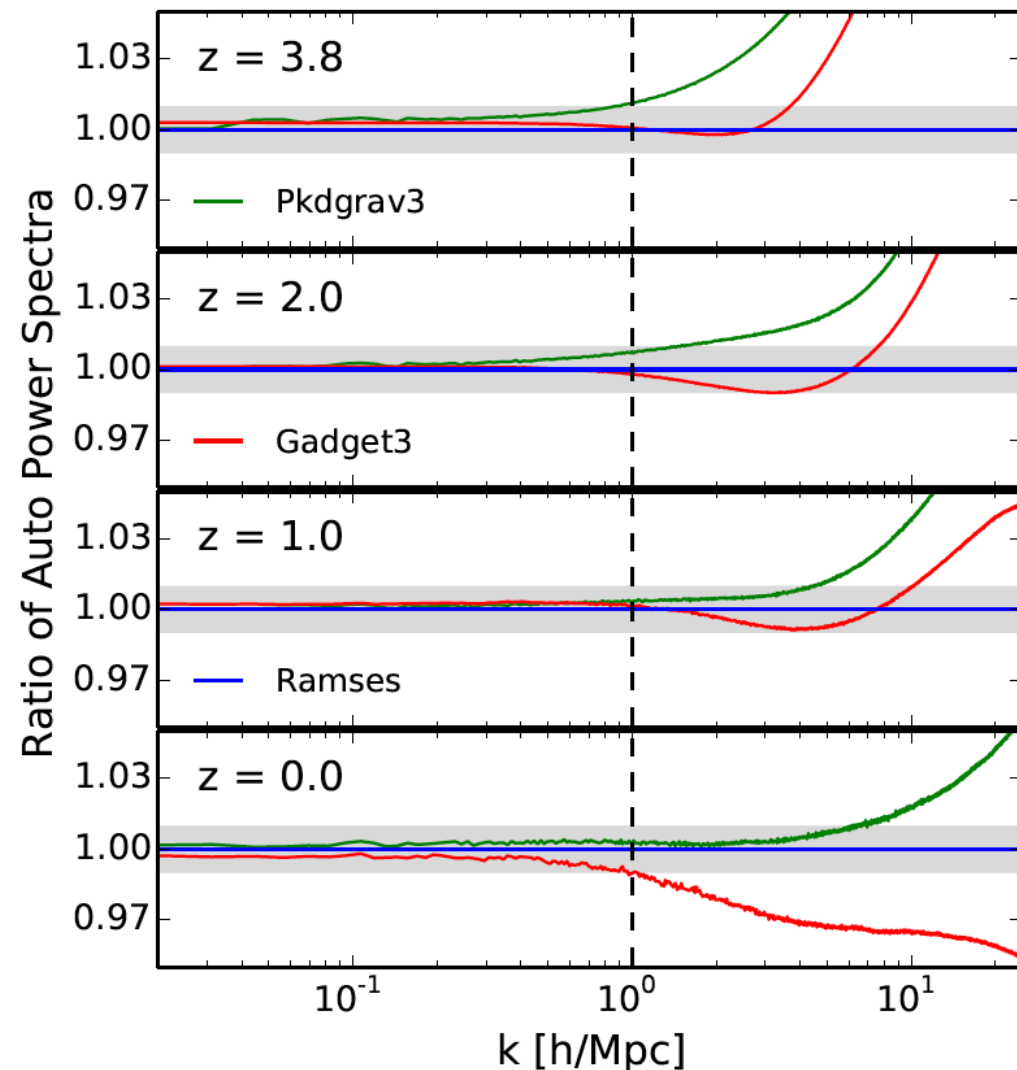


Observational Cosmology Journal Club Dec. 10 2019

Takahiro Nishimichi

	Title	Authors	Reference
[1]	Euclid preparation: II. The EuclidEmulator — A tool to compute the cosmology dependence of the nonlinear matter power spectrum	Euclid Collaboration, M. Knabenhans, J. Stadel, S. Marelli, D. Potter, R. Teyssier, L. Legrand, A. Schneider, B. Sudret , L. Blot and S. Awan	arXiv: 1809.04695
[2]	Learning to Predict the Cosmological Structure Formation	S. He, Y. Li, Y. Feng, S. Ho, S. Ravanbakhsh, W. Chen and B. Póczos	arXiv: 1811.06533
[3]	CosmoFlow: Using Deep Learning to Learn the Universe at Scale	A. Mathuriya*, D. Bard, P. Mendygral†, L. Meadows*, J. Arnemann, L. Shao*, S. He, T. Karna*, D. Moise†, S. J. Pennycook*, K. Maschhoff†, J. Sewall*, N. Kumar*, S. Ho, M. F. Ringerburg†, Prabhat and V. Lee*	arXiv: 1808.04728

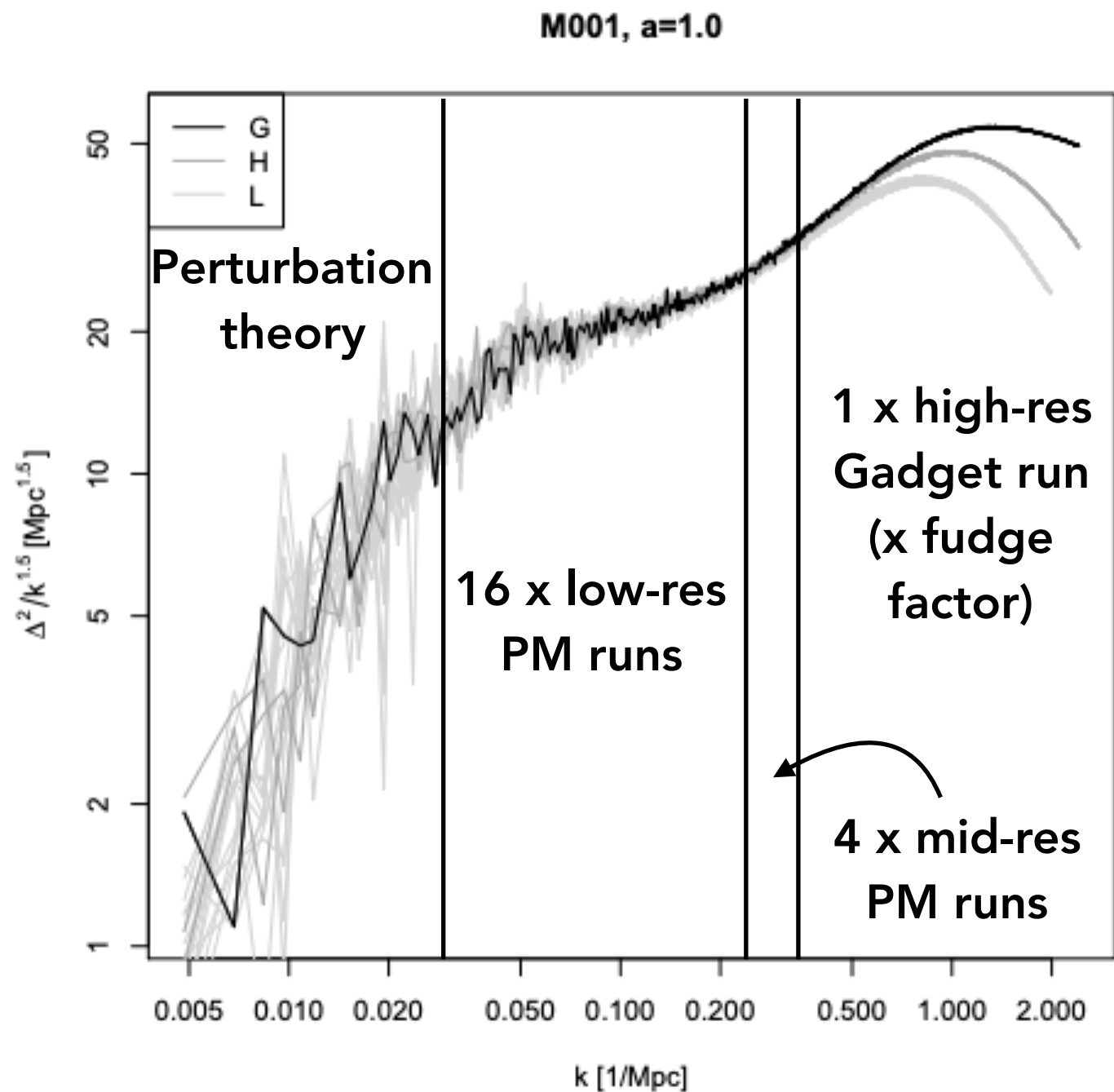
[1] Euclid preparation: II. The EuclidEmulator



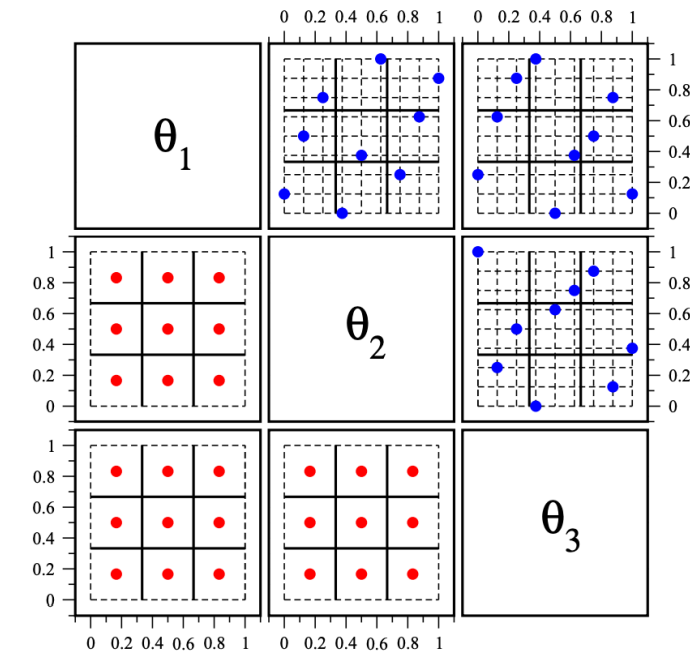
From arXiv:1503.05920

- ▶ Eventual goal: 1% accuracy on $P(k)$ up to $k \sim 10$ h/Mpc
- ▶ Euclid "flagship" simulations ongoing
- ▶ Prepare a framework to interpolate across the cosmological parameter space efficiently and accurately
 - ▶ For the moment, using cheaper but still big simulation suite
 - ▶ \longrightarrow Emulator for matter $P(k)$

