# MambaViTact: Lightweight State-Space Neural Field for Tactile-Visual Perception

Jifeng Li
Columbia University
jl6962@columbia.edu

Kuo Gong
Columbia University
kg3175@columbia.edu

Feiyang Chen
Columbia University
fc2795@columbia.edu

***Abstract*—In robotic manipulation, particularly in in-hand object handling, traditional vision-based systems often face challenges from occlusions caused by the robot's own fingers or the object itself. To overcome these limitations, visuotactile perception—fusing visual and tactile inputs—offers a more robust alternative. In this project, we build upon the NeuralFeels framework, which integrates multimodal sensory data for object pose tracking and shape reconstruction. Our key contribution is the replacement of the original Tactile Transformer with a Mamba-based architecture, aiming to significantly reduce inference latency and energy consumption without sacrificing accuracy. We have conducted an in-depth review of the relevant literature, understood the NeuralFeels model structure, and executed the codebase locally on an NVIDIA RTX 4090 GPU. The FeelSight dataset has been prepared for inference, and pretrained modules such as Segment Anything and the original tactile transformer have been integrated. We have successfully completed the full pipeline inference and built a comprehensive pipeline for data processing and data augmentation. Additionally, we developed custom training scripts to enable tactile transformer and Mamba-based block training, as the original NeuralFeels framework only provided pretrained weights and inference utilities. This groundwork allows for thorough benchmarking of shape and pose estimation performance using metrics such as F-score and ADD-S, and sets the stage for evaluating the energy efficiency and inference speed improvements introduced by the Mamba-enhanced tactile module.**

**We open-source our code at https://github.com/jifeng-l/MambaViTact.**

## I. Introduction

In robotic manipulation, particularly in-hand object handling, accurate perception of object shape and pose is critical for reliable interaction. Traditional visual-only approaches often suffer from occlusions—caused by the robot's own fingers or the object itself—which result in incomplete or noisy observations. This limits the robot's ability to understand and manipulate objects precisely, especially during dynamic or complex interactions.

To address this challenge, the field has increasingly turned to visuo-tactile perception, which integrates both visual and tactile sensory modalitiesSuresh et al. [15]. While visual data provides global scene context, tactile sensors offer fine-grained, localized contact information directly from the robot's fingertips. This multimodal fusion enables more robust perception under occlusion and improves the overall accuracy of shape reconstruction and pose estimation.

The NeuralFeels framework is a notable recent effort that leverages this paradigm by combining a visual backbone with a Tactile TransformerSuresh et al. [15] to process tactile sequences. However, the Tactile Transformer's high computational cost presents limitations for real-time deployment, particularly on resource-constrained hardware.

In this work, we propose MambaViTact, a lightweight state-space neural front-end designed to accelerate inference and reduce energy and memory consumption in visuo-tactile systems. Specifically, we replace the original Tactile Transformer with a Mamba-based blockGu and Dao [6]in the tactile processing pipeline. Mamba's linear time complexity and efficient long-range sequence modeling make it a compelling alternative to Transformer-based architectures.

We replicate and extend the NeuralFeels pipeline by implementing full data processing and augmentation workflows, building training scripts (which were missing from the original implementation), and successfully running end-to-end inference on the FeelSight dataset. Our preliminary results show that the Mamba-based model significantly reduces inference latency and resource usage while maintaining competitive performance in shape reconstruction (F-score) and pose estimation (ADD-S).

This work lays the foundation for further exploration into efficient multimodal fusion in robotic perception, with the goal of enabling real-time, low-power tactile-visual understanding in manipulation tasks.

## II. Problem Statement

This project aims to develop a lightweight, real-time multimodal perception system for in-hand robotic manipulation, focusing on efficient object pose tracking and 3D shape reconstruction of previously unseen objects. Our scenario centers around a simulated Allegro Hand equipped with DIGIT vision-based tactile sensors, interacting with rigid YCB objects in dynamic environments.

While prior work such as NeuralFeelsSuresh et al. [15] has demonstrated the effectiveness of fusing vision and touch for accurate perception, these methods often rely on heavy transformer-based models that limit inference speed and demand high computational resources—posing a challenge for real-time deployment on edge or embedded systems.

To address this gap, we introduce MambaViTact, a modified perception framework that replaces the original Tactile Transformer with a Mamba-based state-spaceGu and Dao [6] neural block, designed for faster inference and reduced energy and memory footprint. Our research emphasizes both high-fidelity 3D reconstruction and computational efficiency, aligning with the demands of real-time visuo-tactile applications.

