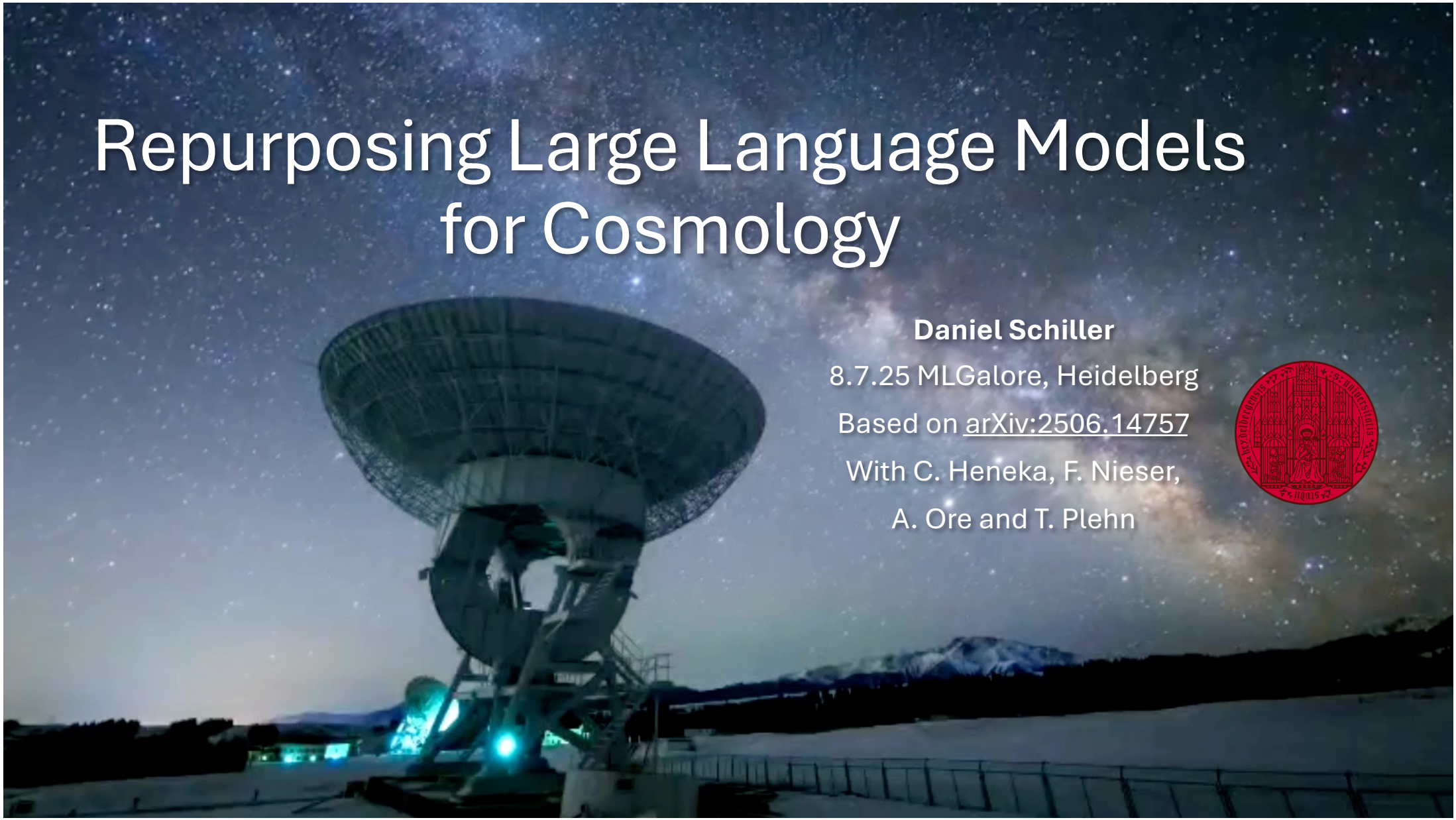


Repurposing Large Language Models for Cosmology

Daniel Schiller

8.7.25 MLGalore, Heidelberg

Based on arXiv:2506.14757

With C. Heneka, F. Nieser,
A. Ore and T. Plehn

Foundation Models for Physics Data

- **Foundation Model: long** pretraining + **short** finetuning

Foundation Models for Physics Data

- **Foundation Model: long** pretraining + **short** finetuning
- **Pretraining:**
  - Pretraining task on a **large dataset**
  - Learn **self-supervised** a good **data representations**, e.g. symmetry-aware

Foundation Models for Physics Data

- **Foundation Model: long** pretraining + **short** finetuning
- **Pretraining:**
  - Pretraining task on a **large dataset**
  - Learn **self-supervised** a good **data representations**, e.g. symmetry-aware
- **Finetuning:**
  - Train on downstream task
  - Especially useful for **small datasets**

LLMs are Foundation Models

LLMs are Foundation Models

- **Pretraining:** complete sentences

LLMs are Foundation Models

- **Pretraining:** complete sentences
- Sentences are split into **tokens** by a predefined algorithm

$$\text{sentence} \leftrightarrow (t_1, \ldots, t_n)$$

LLMs are Foundation Models

- **Pretraining:** complete sentences
- Sentences are split into **tokens** by a predefined algorithm

$$\text{sentence} \leftrightarrow (t_1, \ldots, t_n)$$

- Example: “Hello Word” ↔ ([Hello],[World])

LLMs are Foundation Models

- **Pretraining:** complete sentences
- Sentences are split into **tokens** by a predefined algorithm

$$\text{sentence} \leftrightarrow (t_1, \dots, t_n)$$

- Example: “Hello Word” ↔ ([Hello],[World])
- Pretraining = **Next token prediction**
- Example: $p_\theta([\text{World}]|[\text{Hello}]) \approx p([\text{World}]|[\text{Hello}])$

How do LLMs come into play?

How do LLMs come into play?

- Network sizes: vs LLMs:
  - Physics Foundation Models: $\sim\mathbf{1B}$ parameters
  - LLMs: $\sim\mathbf{100B}$ parameters
- Language modeling is a hard task

How do LLMs come into play?

- Network sizes: vs LLMs:
  - Physics Foundation Models: $\sim\mathbf{1B}$ parameters
  - LLMs: $\sim\mathbf{100B}$ parameters
- Language modeling is a hard task
- LLM = **highly structured function**

How do LLMs come into play?

- Network sizes: vs LLMs:
  - Physics Foundation Models: $\sim\mathbf{1B}$ parameters
  - LLMs: $\sim\mathbf{100B}$ parameters
- Language modeling is a hard task
- LLM = **highly structured function**
- **Idea: Repurpose this structure**

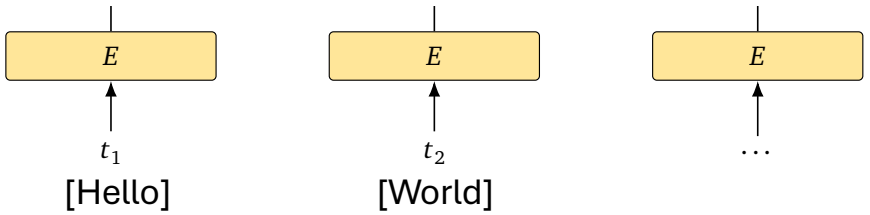
LLM architecture

LLM architecture

t_1 t_2 $\dots$

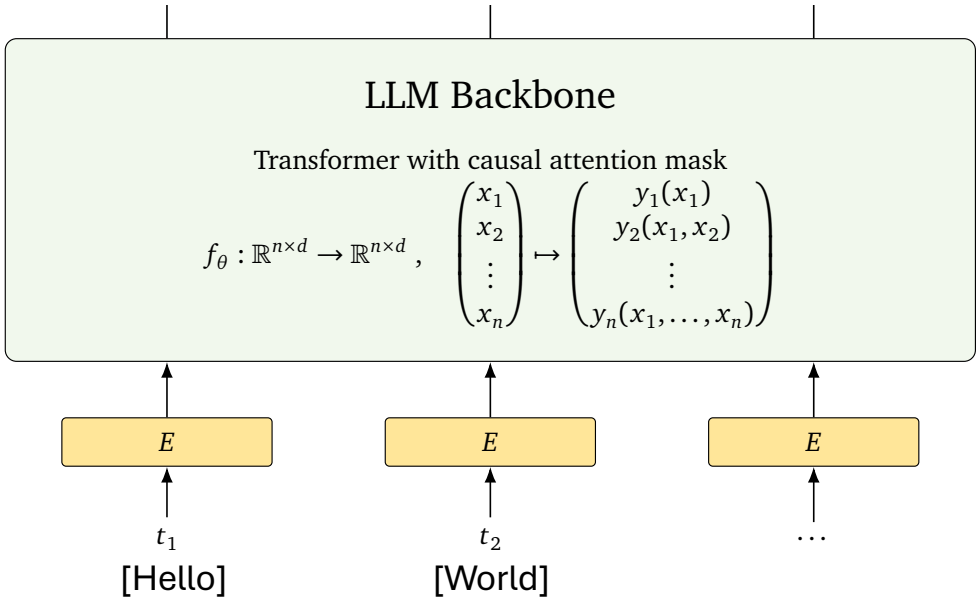
[Hello] [World]

