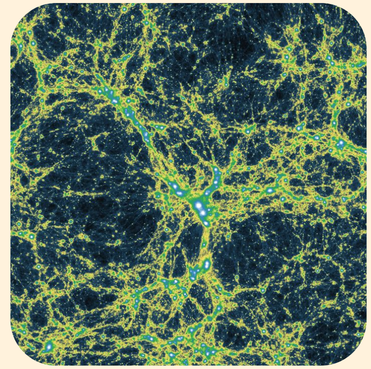
BERKELEY CENTER *for* COSMOLOGICAL PHYSICS

# AI-assisted emulator for cosmological simulations

Linda Zhenyu Jin
Berkeley Center for Cosmological Physics, UC Berkeley

# Universe(s)





Gas density map for a hydrodynamical simulation with the same random seed (TNG300-1 from TNG website).

## Gas and stellar particles (baryons) are physics we don't know – unfeasibly expensive*

(Magneto)hydrodynamic simulations

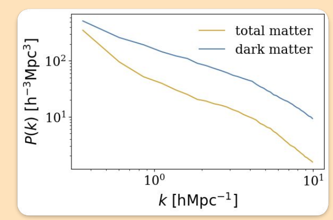
**15%**

~20M node-hours for a reasonable scale to study the large scale structure (100 Mpc)

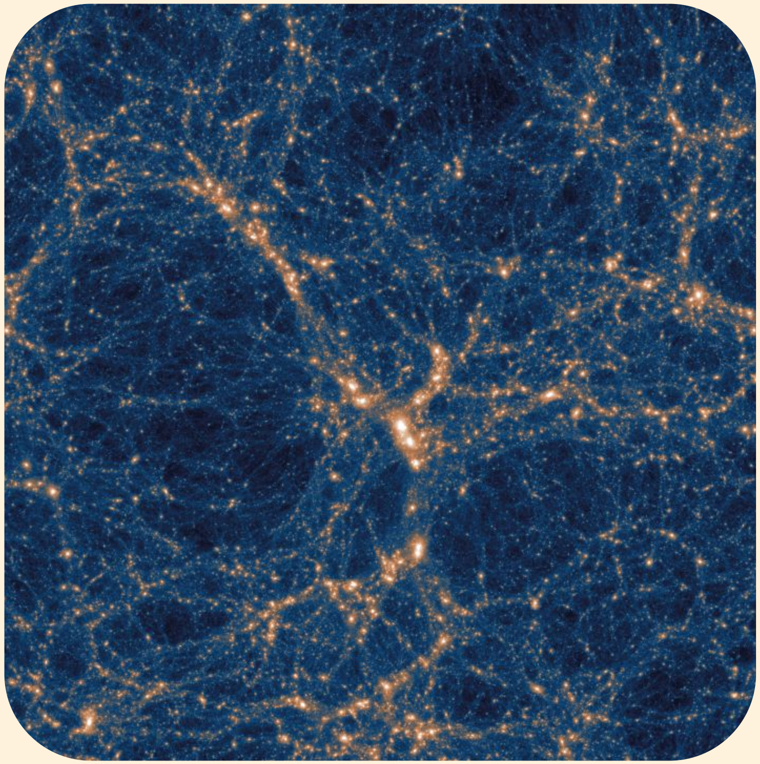
**this work!**

### Modeling baryonic process

An HPC solution to paint gas and stellar properties onto the dark matter field from N-body simulations.




We observe suppression in power spectra that comes from baryons.



Dark matter density map from a N-body simulation at z=0.0 (TNG300-1-Dark from TNG website).

