentations significantly enhance performance underscores this point, emphasizing the necessity of tailored training methods for robotic tactile systems. Future work in embodied robotic systems that leverage these insights could potentially overcome existing sensor limitations, including scaling complexity, stimulus discrimination challenges, and restricted sensing surfaces, thus achieving more robust and nuanced environmental interactions similar to biological organisms.

Limitations and Future Directions. Our findings do not represent the final word, but rather the beginning of an improved understanding of tactile sensory processing in both animals and machines. Most importantly, currently available tactile neural datasets remain severely limited in stimulus diversity with the number of object conditions tested, restricting the captured neural variability, and could be the reason why inter-animal consistency values are lower for the statistical average between animals than the current pointwise empirical maximum of 1.3 (Fig. 4b), indicated by the NeuroAI Turing Test. Expanding neural datasets to include broader sets of tactile stimuli and additional animals will thus be crucial for future work, enabling models to approach and potentially surpass this theoretical ceiling of neural predictivity.

In the longer term, incorporating multimodal sensory integration will be important–particularly combining tactile inputs with other modalities, such as proprioception or vision, through biologically-inspired fusion operations that integrate signals extensively across intermediate layers, as is observed in the brain [Navarro-Guerrero et al., 2023], rather than relying solely on last-layer concatenation, as is commonly done now [Yang et al., 2024]. Exploring precisely *where* and *how* these multimodal signals are best fused, using diverse fusion operations (e.g., concatenation, attention, routing), is already supported by our `PyTorchTNN` library and will yield stronger constraints on models, potentially enabling more robust tactile-driven decision-making in genuinely unstructured scenarios as well as animals do.

Broader Impacts. Improving robotic tactile sensing to not only match human *capabilities*, but also quantitatively validating, as we explicitly do, shared processing of object properties with human collaborators at the level of *internal* representations, could enhance human safety and efficiency in environments such as healthcare, assistive technologies, and manufacturing in the future. However, advancements in tactile-enabled robotics might also disrupt employment in sectors relying heavily on manual labor. It will therefore be important to thoughtfully manage these technological transitions through proactive policy-making and workforce training.

Acknowledgements

We thank Katerina Fragkiadaki, Albert Gu, and Roberta "Bobby" Klatzky for helpful discussions. A.N. thanks the Burroughs Wellcome Fund and Google Robotics Award for funding.

References

A. S. Ahl. The role of vibrissae in behavior: a status review. *Veterinary research communications*, 10(1): 245–268, 1986.

M. Armstrong-James, K. Fox, and A. Das-Gupta. Flow of excitation within rat barrel cortex on striking a single vibrissa. *Journal of neurophysiology*, 68(4):1345–1358, 1992.

T. Assaf, E. D. Wilson, S. Anderson, P. Dean, J. Porrill, and M. J. Pearson. Visual-tactile sensory map calibration of a biomimetic whiskered robot. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 967–972. IEEE, 2016.

J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization, 2016. URL `https://arxiv.org/abs/1607.06450`.

S. Bakhtiari, P. Mineault, T. Lillicrap, C. Pack, and B. Richards. The functional specialization of visual cortex emerges from training parallel pathways with self-supervised predictive learning. *Advances in Neural Information Processing Systems*, 34:25164–25178, 2021.

H. M. Belli, A. E. Yang, C. S. Bresee, and M. J. Hartmann. Variations in vibrissal geometry across the rat mystacial pad: base diameter, medulla, and taper. *Journal of Neurophysiology*, 117(4):1807–1820, 2017.

H. M. Belli, C. S. Bresee, M. M. Graff, and M. J. Hartmann. Quantifying the three-dimensional facial morphology of the laboratory rat with a focus on the vibrissae. *PLoS One*, 13(4):e0194981, 2018.

R. Bhirangi, T. Hellebrekers, C. Majidi, and A. Gupta. Reskin: versatile, replaceable, lasting tactile skins. *arXiv preprint arXiv:2111.00071*, 2021.

R. Bhirangi, V. Pattabiraman, E. Erciyes, Y. Cao, T. Hellebrekers, and L. Pinto. Anyskin: Plug-and-play skin sensing for robotic touch. *arXiv preprint arXiv:2409.08276*, 2024a.

R. Bhirangi, C. Wang, V. Pattabiraman, C. Majidi, A. Gupta, T. Hellebrekers, and L. Pinto. Hierarchical state space models for continuous sequence-to-sequence modeling. *arXiv preprint arXiv:2402.10211*, 2024b.