LLM architecture



Embedding:
Maps discrete tokens to a latent representation

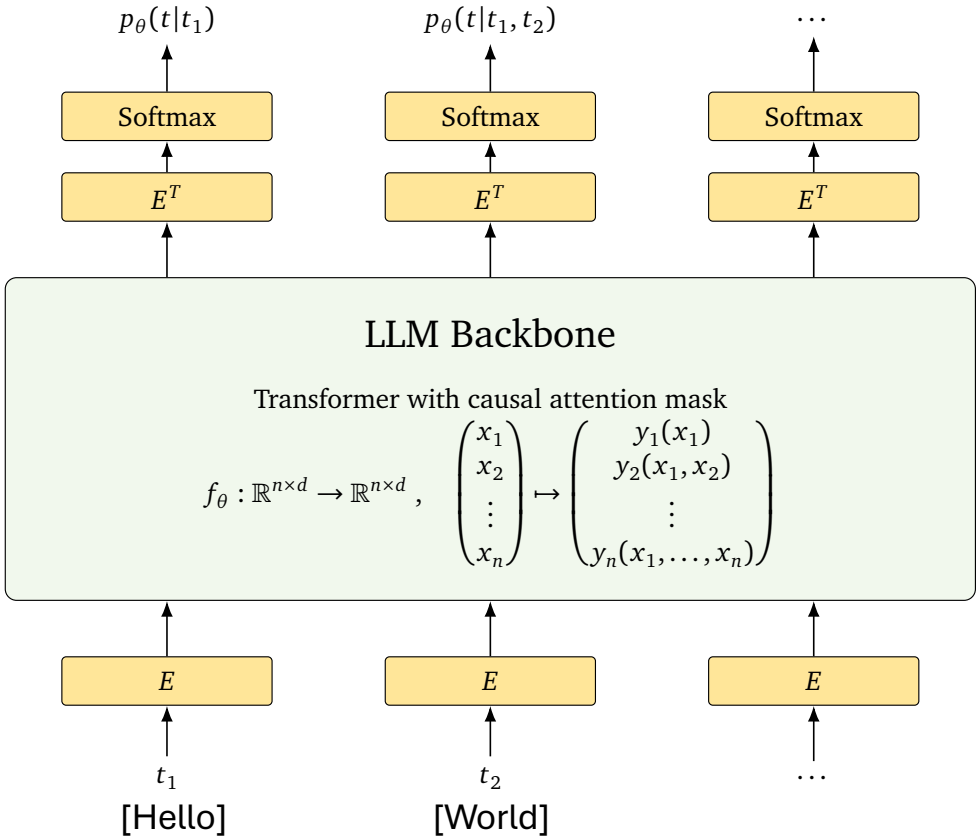
LLM architecture



LLM backbone:
Learns causal correlations between representations

Embedding:
Maps discrete tokens to a latent representation

LLM architecture

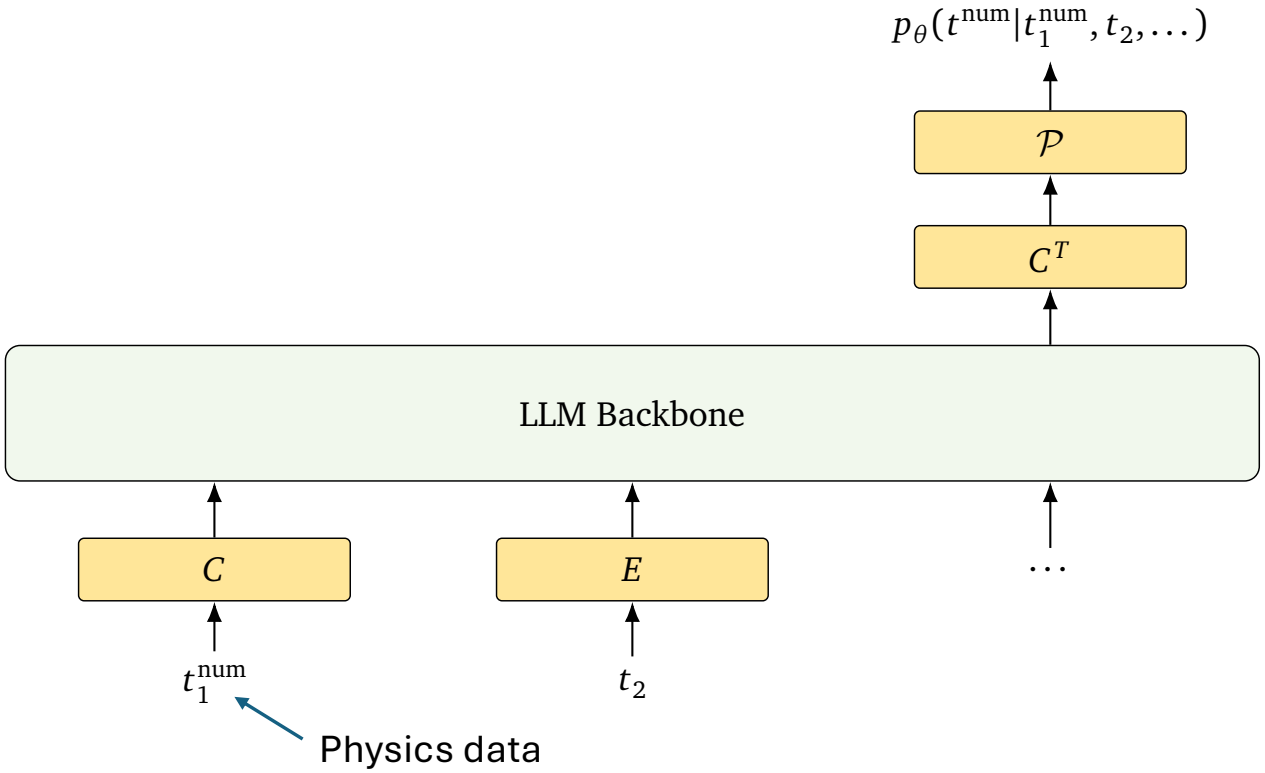


Unembedding & Softmax:
Maps representations to a categorical distribution

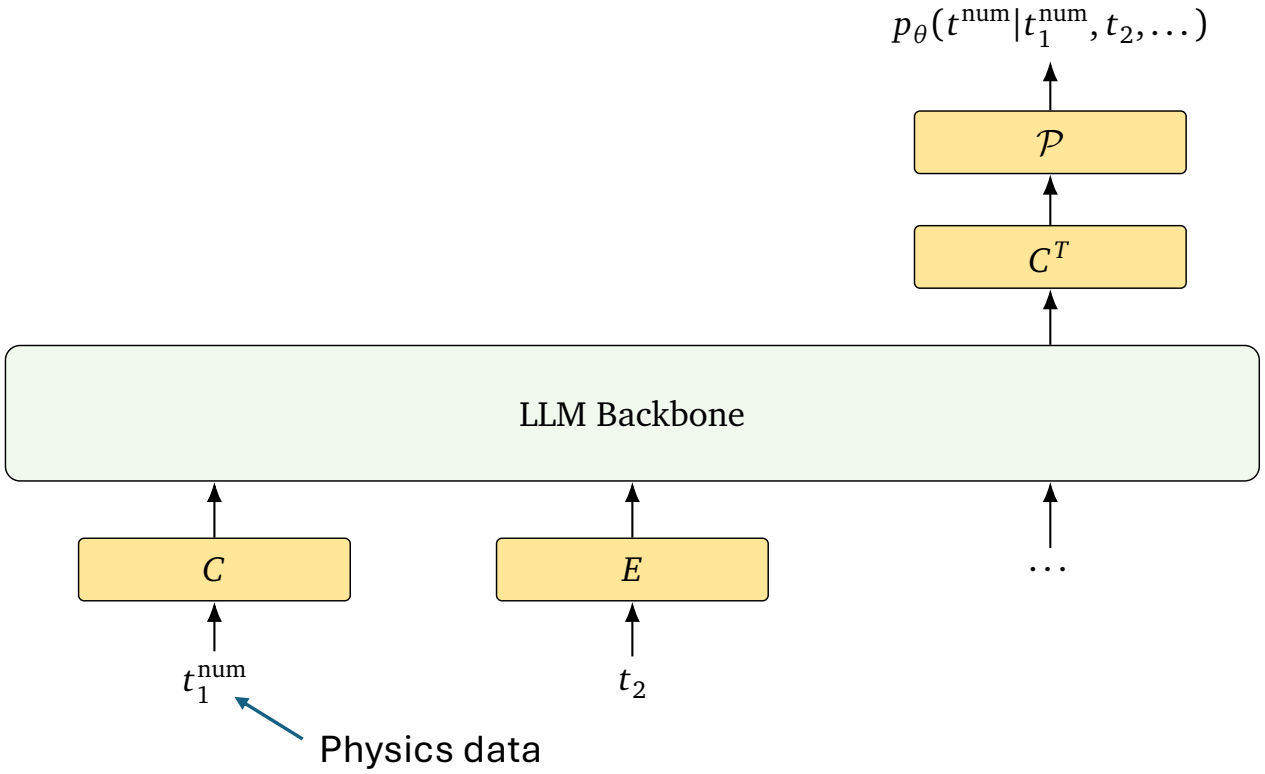
LLM backbone:
Learns causal correlations between representations