- c.f. Previous example (Coyote Universe; Heitmann+'09, 10, Lawrence et al. '10)



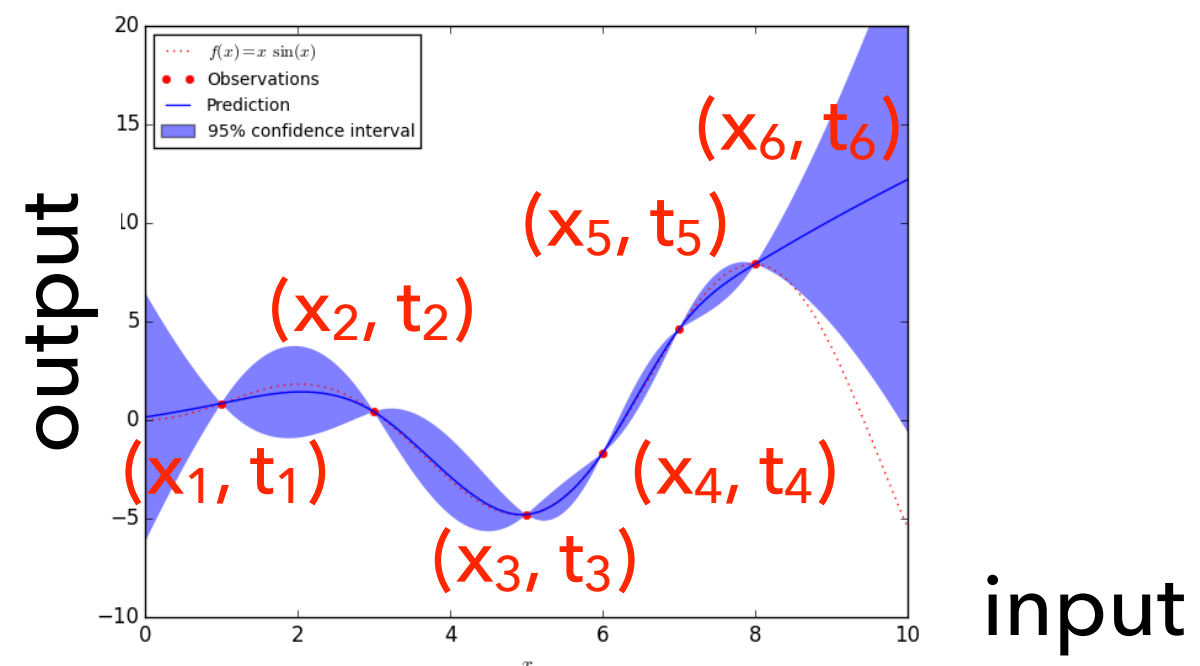
- LH design



- SVD decomposition

$$\mathcal{P}(k; z; \theta) = \sum_{i=1}^{n_P} \phi_i(k; z) w_i(\theta), \quad \theta \in [0, 1]^{n_\theta},$$

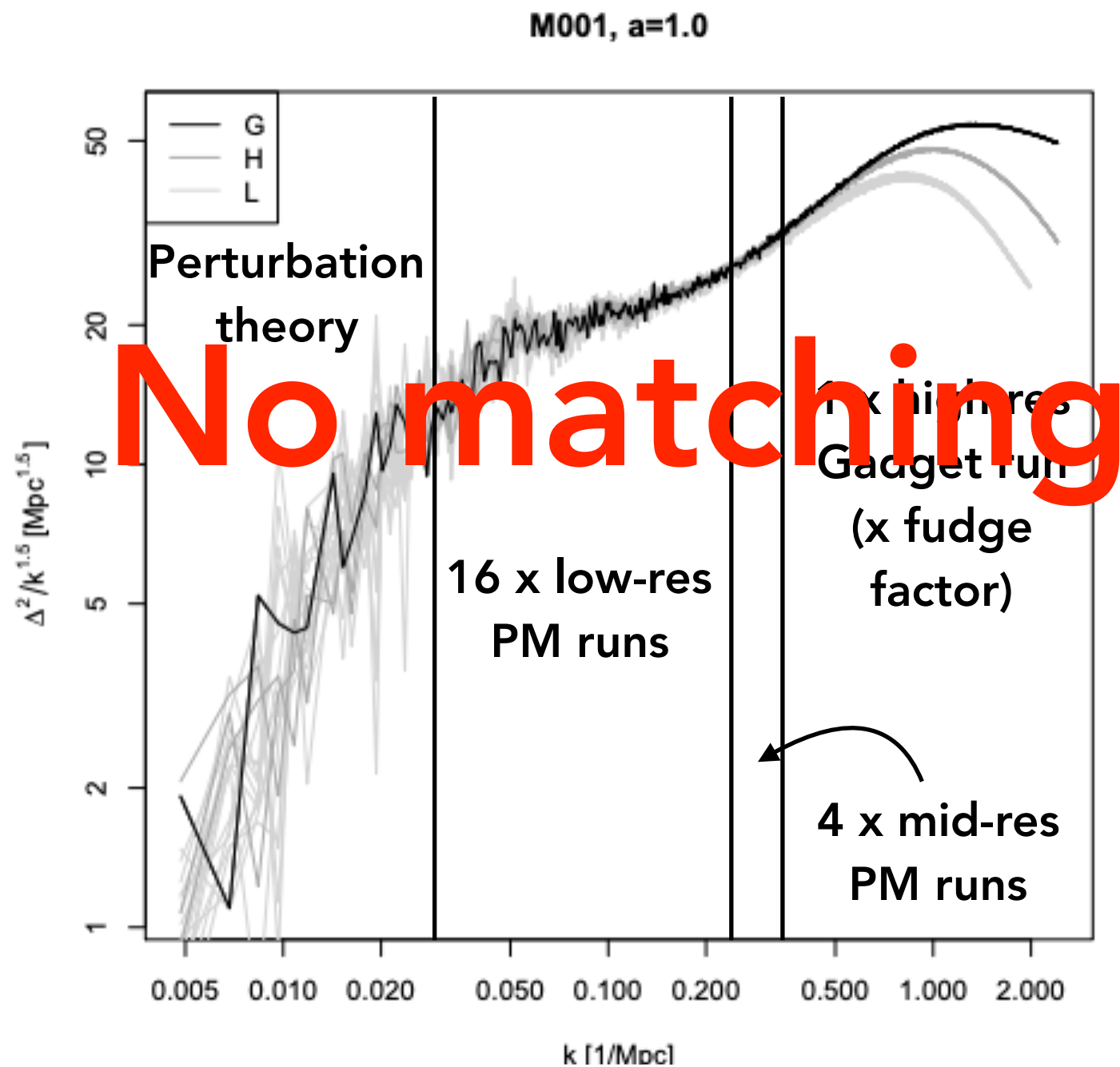
- Gaussian Process



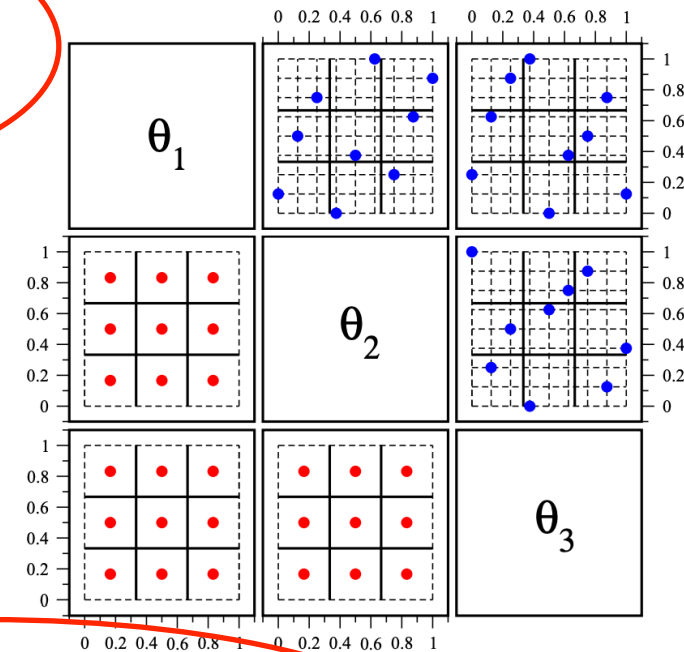
$$P(t_{N+1} | \mathbf{t}_N) \propto \exp \left[-\frac{1}{2} [\mathbf{t}_N \ t_{N+1}] \mathbf{C}_{N+1}^{-1} \begin{bmatrix} \mathbf{t}_N \\ t_{N+1} \end{bmatrix} \right].$$

$$\begin{aligned} \hat{t}_{N+1} &= \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \\ \sigma_{\hat{t}_{N+1}}^2 &= \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}. \end{aligned}$$

► Strategy of this work



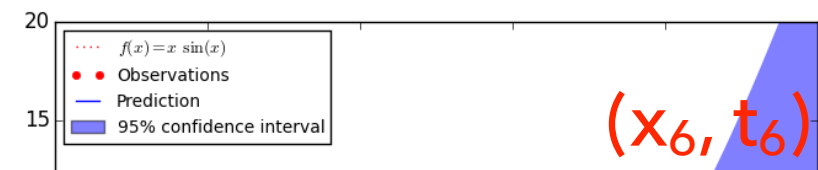
► LH design



► SVD decomposition

$$\mathcal{P}(k; z; \theta) = \sum_{i=1}^{n_P} \phi_i(k; z) w_i(\theta), \quad \theta \in [0, 1]^{n_\theta},$$