L. W. Bosman, A. R. Houweling, C. B. Owens, N. Tanke, O. T. Shevchouk, N. Rahmati, W. H. Teunissen, C. Ju, W. Gong, S. K. Koekkoek, et al. Anatomical pathways involved in generating and sensing rhythmic whisker movements. *Frontiers in integrative neuroscience*, 5:53, 2011.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010.

C. S. Bresee, H. M. Belli, Y. Luo, and M. J. Z. Hartmann. Comparative morphology of the whiskers and faces of mice (mus musculus) and rats (rattus norvegicus). *Journal of Experimental Biology*, 226(19):jeb245597, 10 2023. ISSN 0022-0949. doi: 10.1242/jeb.245597. URL `https://doi.org/10.1242/jeb.245597`.

R. Cao and D. Yamins. Explanatory models in neuroscience, part 2: Functional intelligibility and the contravariance principle. *Cognitive Systems Research*, 85:101200, 2024.

A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

J. Cheung, P. Maire, J. Kim, J. Sy, and S. A. Hires. The sensorimotor basis of whisker-guided anteroposterior object localization in head-fixed mice. *Current Biology*, 29(18):3029–3040, 2019.

K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014. URL `https://arxiv.org/abs/1406.1078`.

J. Collins, J. Sohl-Dickstein, and D. Sussillo. Capacity and trainability in recurrent neural networks. In *ICLR*, 2017.

E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2021.

R. S. Dahiya, G. Metta, M. Valle, and G. Sandini. Tactile sensing—from humans to humanoids. *IEEE transactions on robotics*, 26(1):1–20, 2009.

G. Dehnhardt, B. Mauck, W. Hanke, and H. Bleckmann. Hydrodynamic trail-following in harbor seals (phoca vitulina). *Science*, 293(5527):102–104, 2001.

J. Feather, A. Durango, R. Gonzalez, and J. McDermott. Metamers of neural networks reveal divergence from human perceptual systems. In *Advances in Neural Information Processing Systems*, pages 10078–10089, 2019.

J. Feather*, M. Khosla*, N. Murty*, and A. Nayebi*. Brain-model evaluations need the neuroai turing test. *arXiv preprint arXiv:2502.16238*, 2025.

D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex (New York, NY: 1991)*, 1(1):1–47, 1991.

P. W. Freeman and C. A. Lemen. A simple morphological predictor of bite force in rodents. *Journal of Zoology*, 275(4):418–422, 2008.

R. A. Grant. Can we study whisker movements to gain insights into the natural sensory behaviours of mammals? *The Journal of Physiology*, 2025.

A. Gu and T. Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

A. Gu, K. Goel, and C. Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

S. Haldar, Z. Peng, and L. Pinto. Baku: An efficient transformer for multi-task policy learning. *arXiv preprint arXiv:2406.07539*, 2024.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

S. A. Hires, D. A. Gutnisky, J. Yu, D. H. O'Connor, and K. Svoboda. Low-noise encoding of active touch by layer 4 in the somatosensory cortex. *elife*, 4:e06619, 2015.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

A. J. Kell*, D. L. Yamins*, E. N. Shook, S. V. Norman-Haignere, and J. H. McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

T. A. Kent and S. Bergbreiter. Flow shadowing: A method to detect multiple flow headings using an array of densely packed whisker-inspired sensors. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17843–17849. IEEE, 2024.

T. A. Kent, H. Emnett, M. Babaei, M. J. Hartmann, and S. Bergbreiter. Identifying contact distance uncertainty in whisker sensing with tapered, flexible whiskers. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 607–613. IEEE, 2023.

J. N. Kerr, C. P. De Kock, D. S. Greenberg, R. M. Bruno, B. Sakmann, and F. Helmchen. Spatial organization of neuronal population responses in layer 2/3 of rat barrel cortex. *Journal of neuroscience*, 27(48):13316–13328, 2007.

S.-M. Khaligh-Razavi and N. Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915, 2014.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

P. M. Knutsen, A. Biess, and E. Ahissar. Vibrissal kinematics in 3d: tight coupling of azimuth, elevation, and torsion across different whisking modes. *Neuron*, 59(1):35–42, 2008.

N. Kriegeskorte, M. Mur, and P. A. Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

S. J. Lederman and R. L. Klatzky. Haptic perception: A tutorial. *Attention, Perception, & Psychophysics*, 71(7): 1439–1459, 2009.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization, 2019. URL `https://arxiv.org/abs/1711.05101`.