Embedding:
Maps discrete tokens to a latent representation

Ansatz


$p_\theta(t^{\text{num}}|t_1^{\text{num}}, t_2, \ldots)$
$\mathcal{P}$
C^T
LLM Backbone
C
E
$\ldots$
t_1^{num}
t_2
Physics data

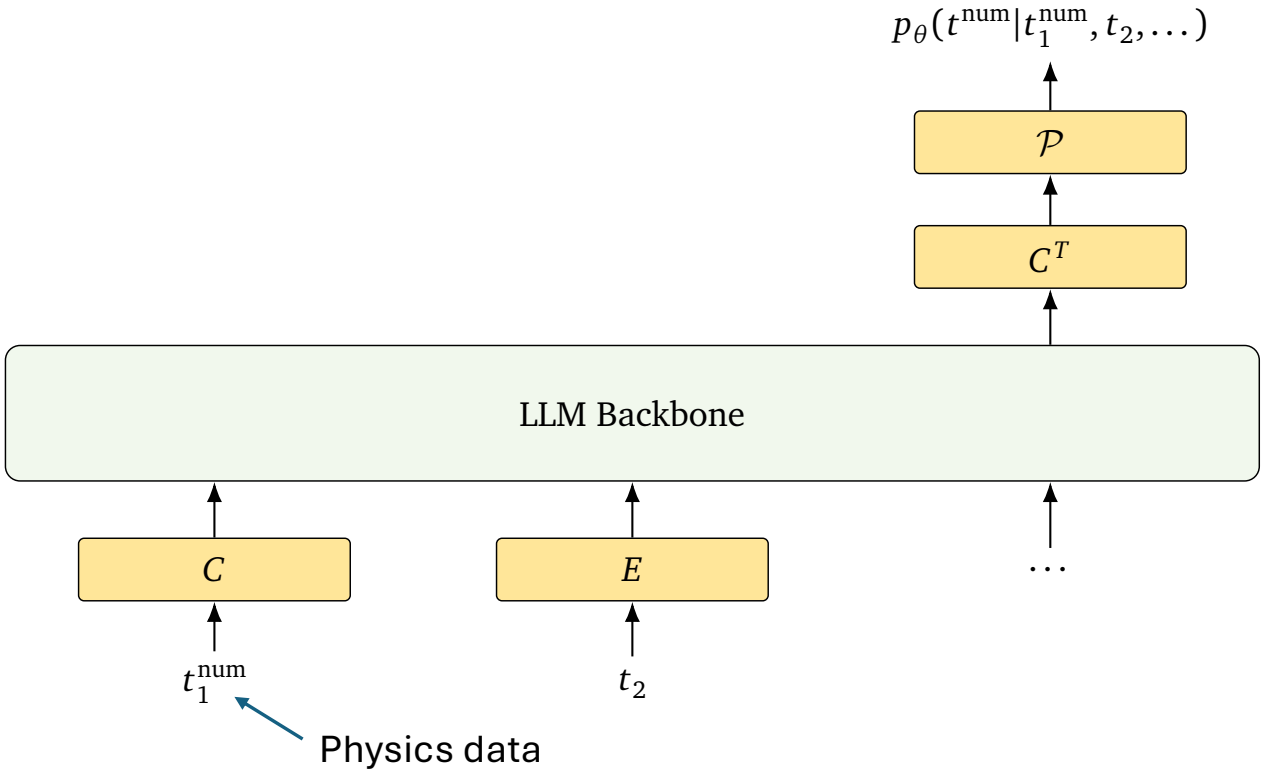
Ansatz



C^{T}: output connector network

C: input connector network

Ansatz



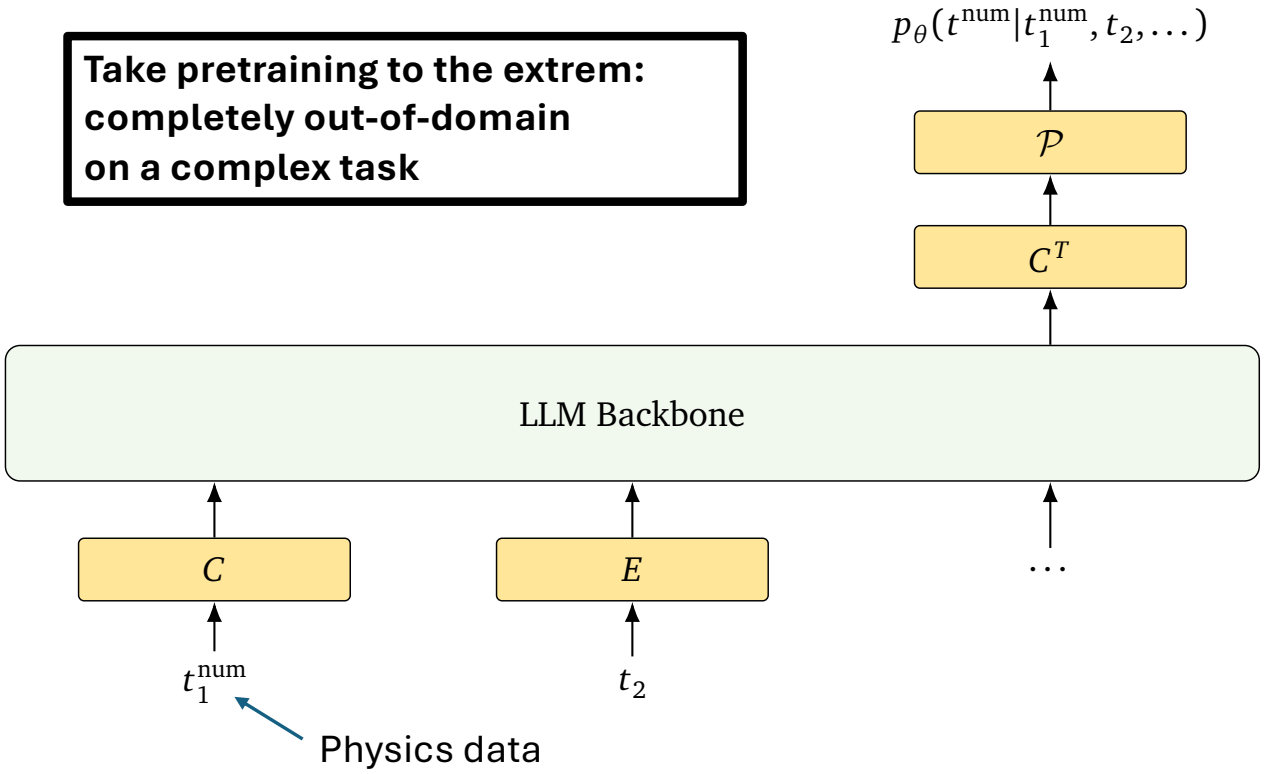
$\mathcal{P}$: Softmax, Gaussian, ...

C^{T}: output connector network

C: input connector network

Ansatz

Take pretraining to the extrem:
completely out-of-domain
on a complex task



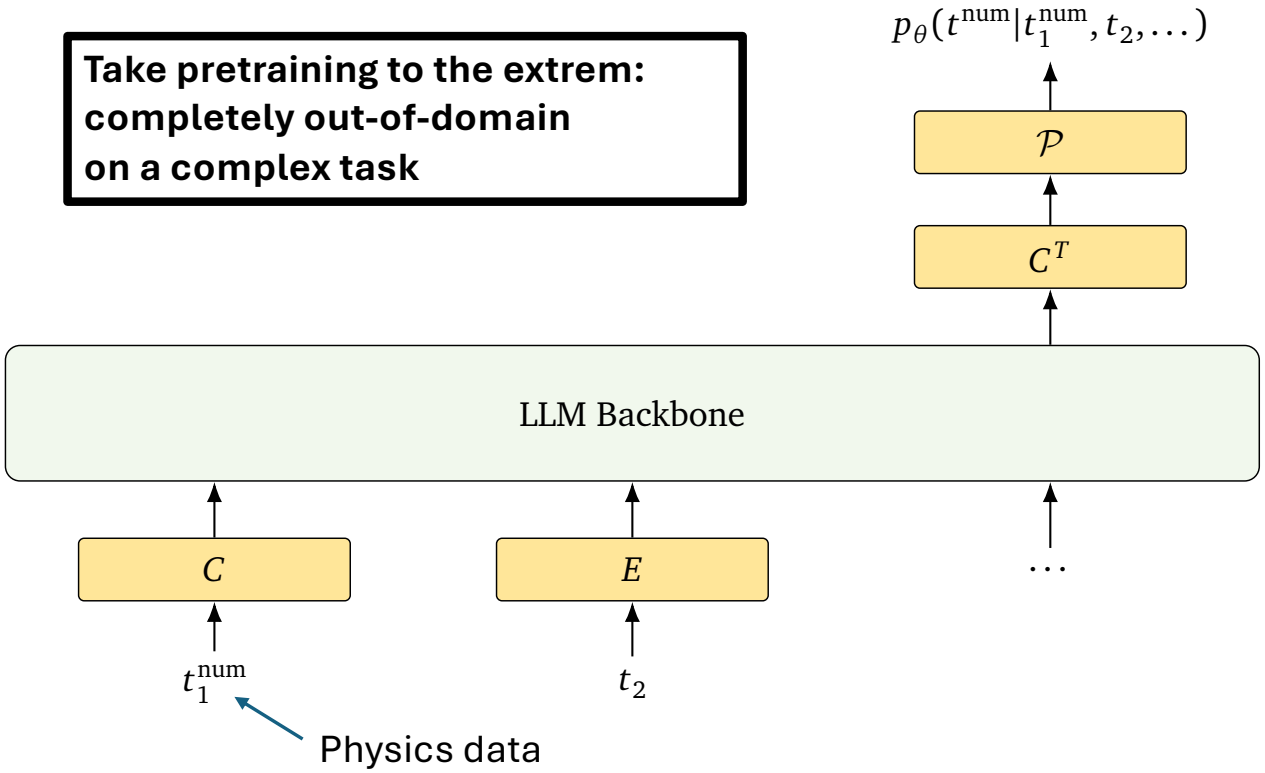
$\mathcal{P}$: Softmax, Gaussian, ...

