Cosmological Parameter Estimation with a Denoising U-Net and Patch-Based CNNs

Andry Rafaralahy

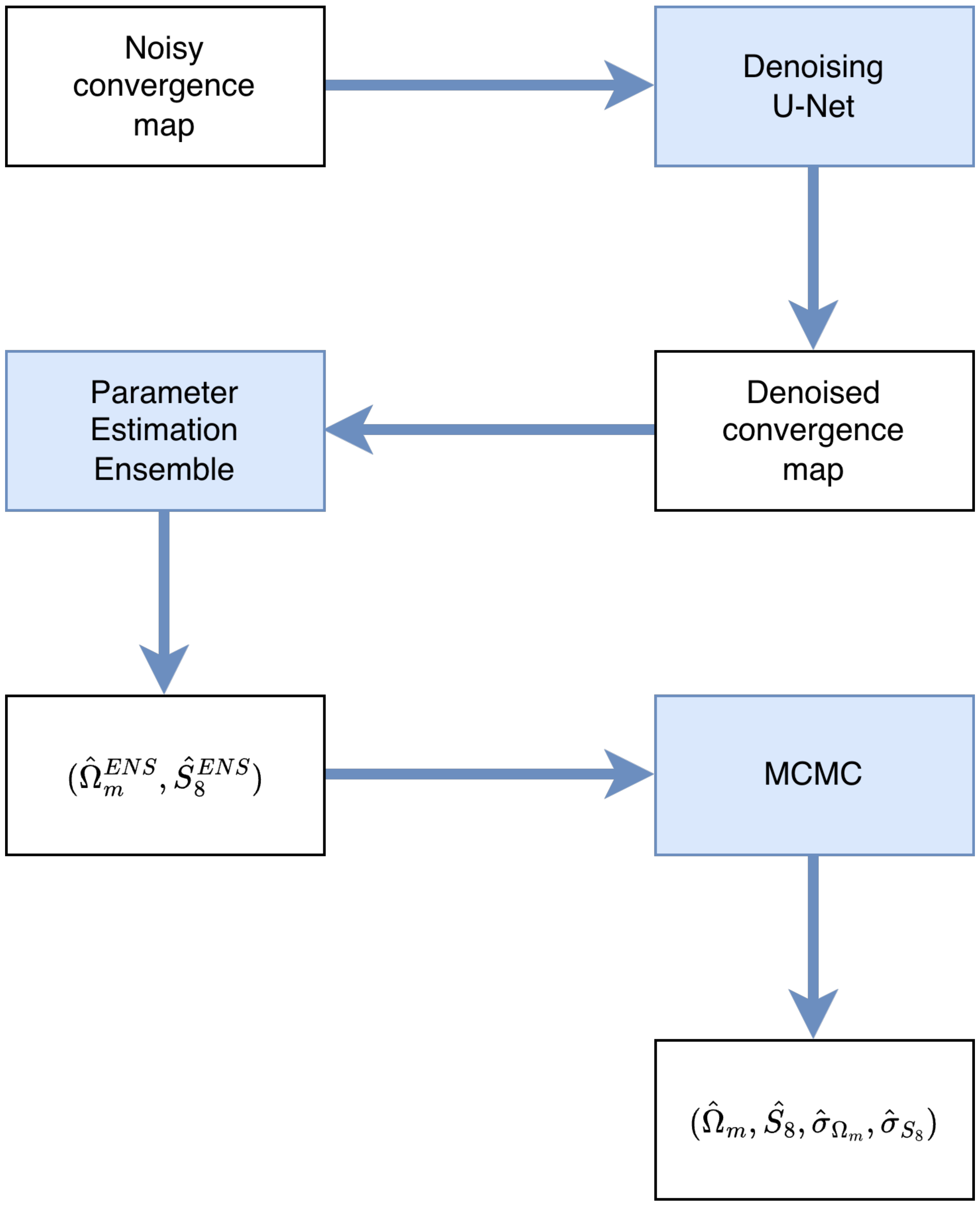
Introduction & Overview

Challenge Goal

- Predict Ω_m, S_8 and uncertainties σ_{Ω_m}, σ_{S_8} from noisy weak-lensing convergence maps