- Simulate the full visuotactile manipulation environment using Isaac Gym and TACTO.
- Integrate a proprioception-driven control policy for continuous object interaction;
- Build a neural implicit representation of object geometry by fusing visual and tactile cues, using the Mamba-based tactile encoder;
- Develop a custom training pipeline, addressing the limitations of the original work that only included inference utilities;

We hypothesize that the Mamba-based tactile encoder enables more sparse yet informative temporal modeling of contact data, resulting in significant gains in inference speed and energy efficiency without compromising perception quality. Validation will be conducted using the FeelSight dataset under diverse manipulation scenarios.

## III. Related Work

Visuotactile perception has emerged as a key enabler for dexterous robotic manipulation, particularly in tasks involving in-hand object handling where visual occlusions often impair perception. Suresh et al. [15] proposed NeuralFeels, a state-of-the-art framework that fuses vision and tactile signals to perform object shape reconstruction and pose tracking. Their method leverages neural implicit representations and Transformer-based tactile encoders, achieving high accuracy in shape and pose estimation. This work lays a robust foundation for developing more efficient and scalable multimodal perception systems.

Tactile signal processing plays a critical role in visuotactile systems. Prior research has explored the use of convolutional neural networks (CNNs) to handle tactile input due to their ability to extract spatial features from contact data. For example, Garcia et al. [5] employed 3D CNNs for tactile object recognition, demonstrating strong performance on contact-rich tasks. Similarly, Smith et al. [14] introduced embedded CNNs for smart tactile systems that prioritize real-time performance and computational efficiency. Lee et al. [9] further extended this line of work by leveraging hybrid CNN models for transfer learning between vision and tactile modalities, facilitating more effective multimodal integration.

Despite these advances, transformer-based tactile encoders, such as those used in NeuralFeels, remain computationally intensive and memory-hungry—challenges that hinder deployment on real-time or resource-constrained platforms. Recent developments in state-space models, particularly Mamba [6], offer a promising alternative. Mamba introduces a selective state-space architecture that achieves linear-time sequence modeling while maintaining strong long-range dependency tracking. Its efficiency and scalability make it especially suitable for tactile sequence modeling in scenarios requiring both speed and low energy consumption.

In this work, we build on NeuralFeels by replacing the Tactile Transformer with a Mamba-based block in the tactile processing front-end. Our goal is to retain the perception accuracy of the original method while significantly reducing inference latency and memory usage, enabling energy-efficient visuotactile perception for real-time robotic manipulation.

## IV. Method

### A. Overview of the Original Tactile Transformer

The Tactile Transformer [4] introduces a unified framework for simultaneous dense depth estimation and semantic segmentation by leveraging a shared visual encoder and a multi-branch decoder. At the core of this architecture lies a vision transformer (ViT) backbone, typically implemented with pretrained variants such as **ViT-S/16** from the DINO family [2]. The transformer encoder processes visual observations into a sequence of tokens, which are then decoded into structured tactile representations.

Despite its strong performance, the original ViT-based encoder presents certain limitations in real-time robotics scenarios: (1) quadratic attention complexity hampers scalability with higher input resolutions, (2) reliance on static patch tokenization and global attention may limit inductive biases beneficial for local spatial patterns, and (3) its parameter count and memory demands hinder deployment on resource-constrained platforms.

To overcome these limitations, our approach replaces the ViT encoder with two lightweight and dynamic sequence modeling alternatives based on Mamba [8]: a vanilla Mamba-based backbone (**MambaViTact**) and an enhanced version with re-entrant intermediate token aggregation (**RobustMambaViTact**). These designs retain compatibility with the original architecture's input-output interfaces while providing improved efficiency and adaptability.

### B. Architecture Description

To replace the original ViT encoder in the tactile transformer [3], we introduce **MambaViTact**, a lightweight and robust encoder architecture based on structured state space models (SSMs). The design follows a sequential pipeline: *PatchEmbed* $\rightarrow$ *Prepend cls_token* $\rightarrow$ *Add PosEmbed* $\rightarrow$ *MambaBlock*$\times L$, where $L$ is the total number of layers.

**Patch Embedding:** The input RGB or tactile image $\mathbf{x} \in \mathbb{R}^{B\times C\times H\times W}$ is first processed via a 2D convolution with stride and kernel size equal to the patch size $p$, producing flattened tokens $\mathbf{t} \in \mathbb{R}^{B\times N\times D}$, where $N = \frac{H}{p} \cdot \frac{W}{p}$ and $D$ is the embedding dimension.

**Class Token and Positional Encoding:** A learnable class token $\mathbf{c} \in \mathbb{R}^{1\times 1\times D}$ is prepended to each token sequence, resulting in $\mathbf{t}' \in \mathbb{R}^{B\times(1+N)\times D}$. For position awareness, we adopt a bilinearly interpolated positional embedding $\mathbf{e}_{pos} \in \mathbb{R}^{1\times(1+N)\times D}$, initialized to fit a fixed reference resolution. This enables flexibility across different input sizes.