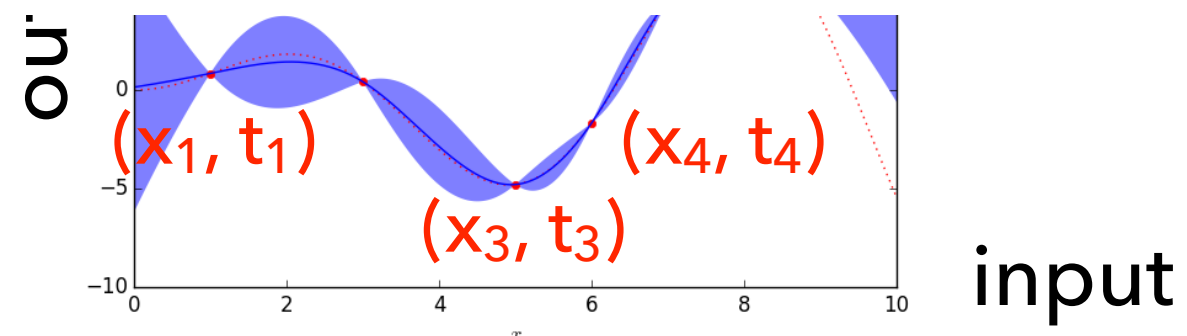
► ~~Gaussian Process~~



—> **UQLab (uncertainty qualification software)**

$$P(t_{N+1} | \mathbf{t}_N) \propto \exp \left[-\frac{1}{2} [\mathbf{t}_N \ t_{N+1}] \mathbf{C}_{N+1}^{-1} [t_{N+1}] \right].$$

$$\begin{aligned} \hat{t}_{N+1} &= \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t}_N \\ \sigma_{\hat{t}_{N+1}}^2 &= \kappa - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}. \end{aligned}$$



► Simulations

Simulation identifier	L [h^{-1} Mpc]	N	PF	number of runs	total runtime [node hours]
000	1250	2048	yes	2	1 904
001-100	1250	2048	yes	200	190 200
101-136	1250	2048	yes	72	68 472
Euclid Flagship High Resolution (EFHR)	1920	8000	no	1	93 600 [★]
Large Volume (LV)	4000	4089	yes	2	14 696
Convergence Test 1 (CT ₁)	256	262	yes	2	23
CT ₂	512	524	yes	2	50
CT ₃	640	655	yes	2	80
CT ₄	960	983	yes	2	236
CT ₅	1250	1280	yes	2	508
CT _{6abc} (a: $R_{\text{grid}} = 1$, b: $R_{\text{grid}} = 2$, c: $R_{\text{grid}} = 7$)	480	1024	yes	2	378
CT ₇	480	1536	yes	2	1 402
CT _{8abc}	640	1024	yes	2	308
CT ₉	640	1536	yes	2	1 204
CT _{10abc}	960	1024	yes	2	240
CT ₁₁	960	1536	yes	2	896
CT ₁₂	960	2048	yes	2	2 274
CT _{13abc}	1440	1024	yes	2	209
CT ₁₄	1440	1536	yes	2	752
CT ₁₅	1440	1920	yes	2	1 953
CT _{16abc}	1920	1024	yes	2	184
CT ₁₇	1920	1536	yes	2	633
CT ₁₈	1920	1920	yes	2	1529

Training

Validation

Volume requirement

Resolution requirement

★ with GPUs

✂ PF: “Pairing and Fixing” simulations (Angulo & Pontzen ’16)

$$P(|\delta_{i,\text{lin}}|, \theta_i) = \prod_i \frac{1}{2\pi} \delta_{\text{D}}(|\delta_{i,\text{lin}}| - \sqrt{P_i}),$$

$$\delta_{i,\text{lin}} = \sqrt{P_i} e^{i\theta_i}.$$

$$\theta_{i,2} = \theta_{i,1} + \pi$$

► Convergence study

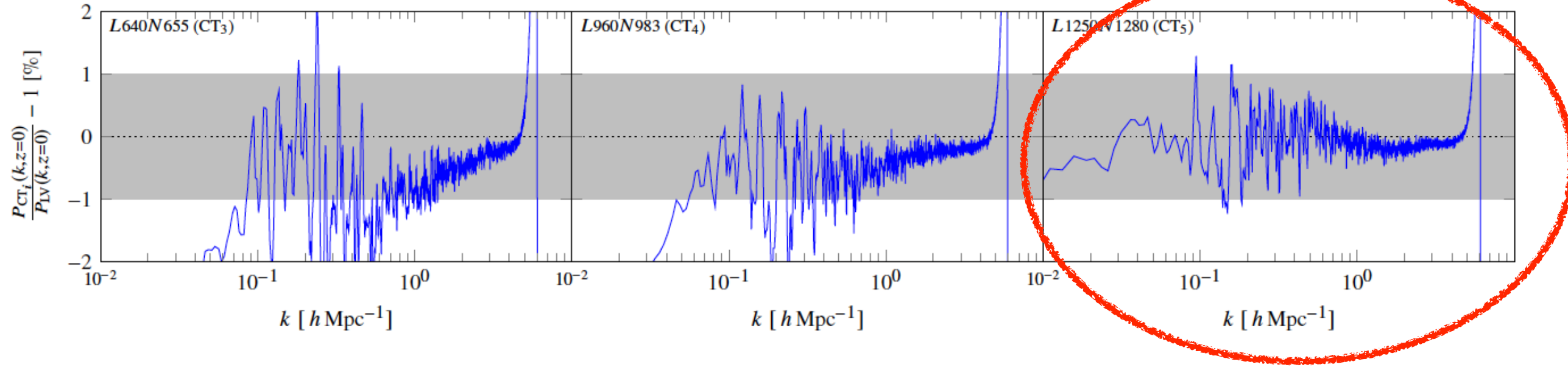


Figure C1. Comparison of three different runs with different box sizes but equal mass resolution $\ell^{-1} = 1.024 h \text{ Mpc}^{-1}$ against a large volume (LV) reference simulation with a box of $L = 4000 h^{-1} \text{ Mpc}$ side length and $N = 4096$ particles per dimension. Only simulation volumes of at least $L^3 = (1250 h^{-1} \text{ Mpc})^3$ (right panel) allow a power spectrum measurement that agrees with the one in a large volume at the 1% level. Smaller simulation volumes lead to a power deficit on large scales (left and middle panel) and increasingly larger deviations from the reference on mildly nonlinear scales.

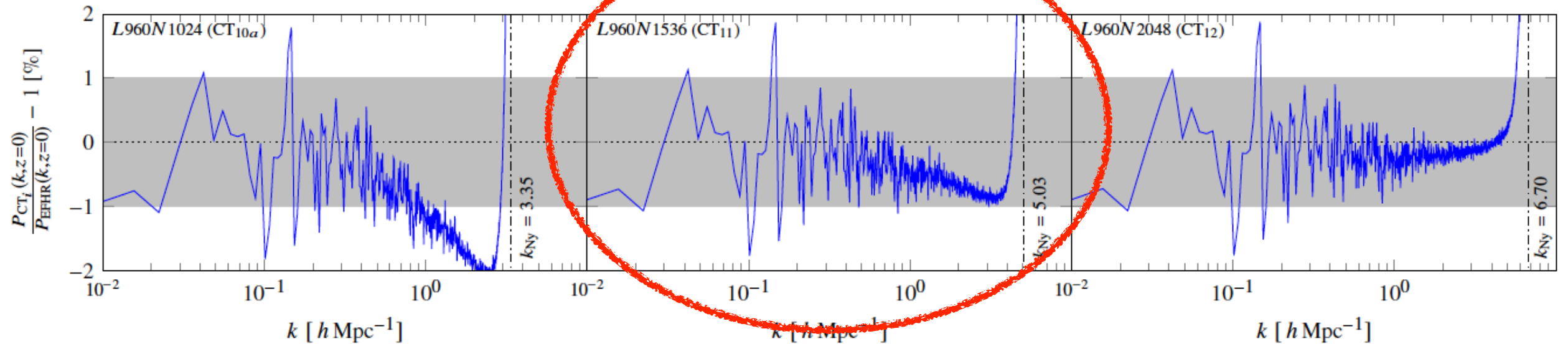


Figure C2. Effect of measuring the power spectrum in boxes of different size and with different number of particles ($R_{\text{grid}} = 1$), i.e. in the L - N -plane. All panels belong to the same box size $L = 960 h^{-1} \text{ Mpc}$. From left to right, $N = 1024, 1536$ and 2048 particles per dimension were used. We thus span an inverse mean inter-particle distance range of $1.07 h \text{ Mpc}^{-1} \leq \ell^{-1} \leq 2.13 h \text{ Mpc}^{-1}$. The turn-up of the error curve at high k -modes indicates the location of the Nyquist frequency (dash-dotted line) of the mass assignment grid.

► Data flow

$$B(k, z) := \frac{P_{\text{nl}}(k, z)}{P_{\text{lin}}(k, z)},$$

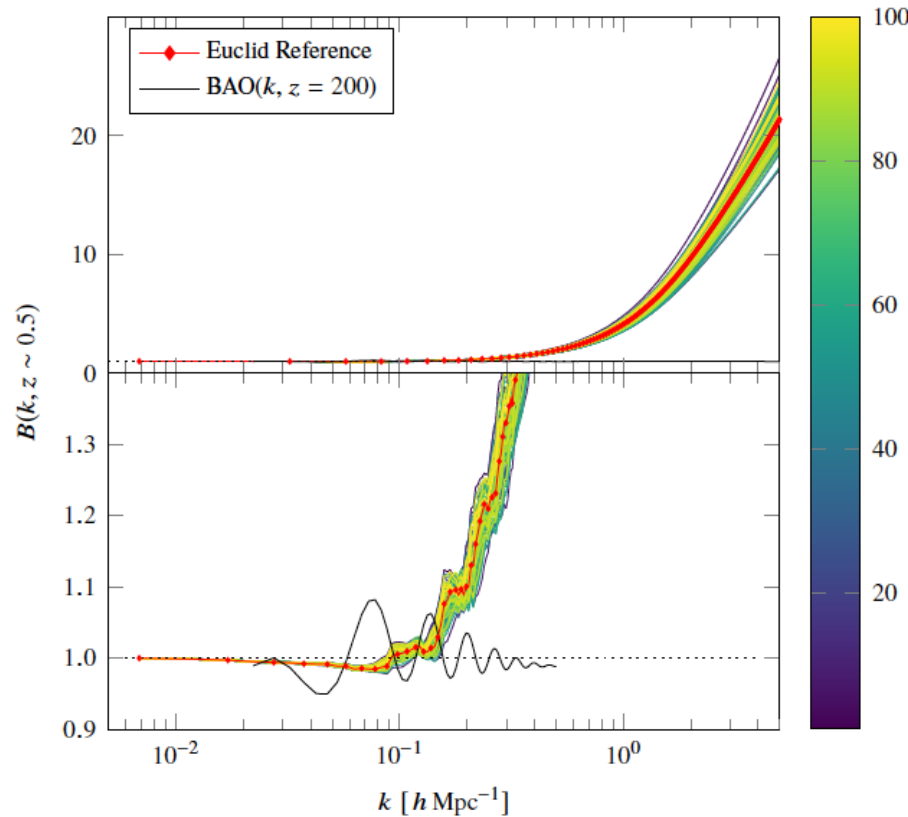


Figure 2. The experimental design used as input for the construction of the **EuclidEmulator**: boost factor curves for each input cosmology and the Euclid reference cosmology (red line with diamond markers) at output step 60 (roughly $z \sim 0.5$ depending slightly on the cosmology). The color bar labels the 100 different cosmologies in the **ED** via their IDs. It is clearly visible that the Euclid reference cosmology is reasonably near the center of the parameter space as all boost factors of the input cosmologies scatter nicely around it. The black oscillatory curve in the lower panel indicates the **BAO** at high redshift, as in **Fig. 1**.