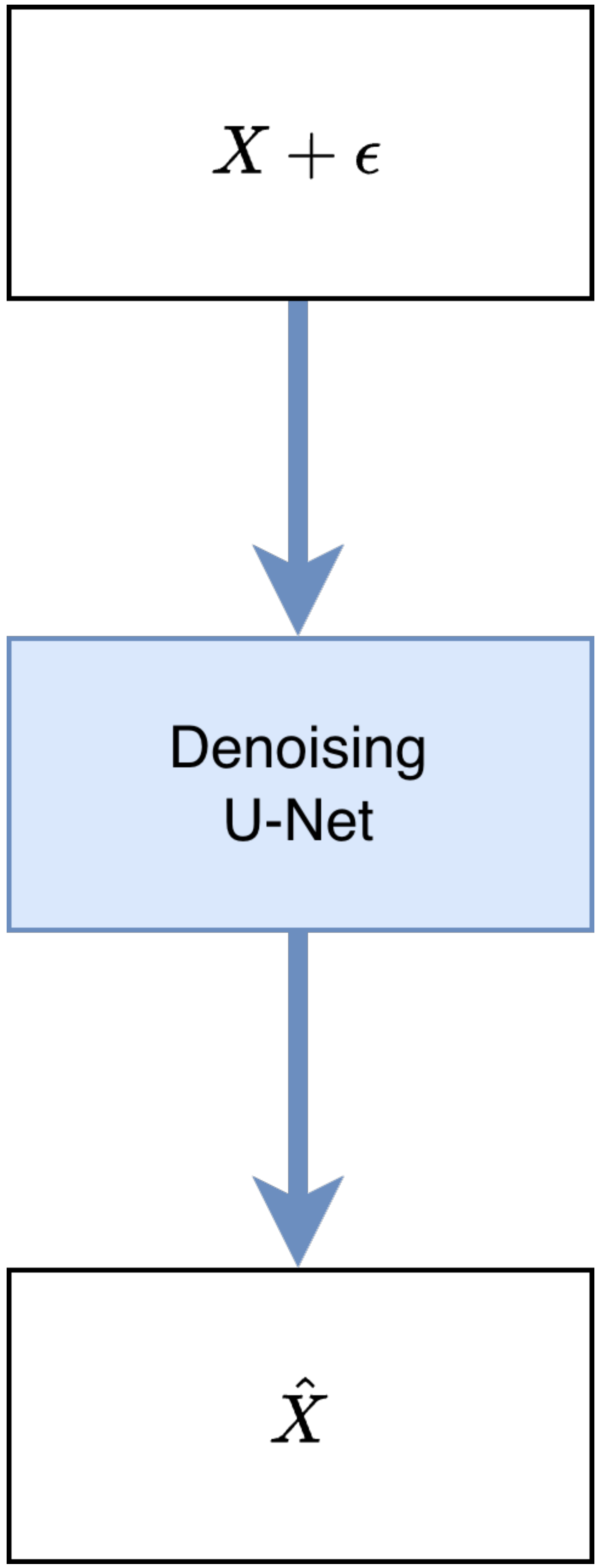
My Approach (8th place):

1. **Denoising U-Net** → denoise the convergence maps
2. **Parameter Estimation Ensemble** → ensemble of CNNs to predict Ω_m and S_8
3. **MCMC** → estimate uncertainties