C^{T}: output connector network

C: input connector network

Ansatz

Take pretraining to the extrem:
completely out-of-domain
on a complex task



$\mathcal{P}$: Softmax, Gaussian, ...

C^{T}: output connector network

Here: **Qwen2.5** (360M parameters)
C & C^{T}: affine layers

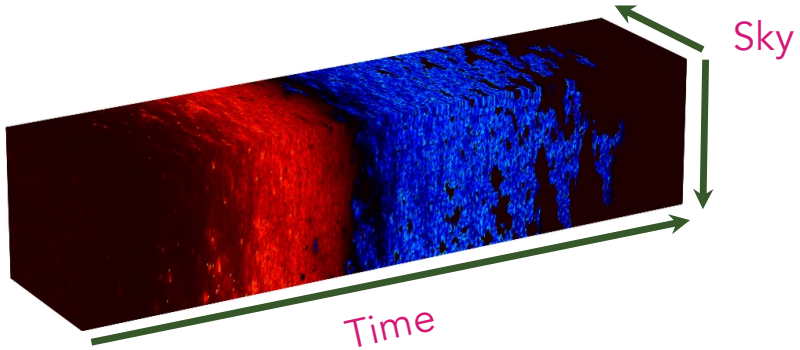
C: input connector network

Lightcone Large Language Model (L3M)

- Data: simulated lightcones

Lightcone Large Language Model (L3M)

- Data: simulated lightcones

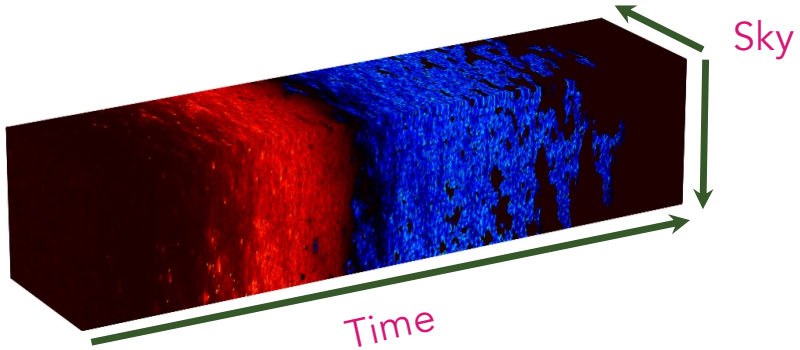


Intensity of Hydrogen Spin Flip Emission/Absorption relative to CMB

Brightness Temperature δT_{21}

Lightcone Large Language Model (L3M)

- Data: simulated lightcones



Intensity of Hydrogen Spin Flip Emission/Absorption relative to CMB

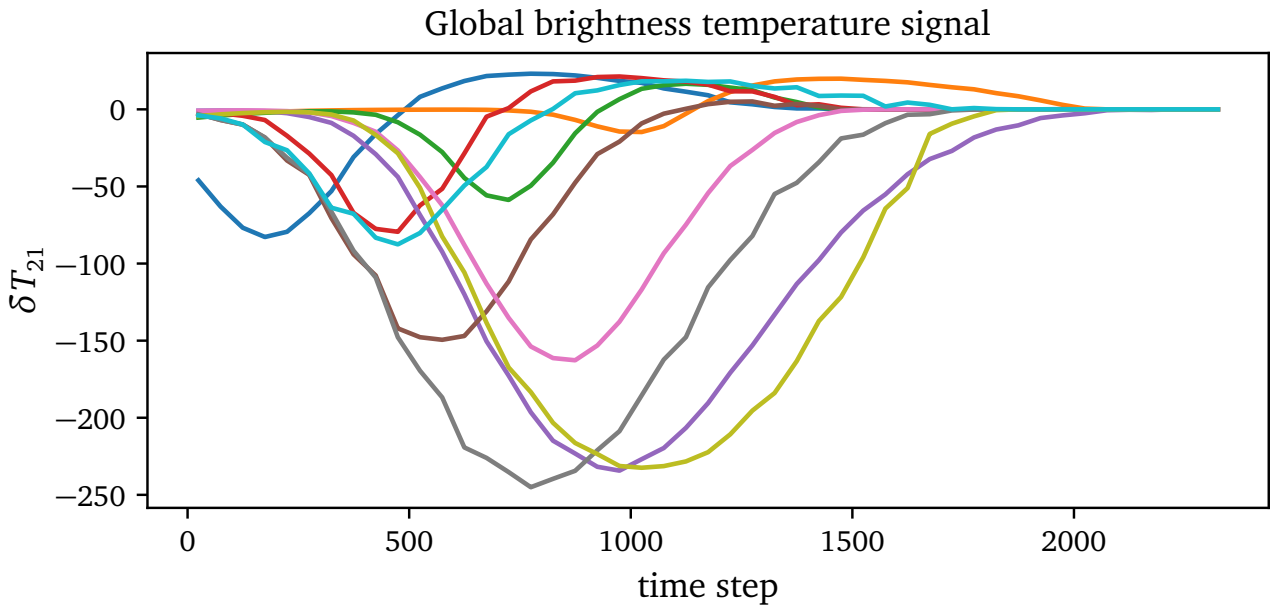
Brightness Temperature δT_{21}

- Depend on 6 parameters and initial conditions

Regression with frozen backbone

Regression with frozen backbone

- Regress 6 simulation parameters from global δT_{21} signal



Regression with frozen backbone

Minimal prompting:

t_1^{BT} … t_n^{BT}<|param|>

Global signal at time step 1

Added token to output prediction

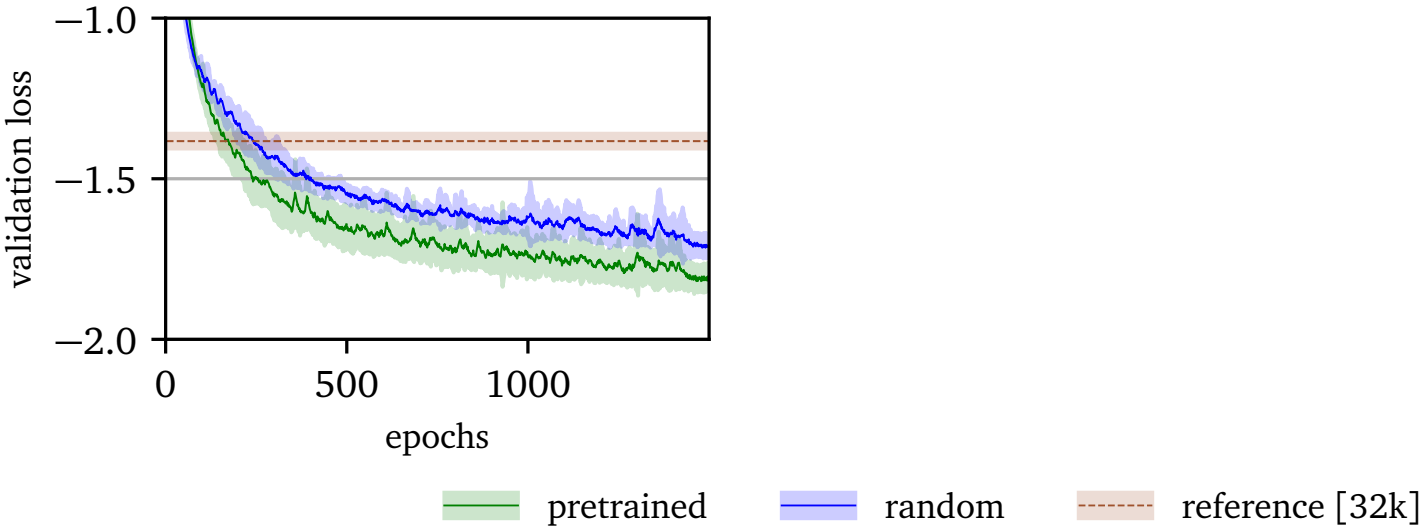
Regression with frozen backbone

Minimal prompting:

t_1^{BT} … t_n^{BT}<|param|>

Global signal at time step 1

Added token to output prediction



Regression with frozen backbone

Minimal prompting:

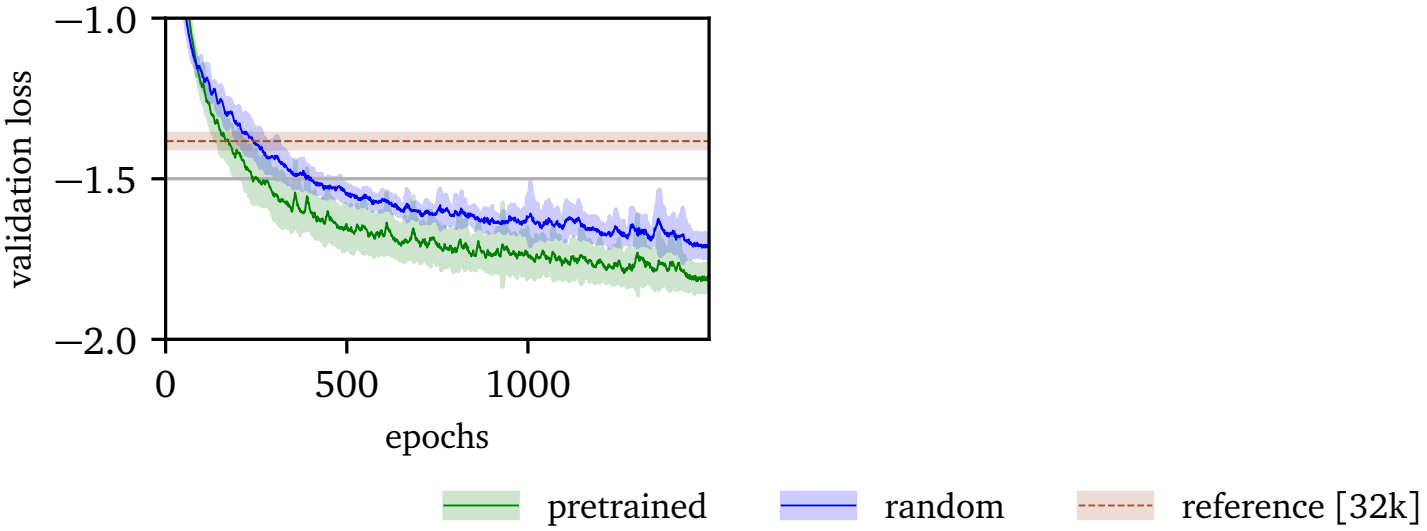
t_1^{BT} … t_n^{BT}<|param|>

Global signal at time step 1

Added token to output prediction

Chat inspired prompting:

<|start|>**system**
<|end|>
<|start|>**user**
t_1^{BT} … t_n^{BT}<|end|>
<|start|>**assistant**
<|param|><|end|>



Regression with frozen backbone

Minimal prompting:

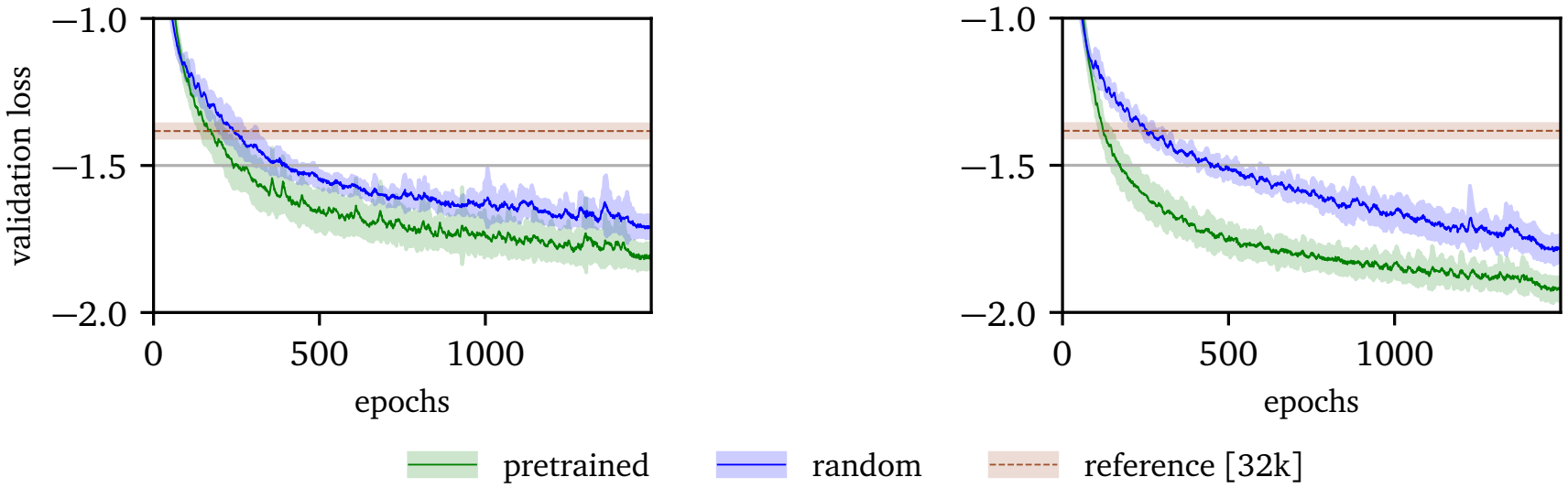
t_1^{BT} … t_n^{BT}<|param|>

Global signal at time step 1

Added token to output prediction

Chat inspired prompting:

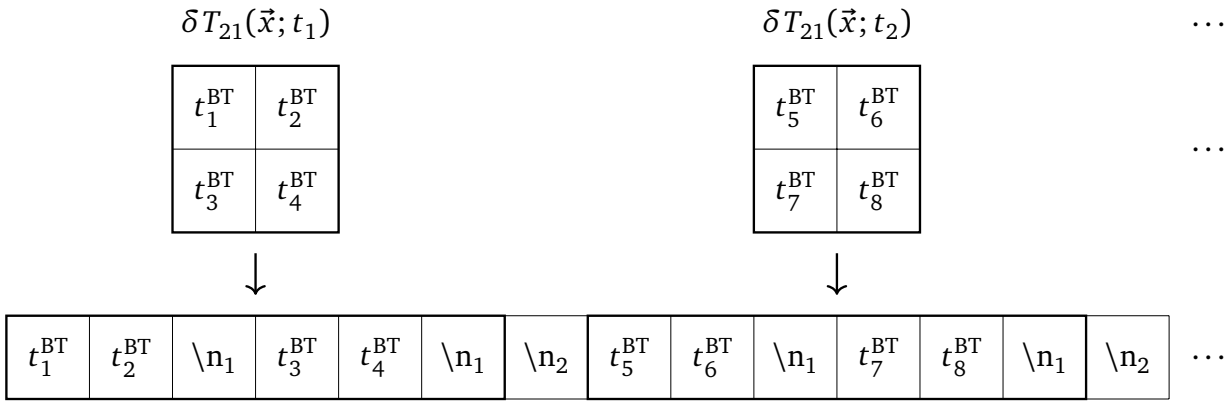
<|start|>**system**
<|end|>
<|start|>**user**
t_1^{BT} … t_n^{BT}<|end|>
<|start|>**assistant**
<|param|><|end|>

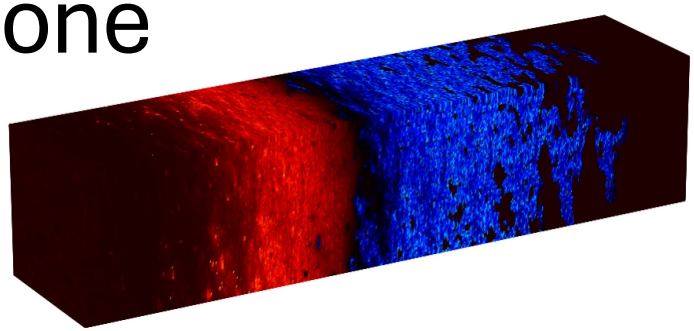


Generation with finetuned backbone

Generation with finetuned backbone

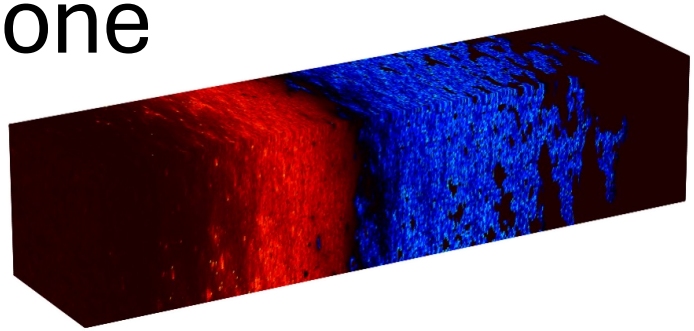
- Represent lightcone as a sequence of tokens