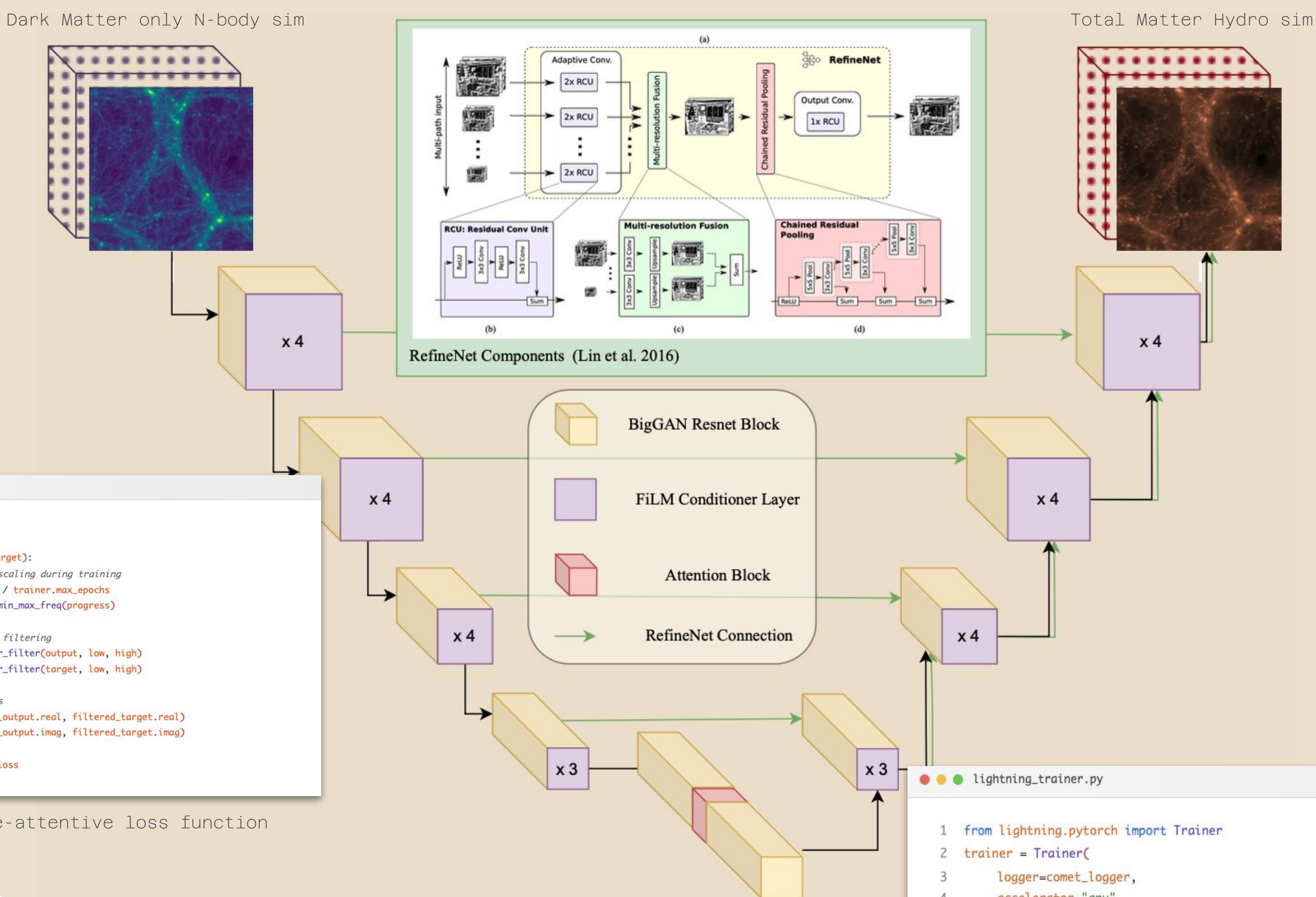
**85%**

## of the total matter content in the Universe is dark matter

Gravity-only N-body simulations

~1M node-hours

*For full simulations over gigaparsec volume with a reasonable resolution.




fourier_loss.py

```python
loss_fn = fourier_loss

def fourier_loss(output, target):
    # Progressive frequency scaling during training
    progress = current_epoch / trainer.max_epochs
    low, high = get_current_min_max_freq(progress)

    # Apply frequency-domain filtering
    filtered_output = fourier_filter(output, low, high)
    filtered_target = fourier_filter(target, low, high)

    # Complex-valued MSE loss
    real_loss = MSE(filtered_output.real, filtered_target.real)
    imag_loss = MSE(filtered_output.imag, filtered_target.imag)

    return real_loss + imag_loss
```