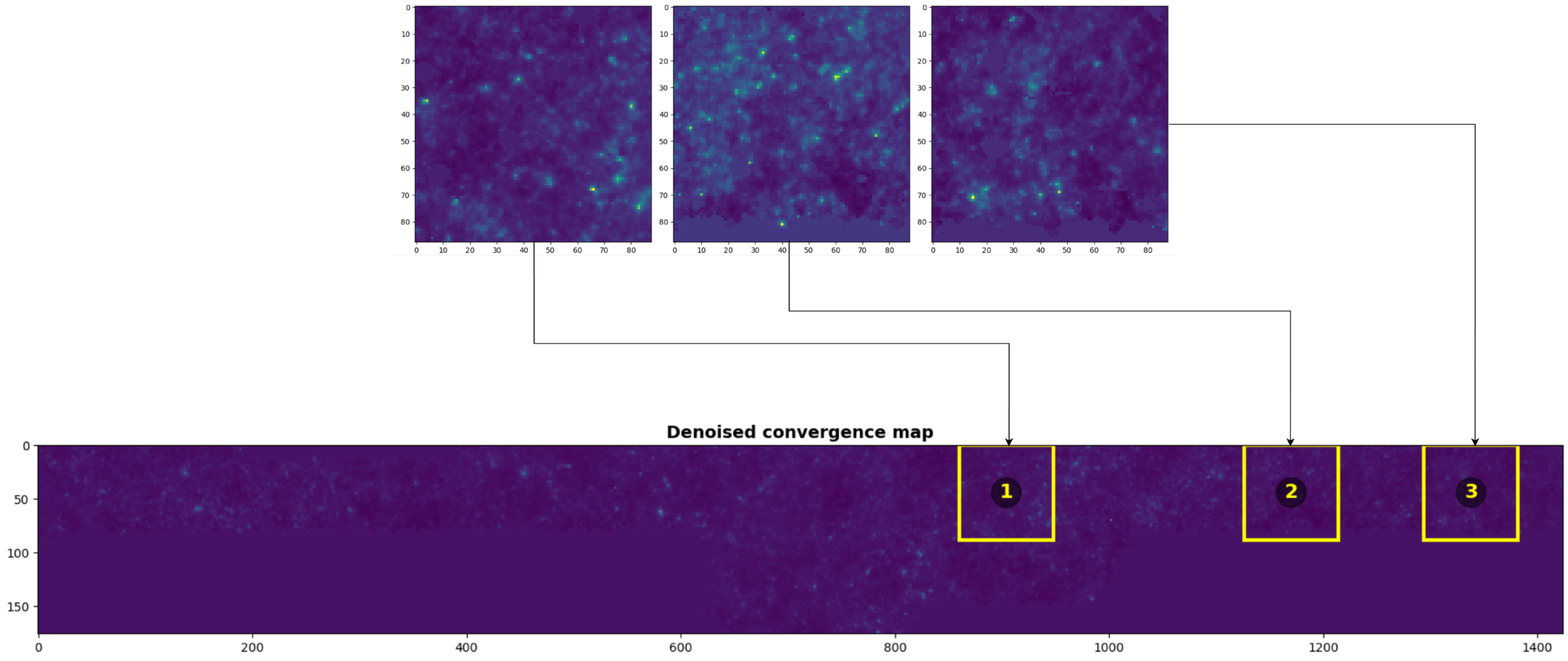
Denoising U-Net

- Standard U-Net with encoder–decoder + skip connections
- Input = noisy map $X + \epsilon$, Output = denoised map $\hat{X}$
- Loss: MSE $||\hat{X} - X||_2^2$



