SKATR

A self-supervised summary transformer for the Square Kilometre Array

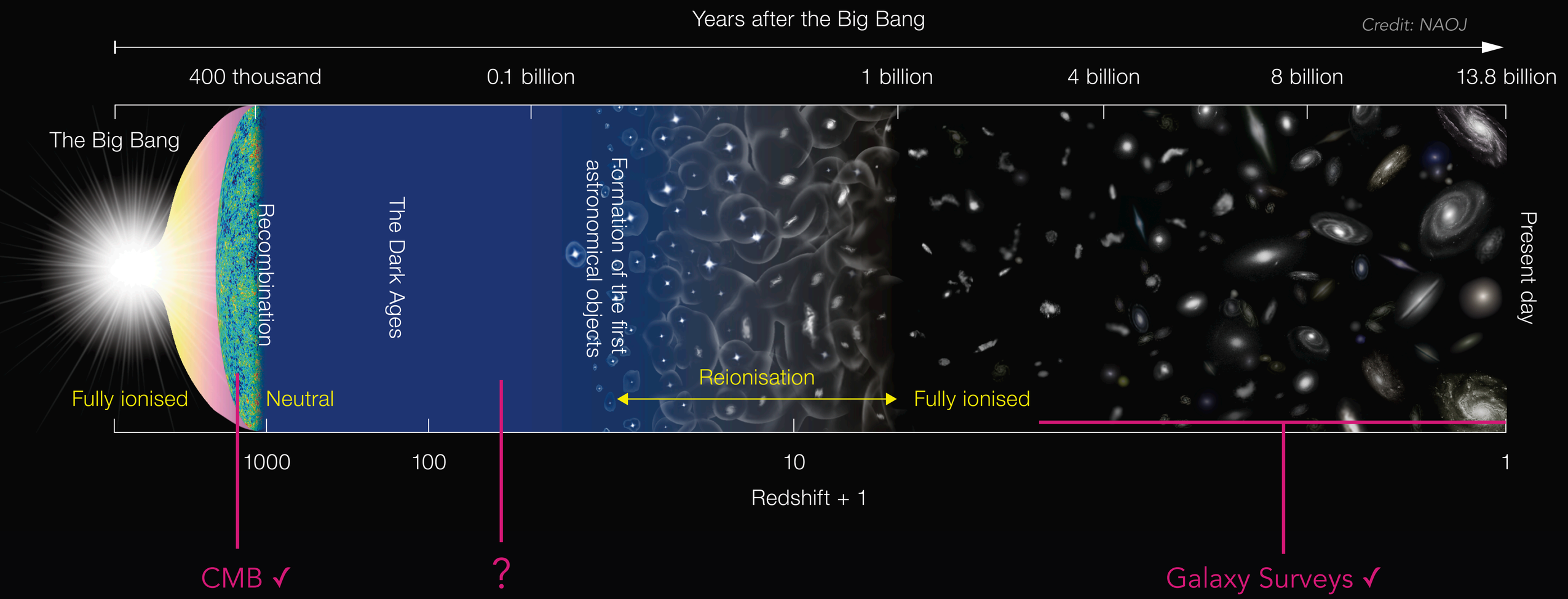
Ayodele Ore

EUCAIFCon 2025, Cagliari

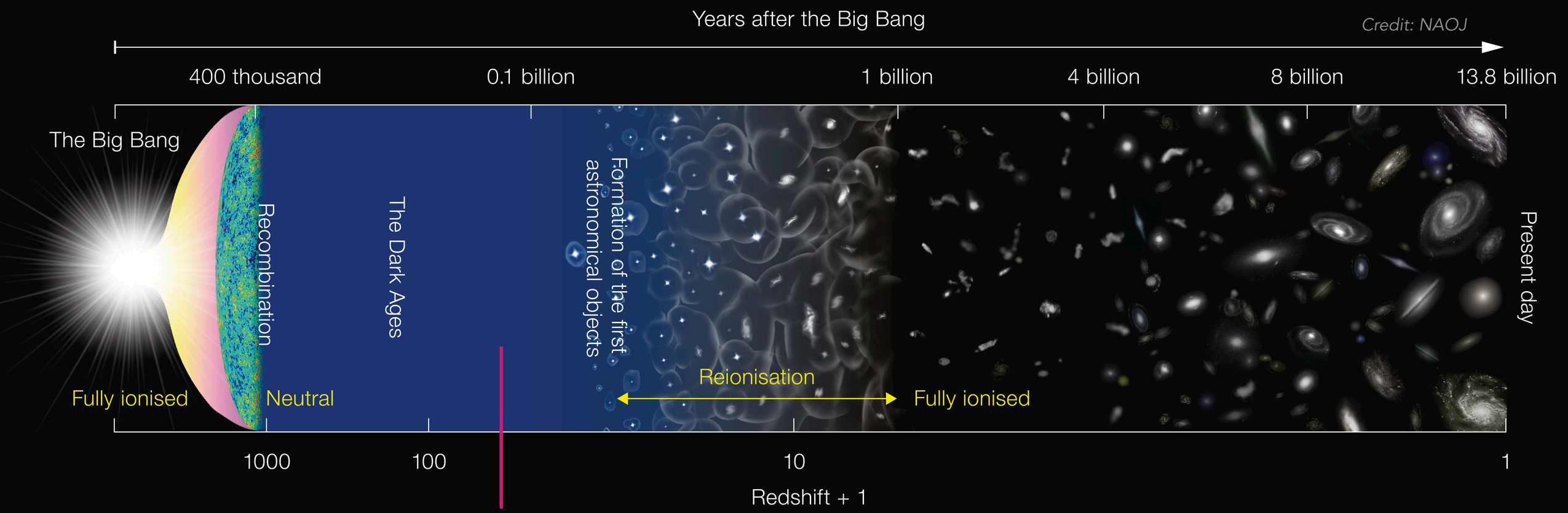
Based on arXiv:2410.18899 with Caroline Heneka and Tilman Plehn

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Observations of the Cosmos



Observations of the Cosmos



Neutral H

Structure traceable via rare 21cm emissions: $\left(|\uparrow\uparrow\rangle \rightarrow |\uparrow\downarrow\rangle + \gamma \right)$

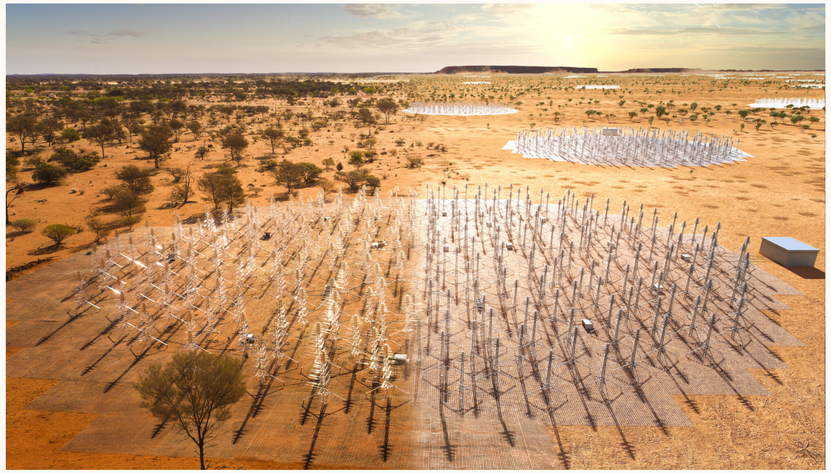
The Square Kilometre Array: 21cm imaging

- The Square Kilometre Array (SKA) Observatory is a pair of radio telescopes. Located in South Africa and Australia

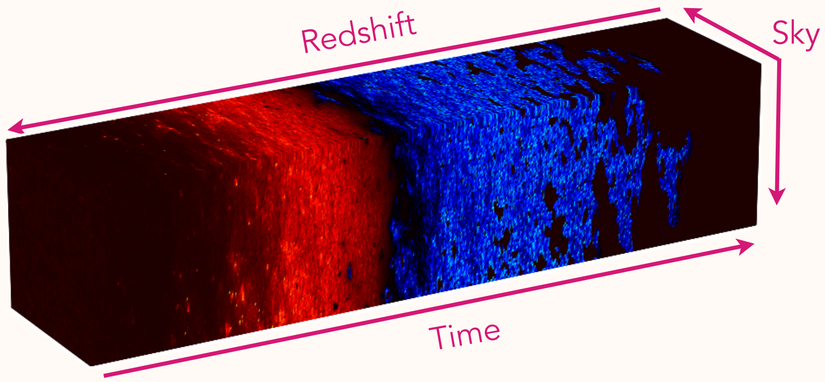
- SKA will image ***3D maps*** of 21cm intensity
  - Redshift range capturing Reionisation
  - Huge data rate: Few TB/s, ***8 EB*** archived total

- Will inform us on:
  - Matter power spectrum
  - Deviations from GR
  - Inflationary scenarios
  - Structure formation
  - Dark energy EoS
  - … and lots more

- Task: Predict physics parameters given an image
  → ***Regression / Inference***



Credit: skao.int



A data problem

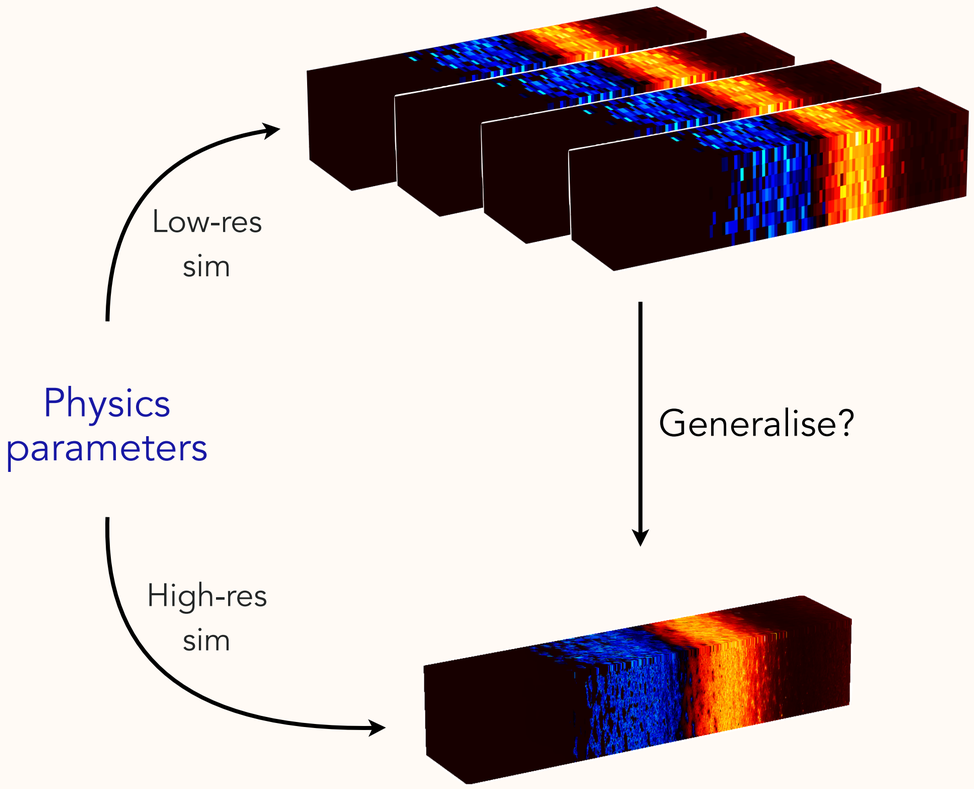
- Lightcones are expensive to simulate and huge
  - → Training data limited by time and memory
  - → But these trade-off with simulation quality (resolution)

- ***Can large datasets of cheap images help?***

  i.e. Pretrain network on low-res, adapt to high-res

- Need to avoid overfitting to mis-modelled physics
  - → ***Self-supervised learning:***

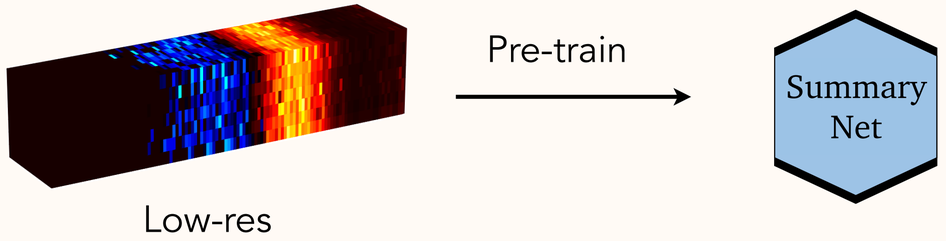
    Train a network to produce informative representations without using labels (physics parameters)



