We use **mechanistic understanding** to compress **proofs of model performance** on toy transformers

Compact Proofs of Model Performance via Mechanistic Interpretability

We **formalize** post-hoc mechanistic interpretability as proving worst-case generalization bounds on the performance of models.

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[f(\mathcal{M}(\mathbf{x}))] \geq b$$

We define a bound approximation algorithm; a proof is the trace of running the algorithm along with an explanation that it provides valid lower bounds on every input.

Algorithm 1 Counting Correct Sequences By Brute Force

```
function CORRECTNESS(M, input-sequence)
    return MODEL-BEHAVIOR(M, input-sequence) == MAX(input-sequence)
end function
function BRUTE-FORCE(d_vocab, n_ctx, M)
    return 1/(d_vocab)^n_ctx SUM(CORRECTNESS(M, tokens) for tokens ∈ (RANGE(d_vocab))^n_ctx)
end function
```

We can run inference on various inputs to guarantee model performance on those inputs. Can mechanistic understanding provide a **compression** of model behavior that beats this inefficient baseline?

We **prototype** the proofs-based approach to formal interpretability on transformer models trained on the the max_of_k task.

We implement many proof strategies, which incorporate varying degrees of **mechanistic detail**...

... and find that more mechanistic detail results in **tighter performance bounds**. Moreover, for any given proof length, proving a tighter performance bound requires more mechanistic detail.

We identify **compounding approximation error** in post-hoc analysis as a key challenge to proving worst-case bounds.

Approximation Strategy	Result	Complexity
(exact) max row diff	≈ 1.8	$(O(d_{\text{vocab}}{}^2 d_{\text{model}}))$
2 · (max abs value)	≈ 2.0	$(O(d_{\text{vocab}}{}^2 d_{\text{model}}))$
max row diff on subproduct	≈ 5.7	$(O(d_{\text{vocab}} d_{\text{model}}{}^2))$
recursive max row diff	≈ 97	$(O(d_{\text{vocab}} d_{\text{model}}))$

Jason Gross, Rajashree Agrawal, Thomas Kwa, Euan Ong, Chun Hei Yip, Alex Gibson, Soufiane Noubir, Lawrence Chan

Finite MLPs can be treated as **analytic approximations** of **infinite width MLPs**

ReLU MLPs Can Compute Numerical Integration Mechanistic Interpretation of a Non-linear Activation

We build upon Nanda 2023 and Zhong 2023 interpretations of the "pizza" modular addition transformer model, which has a **black-box treatment of its MLP**.

While we cannot compactly describe the behavior of 128 MLP neurons individually, we look for continuous functions capturing the **aggregate input behavior**, treating the finite-width MLP as an approximation of some **infinite-width** counterpart.

We apply **amplitude-phase Fourier transforms** to rewrite each neuron's input and output maps. Neurons are **single frequency** with $k_{\text{in}} = k_{\text{out}}$, output map phases are **2×** the input map phase, and the phases are **uniformly distributed**.

$$(A_x)_i \approx a_i \cos(k_i x + \phi_i)$$
$$(B_{xy})_i = \tfrac{1}{2}(A_x + A_y)$$
$$(C_{xy})_i \approx \mathrm{ReLU}((B_{xy})_i)$$
$$D_{xyz} \approx \textstyle\sum_i (C_{xy})_i \cos(k_i z + \psi_i) d_i$$

$$D_{xyz} \approx \sum_i \left|\cos\left(\tfrac{k_i}{2}(x-y)\right)\right| \underbrace{\mathrm{ReLU}\left[\cos\left(\tfrac{k_i}{2}(x+y)+\phi_i\right)\right]}_{g(x+y,i)} \underbrace{\cos(z+2\phi_i)}_{h(z,i)} \underbrace{|a_i| d_i}_{w_i}$$

We can sort the neurons by phase and plot one rectangle for each neuron. Given the input $x + y$, the contributions of neurons to the $z = x + y$ logit look remarkably like **numerical integration**. We plot here $x + y = z = 0$.

$$\int_{-\pi}^{\pi} \underbrace{\mathrm{ReLU}(\cos(k(x+y)/2+\phi))\cos(kz+2\phi)}_{f(\phi)}\,\mathrm{d}\phi = \tfrac{2}{3}\cos(k(x+y-z))$$

We confirm this interpretation by using it to **compactly bound error** in the network approximation.

$$\left|\int_{-\pi}^{\pi} f(x) - f(\phi_i)\,\mathrm{d}x\right| \leq \int_{-\pi}^{\pi} \underbrace{|f(x) - f(\phi_i)|}_{\leq |x-\phi_i| \cdot \underbrace{\sup_x |f'(x)|}_{\leq 2}}\,\mathrm{d}x \leq 2\sum_i \left(\int_{a_{i-1}-\phi_i}^{a_i-\phi_i} |x|\,\mathrm{d}x\right)$$

Error Bound Type \ Freq.	12	18	21	22
Normalised abs cos error	0.04	0.03	0.04	0.03
Normalised abs sin error	0.05	0.05	0.03	0.02
Normalised id cos error	0.06	0.05	0.04	0.04
Normalised id sin error	0.02	0.05	0.04	0.04
Angle approximation error	0.14	0.07	0.06	0.06
Numerical abs $\int_{-\pi}^{\pi}$ bound	0.59	0.52	0.50	0.44
Numerical abs $\int_{0}^{\pi}$ bound	0.23	0.17	0.20	0.17
Total numerical abs $\int_{-\pi}^{\pi}$ bound	0.73	0.59	0.56	0.50
Total numerical abs $\int_{0}^{\pi}$ bound	0.59	0.41	0.46	0.40
Naive abs cos baseline	0.85	0.85	0.85	0.85
Naive abs sin baseline	0.85	0.85	0.85	0.85

Chun Hei Yip, Rajashree Agrawal, Lawrence Chan, Jason Gross