K. Mamou and F. Ghorbel. A simple and efficient approach for 3d mesh approximate convex decomposition. pages 3501–3504, 11 2009. doi: 10.1109/ICIP.2009.5414068.

J. A. Michaels, S. Schaffelhofer, A. Agudelo-Toro, and H. Scherberger. A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. 117(50):32124–32135.

J. D. Moore, N. Mercer Lindsay, M. Deschênes, and D. Kleinfeld. Vibrissa self-motion and touch are reliably encoded along the same somatosensory pathway from brainstem through thalamus. *PLoS biology*, 13(9): e1002253, 2015.

N. Navarro-Guerrero, S. Toprak, J. Josifovski, and L. Jamone. Visuo-haptic object perception for robots: an overview. *Autonomous Robots*, 47(4):377–403, 2023.

A. Nayebi*, D. Bear*, J. Kubilius*, K. Kar, S. Ganguli, D. Sussillo, J. J. DiCarlo, and D. L. Yamins. Task-driven convolutional recurrent models of the visual system. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

A. Nayebi, A. Attinger, M. Campbell, K. Hardcastle, I. Low, C. Mallory, G. Mel, B. Sorscher, A. Williams, S. Ganguli, L. M. Giocomo, and D. L. Yamins. Explaining heterogeneity in medial entorhinal cortex with task-driven neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

A. Nayebi, J. Sagastuy-Brena, D. M. Bear, K. Kar, J. Kubilius, S. Ganguli, D. Sussillo, J. J. DiCarlo, and D. L. Yamins. Recurrent connections in the primate ventral visual stream mediate a tradeoff between task performance and network size during core object recognition. *Neural Computation*, 34:1652–1675, 2022.

A. Nayebi*, N. C. Kong*, C. Zhuang, J. L. Gardner, A. M. Norcia, and D. L. Yamins. Mouse visual cortex as a limited resource system that self-learns an ecologically-general representation. *PLOS Computational Biology*, 19, 2023.

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

M. J. Pearson, B. Mitchinson, J. C. Sullivan, A. G. Pipe, and T. J. Prescott. Biomimetic vibrissal sensing for robots. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1581):3085–3096, 2011.

C. C. Rodgers. A detailed behavioral, videographic, and neural dataset on object recognition in mice. *Scientific Data*, 9(1):620, 2022.

M. Schrimpf, J. Kubilius, M. J. Lee, N. A. R. Murty, R. Ajemian, and J. J. DiCarlo. Integrative benchmarking to advance neurally mechanistic models of human intelligence. *Neuron*, 108(3):413–423, 2020.

M. Schrimpf, I. A. Blank, G. Tuckute, C. Kauf, E. A. Hosseini, N. Kanwisher, J. B. Tenenbaum, and E. Fedorenko. The neural architecture of language: Integrative modeling converges on predictive processing. *Proceedings of the National Academy of Sciences*, 118(45):e2105646118, 2021.

K. Shimonomura. Tactile image sensors employing camera: A review. *Sensors*, 19(18):3933, 2019.

N. Simon, A. Z. Ren, A. Piqué, D. Snyder, D. Barretto, M. Hultmark, and A. Majumdar. Flowdrone: wind estimation and gust rejection on uavs using fast-response hot-wire flow sensors. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5393–5399. IEEE, 2023.

N. J. Sofroniew and K. Svoboda. Whisking. *Current Biology*, 25(4):R137–R140, 2015.

C. J. Spoerer, P. McClure, and N. Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Front. Psychol.*, 8:1–14, 2017.

J. F. Staiger and C. C. Petersen. Neuronal circuits in barrel cortex for whisker sensory perception. *Physiological reviews*, 101(1):353–415, 2021.

S. Sterbing-D'Angelo, M. Chadha, C. Chiu, B. Falk, W. Xian, J. Barcelo, J. M. Zook, and C. F. Moss. Bat wing sensors support flight control. *Proceedings of the National Academy of Sciences*, 108(27):11291–11296, 2011.

D. Sussillo, M. M. Churchland, M. T. Kaufman, and K. V. Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. 18(7):1025–1033.

B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora. The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies. *Soft robotics*, 5(2): 216–227, 2018.

D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. 111(23):8619–8624.

F. Yang, C. Feng, Z. Chen, H. Park, D. Wang, Y. Dou, Z. Zeng, X. Chen, R. Gangopadhyay, A. Owens, et al. Binding touch to everything: Learning unified multimodal tactile representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26340–26353, 2024.