Scale-attentive loss function

CNN

# Multi-cascaded Map Reconstruction for Small-scale Physics

RefineNet (Lin et al. 2016)
BigGAN (Brock, Donahue & Simonyan 2018)

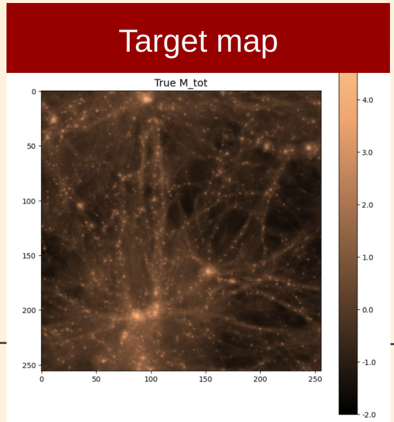
lightning_trainer.py

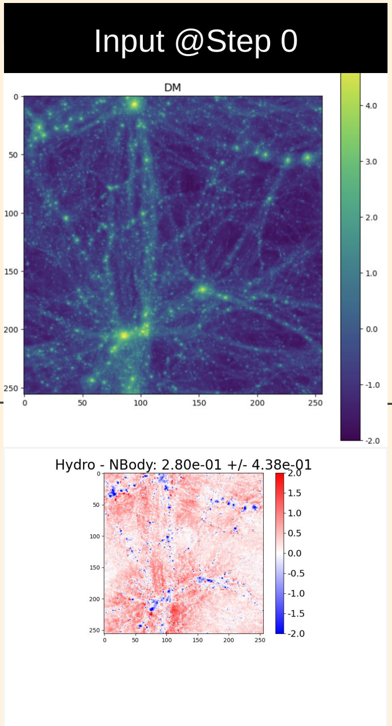
```python
from lightning.pytorch import Trainer
trainer = Trainer(
    logger=comet_logger,
    accelerator="gpu",
    devices=-1, # Uses all available GPUs
    strategy="ddp", # Distributed Data Parallel
    max_epochs=20,
    gradient_clip_val=0.5,
    callbacks=[LearningRateMonitor(),
        latest_checkpoint,
        val_l1_checkpoint
    ],
    # Additional parameters for large dataset handling
    num_nodes=8, # Increase if using multiple nodes
    precision=16, # Mixed precision for memory efficiency
    accumulate_grad_batches=2 # Gradient accumulation for larger effective batch
size
)
```