Initial: 1100 ($k < 5.48 h/\text{Mpc}$)
 $n_{\text{ED}} \times (n_z \cdot n_k)$ matrix of $\log(B_c(k, z))$
 100 designs 100 ($z < 5$)

PCA: $\mathbf{D} = \sum_{i=1}^{n_{\text{ED}}} \lambda_i(\omega_b, \omega_m, n_s, h, w_0, \sigma_8) \text{PC}_i(k, z),$

SPCE: $\lambda_i(\omega_b, \omega_m, n_s, h, w_0, \sigma_8) \approx \sum_{\alpha \in \mathcal{A}} \eta_\alpha \Psi_\alpha(\mathbf{x}),$

$$\Psi_\alpha(\mathbf{x}) = \phi_{\alpha_1}(x_1) \phi_{\alpha_2}(x_2) \dots \phi_{\alpha_6}(x_6) = \prod_{l=1}^6 \sqrt{2\alpha_l + 1} P_{\alpha_l}(x_l).$$

(Sparse Polynomial
Chaos Expansion)

$$\mathcal{A}_{\text{cand}} = \left\{ \alpha : \left(\sum_{i=1}^6 \alpha_i^q \right)^{1/q} \leq p, \quad \sum_{i=1}^6 1_{\alpha_i \neq 0} \leq r \right\}.$$

—> Least Angle Regression (LARS) determines the optimal sparse set of polynomials \mathcal{A} for a given internal parameter set

“p” can be determined automatically by UQLab

$$\begin{aligned} n_{\text{ED}} &= 10, 25, 50, 75, 100, 250, \\ a &= 1 - 10^{-\kappa}, \quad \kappa = 1, 2, \dots, 10, \\ q &= 0.1, 0.2, \dots, 1.0, \\ r &= 2, 3, 4, 5, 6. \end{aligned}$$

Parameter grid search at 3,000 points

► Hyperparameter optimization

Based on $P(k,z)$

Based on $\log(B(k,z))$

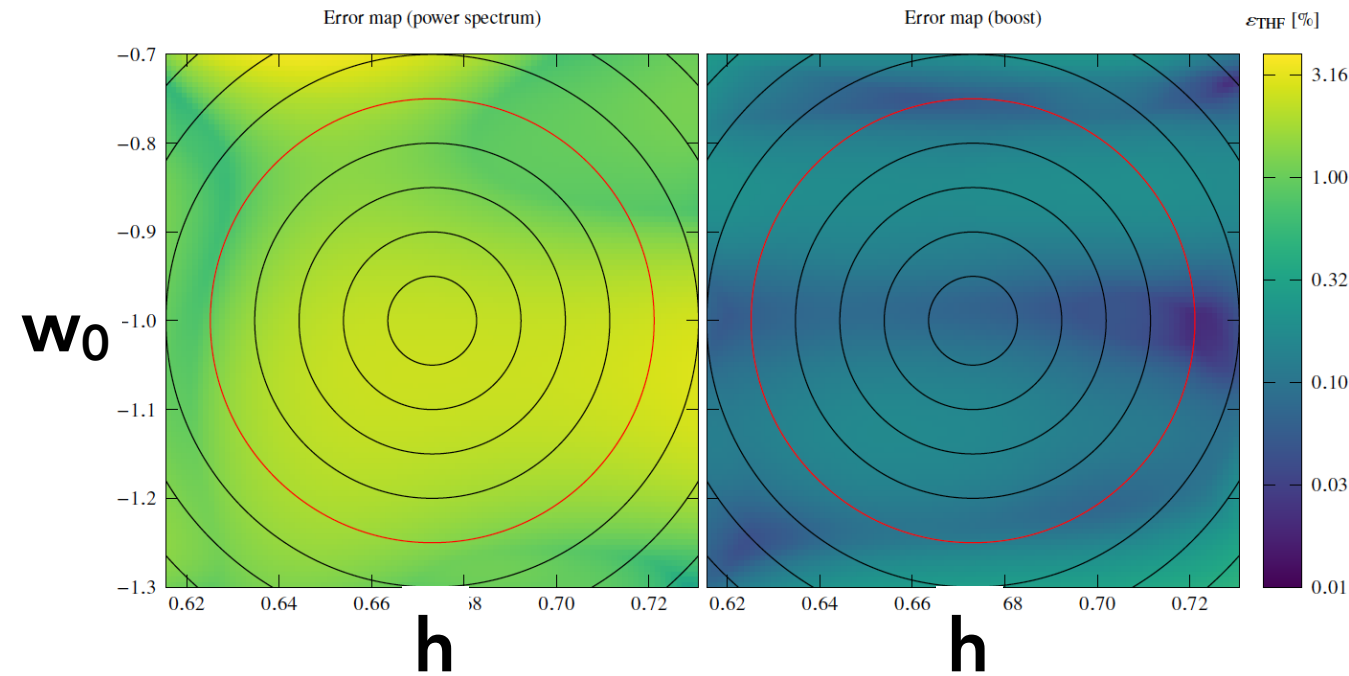


Figure 3. Error maps of the (h, w_0) -plane based on an ED with $n_{ED} = 100$ for direct emulation of the power spectrum (left panel, similar to Heitmann et al. 2010b and Lawrence et al. 2017) and the logarithm of the boost (right panel). We stress that logarithms are just used to construct the emulator and the errors shown here in both panels are based on comparisons of the full nonlinear power spectrum/boost. For the boost emulation, the maximal error over all cosmologies is of order 0.5% (in contrast to 3.7% for direct power spectrum emulation) but if restricted to the 0.83Δ-ellipsoid (red ellipse), the maximal emulation-only error drops to below 0.2% (direct power spectrum emulation: 2.75%). The black circles indicate the different Δ-ellipsoids (the innermost ellipse corresponds to the 0.17Δ, the second to the 0.33Δ etc).

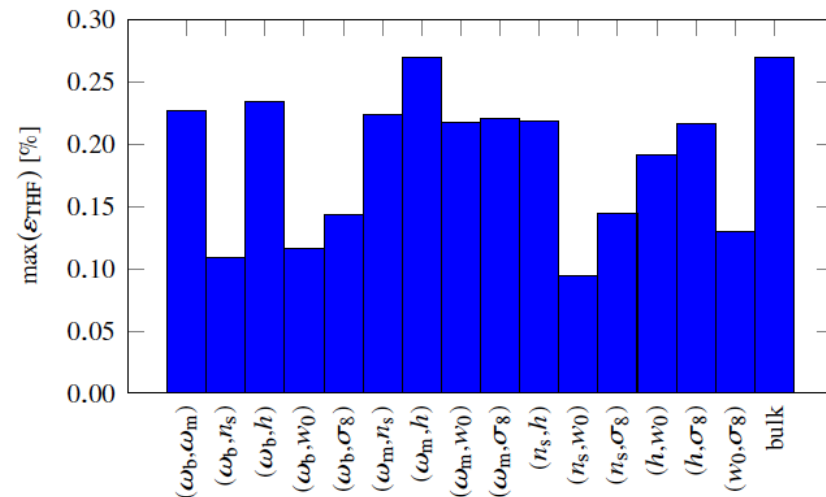


Figure 4. Maximal errors over all cosmologies in all coordinate planes and in the bulk of the space (restricted to the 0.83Δ region). This bar chart shows that a comparison of error maps as in Fig. 3 looks similar for all coordinate planes: the maximal error of EuclidEmulator over all coordinate planes is about 0.27% in contrast to roughly 1% of the FrankenEmu (Heitmann et al. 2010b) and 4% of the CosmicEmu (Lawrence et al. 2017).

- Generate Halofit (w/ Takahashi et al. parameters) template at $16 \times 10,000$ cosmologies to find the best hyper parameters
- “Emulation-Only-Error” (EOE) is estimated as the L^∞ -norm (=max) of

$$EOE_c(k, z) = \frac{B_c^{\text{emulated}}(k, z)}{B_c^{\text{simulated}}(k, z)},$$

► Results

- Cross validation test done at 36 cosmologies outside the design

The test is performed along the 6 coordinate axes. For each of them an N-body simulation of the cosmologies at $\pm 1\sigma$, $\pm 3\sigma$ and $\pm 5\sigma$ (for w_0 we similarly chose $\pm 0.58\sigma$, $\pm 1.75\sigma$ and $\pm 2.92\sigma$) from the center of the parameter range is run resulting in a test set of 36 reference simulations outside the experimental design used for the construction of the emulator.