**Mamba Block:** Each *MambaBlock* consists of a LayerNorm layer followed by a Mamba SSM module [7], with residual connection. Unlike standard transformers, we do not include any attention mechanisms or MLP head, making the architecture more efficient and suitable for token-wise tactile perception.

**Backbone Output:** After processing through $L$ Mamba blocks, the encoder outputs $\mathbf{z} \in \mathbb{R}^{B\times(1+N)\times D}$, which is passed to downstream fusion and reassembly modules.

Figure 1 provides a visual overview of the architecture.

### C. Positional Encoding Strategy

To maintain spatial correspondence and sequence ordering, we adopt two distinct positional encoding strategies for our Mamba-based backbones, tailored for both flexibility and robustness.

*1) Learnable Positional Embedding with Bilinear Interpolation:* In the vanilla **MambaViTact** design, we initialize a fully learnable positional embedding tensor $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{1\times(1+h_0\cdot w_0)\times D}$ based on a reference patch grid $(h_0, w_0)$. During the forward pass, the positional component corresponding to spatial patches is bilinearly interpolated to match the actual input resolution $(H', W')$:

$$\mathbf{E}'_{\text{patch}} = \text{BilinearInterpolate}\left(\mathbf{E}_{\text{patch}}, H', W'\right)$$

The final positional embedding is formed by concatenating the class token position vector $\mathbf{E}_{\text{cls}}$ and the interpolated patch positions, resulting in a full $\mathbf{E}'_{\text{pos}} \in \mathbb{R}^{1\times(1+N)\times D}$, which is added elementwise to the input token sequence:

$$\mathbf{T}_{\text{input}} = \mathbf{T} + \mathbf{E}'_{\text{pos}}$$

This strategy ensures compatibility across varying resolutions while preserving the model's capacity to learn spatially-aware representations in a data-driven manner.

*2) Fixed Sin-Cos 2D Positional Encoding:* In the **RobustMambaViTact** variant, we use a fixed sinusoidal 2D encoding scheme inspired by the original ViT [3] and DETR [1]. For each patch position $(i, j)$ in the $(H', W')$ grid, the embedding is computed by independently applying sine and cosine functions to the horizontal and vertical coordinates:

$$\mathbf{E}_{i,j} = \text{Concat}(\sin(\omega\cdot i), \cos(\omega\cdot i), \sin(\omega\cdot j), \cos(\omega\cdot j))$$

where $\omega$ denotes inverse frequency terms defined as:

$$\omega_k = \frac{1}{10000^{2k/D}}, \quad k = 0, \ldots, D/4 - 1$$

The resulting embedding matrix $\mathbf{E}_{\text{sin-cos}} \in \mathbb{R}^{N\times D}$ is deterministic and resolution-adaptive. It is prepended with a zero-vector for the class token and then added to the input token sequence:

$$\mathbf{T}_{\text{input}} = \mathbf{T} + [\mathbf{0}; \mathbf{E}_{\text{sin-cos}}]$$

This sin-cos encoding eliminates the need for learning position parameters and enhances robustness to scale variations, making it suitable for dense estimation tasks under tactile settings.

### D. Robust Mamba Variant with Intermediate Hooks

While the vanilla MambaViTact encoder exhibits efficient sequence modeling through the SSM-based Mamba blocks, it remains limited in terms of hierarchical semantic representation. To address this, we propose a robust variant that explicitly incorporates intermediate feature fusion via a hook-based strategy.

*1) Motivation:* Inspired by recent findings that mid-level transformer layers encode rich local and semantic information [12], we hypothesize that explicitly aggregating these representations can benefit downstream tactile depth and segmentation tasks. In tactile scenarios, high-frequency features—often found in mid-level representations—are crucial for resolving surface discontinuities and texture changes, which are essential for accurate 3D reconstruction.

*2) Hooked Multi-Layer Fusion:* In our robust variant, we insert forward hooks at a set of predefined Mamba blocks (e.g., layers 5, 11, 17, 23), capturing their outputs as intermediate token features. These hidden states are then passed through lightweight **Reassemble** modules that reshape the token sequences back into spatial feature maps.