CNN generative results

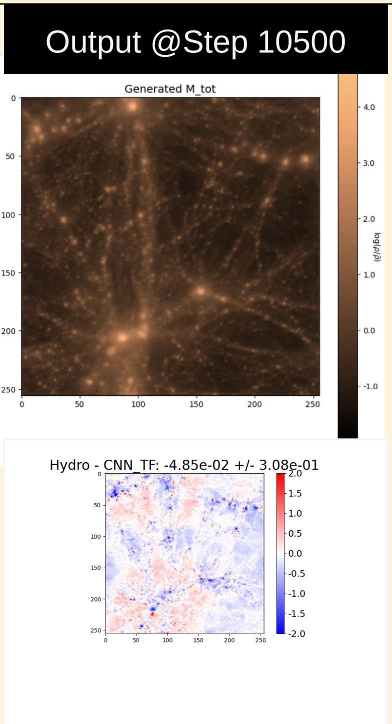
Progress to date

# Results



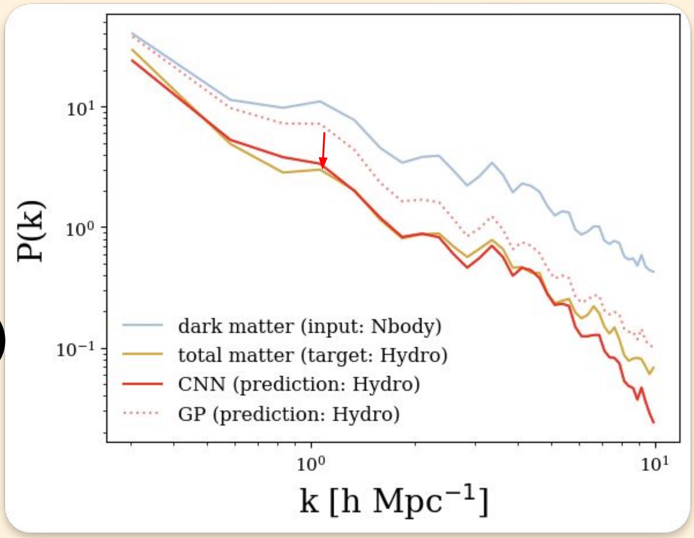



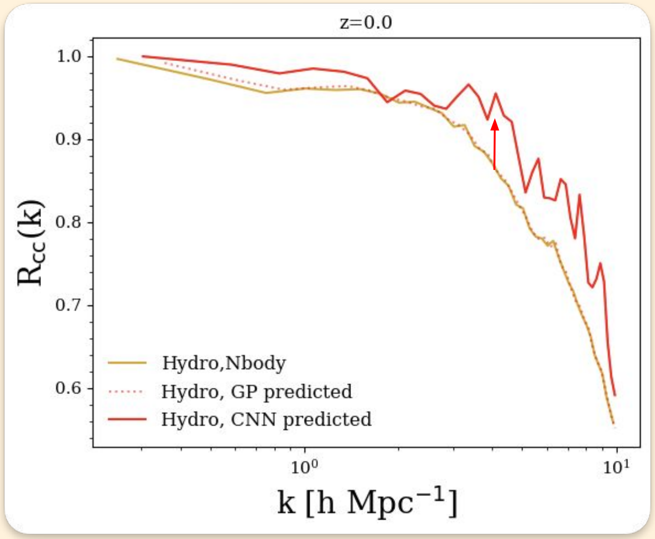

1



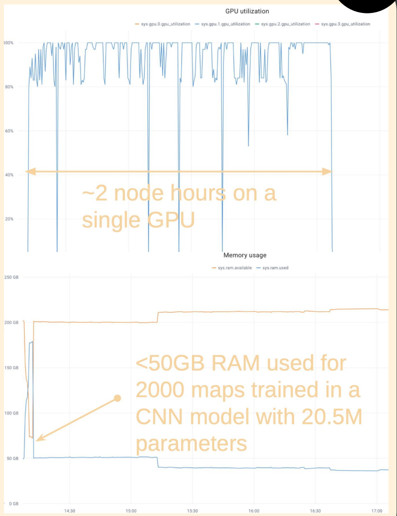

1. Field-level map generation.
2. Performance in fourier space – power spectrum, cross-correlation coefficient with the target hydrodynamical simulation – compared to GP (old model).
3. System metrics – GPU utilization (training for 2D), RAM usage during training.

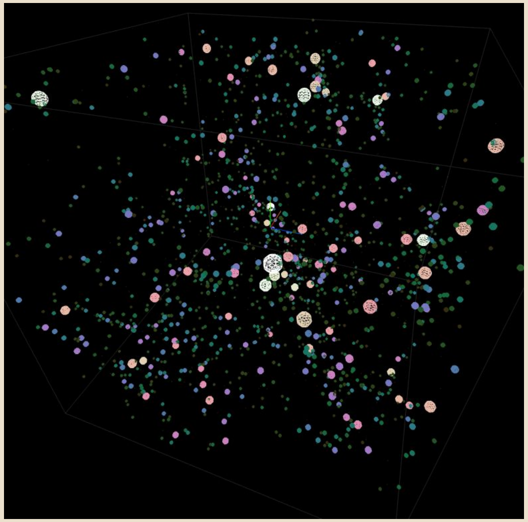
2




3





# Cool 3D Universe Explorer (TNG website)

Questions? Comments?
Please reach out to
lindazjin@berkeley.edu

Thank you!