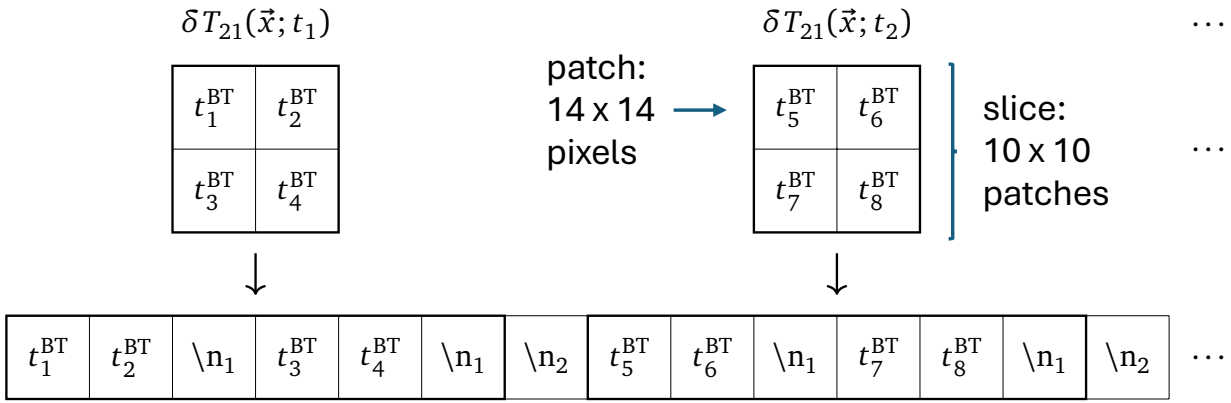




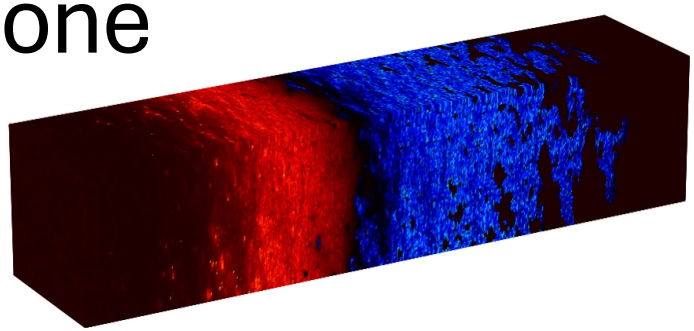
Generation with finetuned backbone

- Represent lightcone as a sequence of tokens

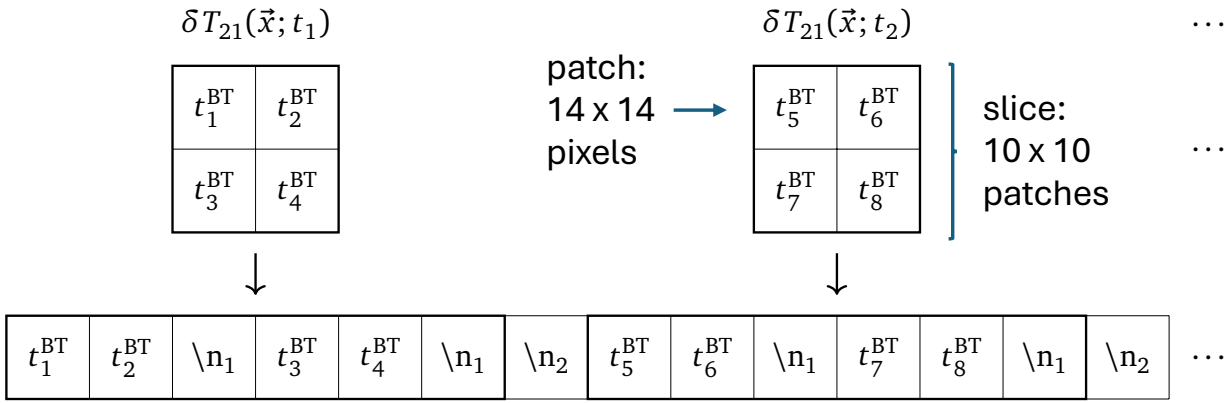




Generation with finetuned backbone



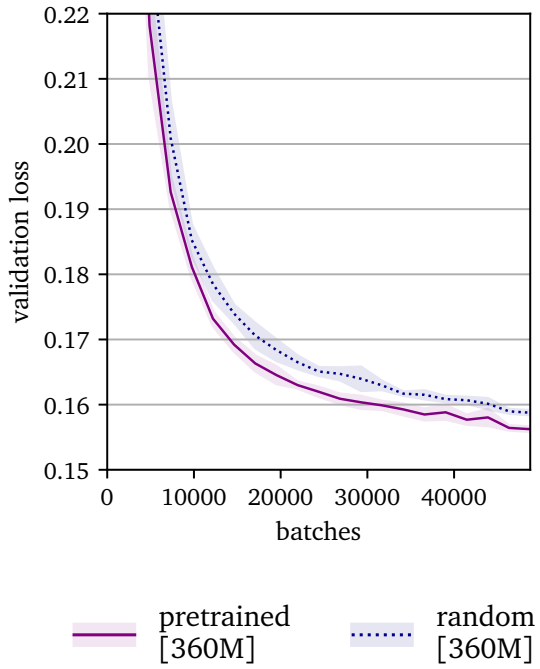
- Represent lightcone as a sequence of tokens



- Next token/patch prediction

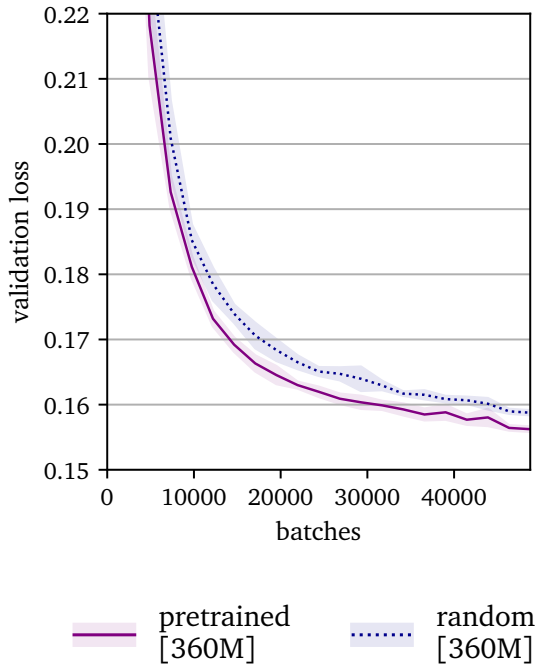
Generation with finetuned backbone

Complete backbone training:



Generation with finetuned backbone

Complete backbone training:



Partial backbone training with LoRA:

Modify linear layers by a small rank:

$$M \in \mathbb{R}^{a\times b} \rightarrow M + \delta M$$

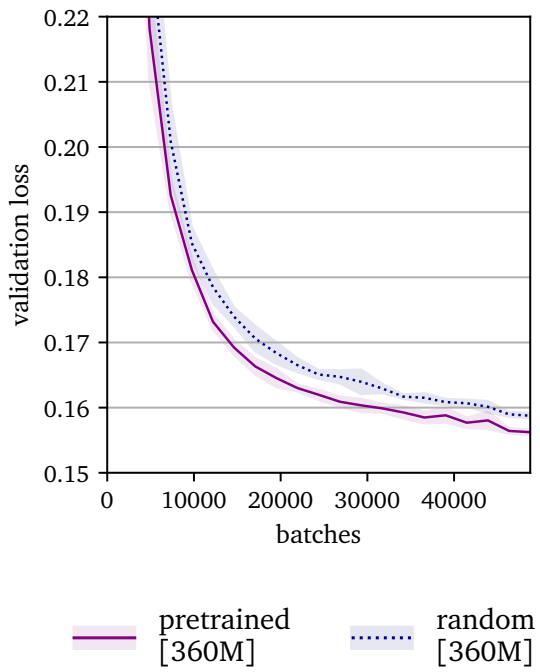
frozen trained

$$\delta M = M_1 M_2$$

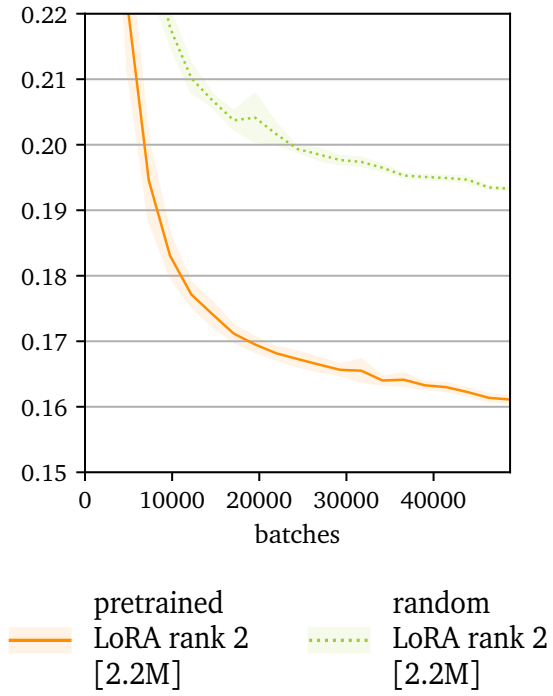
$$M_1 \in \mathbb{R}^{a\times r}, M_2 \in \mathbb{R}^{r\times b}$$