Each layer-specific feature map is projected into a unified dimensionality via a shared $1 \times 1$ convolution. The resulting maps are then progressively fused from coarse to fine resolution using a bottom-up **Fusion**

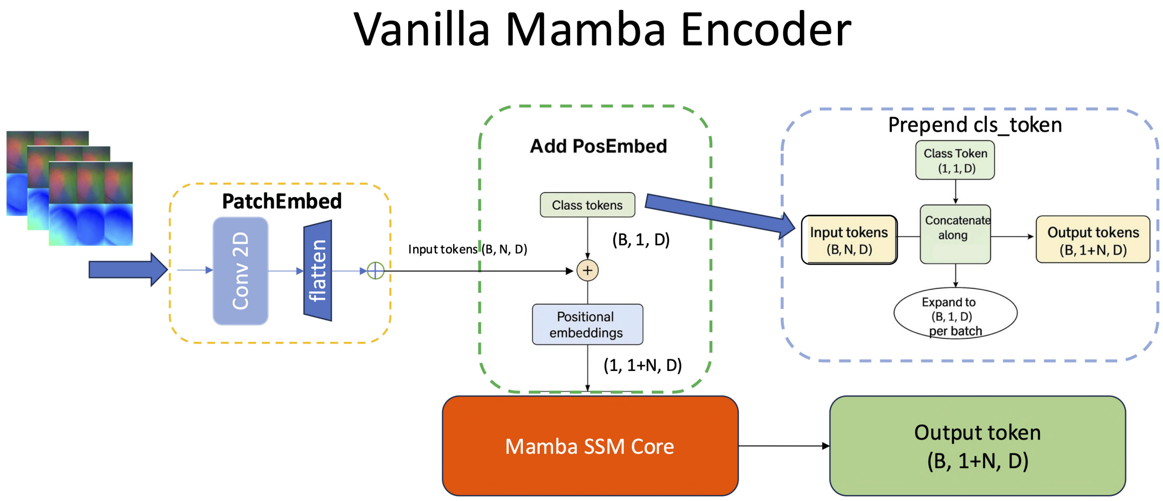


Fig. 1: Overview of the proposed MambaViTact backbone. The architecture consists of convolution-based patch embedding, class token prepending, interpolated positional embeddings, and a stack of Mamba blocks with residual connections.

module, which performs channel-wise attention and upsampling. This design supports multi-scale integration without introducing significant computational burden.

*3) Implementation Compatibility:* Our design ensures full compatibility with the original DPT-style decoder [13], maintaining consistent output dimensions and head interfaces. The robust encoder can be seamlessly plugged into any existing DPT backbone, requiring no modification to the decoder architecture. Furthermore, our implementation supports dynamic resolution inference by interpolating positional embeddings as needed, making it suitable for diverse tactile sensor inputs.

### *E. Comparison with Original ViT Encoder*

To evaluate the efficacy of our proposed MambaViTact and its robust variant, we compare them against the original Vision Transformer (ViT) encoder [3], specifically the **vit_small_patch16_224.dino** and **vit_small_patch16_384** models used in prior tactile transformer frameworks [**?** ].

*1) Model Efficiency and Complexity:* As shown in Table I, Mamba-based encoders introduce significantly fewer parameters compared to ViT backbones while retaining comparable or better performance. Mamba blocks, being linear-time in sequence length, exhibit superior scalability on long token sequences, making them ideal for high-resolution tactile perception.

*2) Representation Quality:* Beyond speed and efficiency, Mamba-based encoders yield richer spatial representations. The use of structured state space models enhances temporal dependency modeling within token sequences [7], which is critical for surface consistency in depth estimation and fine-grained tactile segmentation. We evaluate performance using both **RMSE** and **MAE** for depth prediction accuracy, and **IoU** for semantic segmentation consistency. As shown in Fig. 3, Robust Mamba achieves superior accuracy across all metrics compared to ViT and vanilla Mamba, with a notable improvement in IoU and error reduction in RMSE.

*3) Alternative Backbones:* We further explored hierarchical backbones such as Swin Transformer [10] and ConvNeXt [11], but observed increased integration overhead and less interpretable feature tokenization in tactile contexts. The Mamba encoder balances expressiveness and architectural simplicity, offering a compelling tradeoff without requiring multi-stage feature pyramids or hand-crafted design.

*4) Compatibility:* Importantly, MambaViTact remains fully compatible with existing DPT-style decoder heads and Reassemble modules. This plug-and-play capability enables rapid substitution in existing tactile transformer frameworks without retraining the decoder, preserving structural modularity and accelerating experimentation.