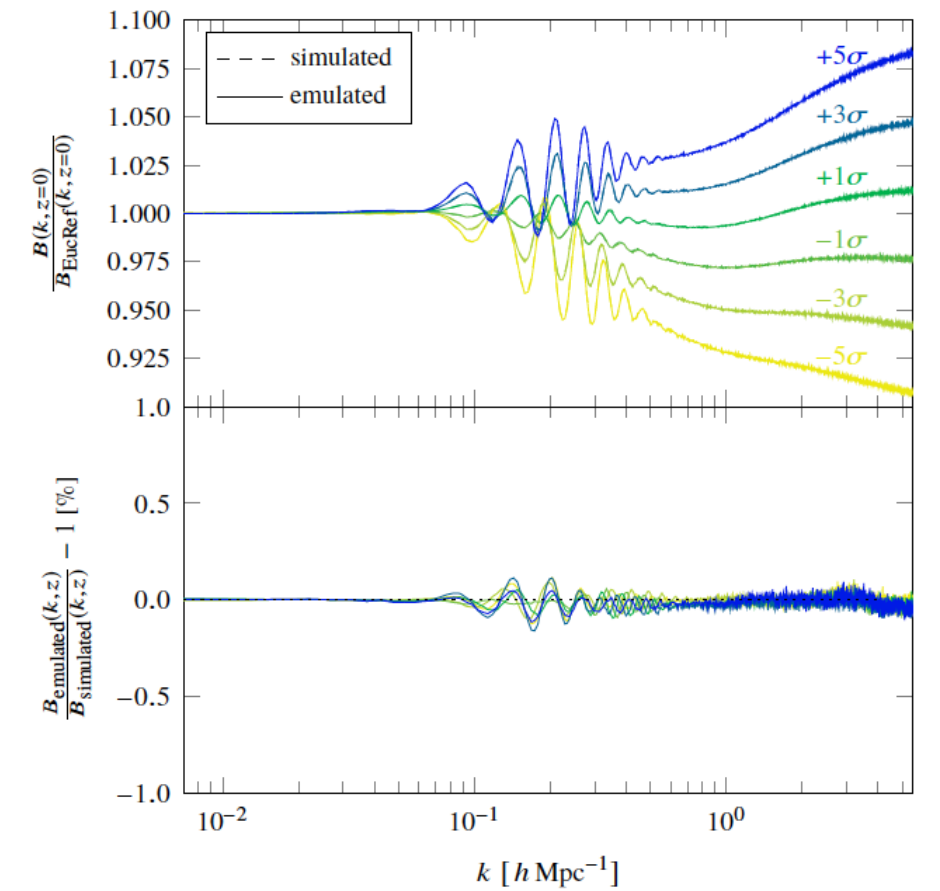


Figure 6. Comparison of the emulated (solid lines) and the simulated (dashed lines) boosts for six different cosmologies at $\pm 1\sigma$, $\pm 3\sigma$ and $\pm 5\sigma$ away from the center of the h -parameter axis (see column 7 of Table 2). In the upper panel, corrections of the boosts themselves relative to Euclid reference (“EucRef”) cosmology are shown. No difference between the emulated and the simulated curves is visible. In the lower panel the relative differences between emulated and simulated boosts are plotted. The entire y-range of the lower subplot corresponds to the accuracy tolerance regime of $\pm 1\%$, while the maximal error is roughly 6 times smaller than this upper limit.

Table 2. In this table we compare boost spectra computed with [EuclidEmulator](#) (EE) on the one hand and with an emulator based on Takahashi’s [HALOFIT](#) (THF) on the other hand to boost spectra of full N-body simulations for 36 cosmologies outside the experimental design over the entire k and z range. The numbers in the table correspond to the relative errors in percent between the emulated and simulated boosts. We find that the errors predicted with the THF emulator are broadly consistent with the EE errors. This implies that the configuration of the EE based on a [HALOFIT](#) emulator is actually valid and that there is no need to make a huge investment to run N-body simulations just for the configuration study of an emulator. This result is particularly important for finding the size of the experimental design (see Fig. 5).

	ω_b		ω_m		n_s		h		w_0		σ_8	
	EE	THF	EE	THF	EE	THF	EE	THF	EE	THF	EE	THF
$\pm 1\sigma$	0.081	0.081	0.093	0.091	0.082	0.076	0.078	0.119	0.082	0.076	0.086	0.097
$\pm 3\sigma$	0.081	0.098	0.105	0.124	0.085	0.064	0.114	0.181	0.076	0.080	0.089	0.067
$\pm 5\sigma$	0.073	0.062	0.166	0.220	0.121	0.069	0.145	0.127	0.081	0.062	0.089	0.096

► Comparison with previous works

@ Euclid reference cosmology

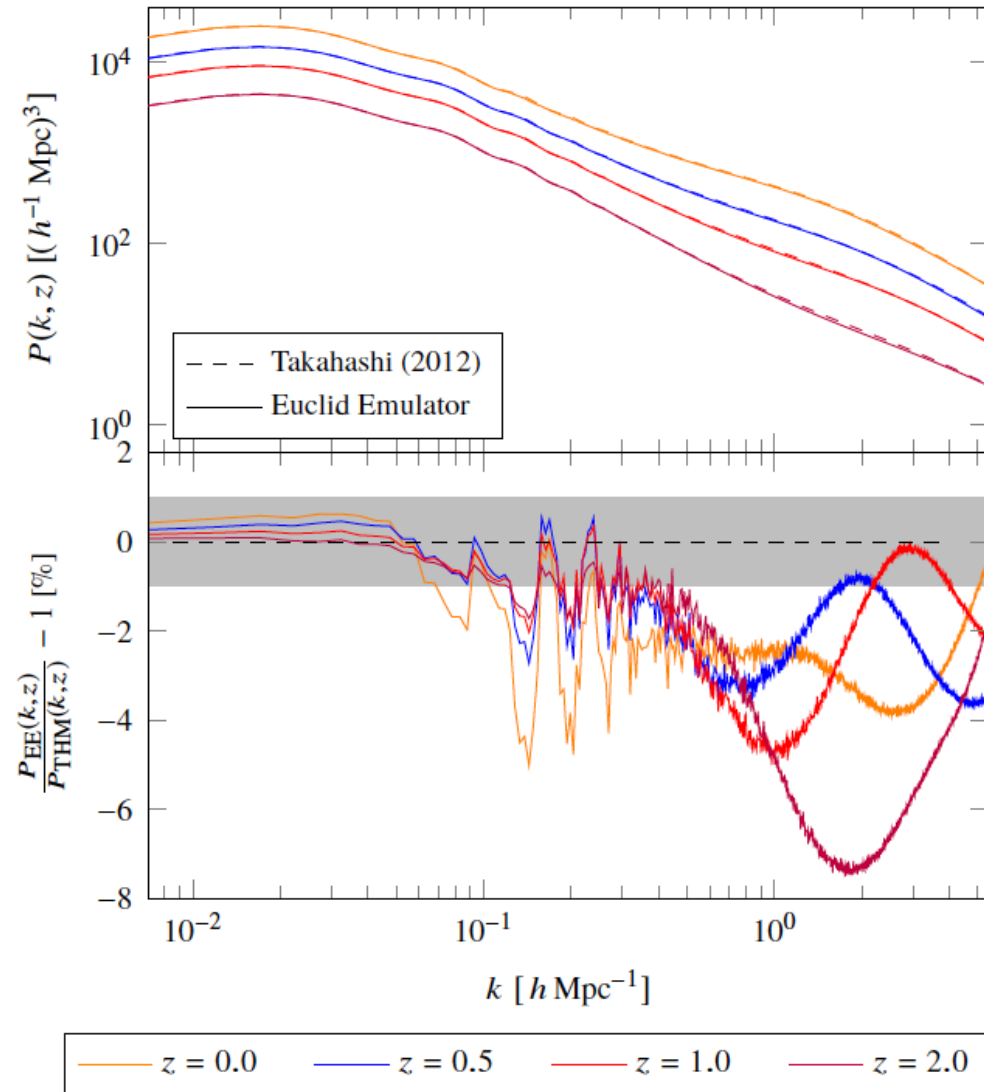


Figure 8. Comparison of nonlinear power spectra computed using **EuclidEmulator** (upper panel, solid curves) and the Takahashi model (upper panel, dashed curves). In the upper panel the absolute power spectra are shown while in the lower panel the relative error curves between the power spectra computed by the two different means are plotted. The agreement on large, linear scales is nearly perfect. On intermediate scales a distinct systematic coming from the BAO signal is observed for all tested redshifts. This shows that the Takahashi model does not capture the BAOs at the desired level of accuracy. The disagreement on the smallest scales is of order of several percent over all redshifts.