Y. You, I. Gitman, and B. Ginsburg. Large batch training of convolutional networks.

W. Yuan, S. Dong, and E. H. Adelson. Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762, 2017.

C. Zhuang, J. Kubilius, M. J. Hartmann, and D. L. Yamins. Toward goal-driven neural network models for the rodent whisker-trigeminal system. *Advances in Neural Information Processing Systems*, 30, 2017.

C. Zhuang, S. Yan, A. Nayebi, M. Schrimpf, M. C. Frank, J. J. DiCarlo, and D. L. Yamins. Unsupervised neural network models of the ventral visual stream. *Proceedings of the National Academy of Sciences*, 118(3), 2021.

N. O. Zweifel, N. E. Bush, I. Abraham, T. D. Murphey, and M. J. Hartmann. A dynamical model for generating synthetic data to quantify active tactile sensing behavior in the rat. *Proceedings of the National Academy of Sciences*, 118(27):e2011905118, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to the Abstract, Introduction, and Discussion sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide limitations in the Discussion section (Limitations and Future Directions.)

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: the paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide our open-sourced code for PyTorchTNN and reference other software used in Methods. The details of creating the whisking dataset with exact parameters used is provided in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: As mentioned in Methods, our PyTorchTNN library is open-sourced. The whisking dataset is also provided.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The 80/10/10 data split, epochs, and other training details are provided in Methods.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The process of calculating these values is described Methods. SEM values are shown on all figures where applicable, which is referenced in Results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Methods, we state the GPUs used (Nvidia A6000s) and approximate time to train per model (8$\sim$16 hours).

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We pledge that this research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We mention broader impacts as an explicit subsection in our Discussion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: To the best of our knowledge, the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original creators of ShapeNet dataset, WHISKiT simulator, and others in the Methods section and respect their licenses in our use/adaptation of the data/software.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Setup and usage documentation is provided for PyTorchTNN and whisking dataset.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects were used.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Appendix

Table of Contents

We detail the contents in the appendix below.

A1 Whisk Dataset Generation

Our whisking dataset uses the same 9981 ShapeNet objects and 117 category labels as in Zhuang et al. [2017], but using an improved whisker model and various sweep augmentations, which are listed in Table 1.

	Sim. Freq.	Speed	Height	Rotation	Distance	Size	Total Sweeps
(1)	1000 Hz	30 mm/s	-5, 0 mm	0, 30, 90, 120°	5, 8 mm	40 mm	316,192
(2)	110 Hz	[30∼60]	-3, 0, 3	[0∼359]	5	[20∼60]	2,076,048

Table 1: **Sweep Augmentations** used for the two whisking datasets, which we refer to as (1) Low-Variation High-Fidelity and (2) High-Variation Low-Fidelity. Simulation Frequency refers to the frequency corresponding to the physics timestep used in Bullet physics engine [Coumans and Bai, 2016–2021], the backend for WHISKiT simulator [Zweifel et al., 2021]. For dataset (2), the speed, rotation, and size is each sampled from the range 26 times. The total number of sweeps equals the number of sweep augmentation combinations $\times$ 9981 objects.

WHISKiT Simulator Modifications. We make a few enhancements to the original WHISKiT simulator [Zweifel et al., 2021]:

- Number of whisker links (where whiskers are modeled as a chain of springs [Zhuang et al., 2017]) are dynamically adjusted by the length of the whisker instead of a fixed number.
- Allow for loading objects with or without convex hull. In the whisk datasets, objects are loaded with convex hull (using V-HACD [Mamou and Ghorbel, 2009]) to keep collisions more stable. In generating the 6 simulated stimuli for model input neural evaluation, convex hull was not used in order to preserve the concavity of the "concave" object. The object was geometrically simple enough to not affect collision stability.
- Add camera settings for viewing in orthographic or perspective projection.
- Load multiple mice/rats at a time.

All our code for the simulation, model training, and neural data analysis is available on GitHub: `https://github.com/neuroagents-lab/2025-tactile-whisking`

A2 Model Definitions

We provide the update rules for UGRNN and IntersectionRNN in our experiments as follows. For other RNN variants, please refer to Nayebi et al. [2022].

