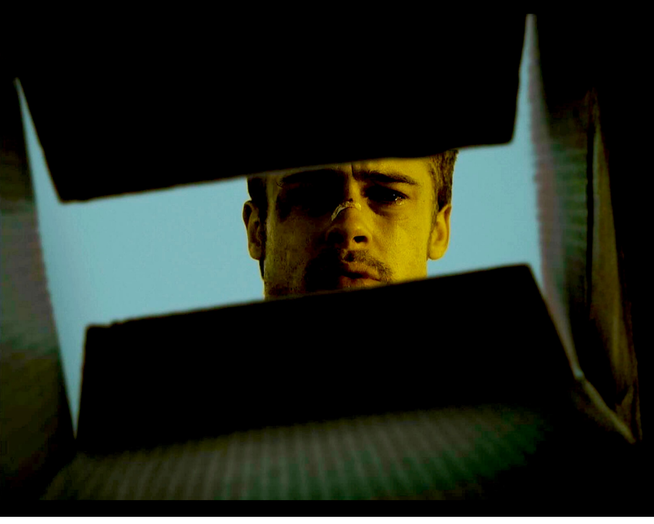


What's in the box?!

ExPLAIND *attributes model behavior to the* ***parameters***, ***data***, *and* ***training steps*** *at* ***local*** *and* ***global*** *granularities in a unified interpretability framework.*

...and all you need is ***gradient descent****!*

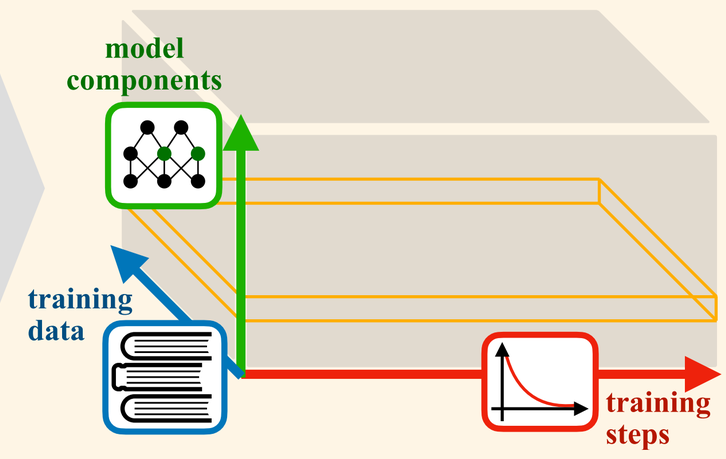
Paper, code & contact

model predictions

$$f_{\theta_N}(x) = f_{\theta_0}(x) - \sum_{s=0}^{N-1}\sum_{i=1}^{D}\sum_{l=1}^{M}\psi(s,\theta^{(i)},x_l,x) - \sum_{s=0}^{N-1}\sum_{i=1}^{D}\psi^{reg}(s,\theta^{(i)},x)$$

decompose into fine-grained ***influence scores**** ψ

which organize in the ***tensor of influences***



and are explained by ***accumulating scores*** *from* ***different perspectives and granularities***