Summary network setup

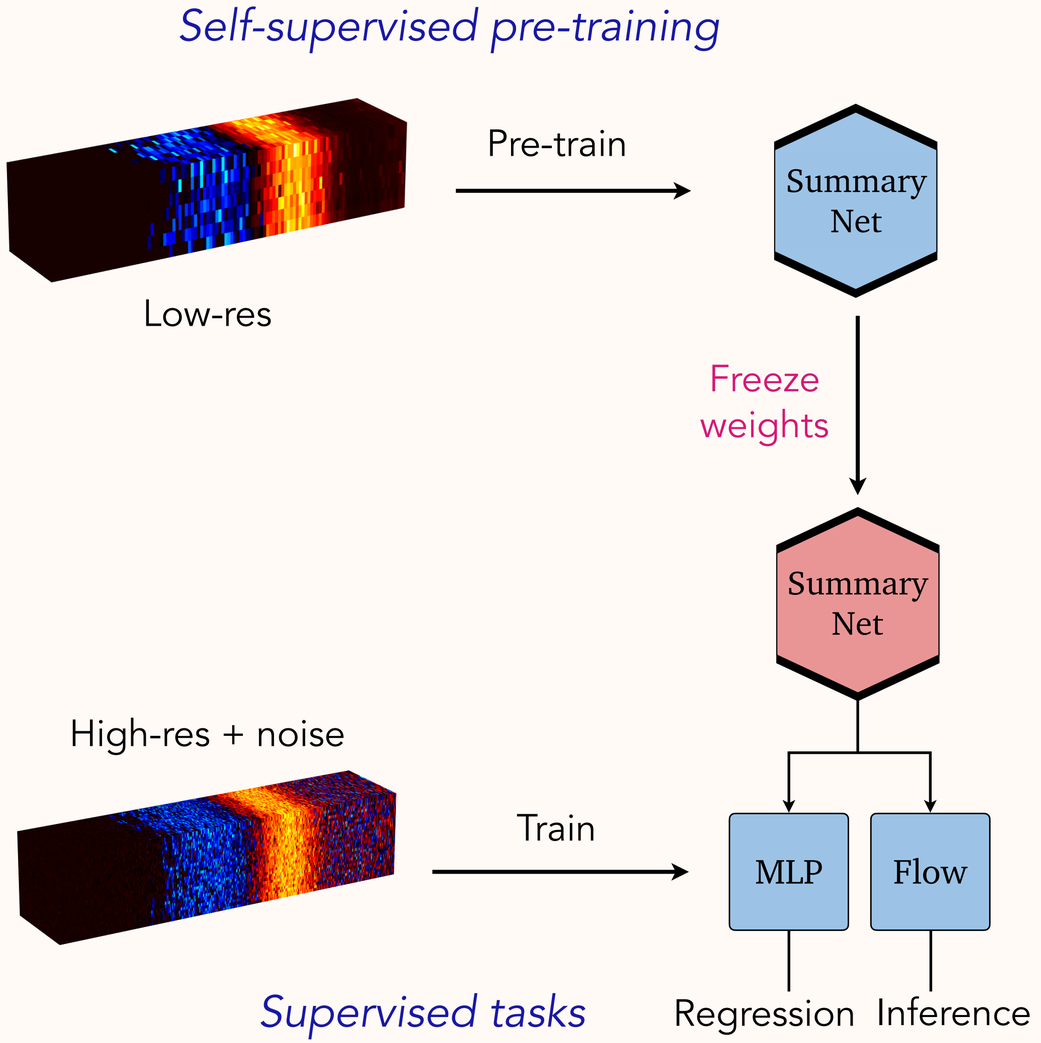
1. Train summary network on low-res simulations

Self-supervised pre-training



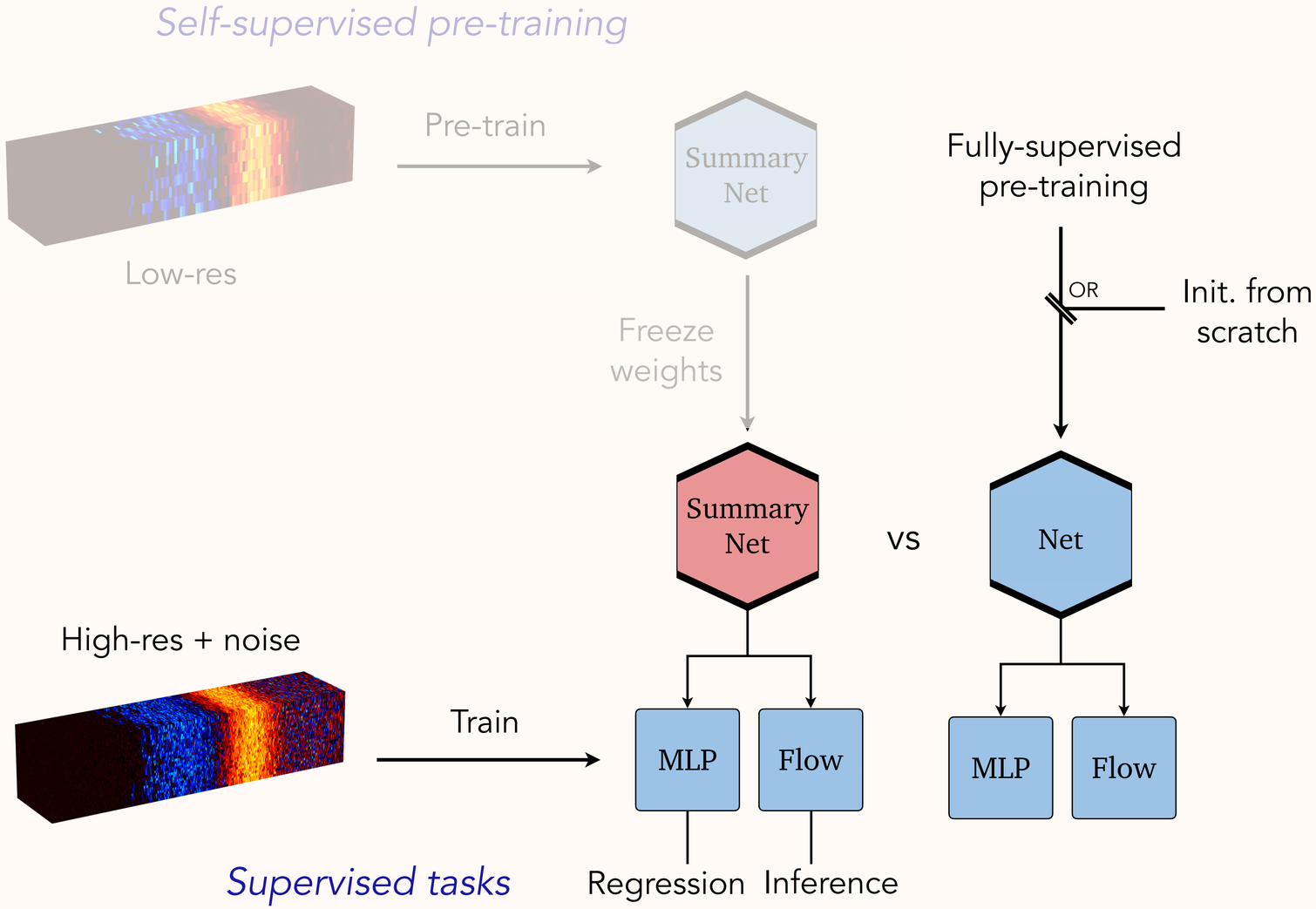
Summary network setup

1. Train summary network on low-res simulations
2. Freeze weights and pair with task head
3. Train on summaries of high-res images



Summary network setup

1. Train summary network on low-res simulations
2. Freeze weights and pair with task head
3. Train on summaries of high-res images
4. Compare to
   - Training from scratch
   - Pre-training with regression

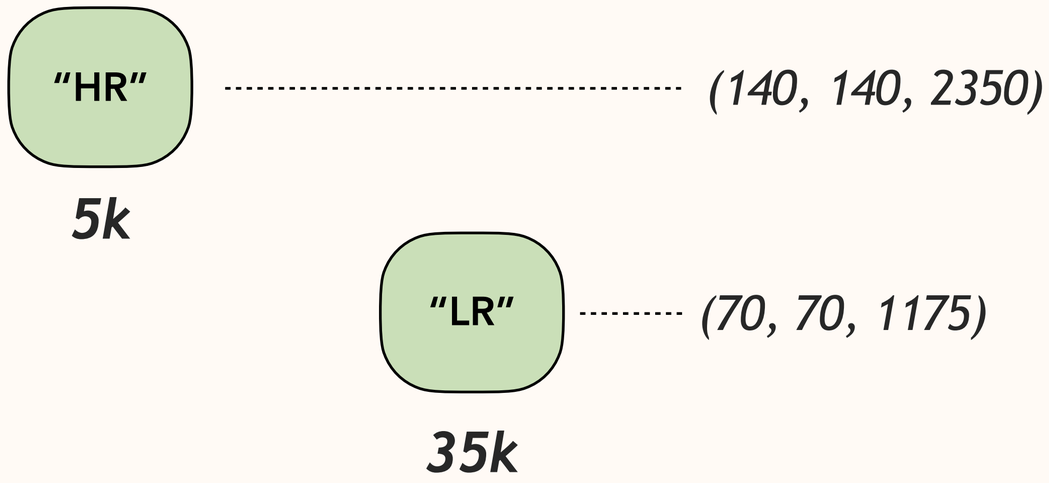


Light cone datasets

- Sample cosmo/astro params from wide priors

$$y \equiv \{m_{\rm WDM}, \Omega_{\rm m}, E_0, L_{\rm X}, T_{\rm vir}, \zeta\}$$

- Simulate lightcones at ***two resolutions***.
  - Box size 200x200 Mpc^2 and 6 < z < 35



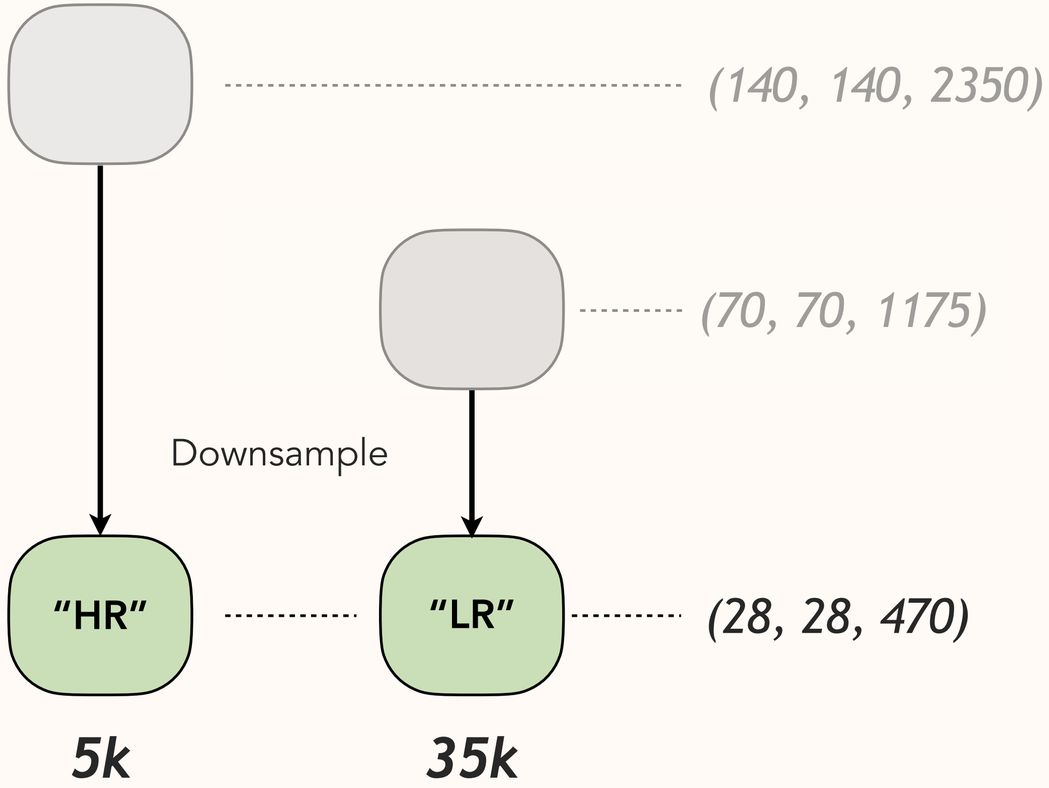
Light cone datasets

- Sample cosmo/astro params from wide priors

$$y \equiv \{m_{\rm WDM}, \Omega_{\rm m}, E_0, L_{\rm X}, T_{\rm vir}, \zeta\}$$

- Simulate lightcones at ***two resolutions***.
  - Box size 200x200 Mpc^2 and $6 < z < 35$