- x_t^ℓ: input at time t and layer ℓ
- s_t^ℓ: state at time t and layer ℓ
- W^ℓ, U^ℓ: learnable weight matrices at layer ℓ
- b^ℓ: bias terms at layer ℓ
- $\circ$: element-wise multiplication
- σ: sigmoid function
- $\tanh$: hyperbolic tangent function
- ReLU: rectified linear unit function
- $*$: linear transformation or convolution (depending on context)

A2.1 UGRNN

$$c_t^\ell = \tanh\big(W_c^\ell * x_t^\ell + U_c^\ell * s_{t-1}^\ell + b_c^\ell\big) \tag{1}$$
$$g_t^\ell = \sigma\big(W_g^\ell * x_t^\ell + U_g^\ell * s_{t-1}^\ell + b_g^\ell + 1\big) \tag{2}$$
$$s_t^\ell = g_t^\ell \circ s_{t-1}^\ell + \big(1 - g_t^\ell\big) \circ c_t^\ell \tag{3}$$
$$h_t^\ell = s_t^\ell \tag{4}$$

A2.2 IntersectionRNN

$$m_t^\ell = \tanh\big(W_m^\ell * x_t^\ell + U_m^\ell * s_{t-1}^\ell + b_m^\ell\big) \tag{5}$$
$$n_t^\ell = \mathrm{ReLU}\big(W_n^\ell * x_t^\ell + U_n^\ell * s_{t-1}^\ell + b_n^\ell\big) \tag{6}$$
$$p_t^\ell = \sigma\big(W_p^\ell * x_t^\ell + U_p^\ell * s_{t-1}^\ell + b_p^\ell + 1\big) \tag{7}$$
$$y_t^\ell = \sigma\big(W_y^\ell * x_t^\ell + U_y^\ell * s_{t-1}^\ell + b_y^\ell + 1\big) \tag{8}$$
$$s_t^\ell = p_t^\ell \circ s_{t-1}^\ell + \big(1 - p_t^\ell\big) \circ m_t^\ell \tag{9}$$
$$h_t^\ell = y_t^\ell \circ x_t^\ell + \big(1 - y_t^\ell\big) \circ n_t^\ell \tag{10}$$

A3 Model Training

All of our experiments are conducted on NVIDIA A6000 GPUs. For supervised learning, we use a batch size of 256 for all the models and train for 100 epochs. For SSL, during the pre-training stage, we use a batch size of 256 for SimCLR and autoencoding (AE), and a batch size of 1024 for SimSiam, following Nayebi* et al. [2023], and train for 100 epochs. During the linear probing stage, we freeze the checkpoint saved with the lowest validation loss and add a trainable linear layer. We further train such a model with labels for 100 epochs, with a batch size of 256, an initial learning rate of 0.1, with the StepLR scheduler, and the SGD optimizer with momentum [Bottou, 2010].

We detail the optimizers [Bottou, 2010, Kingma and Ba, 2015, Loshchilov and Hutter, 2019, You et al.] and schedulers, and their configurations we used in supervised learning and SSL in Table 2.

Optimizer	**Configuration**
SGD	$\texttt{momentum} = 0.9$, $\texttt{weight-decay} = 10^{-4}$
Adam	$\texttt{weight-decay} = 10^{-4}$
AdamW	$\texttt{weight-decay} = 10^{-4}$
LARS	$\texttt{momentum} = 0.9$, $\texttt{weight-decay} = 10^{-4}$
Scheduler	**Configuration**
StepLR	$\texttt{step\_size} = 30$
ConstantLR	fixed learning rate
CosineLR	$\texttt{warmup\_epoch} = 10$
CosineAnnealing	$\texttt{min\_lr} = 0.0$, $\texttt{warmup\_epoch} = 10$, $\texttt{warmup\_ratio} = 10^{-4}$

Table 2: **Optimizers and Schedulers** with default configurations of supervised learning and SSL when training different model architectures.

For supervised learning, we present the model training configurations in Table 3, where we detail the choices of optimizer, scheduler, learning rate, the encoder and attender of our EAD architecture, and we omit the decoder due to space limits, as it is always a linear layer/MLP. As Zhuang+GPT is the best performing supervised model in terms of classification accuracy, we explore different variants of Zhuang's encoder with attender being GPT.

Encoder	**Attender**	**Optimizer**	**Scheduler**	**Learning Rate**
ResNet	None	SGD	StepLR	$10^{\{-1,-2,-3\}}$
Zhuang	None	SGD	StepLR	$10^{\{-1,-2,-3,-4\}}$
		SGD	ConstantLR	$10^{-2}, 5 \times 10^{-3}$
Zhuang-{UGRNN, IntersectionRNN, GRU, LSTM}	None	{SGD, Adam, AdamW}	StepLR	$10^{\{-1,-2,-3,-4\}}$
		(LayerNorm) - AdamW	StepLR	$10^{\{-1,-2,-3,-4\}}$
ResNet, Zhuang, Zhuang-{UGRNN IntersectionRNN-LN, GRU, LSTM}, S4	GPT	AdamW	CosineLR	10^{-4}
		{SGD, AdamW}	StepLR	$10^{\{-1,-2,-3\}}$
ResNet, Zhuang, S4	Mamba	AdamW	CosineLR	10^{-4}
		{SGD, AdamW}	StepLR	$10^{\{-1,-2,-3\}}$