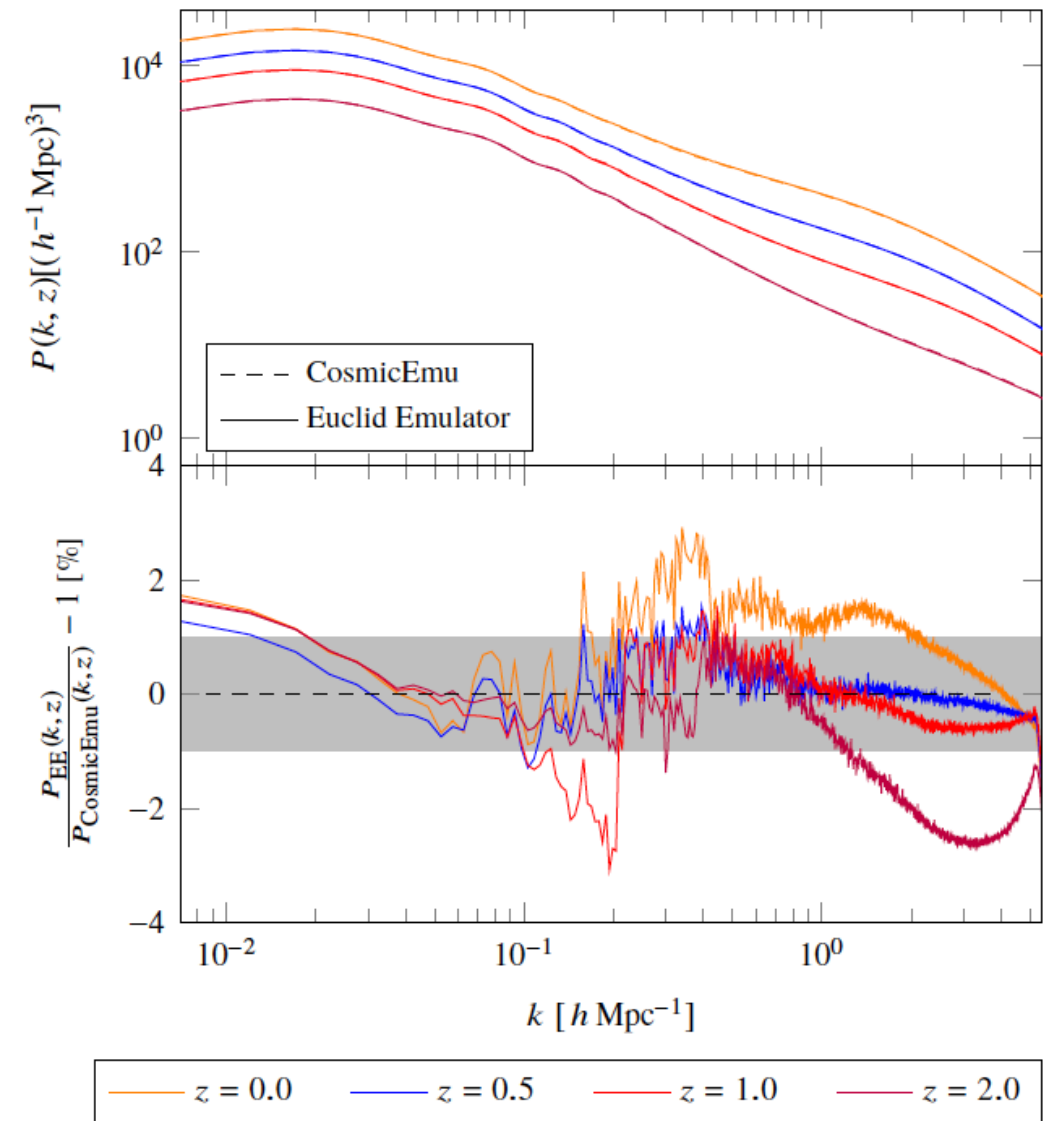
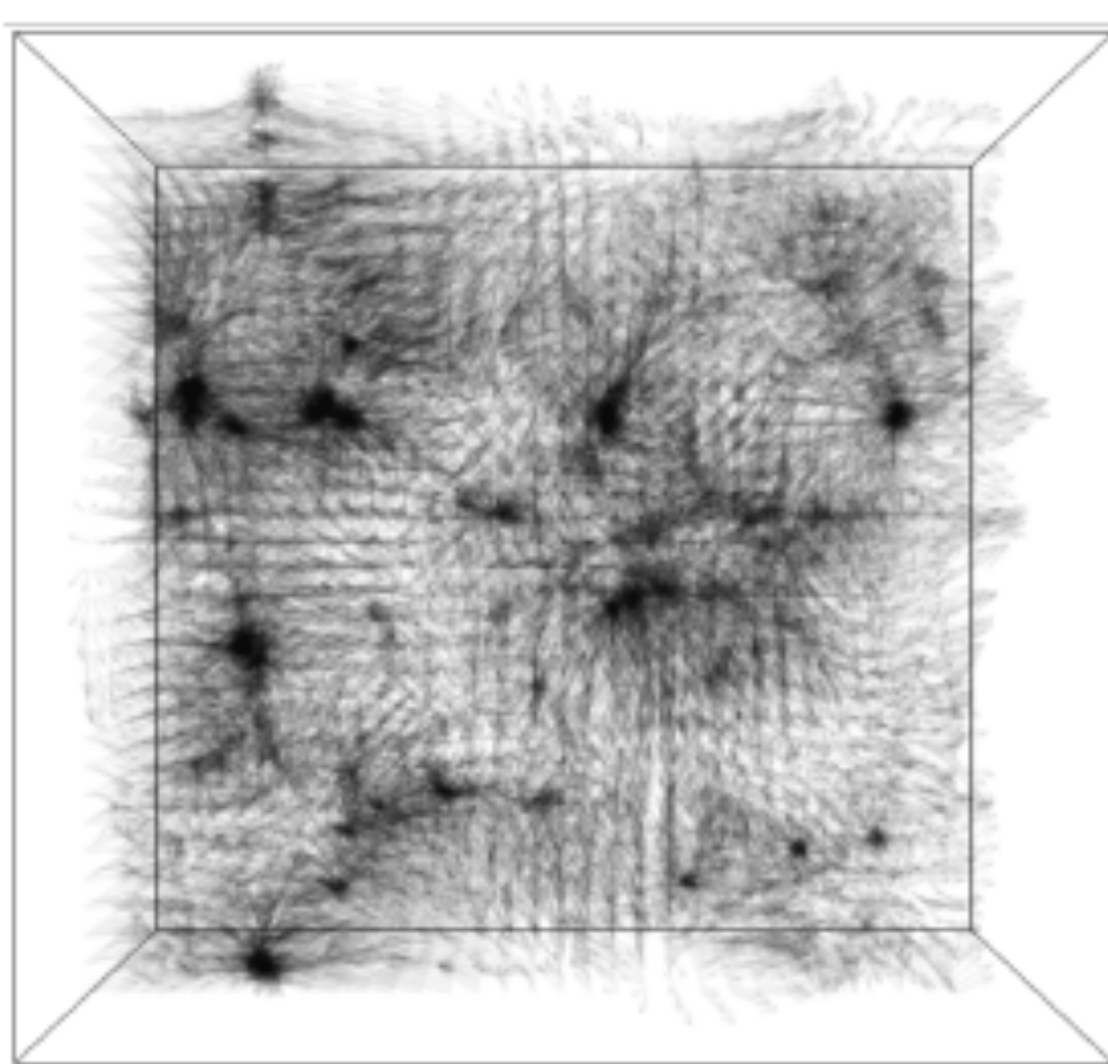


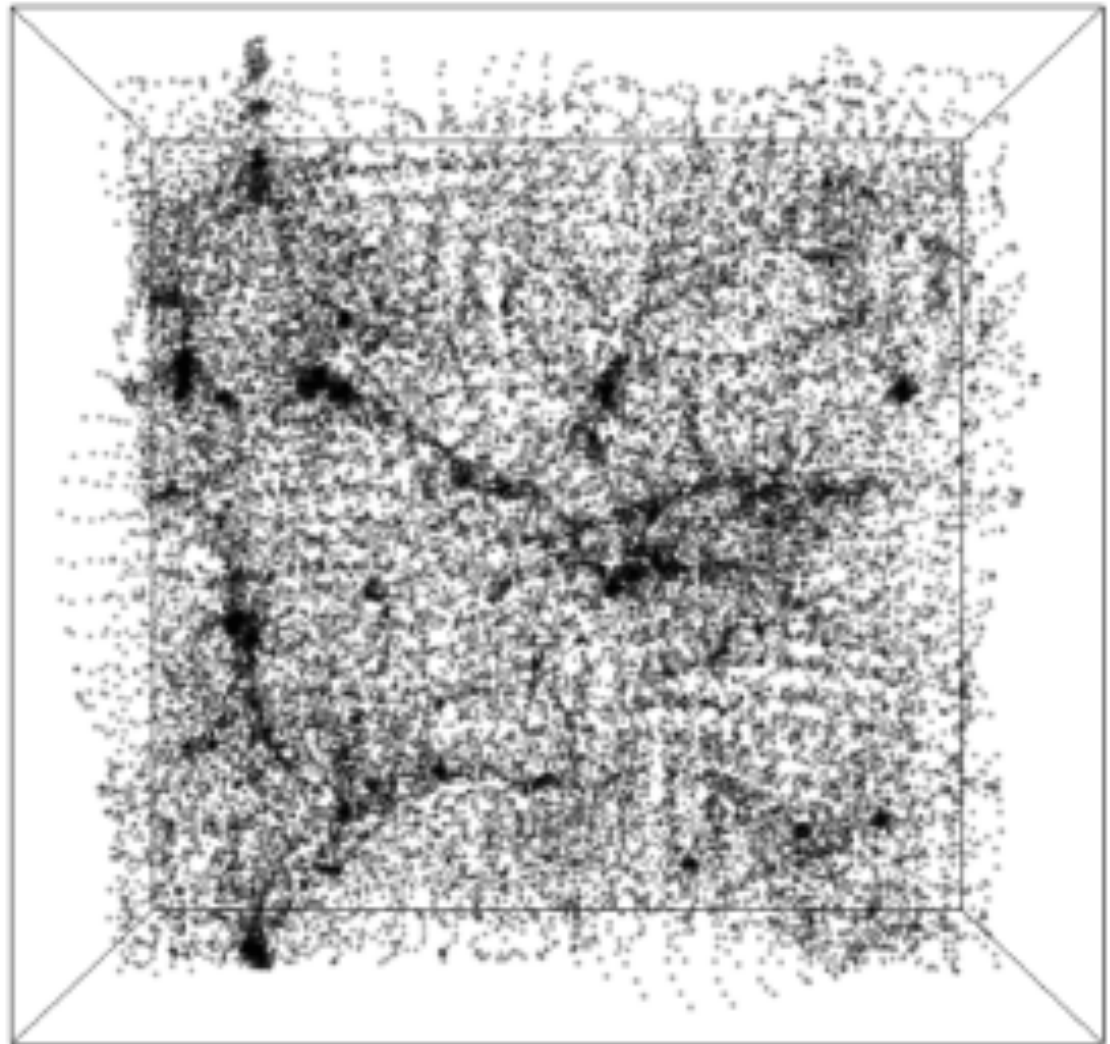
Figure 9. Comparison of nonlinear power spectra computed using **EuclidEmulator** (upper panel, solid curves) and the **CosmicEmu** code (upper panel, dashed curves). The relative errors between the power spectra computed with the two different approaches are again shown in the lower panel. On the largest scales there is a systematic disagreement originating in the **CosmicEmu** data (remember that on these scales **EuclidEmulator** is correct by construction). While on intermediate scales one observes disagreements of the order of a few percent as well, the agreement on small scales is very good over all redshifts.