- Downsample to common low res

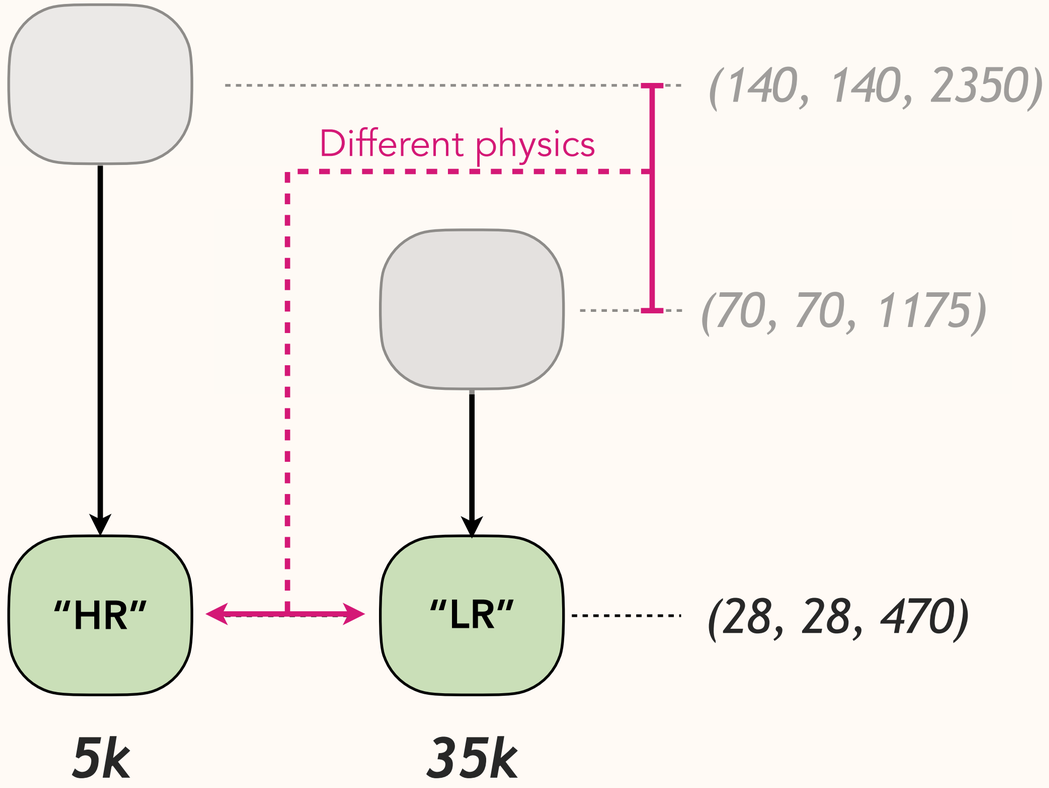


Light cone datasets

- Sample cosmo/astro params from wide priors

$$y \equiv \{m_{\rm WDM}, \Omega_{\rm m}, E_0, L_{\rm X}, T_{\rm vir}, \zeta\}$$

- Simulate lightcones at ***two resolutions***.
  - Box size 200x200 Mpc^2 and $6 < z < 35$
- Downsample to common low res
- ***Note: HR and LR are physically different***
  (Cannot predict $m_{\rm WDM}$ from LR light cone)

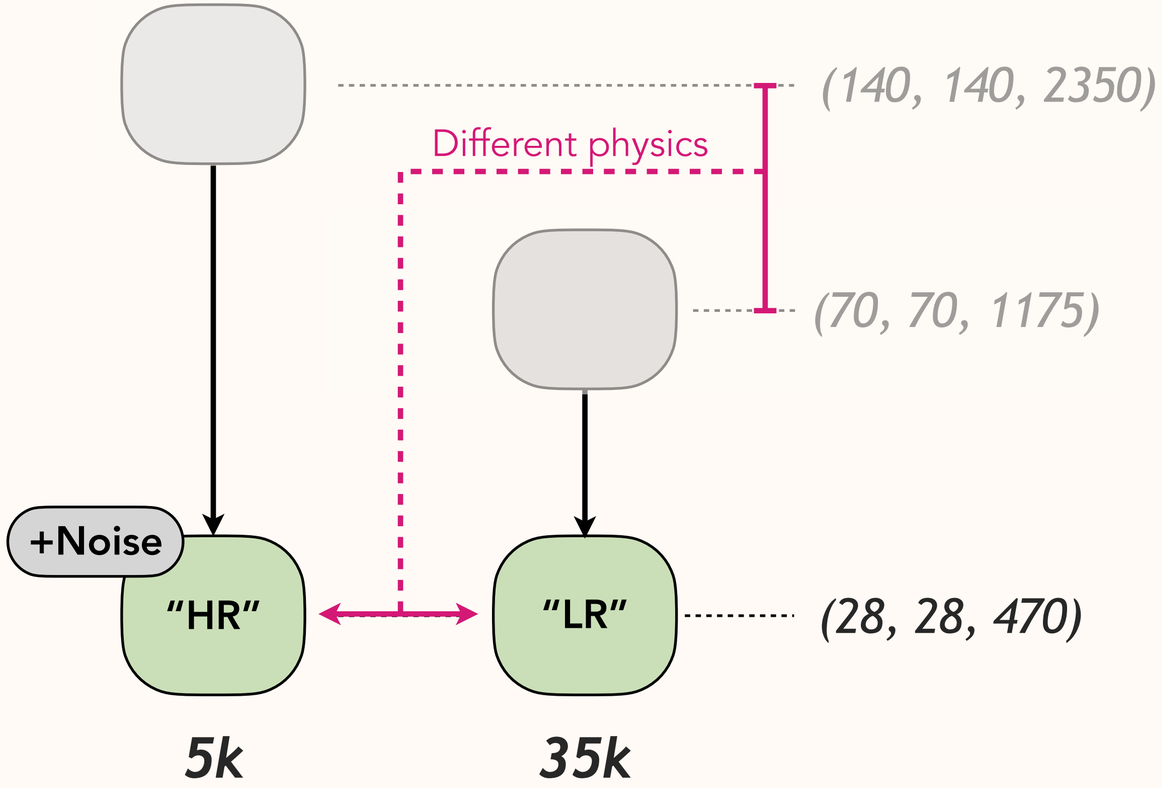


Light cone datasets

- Sample cosmo/astro params from wide priors

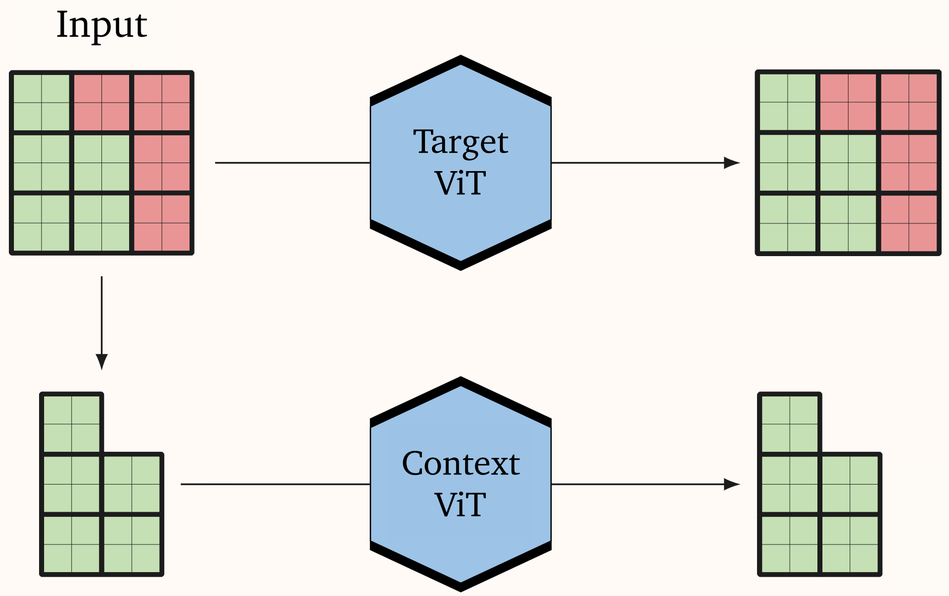
$$y \equiv \{m_{\rm WDM}, \Omega_{\rm m}, E_0, L_{\rm X}, T_{\rm vir}, \zeta\}$$

- Simulate lightcones at ***two resolutions***.
  - Box size 200x200 Mpc^2 and $6 < z < 35$
- Downsample to common low res
- ***Note: HR and LR are physically different***
  (Cannot predict $m_{\rm WDM}$ from LR light cone)
- Noise added and foreground modes removed in HR lightcones



Self-supervised pre-training

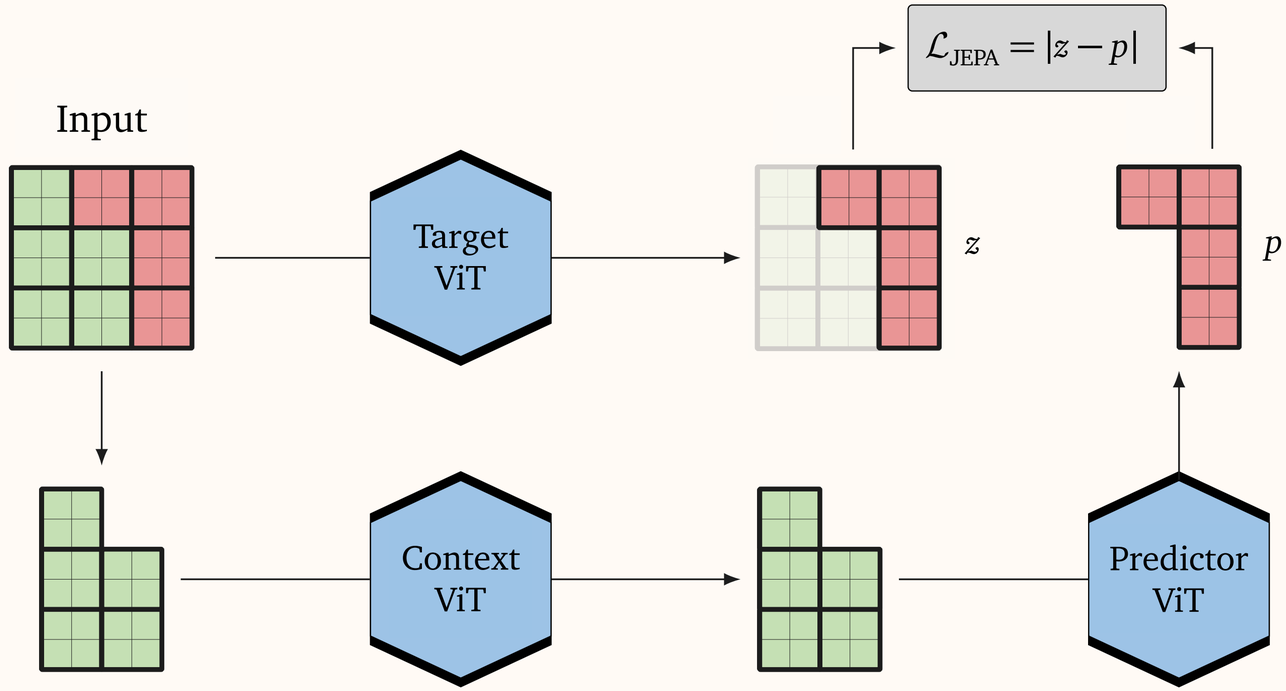
- Twin vision transformers (ViT)
  - "Target": Embed full image
  - "Context": Embed masked image