Generation with finetuned backbone

Complete backbone training:



Partial backbone training with LoRA:



Modify linear layers by a small rank:

$$M \in \mathbb{R}^{a \times b} \rightarrow M + \delta M$$

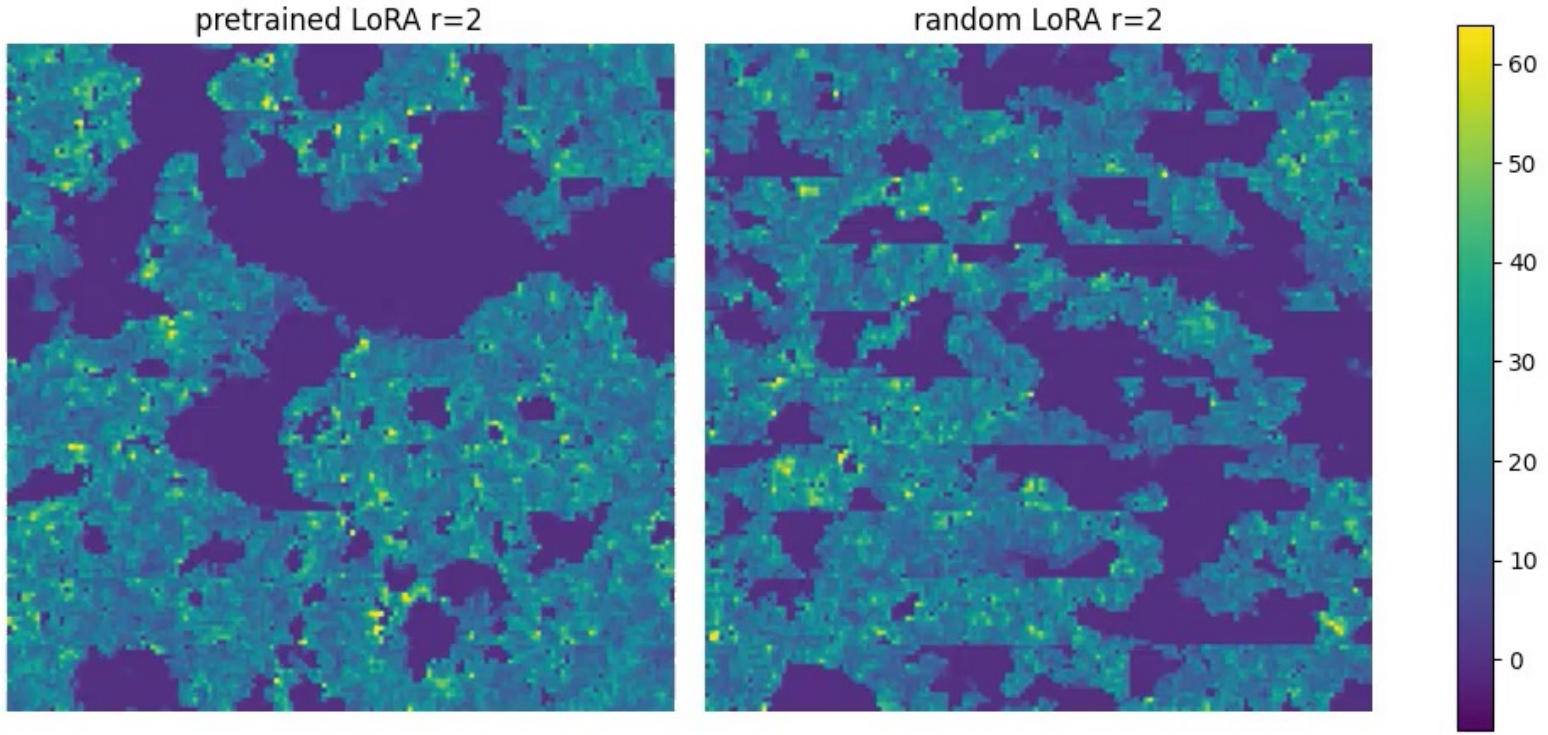
frozen trained

$$\delta M = M_1 M_2$$

$$M_1 \in \mathbb{R}^{a \times r}, M_2 \in \mathbb{R}^{r \times b}$$

Generation with finetuned backbone

Generation with finetuned backbone

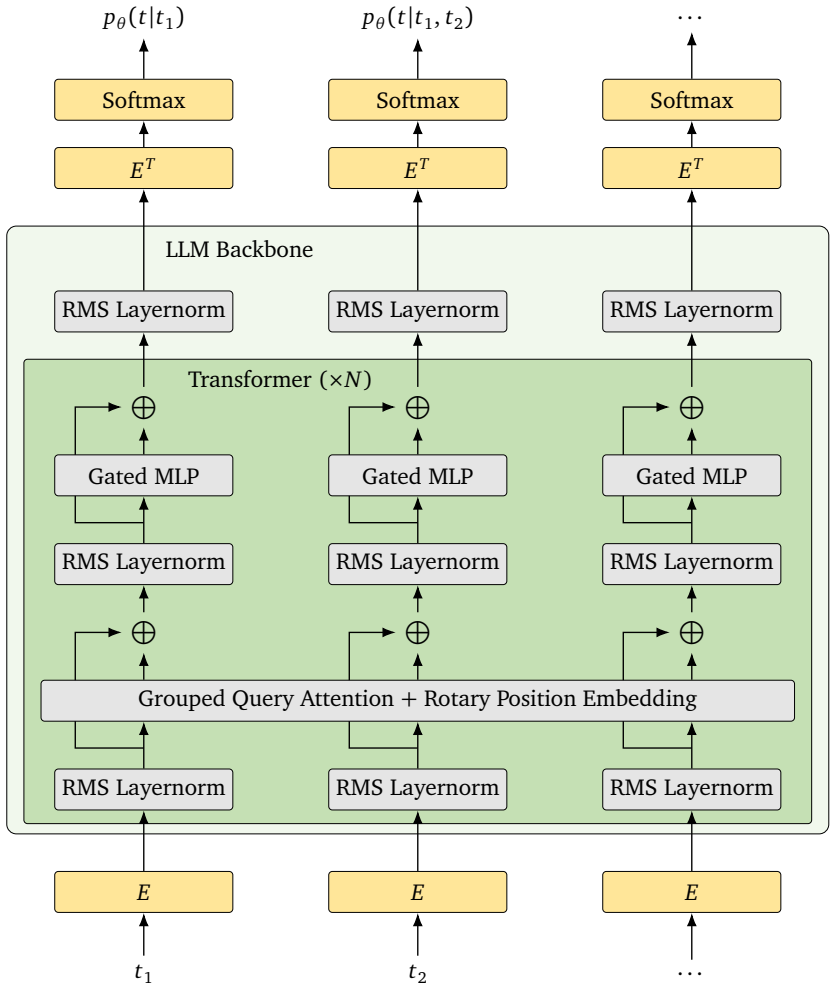


Summary

- Repurposed (open-source) Large Language Models
- Added connector networks and finetuned backbone
- Showed benefit from pretrained structure
- **Novel approach to get large networks for physics**

LLM Architecture

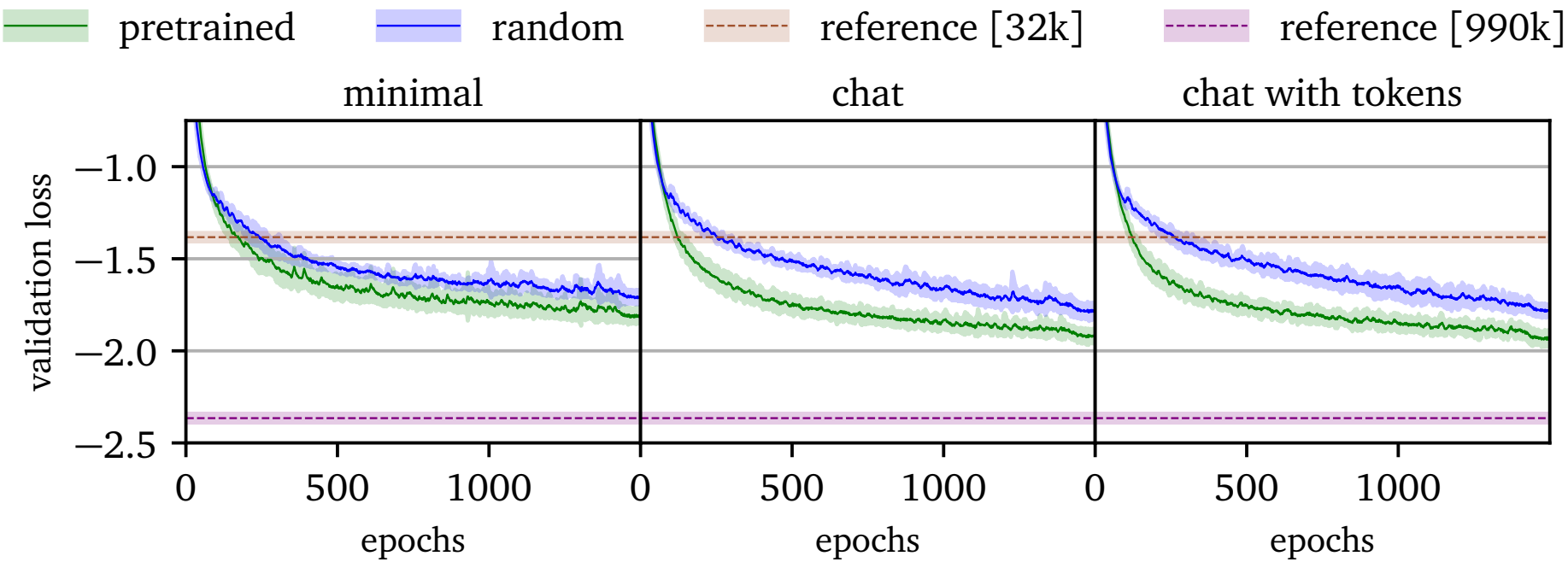
$$p_\theta(t_i|t_1, \ldots, t_{i-1}) \approx p(t_i|t_1, \ldots, t_{i-1})$$