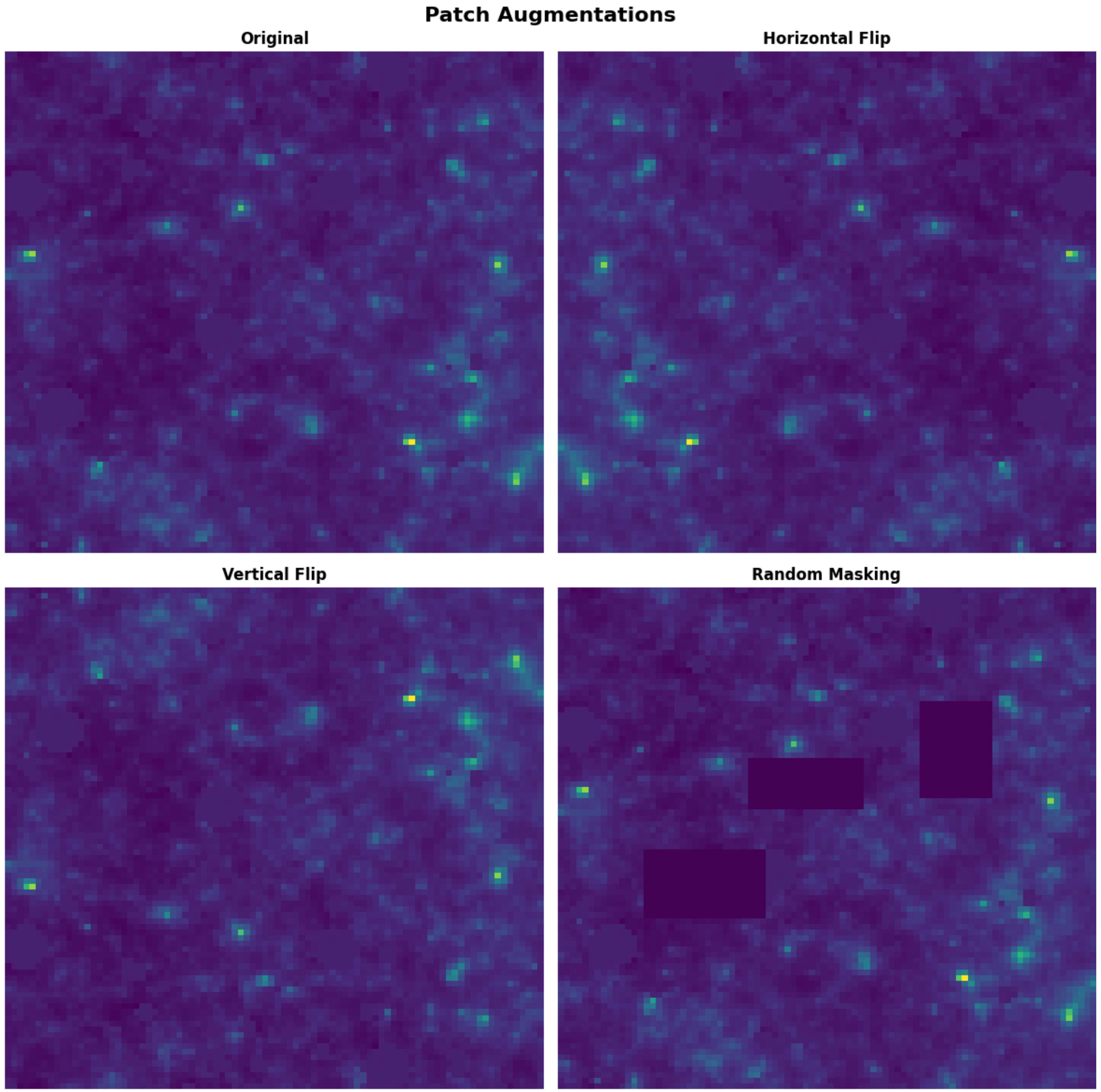
Parameter Estimation Network

- Extract 88x88 patches from denoised convergence map



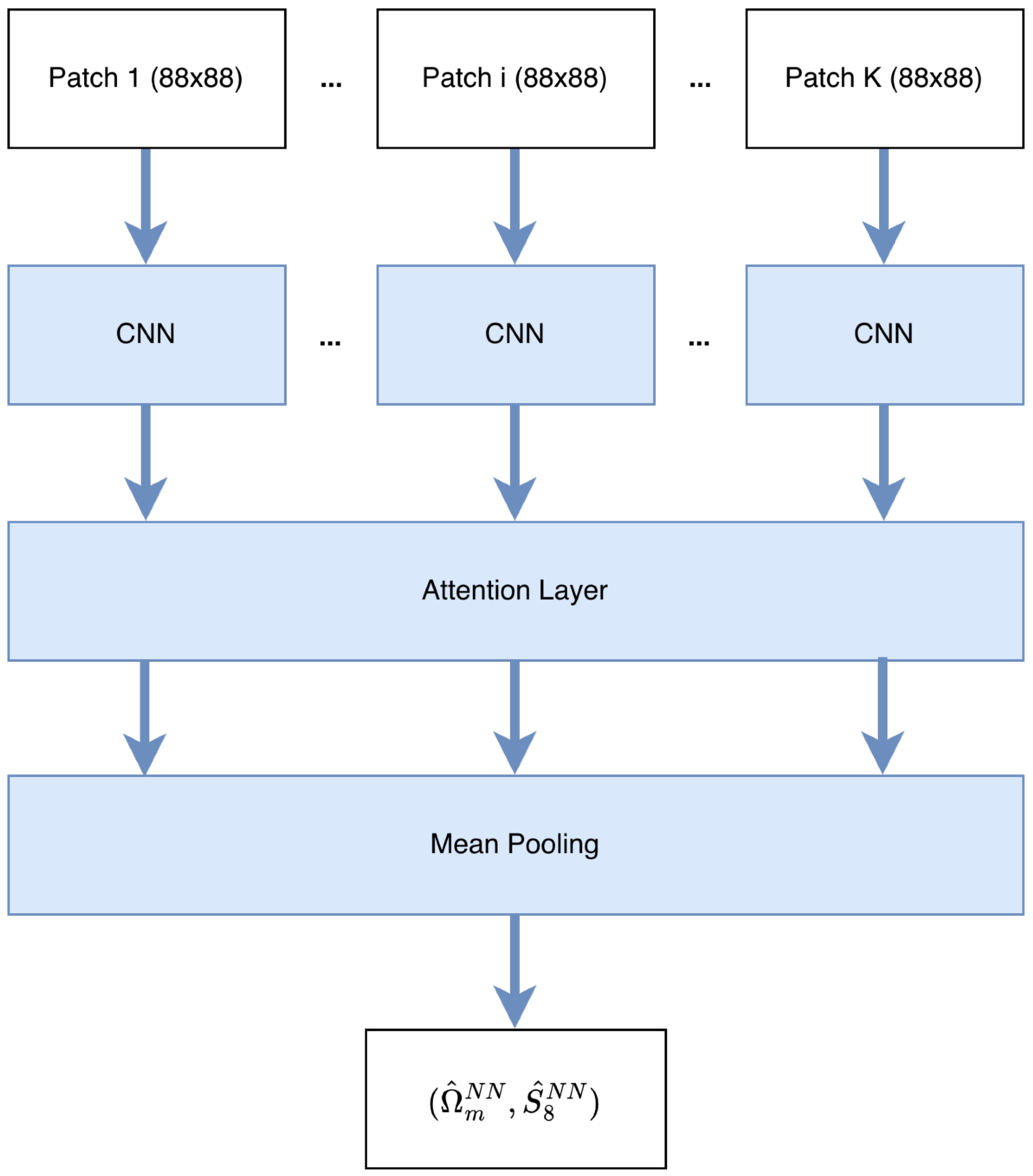
Parameter Estimation Network

- Patch-level data augmentation:
  - Horizontal / vertical / diagonal flips
  - Random masking



Parameter Estimation Network

- CNN + attention + mean pooling
  - Local: patch embeddings
  - Global: aggregated map features
- MLP → $\hat{\Omega}_m^{NN}, \hat{S}_8^{NN}$
- Loss: MSE