[2] Learning to Predict the Cosmological Structure Formation

"D³M" displacement field



"D³M" density field



- ▶ ML for forward modeling of the cosmic large scale structure
- ▶ Develop a deep neural network called "Deep Density Displacement Model (D³M)"
- ▶ Just a proof of concept at this stage

► Dataset

- Pairs of (Zel'dovich Approx. — full N-body simulation)
 - ZA as the input
 - N-body as the output
- Perform 10,000 pairs of particle realizations
 - $L = 128 \text{ Mpc}/h$, $N = 32^3$
 - For one particular cosmology (PLANCK 2015)
 - (training:validation:test) = (80%:10%:10%)
- Additional data for tests
 - 1,000 2LPT realizations
 - Different cosmological parameter sets (discussed later)

► ML Architecture

- U-Net (Ronneberger, Fischer & Brox, "U-Net: Convolutional Networks for **Biomedical Image Segmentation**", arXiv:1505.0459)
- won the ISBI cell tracking challenge 2015 in these categories by a large margin

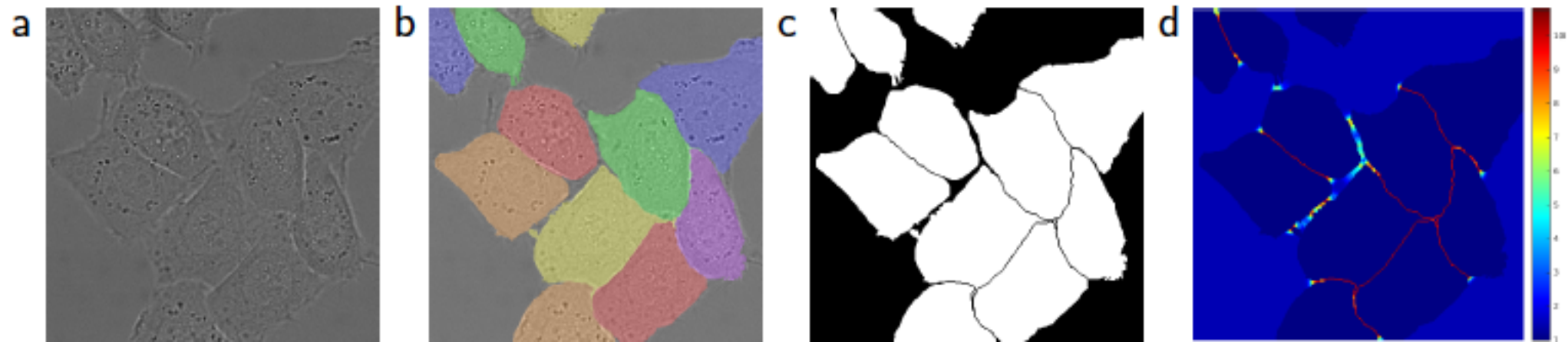
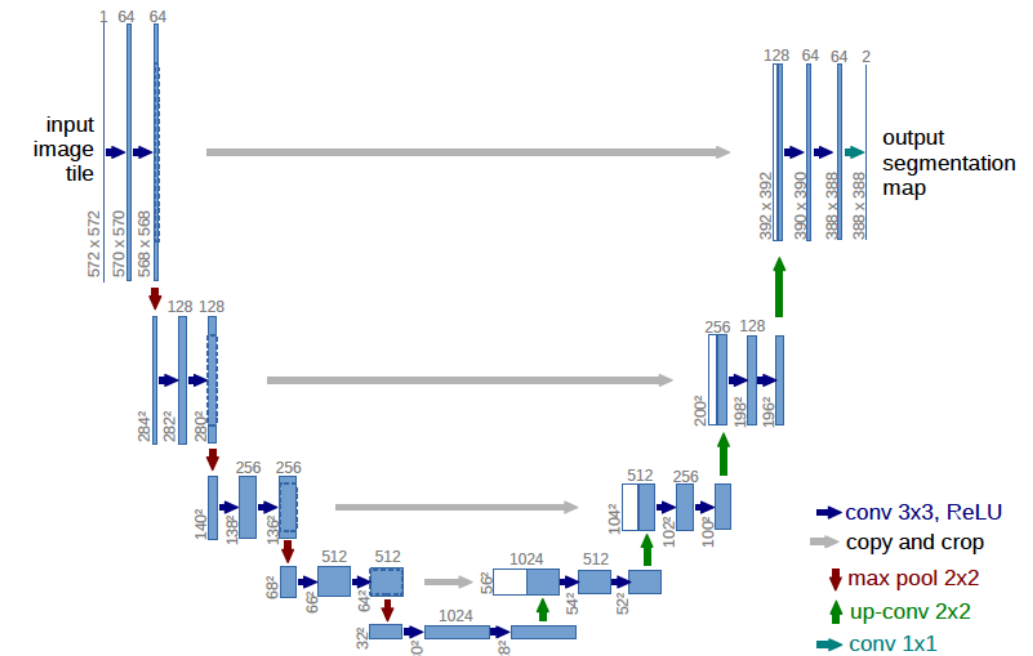
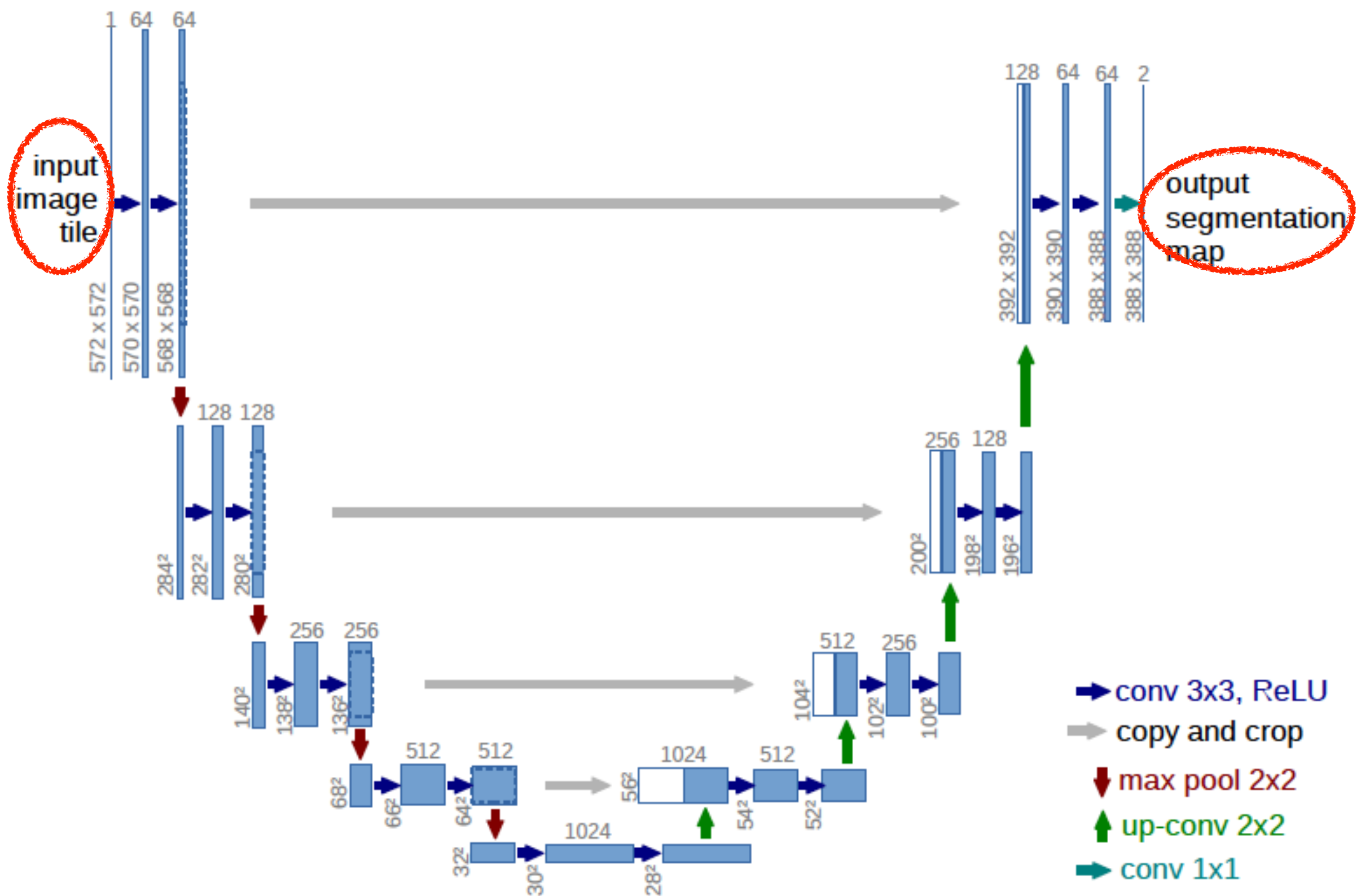
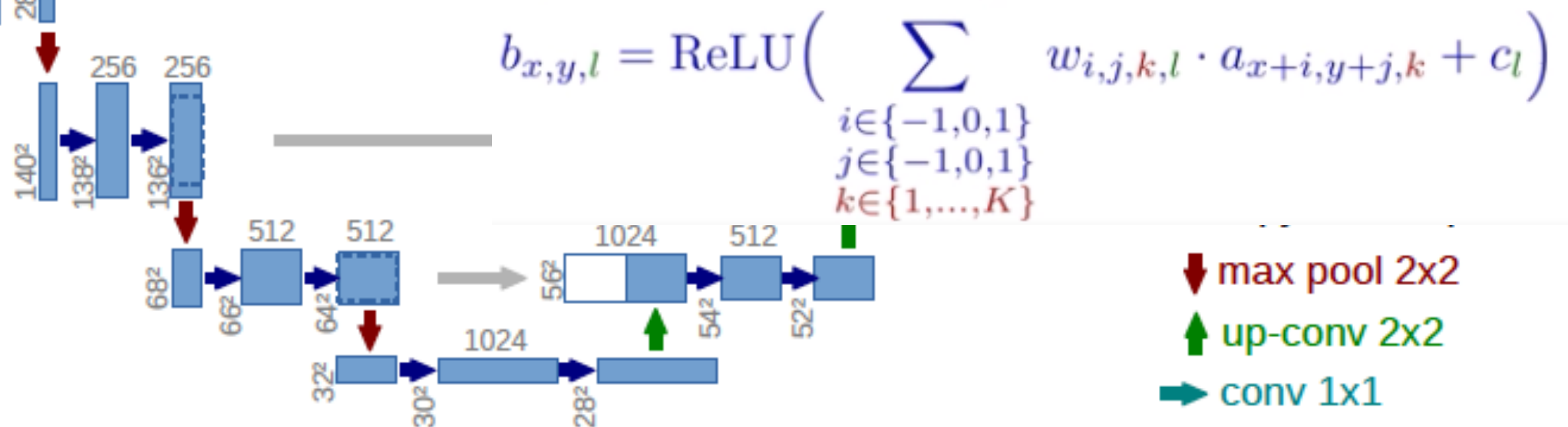
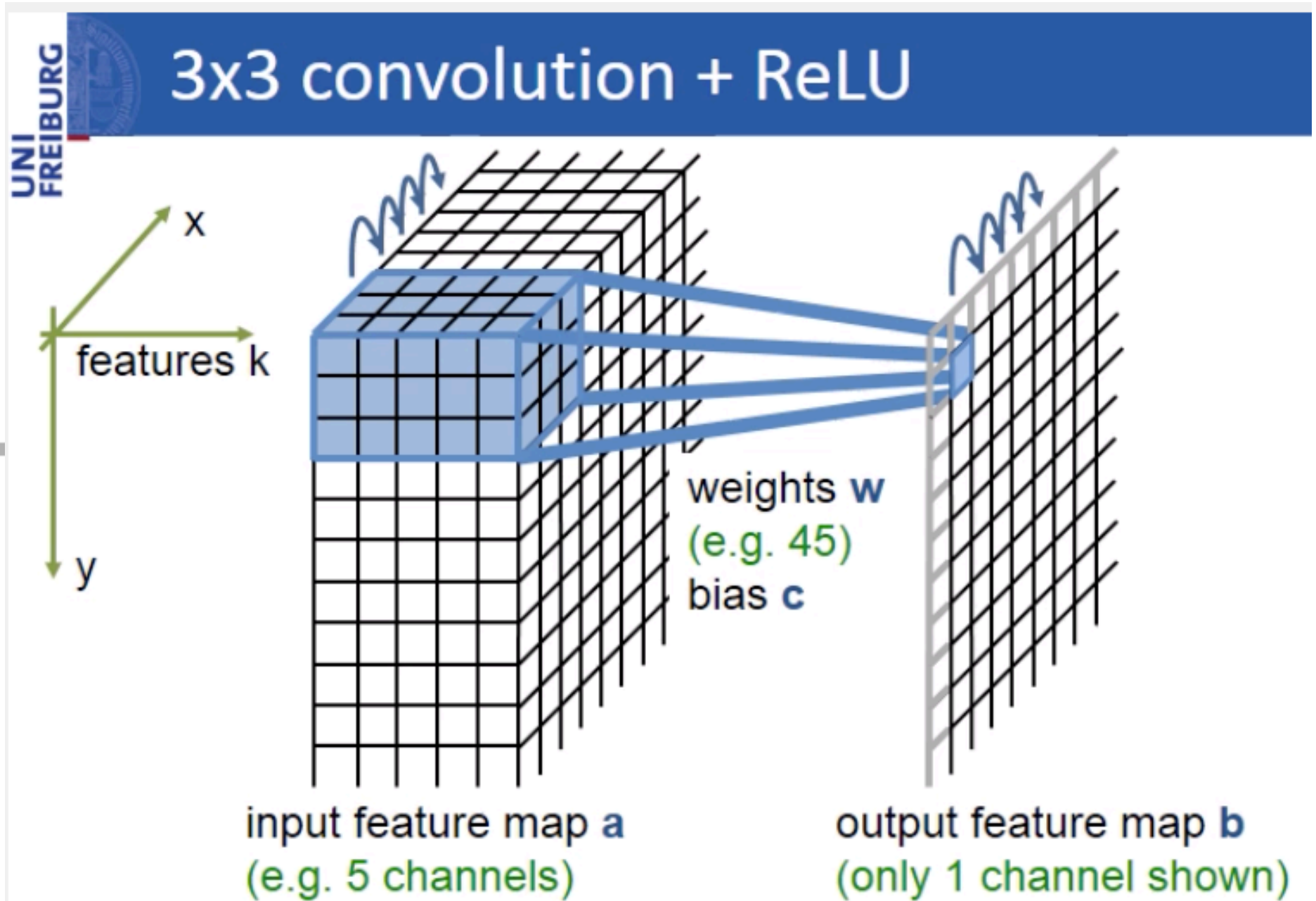
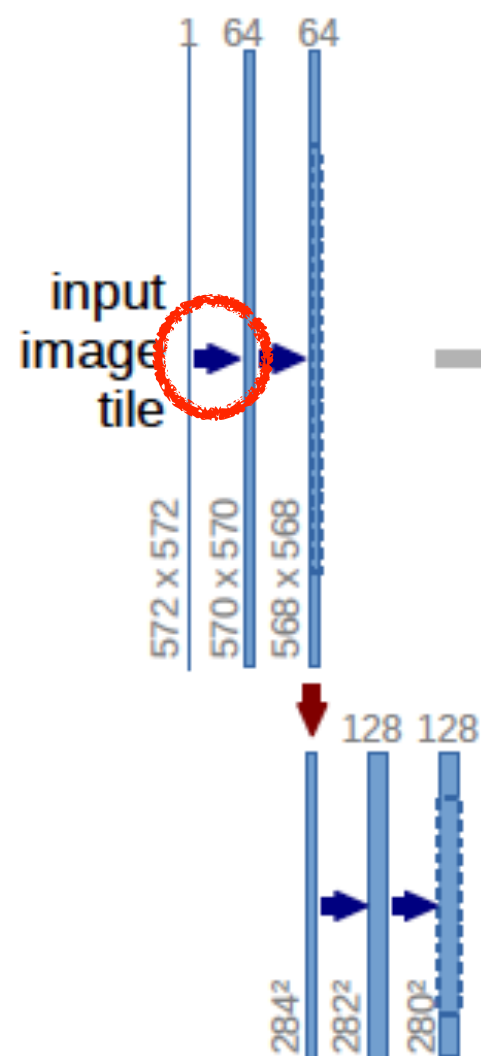