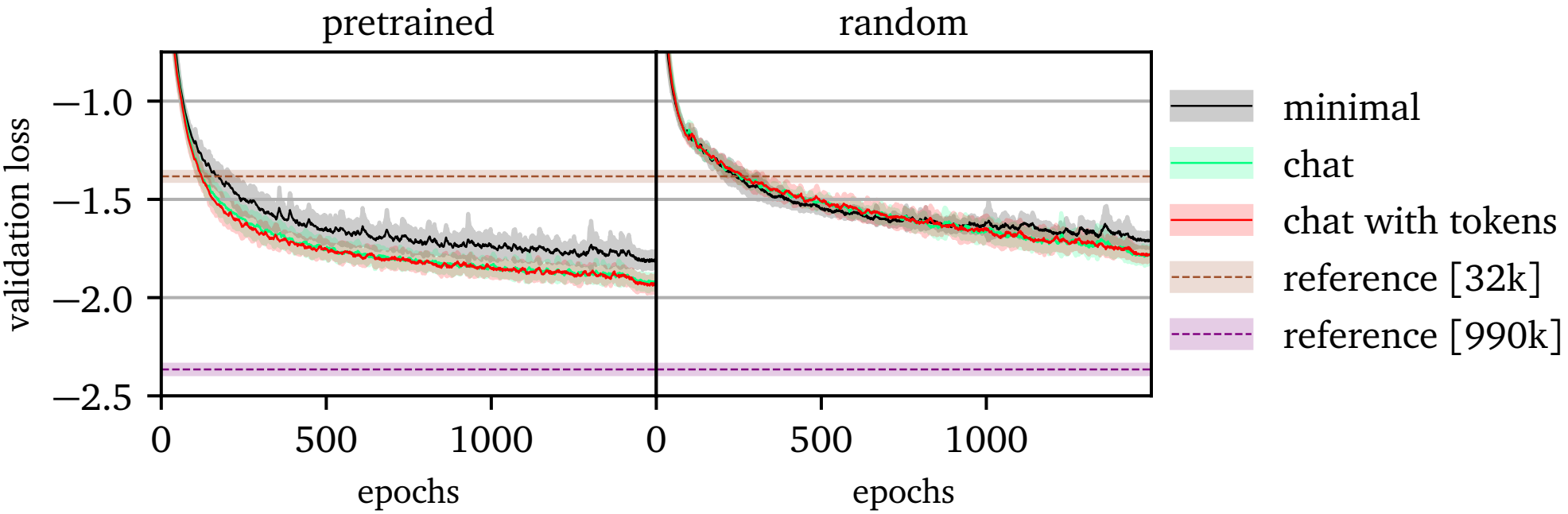
Regression Results

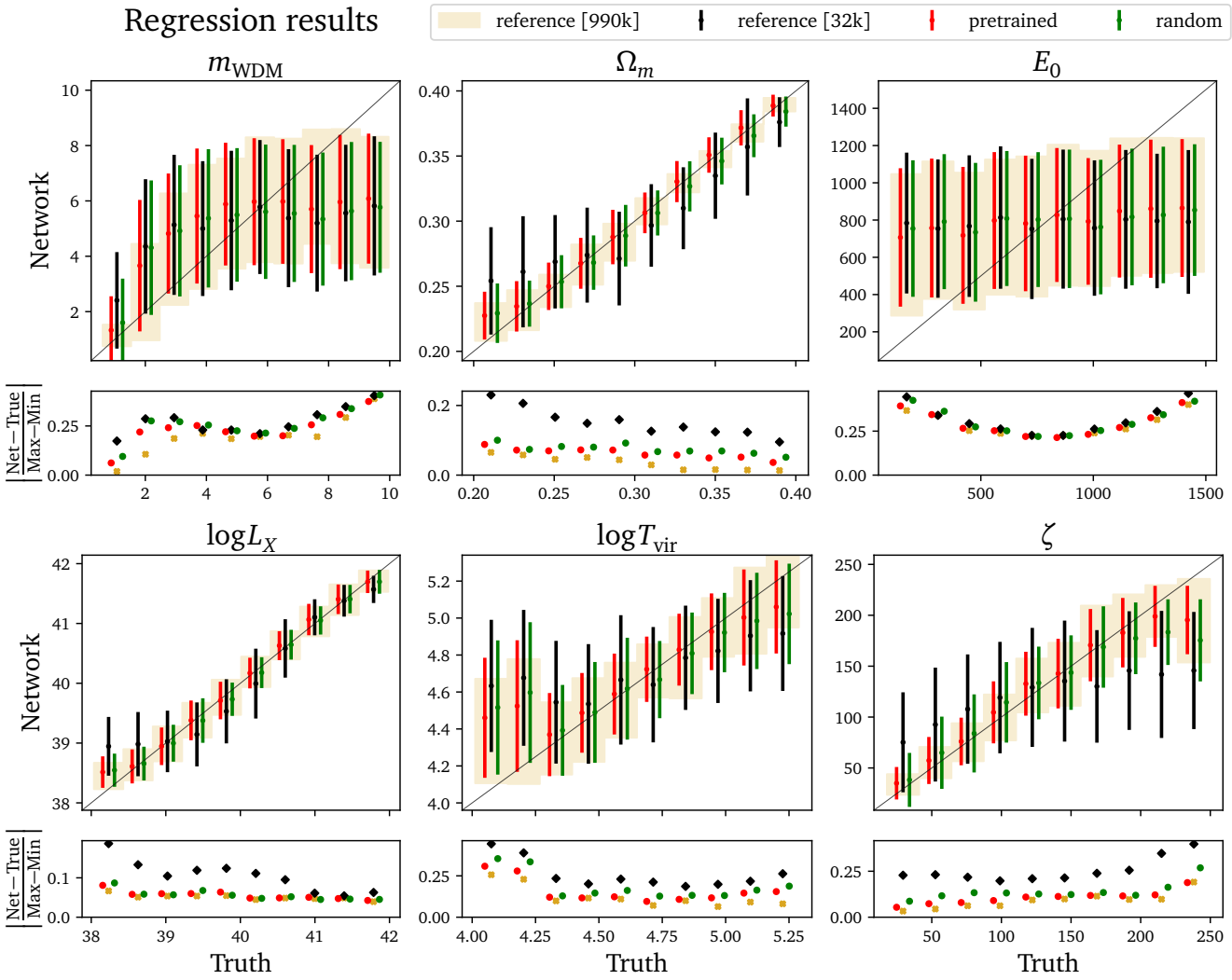
$$\mathcal{P} = \frac{1}{(2\pi)^3\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(\vec{\mu}-\vec{y})^{\mathrm{T}}\Sigma^{-1}(\vec{\mu}-\vec{y})\right)$$

$\vec{\mu}$: mean values, Σ: covariance matrix



Regression Results




Regression results
reference [990k]
reference [32k]
pretrained
random
m_{WDM}
Ω_m
E_0
$\log L_X$
$\log T_{\mathrm{vir}}$
ζ
Network
$\frac{|\mathrm{Net}-\mathrm{True}|}{|\mathrm{Max}-\mathrm{Min}|}$
Truth

Conditional Flow Matching

- Distribution Mapping via velocity field (ODE)

$$\frac{dx}{d\tau} = v(x, \tau, z) \quad \text{with} \quad x(\tau) \sim \begin{cases} \mathcal{N}(\mu = 0, \sigma = 1) & \tau = 0 \\ p\left(t_i^{\mathrm{BT}} \middle| t_{i-1}^{\mathrm{BT}}, \dots\right) & \tau = 1 \end{cases}$$

- Network learns velocity field

$$\mathcal{L} = \left\langle \left(v - v_\theta(x_\tau, \tau, z)\right)^2 \right\rangle \quad \text{with}$$

$$v \equiv t_i^{\mathrm{BT}} - x_0, \qquad x_\tau \equiv (1-\tau)x_0 + \tau t_i^{\mathrm{BT}}, \qquad x_0 \sim \mathcal{N}(\mu = 0, \sigma = 1)$$

- Trained a network with 500k parameters

Generation Results