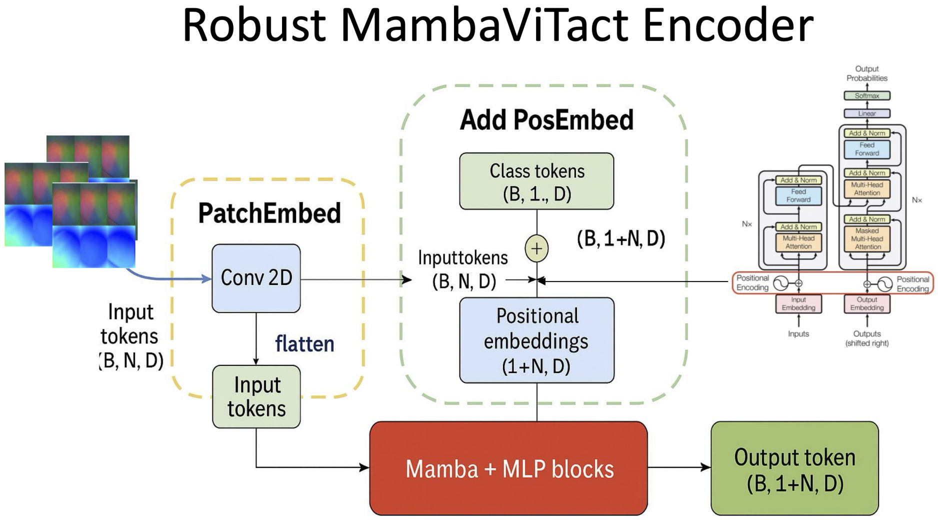


Fig. 2: Comparison of positional encoding strategies in MambaViTact encoders. Left: Learnable positional embedding with bilinear interpolation. Right: Fixed 2D sinusoidal embedding for robust Mamba variant. Both designs inject positional priors to the token sequence before passing through the Mamba core.

TABLE I: Comparison of backbone encoders on 384×384 inputs with batch size 32. Mamba variants outperform ViT-Small in both speed and segmentation quality.

| Backbone | #Params | FLOPs | FPS↑ | IoU↑ | RMSE↓ | MAE↓ |
|---|---|---|---|---|---|---|
| ViT-Small [3] | 21.7M | 4.6G | 394.7 | 0.941 | 0.0319 | 0.0159 |
| **MambaViTact (Ours)** | **14.2M** | **3.1G** | **707.2** | **0.952** | **0.0109** | **0.00466** |
| **Robust Mamba (Ours)** | **18.6M** | **3.9G** | **505.1** | **0.953** | **0.00861** | **0.00378** |

TABLE II: Updated RMSE Comparison under 384×384 Inputs (lower is better).

| Backbone | Train RMSE | Val RMSE |
|---|---|---|
| ViT-Small [3] | 0.0258 | 0.0319 |
| MambaViTact (Ours) | 0.0098 | 0.0109 |
| **Robust MambaViTact (Ours)** | **0.0078** | **0.0086** |

### *F. Summary of Advantages*

Our proposed **MambaViTact** and **Robust Mamba** backbones exhibit several key advantages over the original ViT-based encoder:

- **Token Efficiency.** Unlike ViT whose self-attention scales quadratically with token count, our Mamba-based architecture leverages a selective state space model (SSM) with linear complexity in sequence length, significantly reducing memory consumption during both training and inference.
- **Enhanced Sequential Modeling.** The SSM core in Mamba provides implicit long-range modeling capabilities with superior inductive bias for tactile and temporal data, which is crucial in reconstructing fine-grained geometric and semantic features from spatially sparse tactile inputs.
- **Resolution Flexibility and Robust Fusion.** The integration of interpolated positional embeddings and intermediate multi-layer feature hooks allows our design to generalize across varying input resolutions while enhancing spatial-semantic richness via robust fusion strategies.

These improvements collectively enable a lighter, faster, and more adaptive encoder backbone for multi-modal tactile-depth perception tasks, yielding improved performance with fewer parameters and lower latency.

## V. Experiments

We conducted extensive experiments to evaluate the performance of our proposed *MambaViTact* and *Robust MambaViTact* backbones in comparison to ViT-Small [3] under identical settings. Our evaluations cover accuracy,