Table 3: **Model Training Configurations for Supervised Learning**. We use { } to indicate different choices of a specific component (i.e., encoder, optimizer, learning rate) in the search space. We consider adding layer norm as a variant when searching the best configuration for Zhuang's variants (i.e., the second row), which is denoted as "-LN".

For SSL, we follow Nayebi* et al. [2023] and use specific configurations of optimizer and scheduler for different losses, which are shown in Table 4. For SimSiam, we try the CosineAnnealing scheduler both with and without 10 epochs of warmup to search for better model performance. For AE, we use a 3-layer deconvolution network to decode the sparse latent representation from our EAD framework into the original tactile input, regardless of the choice of model architectures.

Loss	Optimizer	Scheduler	Learning Rate
SimCLR	LARS	CosineAnnealing	$10^{\{-1,-2,-3,-4\}}$
SimSiam	SGD	CosineAnnealing (with & w/o warmup)	$10^{\{-1,-2,-3,-4\}}$
AE	SGD	StepLR	$10^{\{-1,-2,-3,-4\}}$

Table 4: **Optimizer, Scheduler, and Learning Rate Configurations for SSL** when training different model architectures.

We present the model architectures explored for SSL in Table 5, where we present the encoder and attender of our EAD architecture, as during the pre-training stage, only the encoder and attender are used, and during the linear probing stage, the decoder is always a one-layer linear classification head.

Encoder	Attender
Zhuang, Zhuang-{UGRNN, IntersectionRNN-LN, GRU, LSTM}	None
ResNet, Zhuang-IntersectionRNN-LN, Zhuang, S4	GPT
ResNet, Zhuang, S4	Mamba

Table 5: **Model Architecture Configurations for SSL**. We use {} to indicate the different variants of encoders. We consider adding layer norm (LN) as a variant when searching the best configuration for Zhuang's variants.

A4 Neural Evaluation

We use the NeuroAI Turing Test [Feather* et al., 2025] to evaluate the neural similarity of mice whisking to models performing tactile categorization.

RSA Correlation. Due to the low number of stimuli, we use RSA [Kriegeskorte et al., 2008] as our correlation metric. RSA is computed over stimuli and neurons, where the average is over source animals/subsampled source neurons, bootstrapped trials, and train/test splits. This yields a vector of these average values, which we can take median and s.e.m. over, across animals.

For the neurons of animal A to animal B in the set of animals $\mathcal{A}$ we estimate the RSA correlation:

$$\left\langle \mathsf{RSA}\left(\mathrm{t}^{\mathsf{A}}, \mathrm{t}^{\mathsf{B}}\right)\right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}} \sim \left\langle \frac{\mathsf{RSA}\left(\mathrm{s}_1^{\mathsf{A}}, \mathrm{s}_2^{\mathsf{B}}\right)}{\sqrt{\widetilde{\mathsf{RSA}}\left(\mathrm{s}_1^{\mathsf{A}}, \mathrm{s}_2^{\mathsf{A}}\right) \times \widetilde{\mathsf{RSA}}\left(\mathrm{s}_1^{\mathsf{B}}, \mathrm{s}_2^{\mathsf{B}}\right)}} \right\rangle_{\mathsf{A}\in\mathcal{A}:(\mathsf{A},\mathsf{B})\in\mathcal{A}\times\mathcal{A}} . \quad (11)$$

Each neuron in our analysis is associated with a value for when it was a target animal (B), averaged over subsampled source neurons and 1000 bootstrapped trials. This yields a vector of these average values, which we can take median and standard error of the mean (s.e.m.) over, as we do with standard explained variance metrics.

Spearman-Brown Correction. The Spearman-Brown correction can be applied to each of the terms in the denominator individually, as they are each correlations of observations from half the trials of the *same* underlying process to itself (unlike the numerator).

$$\begin{aligned}\widetilde{\mathsf{RSA}}\left(X, Y\right) &:= \widetilde{\mathsf{Corr}}(\mathsf{RDM}(x), \mathsf{RDM}(y)) \\ &= \frac{2\,\mathsf{RSA}\left(X, Y\right)}{1 + \mathsf{RSA}\left(X, Y\right)}.\end{aligned}$$

Inter-Animal Consistency. To estimate the inter-animal consistency, we evaluate the pooled animal consistency for each animal. One animal is held out at a time, then compared to the pseudo-population aggregated across units from the remaining animals. We found the mean pooled animal score was 0.175 with a s.e.m. of 0.161 and maximum score of 1.34.