$$\Psi = \sum \psi(s,\theta,x',x)$$

Does it work? ***Yes!***

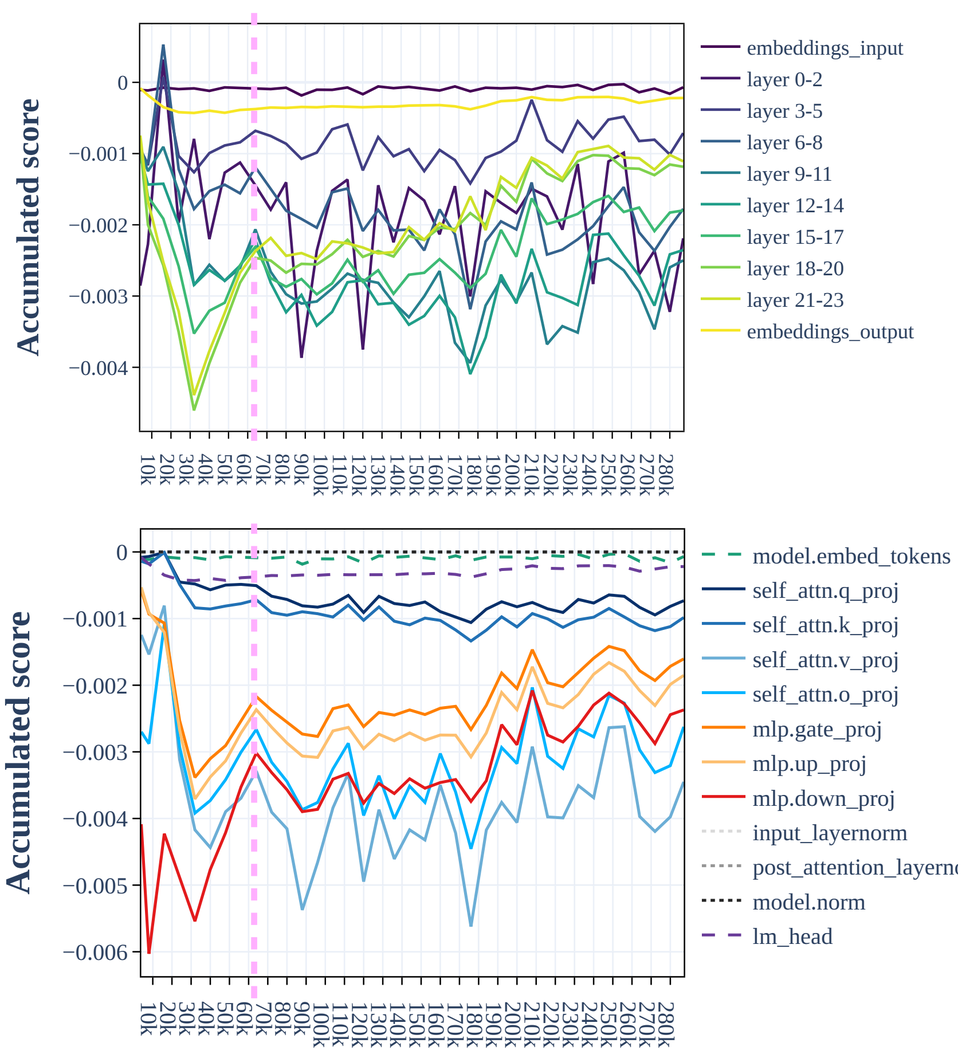
Model	ResNet9		Transformer		CNN	
Data	CIFAR-2		MOD-113		MNIST	
Integr. Steps	10	100	10	100	10	100
EPK Acc.	0.997	1.0	0.748	1.0	0.998	1.0
KL Diverg.	0.0	0.0	0.885	0.0	0.0	0.0

Summing up the scores exactly recovers predictions.

Case Study: EuroLLM-1.7B

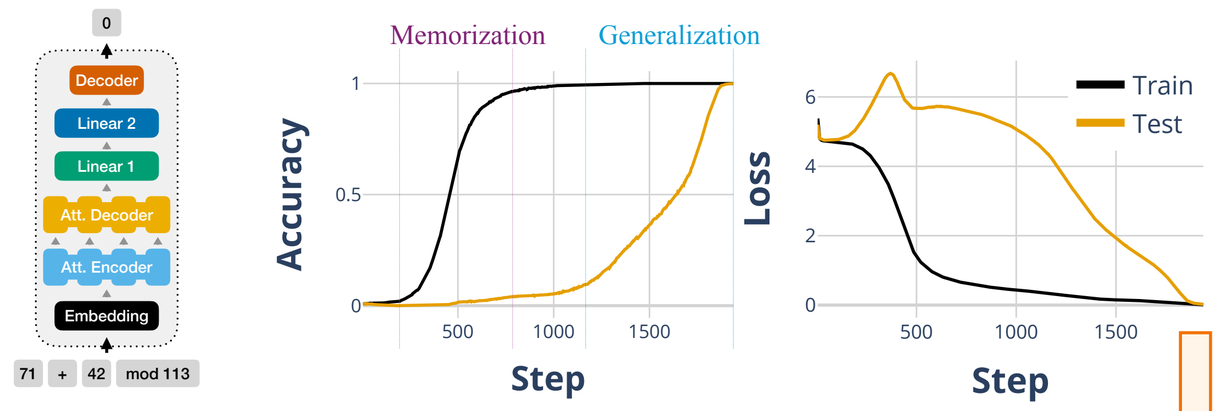
Scaling to larger models requires **subsampling and early gradient accumulation**. This way, one can decompose third-party pretraining checkpoints like those of EuroLLM.

ExPLAIND reveals a **two-phase dynamic in EuroLLM's pretraining**, with the first characterized by outer-layer MLP learning and the second by increased relative influence of intermediate attention layers.

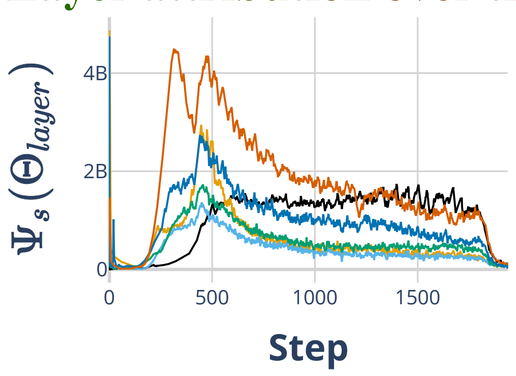


Case study: Grokked Transformer

We train a 1-layer transformer model which exhibits Grokking.