"Joint-embedding predictive architecture" (JEPA) *arXiv:2301.08243*

Self-supervised pre-training

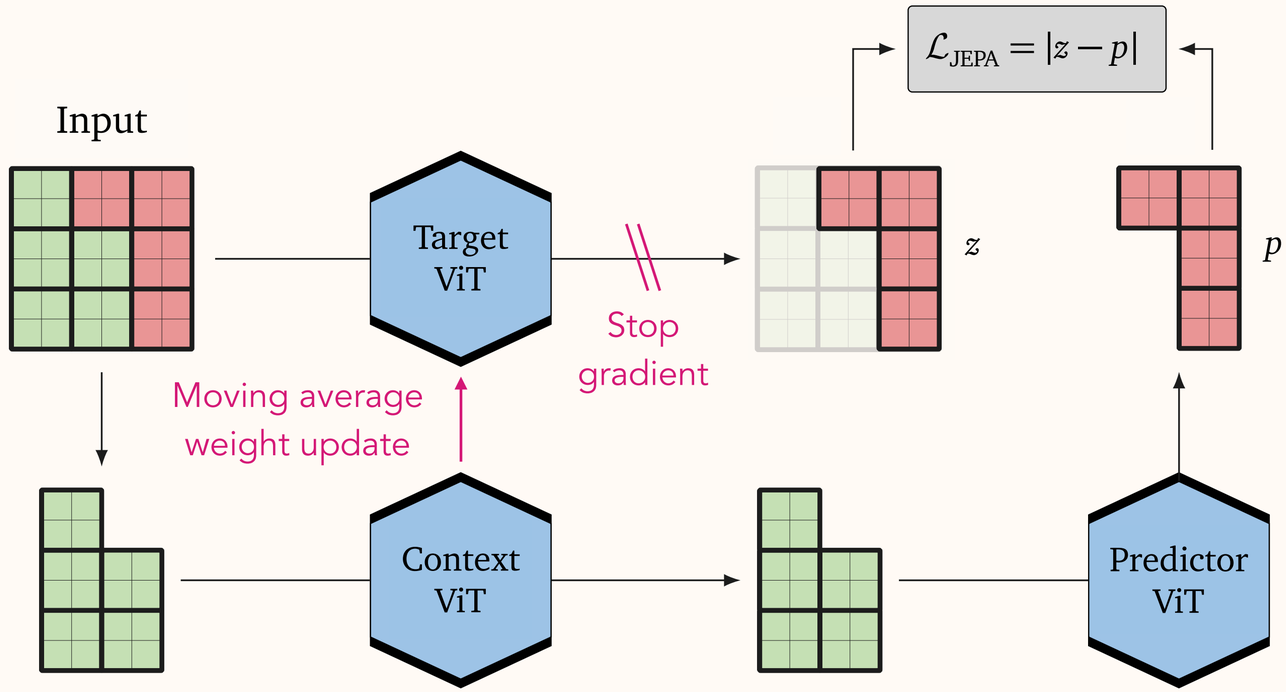
- Twin vision transformers (ViT)
  - "Target": Embed full image
  - "Context": Embed masked image
- Predict embedding of missing patches, given context



"Joint-embedding predictive architecture" (JEPA) *arXiv:2301.08243*

Self-supervised pre-training

- Twin vision transformers (ViT)
  - "Target": Embed full image
  - "Context": Embed masked image
- Predict embedding of missing patches, given context
- Extra mechanisms to prevent collapse

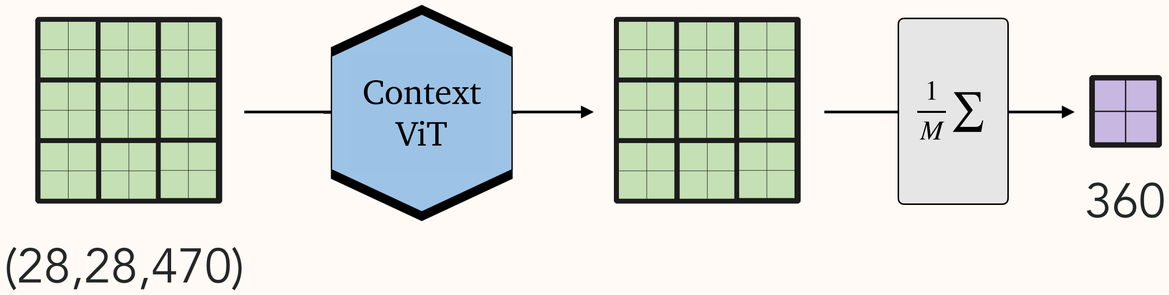


"Joint-embedding predictive architecture" (JEPA) *arXiv:2301.08243*

Self-supervised pre-training

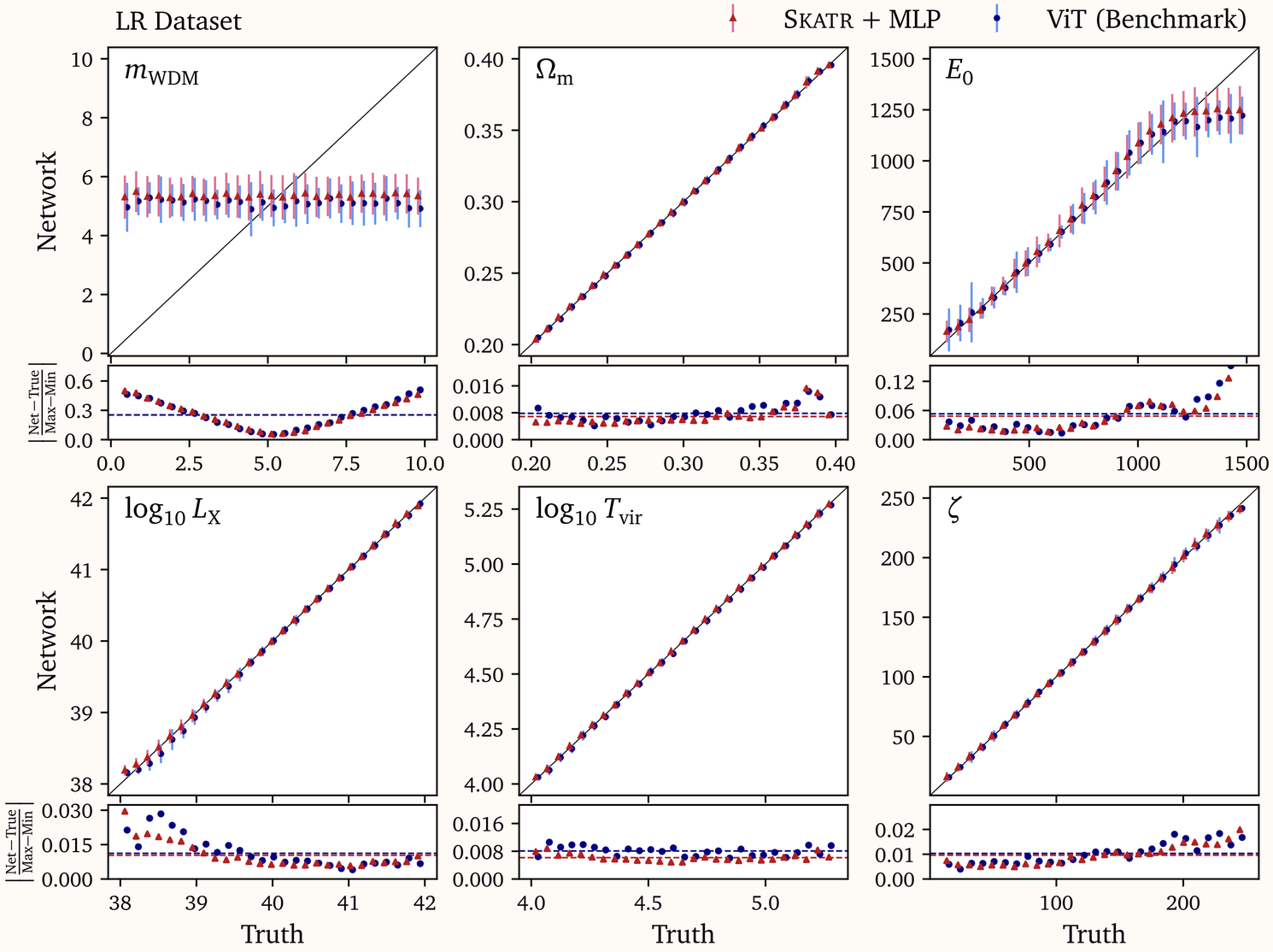
- Twin vision transformers (ViT)
  - "Target": Embed full image
  - "Context": Embed masked image
- Predict embedding of missing patches, given context
- Extra mechanisms to prevent collapse
- Take context ViT as summary network
  - **Compression factor ~ 1000x**

SKATR = "SKA Transformer"



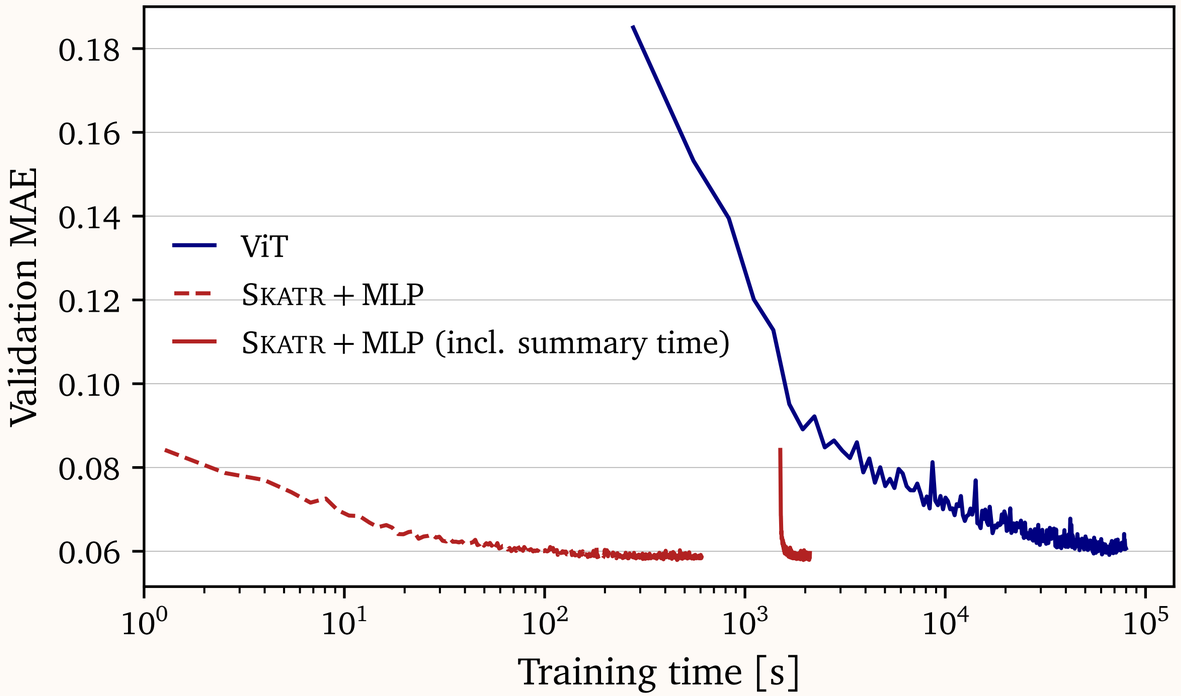
Optimality I: Regression in domain

- Predict parameters using LR images
- ViT regresses perfectly, except for m_{WDM} and E_0
- SKATR matches performance despite being frozen
- Thus all information relevant to regression is retained



Quick aside: Training time

- SKATR calls are amortised (once upfront)
  → Drastic speed up in downstream training
- Pure training is *200x faster*
- Still ***50x faster*** including summarisation
- Fewer trainable parameters
  → Greater stability