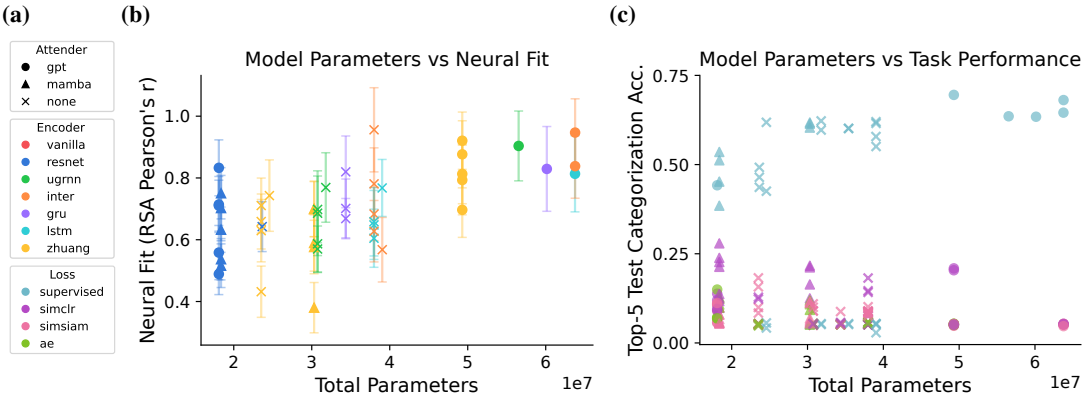

Figure A1: **Model Parameters** compared with categorization performance and neural fit. **(a)** Legends for (b) (Encoder colors) and (c) (Loss colors). **(b)** Models with GPT as an attender have a higher correlation score slightly higher than those without, but high neural fit is still achievable without more parameters as demonstrated by the Inter+SimCLR model (high green "×"). **(c)** Models with GPT as the attender has more parameters and, when trained with supervised learning, tends to have higher top-5 categorization accuracy.

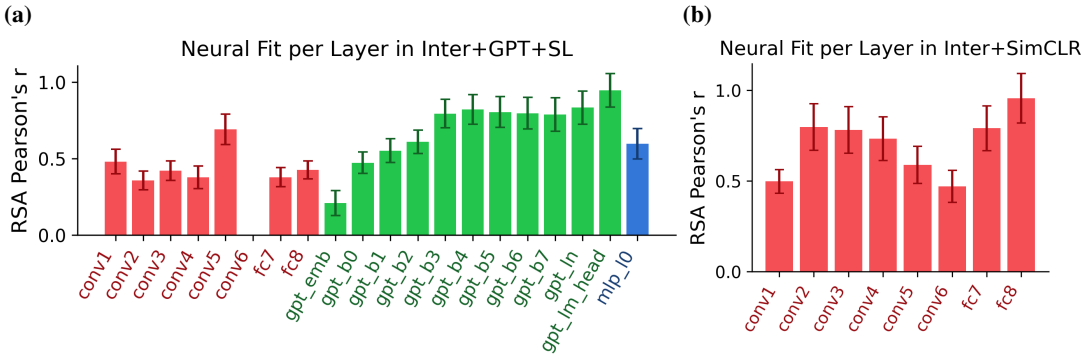

Figure A2: **Neural Fit Score per Model Layers** colored by encoder, attender, and decoder. No bar means the score is NaN for that layer. **(a)** Inter+GPT+SupervisedLearning is the model that scored the highest neural fit out of the supervised models. We observe that later GPT layers perform increasingly better. **(b)** The last fully-connected layer of Inter+SimCLR achieved the highest r value.