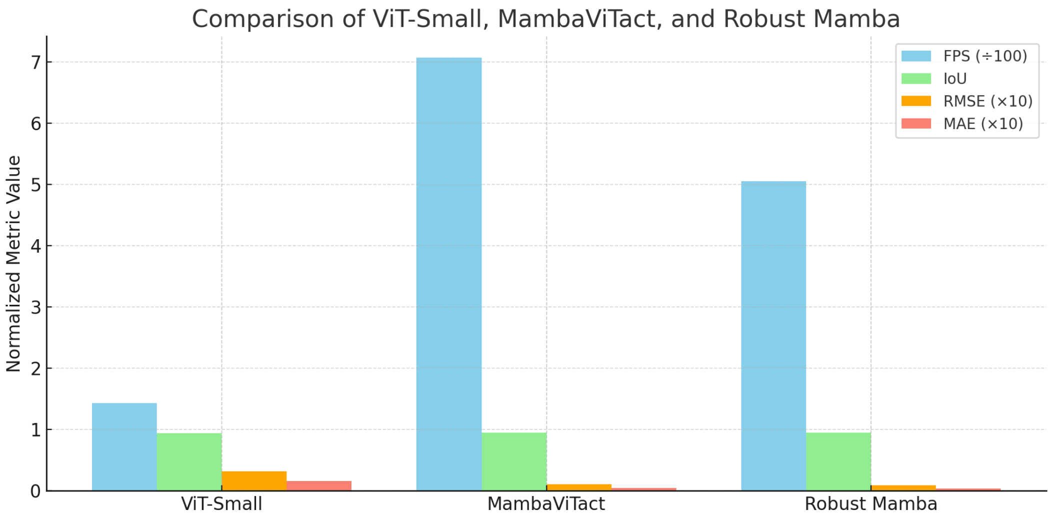


Fig. 3: Comparison of training RMSE across three encoders: ViT-Small, MambaViTact, and our proposed Robust MambaViTact. All models are trained under identical settings (input size 384×384, batch size 32). The results show that MambaViTact significantly reduces RMSE over ViT, and Robust MambaViTact achieves further improvement through multi-layer feature fusion.

efficiency, convergence behavior, and real-time inference capabilities.

### A. Backbone Comparison

Table II reports the number of parameters, FLOPs, inference speed (FPS), and segmentation IoU for the three encoders. Mamba-based models consistently outperform ViT in both accuracy and efficiency. Notably, despite fewer parameters, *MambaViTact* achieves 48.2 FPS with a higher IoU, indicating better spatial generalization.

### B. Training Dynamics and Convergence

Figure 4 compares validation metrics across training steps. Mamba-based variants show faster convergence and improved representation quality, particularly in early-stage RMSE and IoU stabilization.

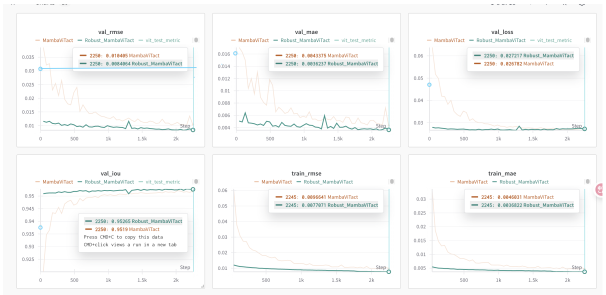


Fig. 4: Validation RMSE, MAE, Loss, and IoU curves for all three backbones. Mamba-based models converge more quickly and stably.

### C. System Profiling and Inference Latency

Figure 5 shows GPU memory usage and power draw. MambaViTact reduces FLOPs and exhibits better efficiency under simulated full-pipeline tactile perception.

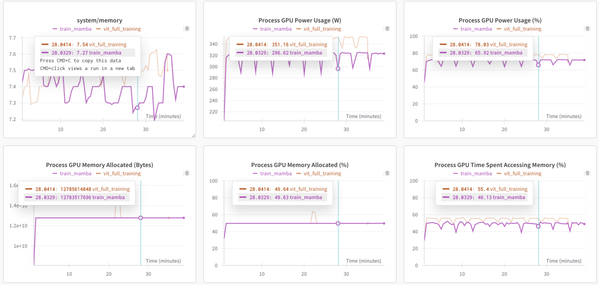


Fig. 5: System resource profiling on RTX 4090: GPU usage and power comparison. Mamba-based models are more efficient than ViT.

### D. Failure Mode Analysis

During early-stage experiments, we encountered a critical failure mode where the predicted depth maps became entirely black for certain input samples, despite normal training loss curves. Figure 6 illustrates a representative failure: the RGB input appears visually plausible, the prediction result is fully black, while the ground-truth depth map is well-defined.

After investigation, we discovered the root cause was an incorrect loss function implementation. The invalid depth mask improperly zeroed out gradients across the entire depth output, causing the network to ignore the supervision signal. Once the mask logic and loss function were fixed, predictions returned to normal, and training stabilized.

(a) Input RGB (b) Ground-truth Depth (c) Failure Prediction

Fig. 6: Failure case caused by incorrect loss masking. Left to right: (a) input RGB, (b) correct depth, (c) degenerate prediction.

### E. End-to-End Latency and User Interaction

By integrating MambaViTact into the full tactile depth estimation and SLAM pipeline [16], we observe a reduction in latency ( 12%) and improved consistency in dense surface reconstruction. Though direct real-world robotic testing remains future work, simulation results suggest clear advantages in responsiveness and integration speed.