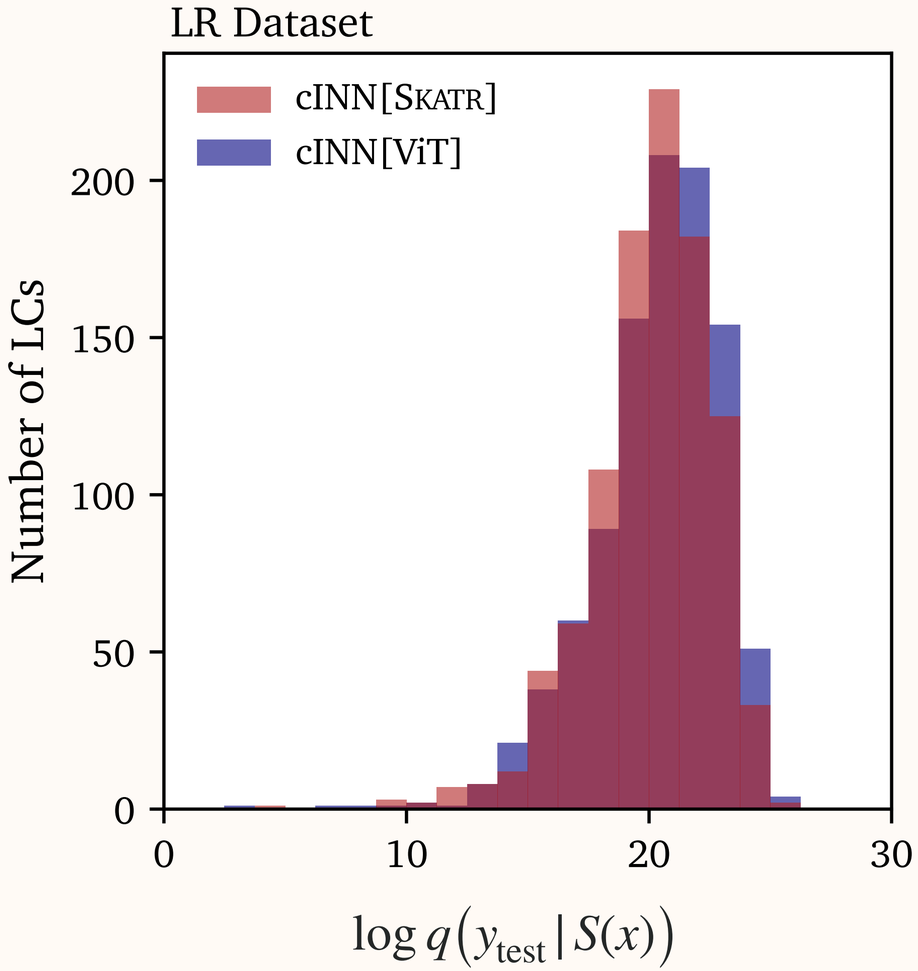
Optimality II: Inference in domain

- Harder task: ***Neural Posterior Estimation***
- Fit ***normalising flow*** to conditional distribution of parameters:

$$L = -\left\langle \log q\big(y \,|\, S(x)\big) \right\rangle_{p_{\text{data}}(x,y)}$$

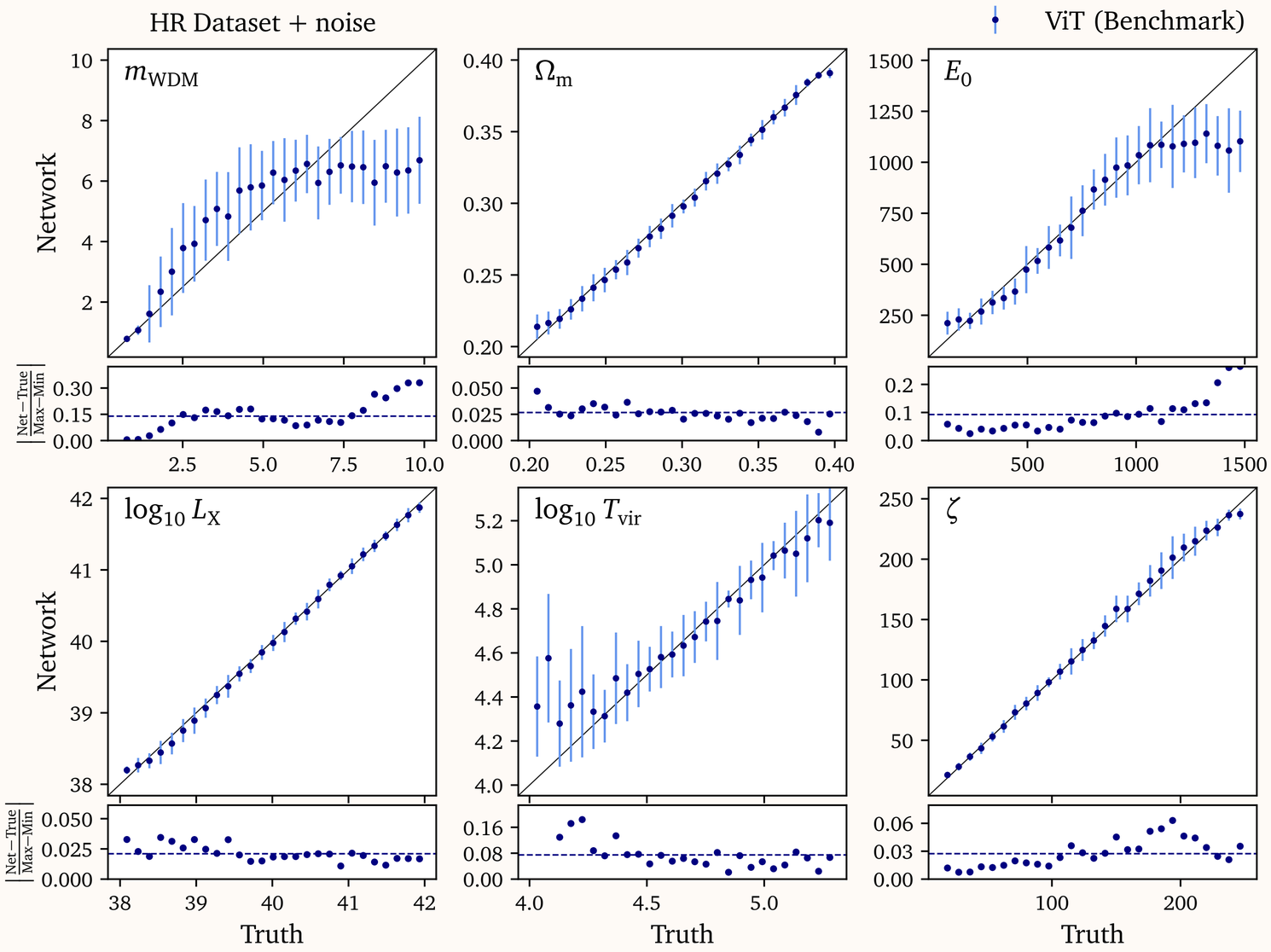
SKATR (frozen) vs ViT (trained)

- Posterior probabilities matched in test set
  - → SKATR summary ***maximally informative***



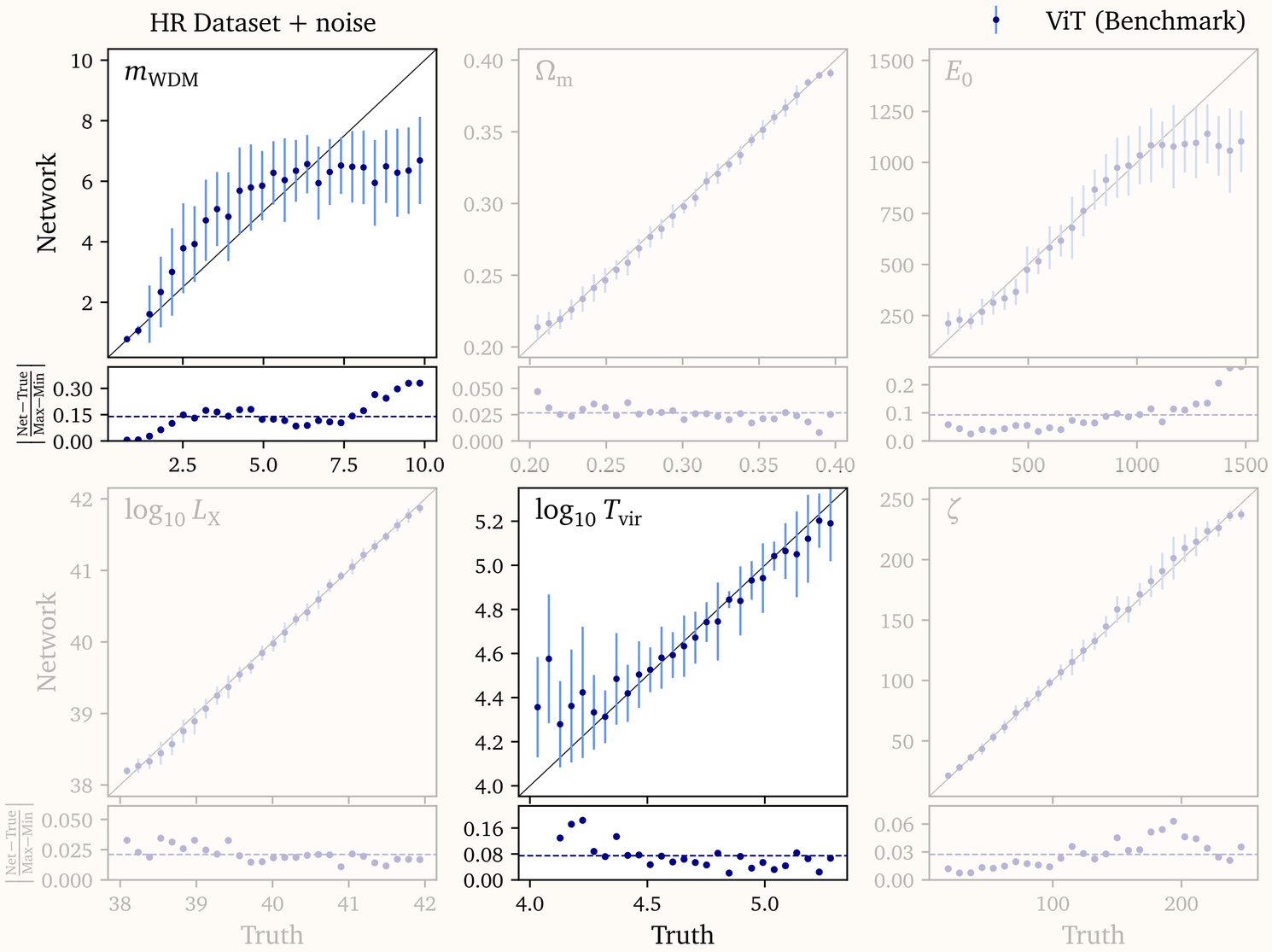
Generalisation: Regression out of domain

- Test regression on HR-simulated data
- New parameter correlations
  - m_{WDM} predictable
  - T_{vir} and m_{WDM} degenerate