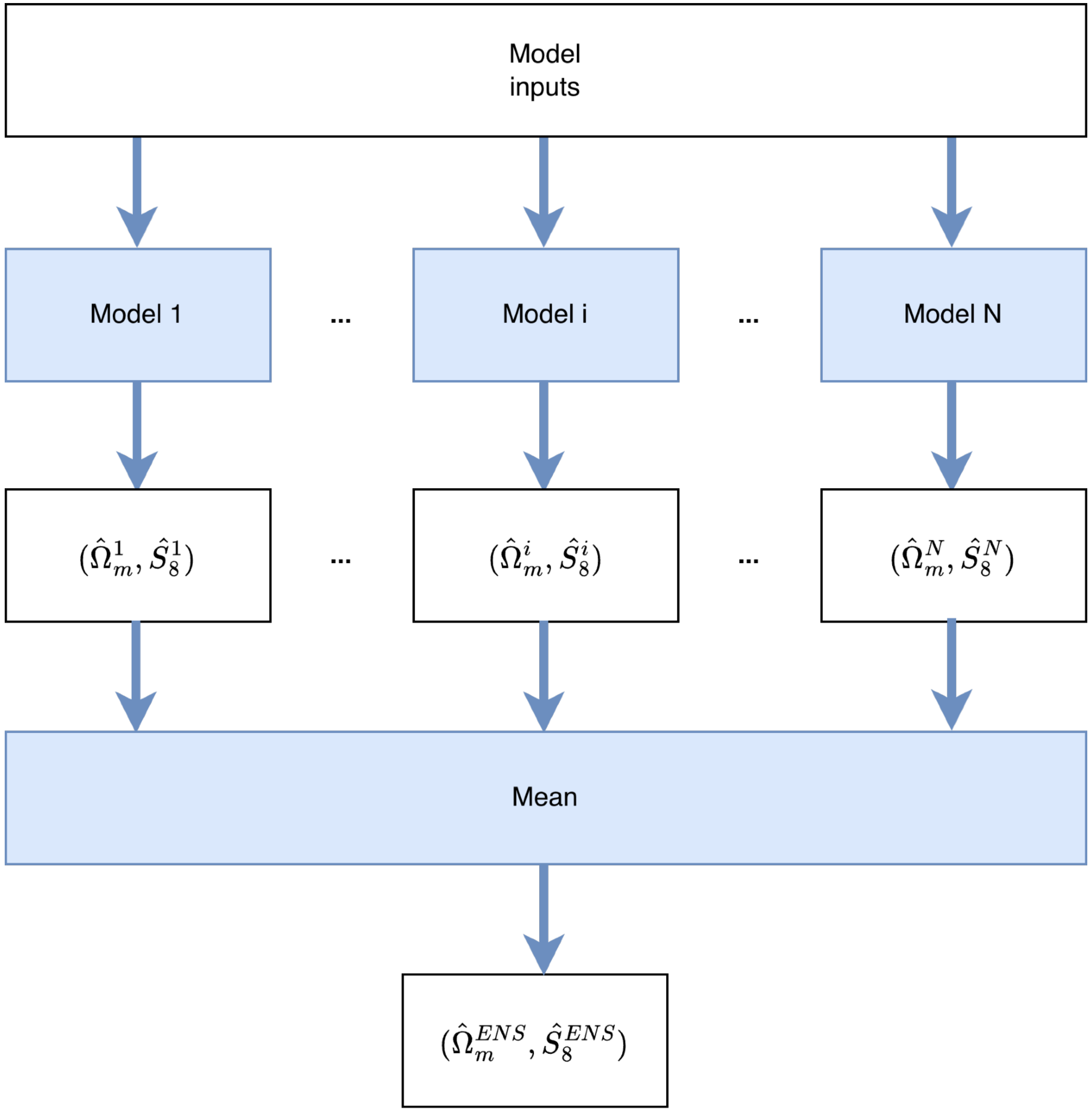


Ensembling

- Ensemble with different pretrained CNN backbones (RegNet and EfficientNet variants)