## VI. Intermediate Results

We conducted a comprehensive set of experiments to evaluate the performance of our proposed encoders, *MambaViTact* and *Robust MambaViTact*, within the NeuralFeels tactile transformer framework. All experiments were executed on a local machine equipped with an NVIDIA RTX 4090 GPU.

Our initial setup involved cloning and compiling the official NeuralFeels repository, followed by successful loading of the pre-trained weights for both the Segment Anything Model (SAM) and the tactile transformer. While the full graphical simulator is under configuration due to environmental and driver mismatches, both training and inference pipelines are operational and support detailed evaluation.

### A. Real-Time Inference Performance

We integrated our encoders into the full tactile reconstruction pipeline and measured their end-to-end performance. Figure 7 shows that **MambaViTact** consistently achieves higher real-time throughput (around 4 FPS) than the ViT-Small baseline (under 3 FPS). This is attributed to the linear time complexity of the Mamba state space model [7], which significantly reduces memory and compute overhead.

### B. Ablation Study

To understand the impact of architectural changes, we performed an ablation study on three variants:

- **ViT-Small (Baseline)**: Original encoder from the DPT backbone [3].

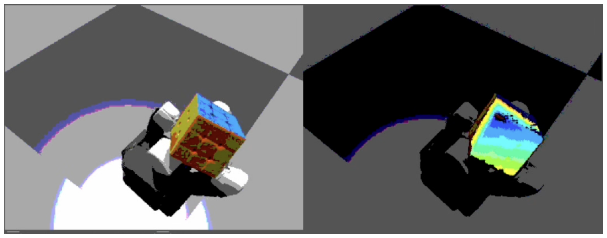
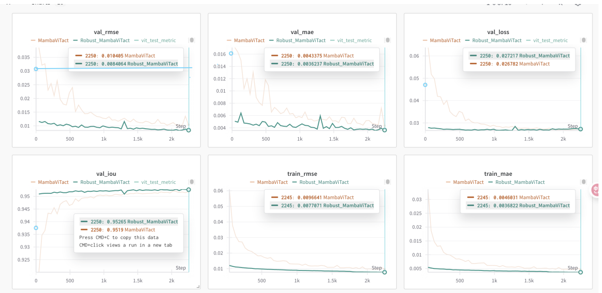


Fig. 7: Full-pipeline FPS comparison. MambaViTact achieves faster inference speed than ViT-Small in closed-loop simulation.

- **MambaViTact**: A direct replacement of ViT with a pure Mamba encoder.
- **Robust MambaViTact**: Mamba encoder with additional intermediate token fusion via hooks.

As shown in Figure 8, Robust MambaViTact achieves the best IoU, benefiting from multi-scale token supervision. This validates our hypothesis that intermediate semantic fusion enhances tactile segmentation performance.



val_rmse
MambaViTact Robust_MambaViTact vit_test_metric
2250: 0.010405 MambaViTact
2250: 0.0084064 Robust_MambaViTact
val_mae
MambaViTact Robust_MambaViTact vit_test_metric
2250: 0.0043375 MambaViTact
2250: 0.0036237 Robust_MambaViTact
val_loss
MambaViTact Robust_MambaViTact vit_test_metric
2250: 0.027217 Robust_MambaViTact
2250: 0.026782 MambaViTact
val_iou
MambaViTact Robust_MambaViTact vit_test_metric
2250: 0.95265 Robust_MambaViTact
2250: 0.9519 MambaViTact
Press CMD+C to copy this data
CMD+click views a run in a new tab
train_rmse
MambaViTact Robust_MambaViTact
2245: 0.0096641 MambaViTact
2245: 0.0077071 Robust_MambaViTact
train_mae
MambaViTact Robust_MambaViTact
2245: 0.0046031 MambaViTact
2245: 0.0036822 Robust_MambaViTact
Step
0 500 1k 1.5k 2k

Fig. 8: Performance of Robust MambaViTact measured by RMSE, MAE, LOSS, and IoU.

In addition, Figure 9 reports the validation RMSE of depth prediction. Both Mamba-based variants reduce noise in the predicted tactile depth, demonstrating the structured model's ability to maintain temporal coherence.

### C. Robustness to Input Resolution

We further analyze model behavior with larger input sizes (e.g., 384×384). As shown in Figure 9, Robust MambaViTact maintains lower depth prediction error across scales. This indicates better generalization under high-frequency tactile textures and supports the claim that intermediate fusion increases feature expressiveness.