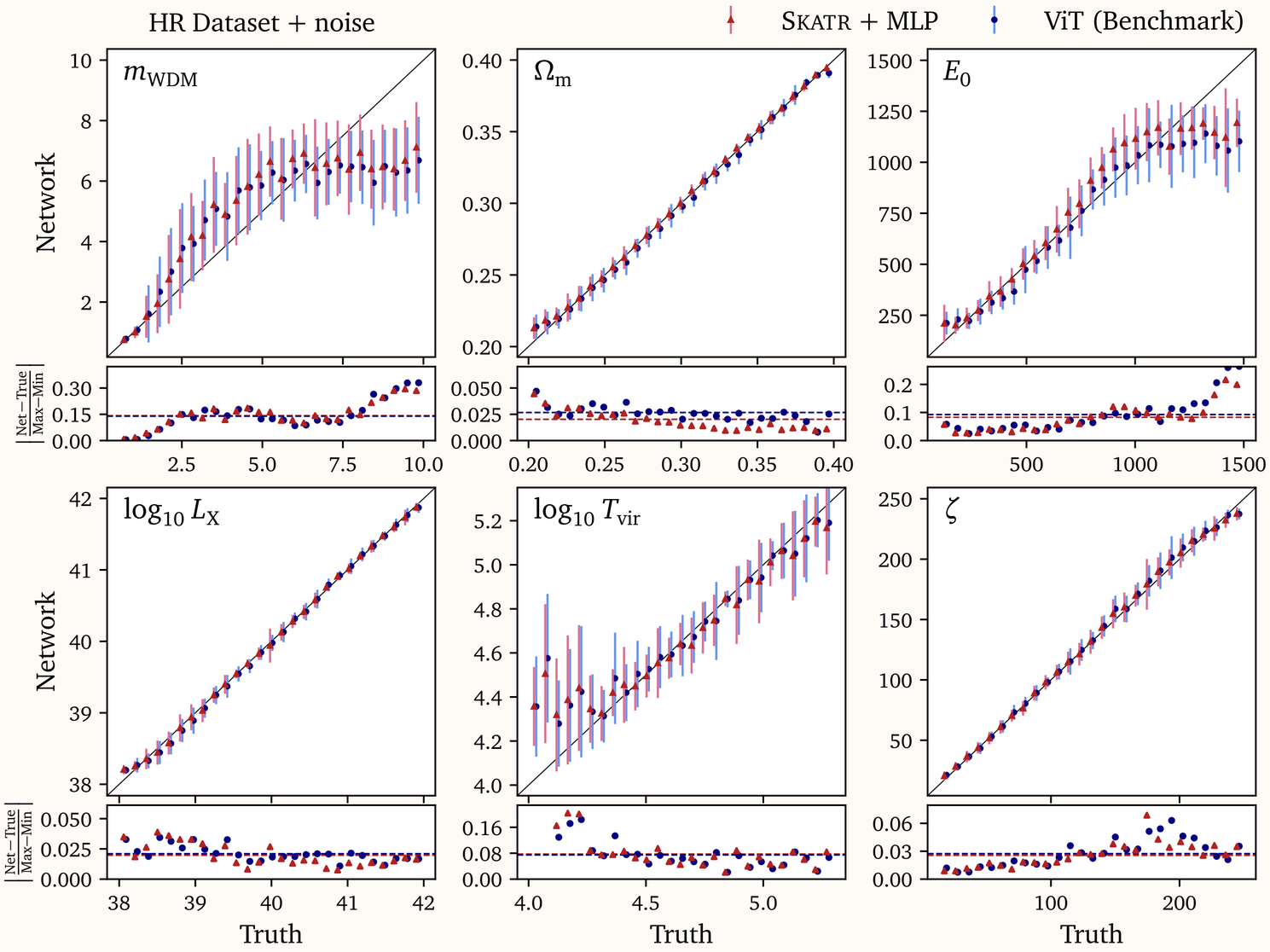
Generalisation: Regression out of domain

- Test regression on HR-simulated data
- New parameter correlations
  - m_{WDM} predictable
  - T_{vir} and m_{WDM} degenerate



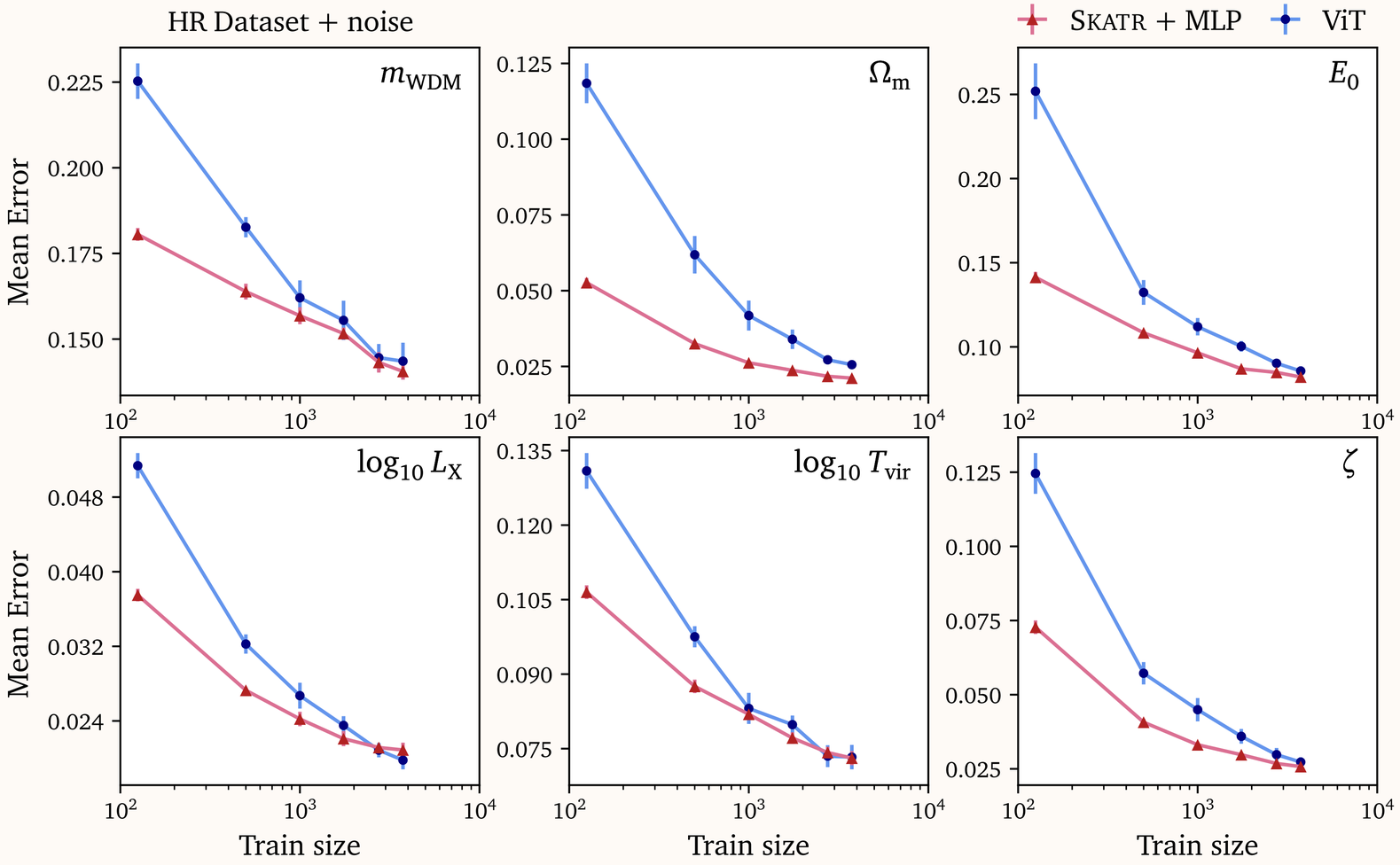
Generalisation: Regression out of domain

- Test regression on HR-simulated data
- New parameter correlations
  - m_{WDM} predictable
  - T_{vir} and m_{WDM} degenerate
- Frozen SKATR matches trained ViT
- Better performance for Ω_{m} explained by large pre-training set.



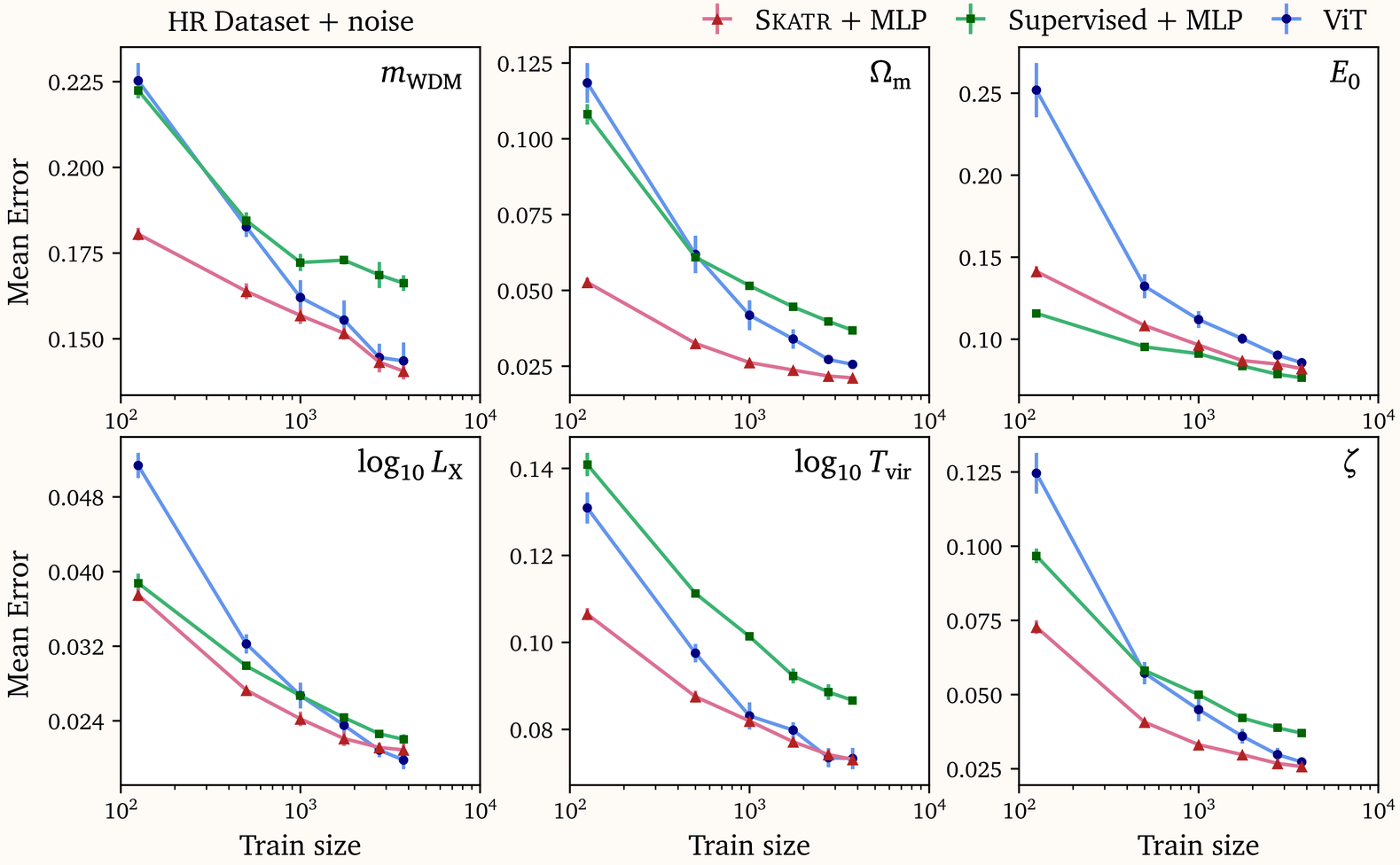
Data efficiency

- Light cone datasets limited by:
  - long simulation time
  - large memory footprint
- SKATR summary best when downstream data is limited