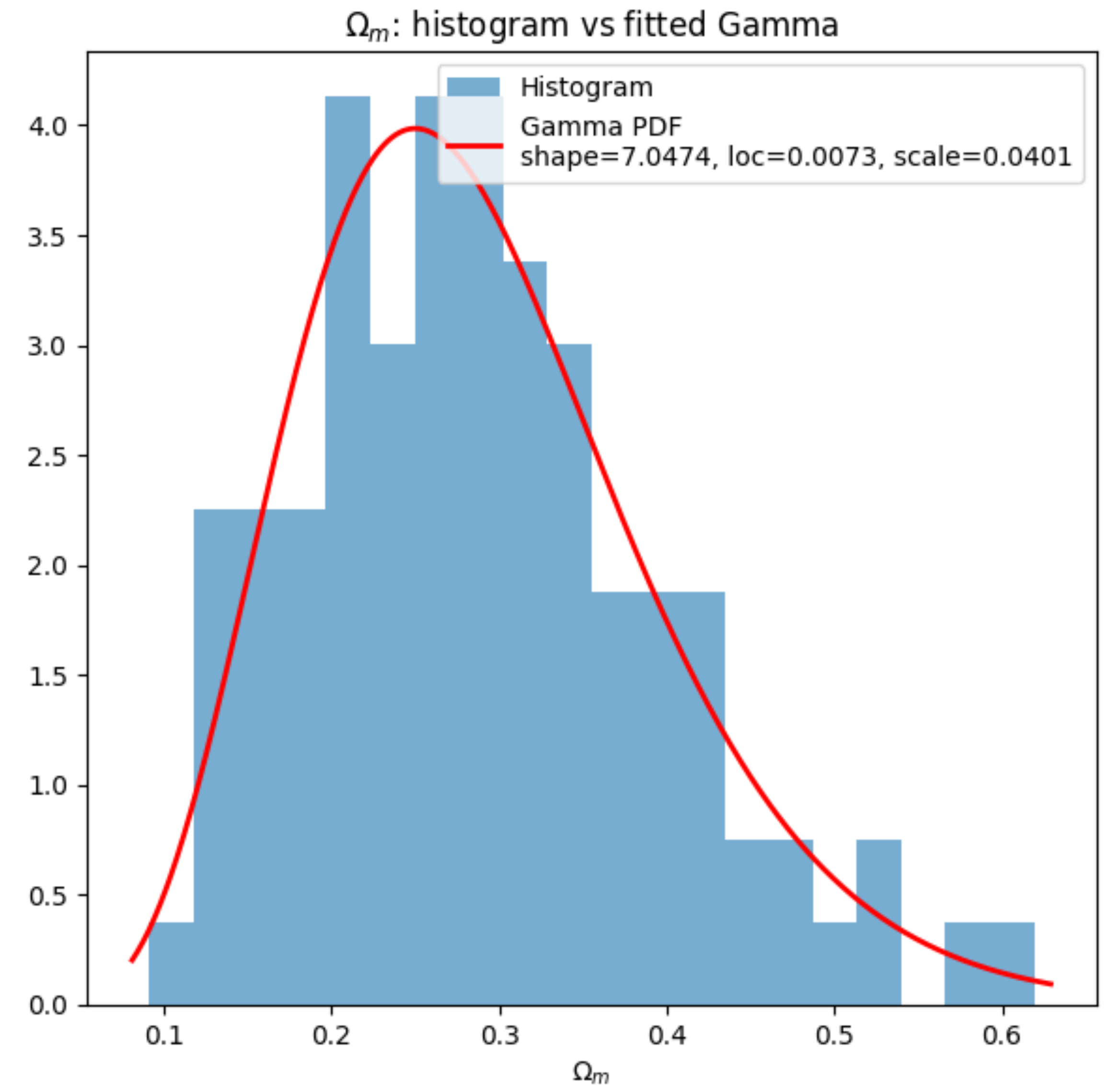
Backbone
efficientnet_b2
efficientnet_b3
regnetz_040.ra3_in1k
regnetv_040.ra3_in1k
regnety_040.ra3_in1k