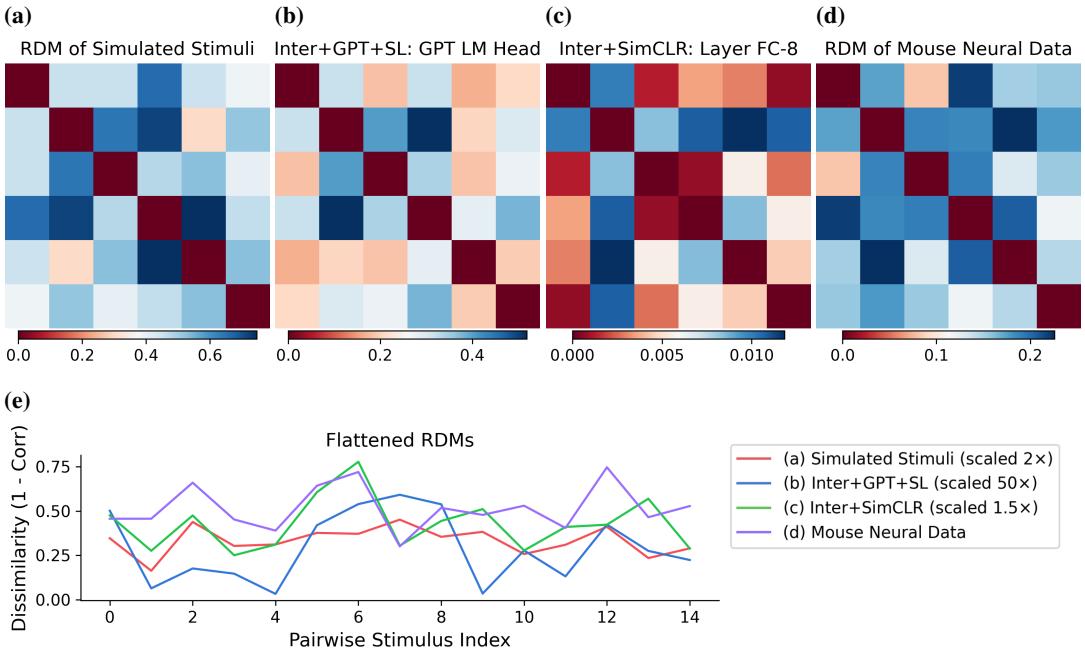

Figure A3: **Representational Dissimilarity Matrices** (RDM) for tactile data and neural evaluation. **(a)** RDM of the 6 simulated stimuli which is used as the model input in neural evaluation. **(b)** The GPT LM Head layer of Inter+GPT+SupervisedLearning is the supervised model with the highest neural fit score. **(c)** The last Fully-Connected layer in Inter+SimCLR achieves the highest neural fit score out of SSL models. **(d)** RDM performed on the 6 stimuli in the mice neural data. **(e)** A visualization of the flattened RDMs, scaled approximately to fit (RSA correlation does not take scale into account).

A5 Additional Experiments

As a basic way to probe the distinguishability of the models' learned representations of stimuli, we performed a decoding test with results shown in Table 6. Note that Inter+SimCLR is the model with the best neural score; Zhuang+GPT+Supervised has the best task score; Inter+GPT+Supervised has best neural score out of supervised models; Resnet+Mamba+SimCLR has best task score out of SSL models.

Rodgers' task variables (convex/concave) are behavioral probes, and are not necessarily the natural computational goals of somatosensory cortex. Our RSA analysis shows that cortical representational geometry is best matched by tactile-optimized models, especially with self-supervised losses, indicating that the cortex builds a general substrate for tactile recognition and discrimination across diverse objects. When reduced to a six-condition probe, this broad representational space may not project well onto the labels, yielding poor decoding despite underlying ethologically aligned representations.

Model	**Shape** (chance=0.5)	**Distance** (chance=0.33)
Inter+SimCLR	0	0
Zhuang+GPT+Supervised	0.33	0.5
Inter+GPT+Supervised	0	0.5
Resnet+Mamba+SimCLR	0	0.67
Animal Neural Data	0.44	0

Table 6: **Stimuli decoding**. For each of the 6 stimuli, we train a logistic regression model on the 5 other stimuli and test on the 1 held-out stimulus to decode either the shape (convex/concave) or the distance (far/medium/near). The stimuli decoding score for Animal Neural Data was obtained by decoding per mouse, then averaging across mice.

We have also tried adding temporal masking, where we randomly mask 75% of the as a tactile augmentation, but it did not significantly improve the task performance or neural fit score (Table 7).

Model	**Top-5 Cat. Accuracy**	**Neural Fit**
Inter+SimCLR	0.15	0.96
Inter+SimCLR **with Temporal Masking**	0.18	0.42
Resnet+Mamba+SimCLR	0.23	0.70
Resnet+Mamba+SimCLR **with T. Masking**	0.28	0.67

Table 7: **Temporal Masking**. Results from retraining two models with temporal masking as an additional tactile augmentation. Top-5 categorization task performance improved slightly (~5%), but neural fit scores decreased.