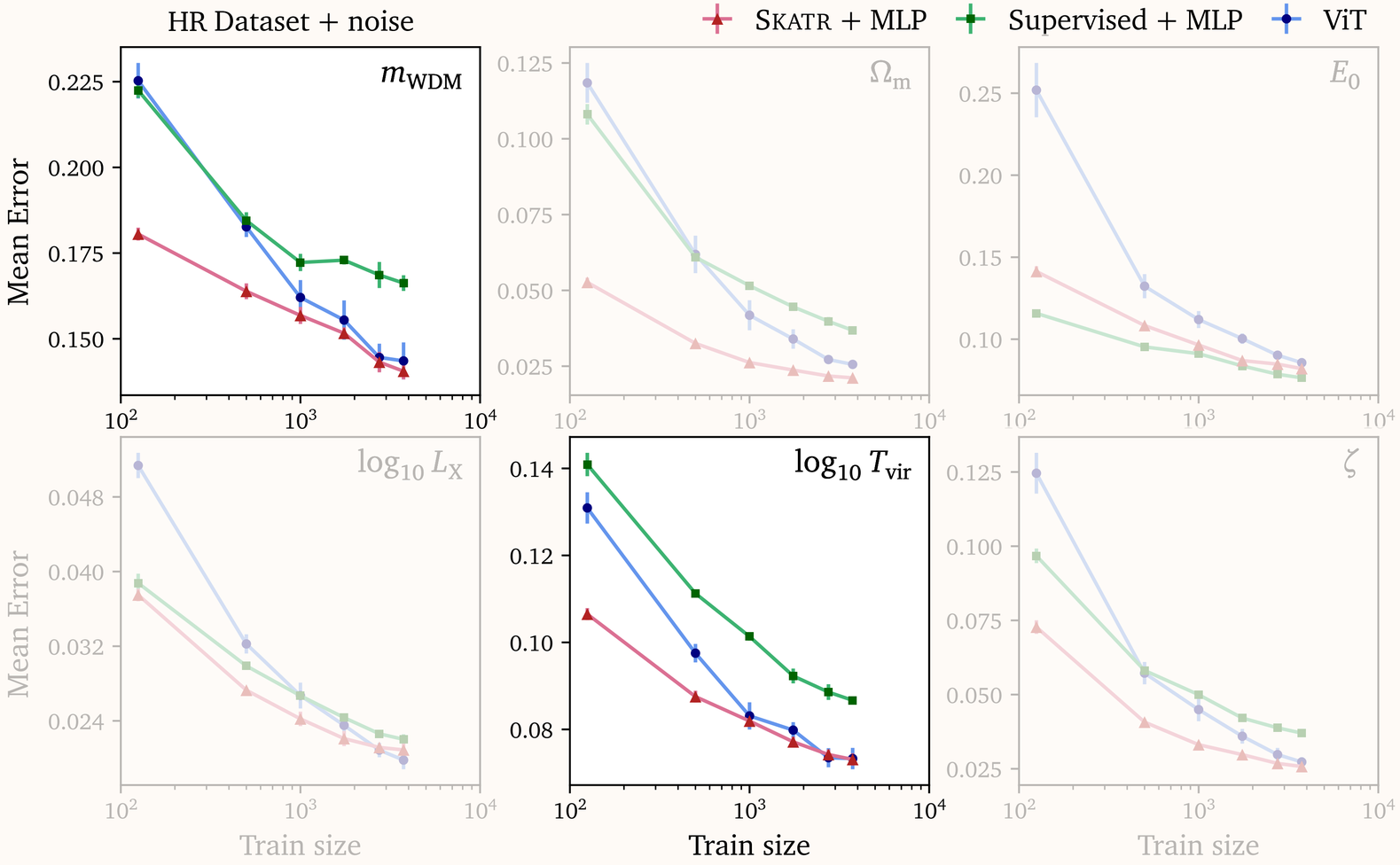
Data efficiency

- Light cone datasets limited by:
  - long simulation time
  - large memory footprint
- SKATR summary best when downstream data is limited
- Regression-pretrained summary fails to generalise
  - Worst for m_{WDM} and T_{vir} .



Data efficiency

- Light cone datasets limited by:
  - long simulation time
  - large memory footprint
- SKATR summary best when downstream data is limited
- Regression-pretrained summary fails to generalise
  - Worst for m_{WDM} and T_{vir} .

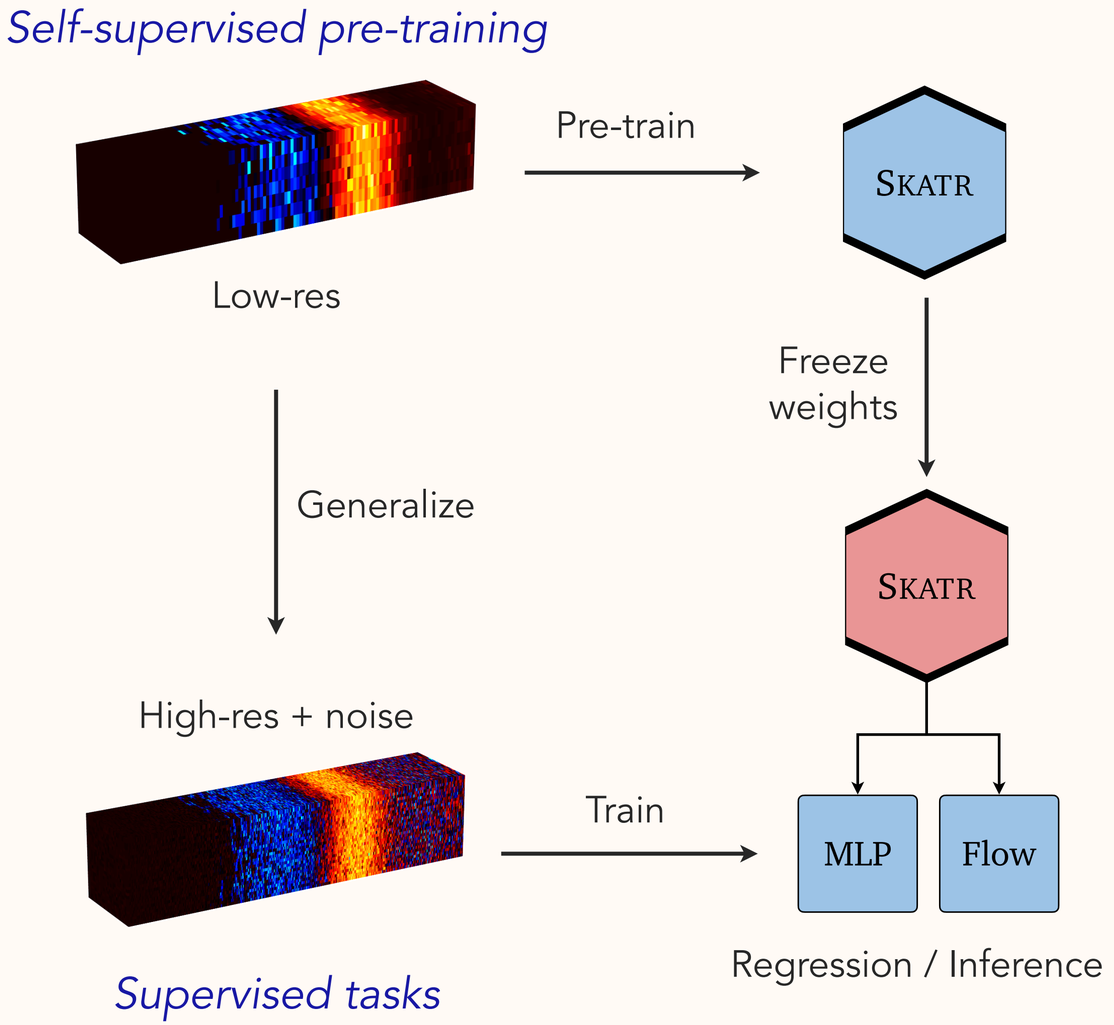


Conclusions and Outlook

- Developed ***SKATR***
  A self-supervised vision transformer for 21cm images

- While frozen, SKATR summary…
  - ★ *Retains physical information*
  - ★ *Allows fast training*
  - ★ *Generalises*
  - ★ *Copes with limited data*
  - ★ *Outperforms fully-supervised summary*

- Read more:
  - Paper: *SciPost Phys. 18, 155* (*arXiv:2410.18899*)
  - Code: *github.com/heidelberg-hepml/skatr*

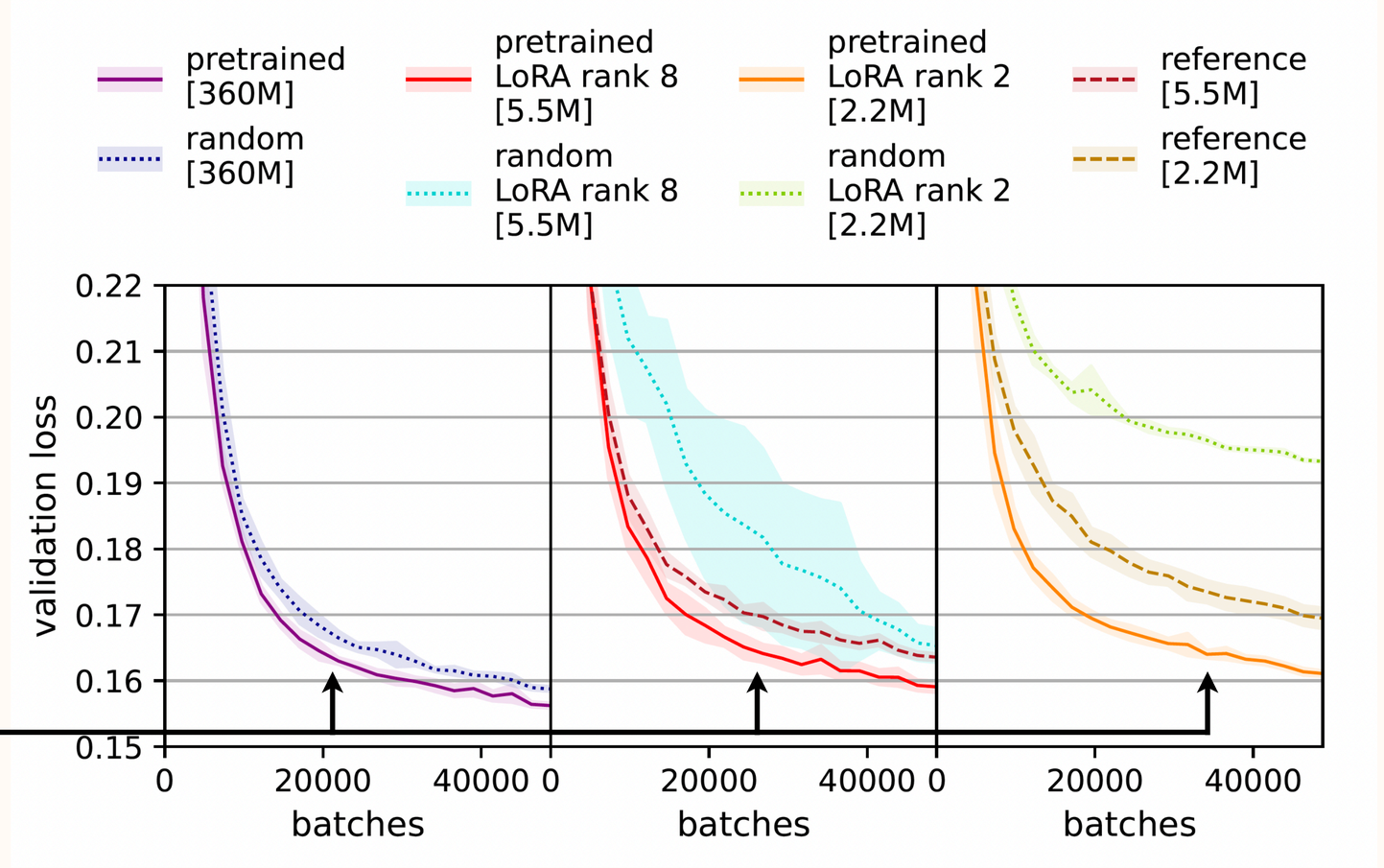


Paper Advertisement

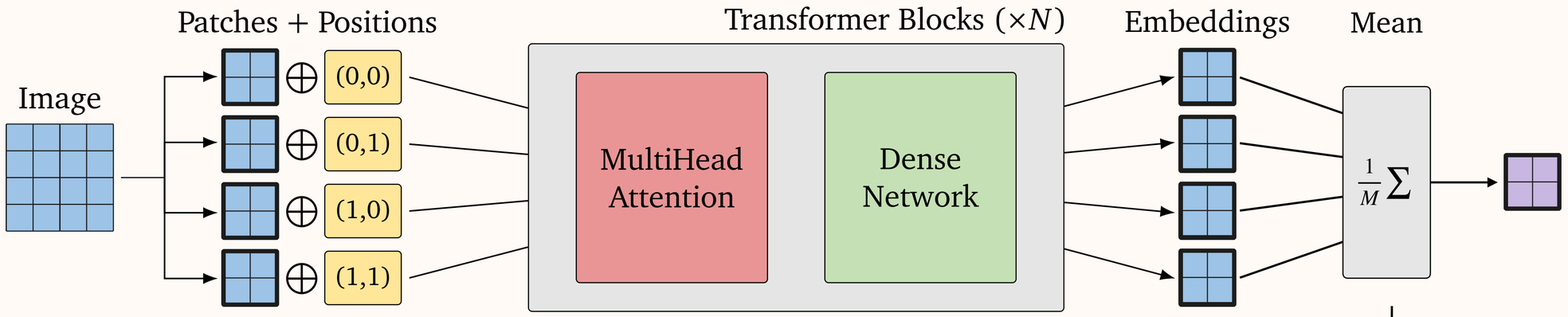
Work with Daniel Schiller et al.: *"Large Language Models — the Future of Fundamental Physics?"*