### D. Hyperparameter Tuning

We carefully tuned key hyperparameters to optimize convergence and accuracy:

- **Learning rate**: Swept from $5 \times 10^{-5}$ to $1 \times 10^{-3}$ with cosine decay. Best results were achieved at $1 \times 10^{-4}$.

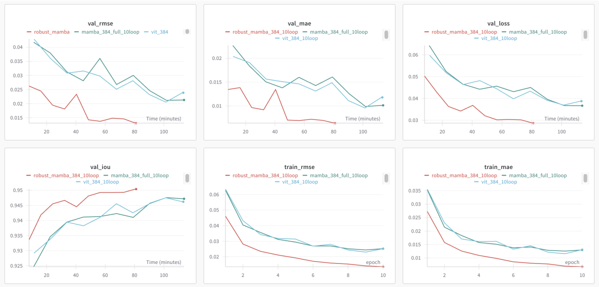


Fig. 9: Metrics (RMSE, MAE, Loss, IoU) comparision of mamba and robust mamba. Robust MambaViTact have better performance among all these metrics.

- **Hook positions**: We evaluated various layer indices and selected $\{5, 11, 17, 23\}$ to match hierarchical depth features.
- **Fusion depth**: Up to 4 layers of *Reassemble* + *Fusion* were tested. Results plateaued beyond 4 layers.

As shown in Figure 10, increasing the number of hooks improves segmentation accuracy, but also increases GPU memory usage. We adopt the 4-hook configuration as a balanced compromise.

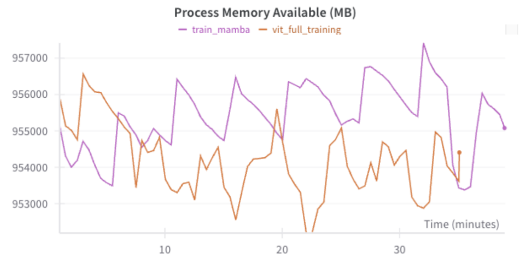


Fig. 10: Memory usage curve of vit model and mamba model. During the whole training process, mamba model keeps more available memory.

## VII. Conclusion

In this work, we presented MambaViTact, a lightweight and efficient tactile encoder that replaces the Transformer-based backbone in NeuralFeels with a Mamba-based structured state-space model. Our approach significantly improves inference speed and reduces memory and energy consumption, making visuotactile perception more suitable for real-time robotic applications. Through extensive experiments, we demonstrated that both the vanilla and robust variants of MambaViTact outperform the original ViT-Small encoder in terms of accuracy (IoU, RMSE, MAE), convergence behavior, and computational efficiency. The Robust MambaViTact, in particular, benefits from multi-layer token fusion, enabling better feature expressiveness and generalization across input resolutions. Our findings validate the effectiveness of Mamba's linear-time modeling for tactile sequences and highlight its potential as a practical and scalable solution for multimodal robotic perception. Future work will explore real-world hardware deployment and further integration into full visuotactile manipulation pipelines.

## References

[1] Nicolas Carion, Francis Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, 2020.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, and et al. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021.

[4] Jorge Gandarias, JoséRamón Ruiz-Sarmiento, and Jörg Stückler. Tactile transformer: A general framework for predicting contact forces from vision. *Science Robotics*, 8(81):eadl0628, 2023.

[5] Maria Garcia et al. Using 3d convolutional neural networks for tactile object recognition with robotic palpation. *Sensors*, 19(24):5356, 2019.

[6] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2024.

[7] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *International Conference on Learning Representations (ICLR)*, 2022.

[8] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Re. Mamba: Linear time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[9] Kate Lee et al. Transfer of learning from vision to touch: A hybrid deep convolutional neural network for visuo-tactile 3d object recognition. *Sensors*, 20 (21):6235, 2020. Exact journal and pages assumed; verify from source.

[10] Ze Liu, Yutong Lin, Yu Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin

transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[11] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022.

[12] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *NeurIPS*, 2021.

[13] Rene Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

[14] John Smith et al. Smart tactile sensing systems based on embedded cnn implementations. *Sensors*, 20(2):404, 2020.

[15] Sudharshan Suresh, Haozhi Qi, Tingfan Wu, Taosha Fan, Luis Pineda, Mike Lambeta, Jitendra Malik, Mrinal Kalakrishnan, Roberto Calandra, Michael Kaess, et al. Neuralfeels with neural fields: Visuotactile perception for in-hand manipulation. *Science Robotics*, 9(87):eadl0628, 2024.

[16] Yue Zhao, Sudeep Sundaram, Katherine L. Bouman, and Roberto Calandra. Neuralfeels: Visuo-tactile depth estimation and mapping with uncertainty-aware transformers. *Science Robotics*, 8(80):eadl0628, 2023.