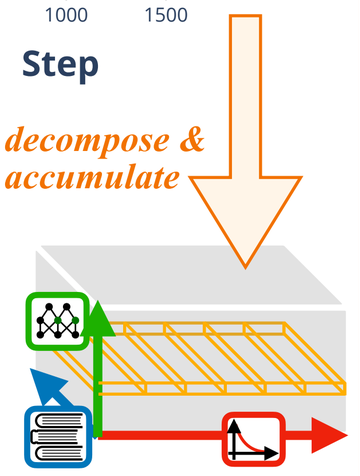


Layer attribution over time:

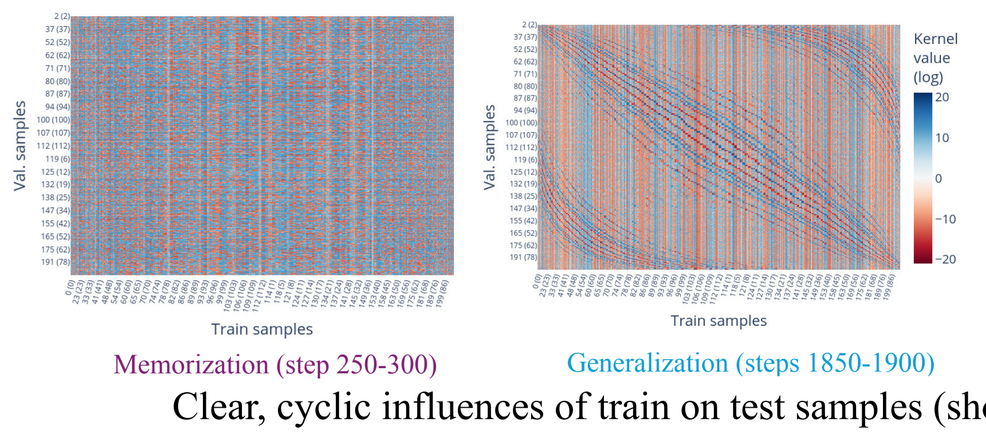


Memorization phase is influenced by all layers, whereas **generalization is driven by alignment of embeddings and decoder around a representation pipeline**. We confirm this with causal ablations.

decompose & accumulate



Layer-wise data attribution over time:



Memorization (step 250-300) Generalization (steps 1850-1900)

Clear, cyclic influences of train on test samples (shown: decoder).

...and much more (see paper)!

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a model ***with parameters*** *$\theta \in \mathbb{R}^D$, where $\mathcal{X} \subseteq \mathbb{R}^I$ and $\mathcal{Y} \subseteq \mathbb{R}^O$, and $\mathcal{D} = \{(x_1, y_1), \ldots, (x_M, y_M)\}$* ***be a dataset.*** *Further, let θ_N* ***be obtained from*** *θ_0* ***by*** *N* ***steps of AdamW*** *with learning rates α_s, weight decay λ, and loss L. Let $x \in \mathcal{X}$ be a model input. Then the predictions $f_{\theta_N}(x)$ decompose as*

$$f_{\theta_N}(x) = f_{\theta_0}(x) - \sum_{s=0}^{N-1}\sum_{i=1}^{D}\sum_{l=1}^{M}\psi(s,\theta^{(i)},x_l,x) - \sum_{s=0}^{N-1}\sum_{i=1}^{D}\psi^{reg}(s,\theta^{(i)},x)$$

where

$$\psi(s,\theta^{(i)},x_k,x)_j := \phi_s^{test}(x)_{j,i} \cdot \phi_s^{train}(x_k)_i \in \mathbb{R}$$

$$\phi_s^{test}(x) := \int_0^1 \nabla_\theta f_{\theta_s(t)}(x)\, dt \in \mathbb{R}^{O\times D},$$

$$\phi_s^{train}(x_k) := \sum_{i=0}^{s} \alpha_{s,i}\, \mathbf{1}_{k\in Batch_i}\, w_{i,k}\, \frac{\nabla_\theta\left(L(f_{\theta_i}(x_k), y_k)\right)}{\sqrt{\hat{v}_{s+1}} + \epsilon} \in \mathbb{R}^D$$

$$\mathbf{r}_s := \alpha_s \lambda \theta_s \in \mathbb{R}^D,$$

$$\theta_s(t) := \theta_s + t(\theta_{s+1} - \theta_s) \in \mathbb{R}^D,$$

$$\alpha_{s,i} := \alpha_s(1-\beta_1)\beta_1^{s-i}(1-\beta_1^{s+1})^{-1} \in \mathbb{R},$$

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

MCML Munich Center for Machine Learning

ExPLAIND:

Unifying Model, Data, and Training Attribution to Study Model Behavior

Florian Eichin, Yupei Du, Philipp Mondorf, Maria Matveev, Barbara Plank, Michael A. Hedderich