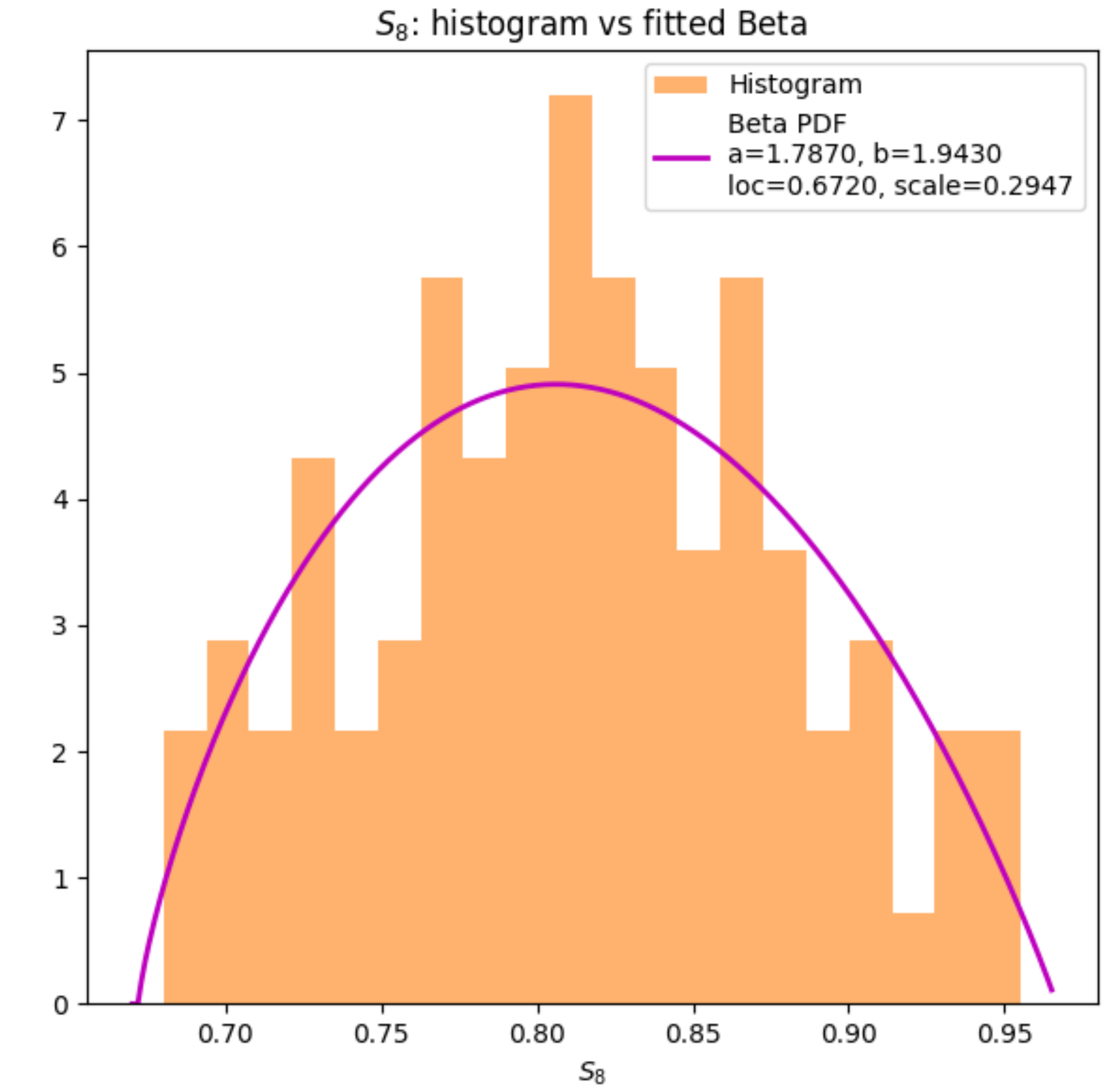
Backbone Architectures Used for Parameter Prediction



MCMC Posterior Estimation

- Use $(\hat{\Omega}_m^{ENS}, \hat{S}_8^{ENS})$ as summary statistics
- Independent priors for each parameter
  - Gamma for Ω_m
  - Beta for S_8
- Gaussian likelihood
- Output: $(\hat{\Omega}_m^{MCMC}, \hat{S}_8^{MCMC}, \hat{\sigma}_{\Omega_m}, \hat{\sigma}_{S_8})$





Results

Comparisons:

- With vs. without denoising
- Ensemble vs. single model

Model	Without Denoising	With Denoising
Ensemble	10.779	11.0551
efficientnet_b2	10.401	10.801
efficientnet_b3	10.360	10.813
regnetz_040.ra3_in1k	10.455	10.890
regnetv_040.ra3_in1k	10.492	10.939
regnety_040.ra3_in1k	10.466	10.891

Model performance on the holdout dataset

Conclusion

- Denoising and ensembling significantly improve performance
- Top performance likely requires stronger foundational methods
- May provide further gains when combined with stronger models ?

Thank you!