Forecasting lightcones
with next-token prediction

Read the details
arXiv:2506.14757



Backup: Vision transformer



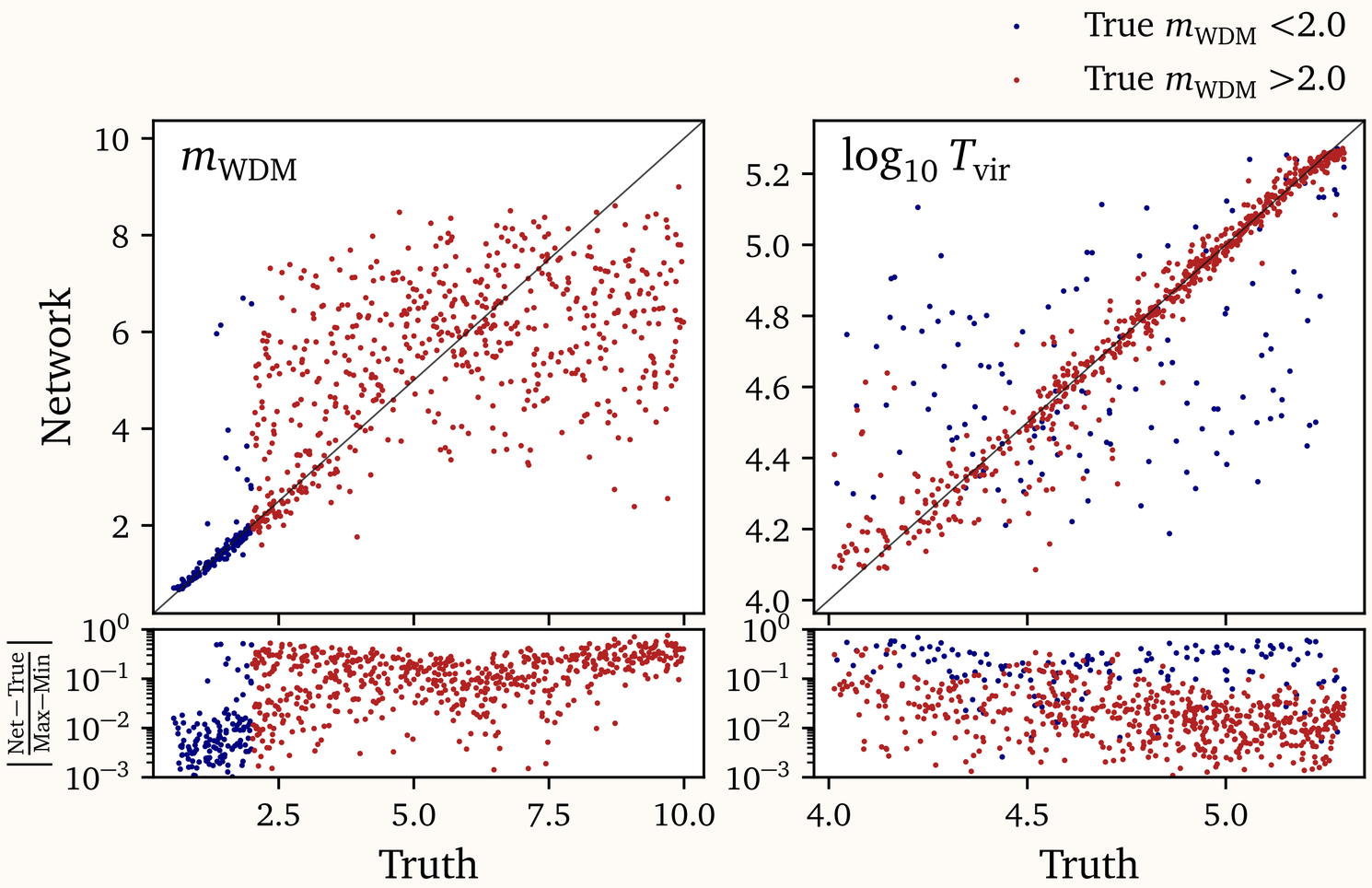
- Divide image into ***patches***
- Shared embedding into ***d*** dimensions
- Encode patch locations into embeddings
- Process with transformer blocks

Probe long-range correlations

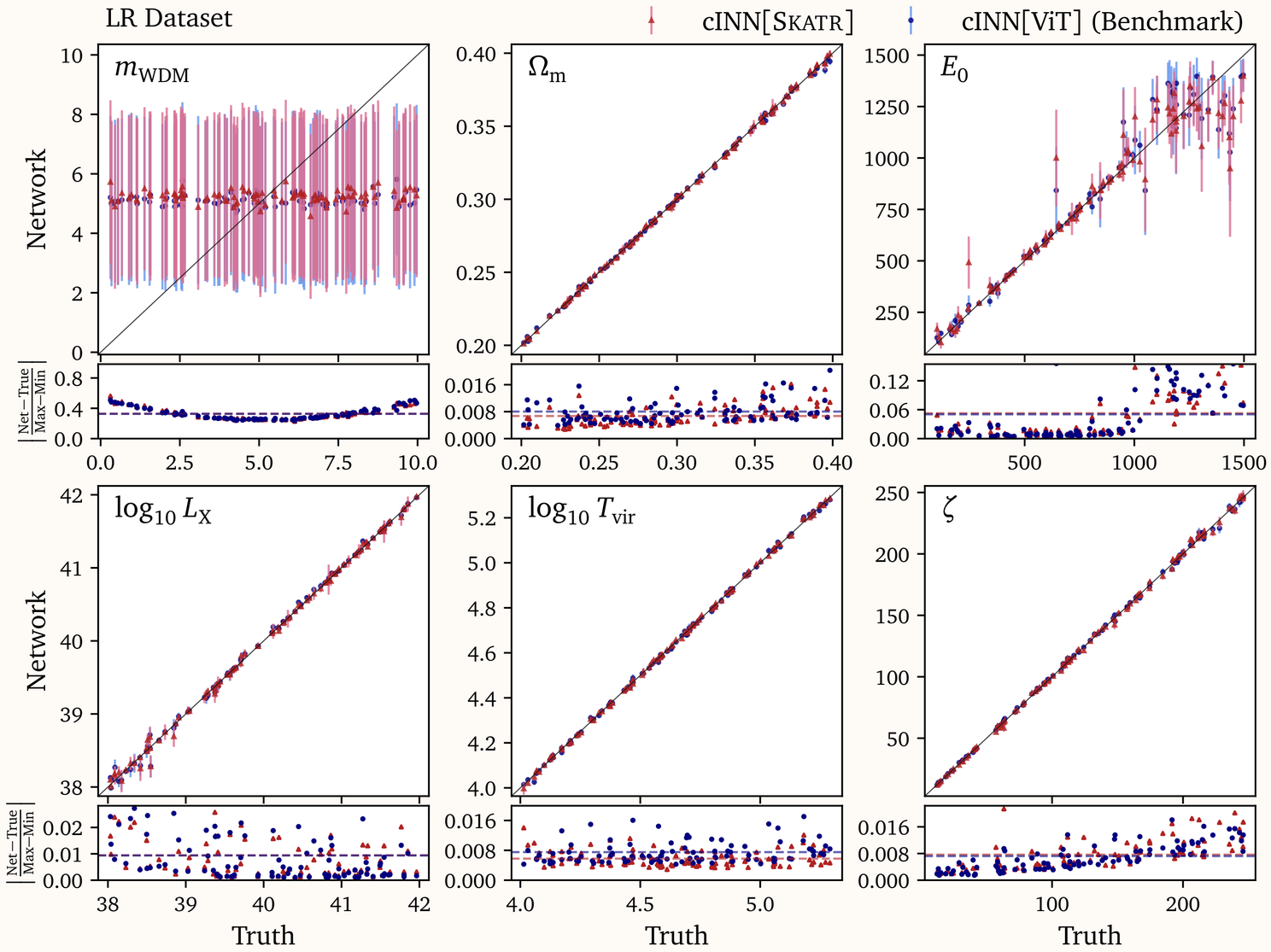
Output is a set of embeddings

Global feature by averaging patches
(e.g. for regression)

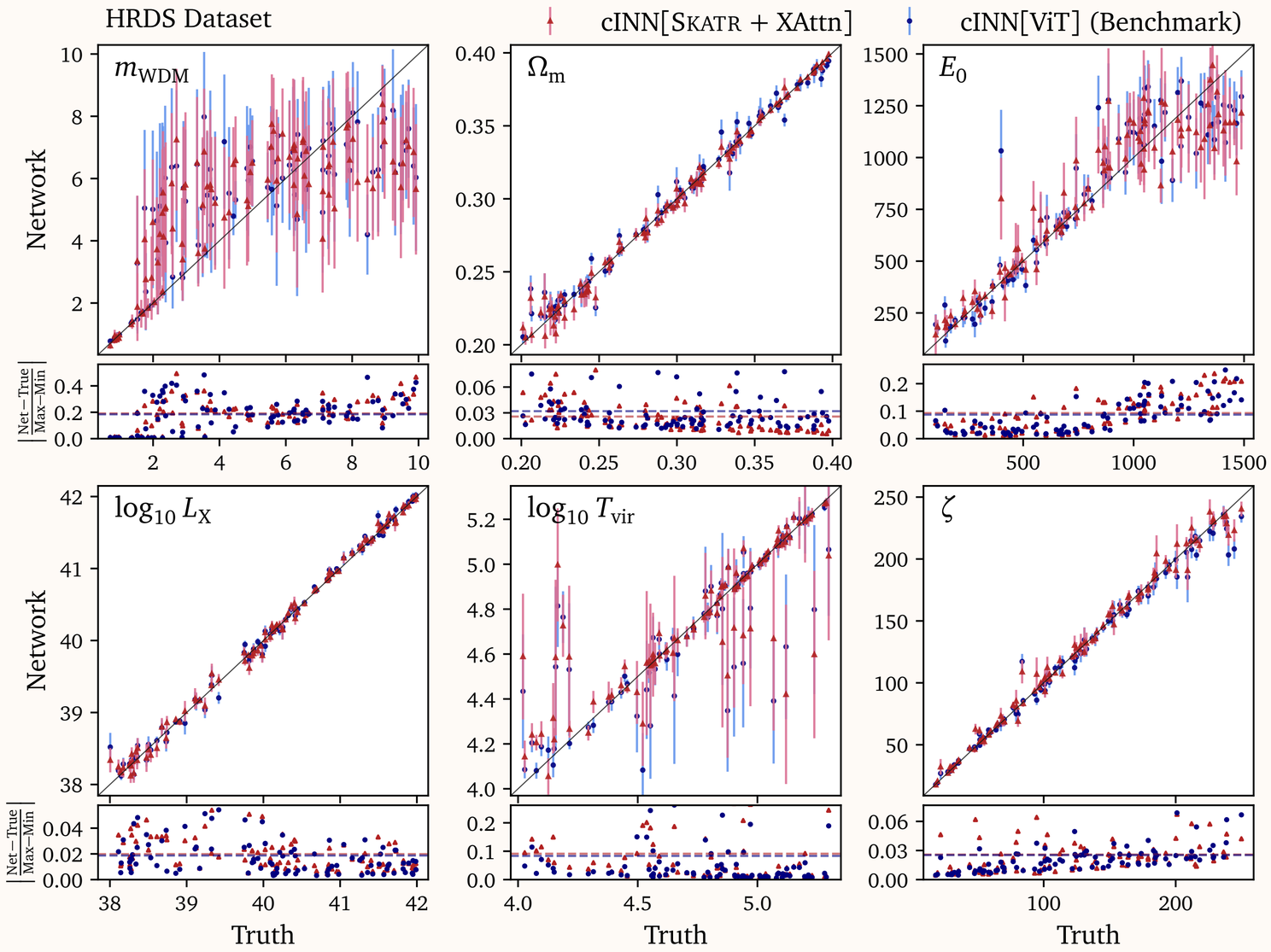
Backup: Parameter degeneracy at HR



Backup: 1D Maringal Posteriors (LR)



Backup: 1D Maringal Posteriors (HR)



Backup: Adapting to full resolution

- Evaluating SKATR on full res (140,140,2350) light cones possible, but
  - attention becomes expensive
  - new physical scale for patches
- Fixing physical size of patches requires training an embedding layer, which is inefficient
- Solution: Upsample LR light cones to HR during pre-training
- Most parameters still recovered well, but ζ is difficult