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (a) raw image. (b) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (c) generated segmentation mask (white: foreground, black: background). (d) map with a pixel-wise loss weight to force the network to learn the border pixels.

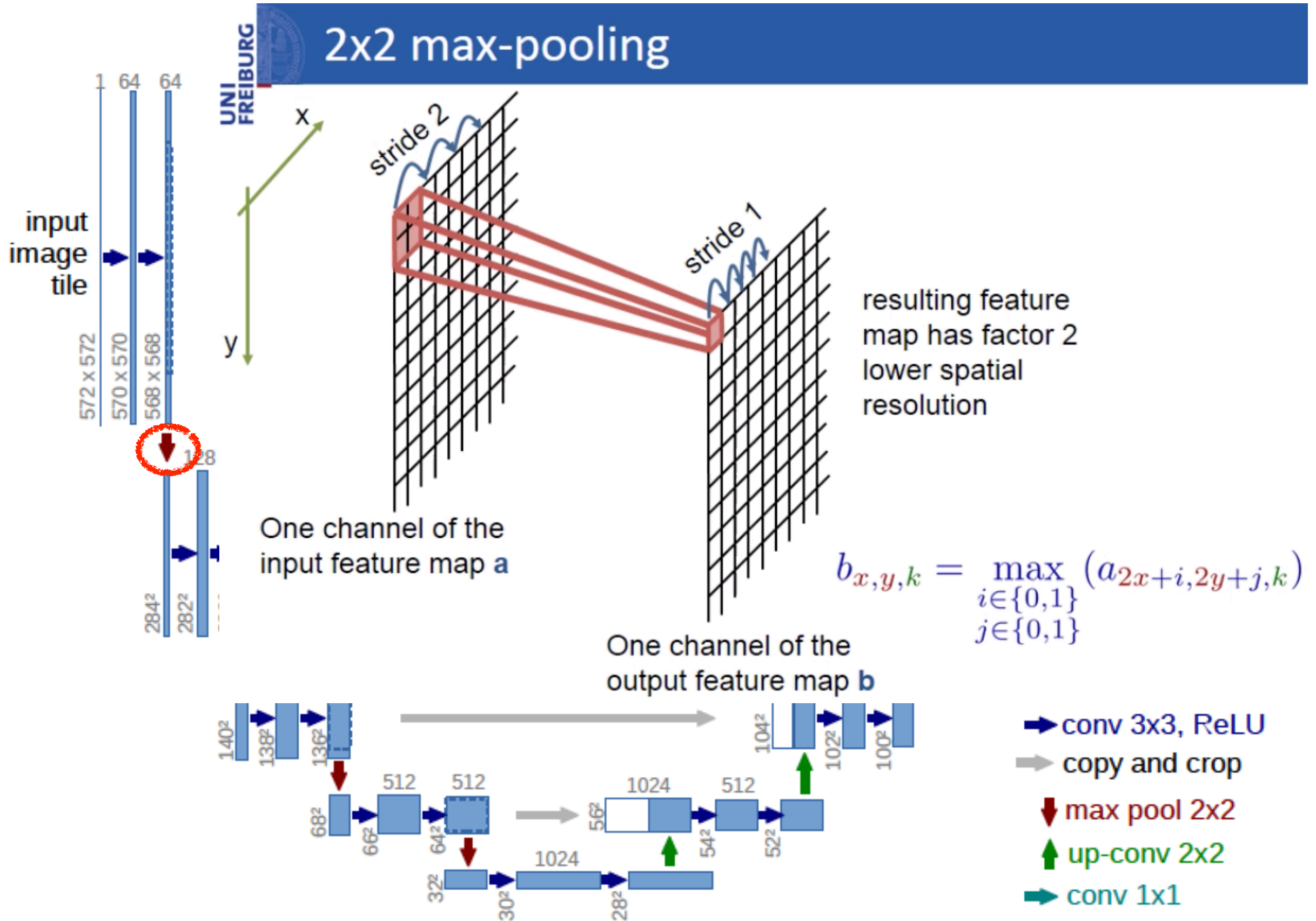
► U-Net



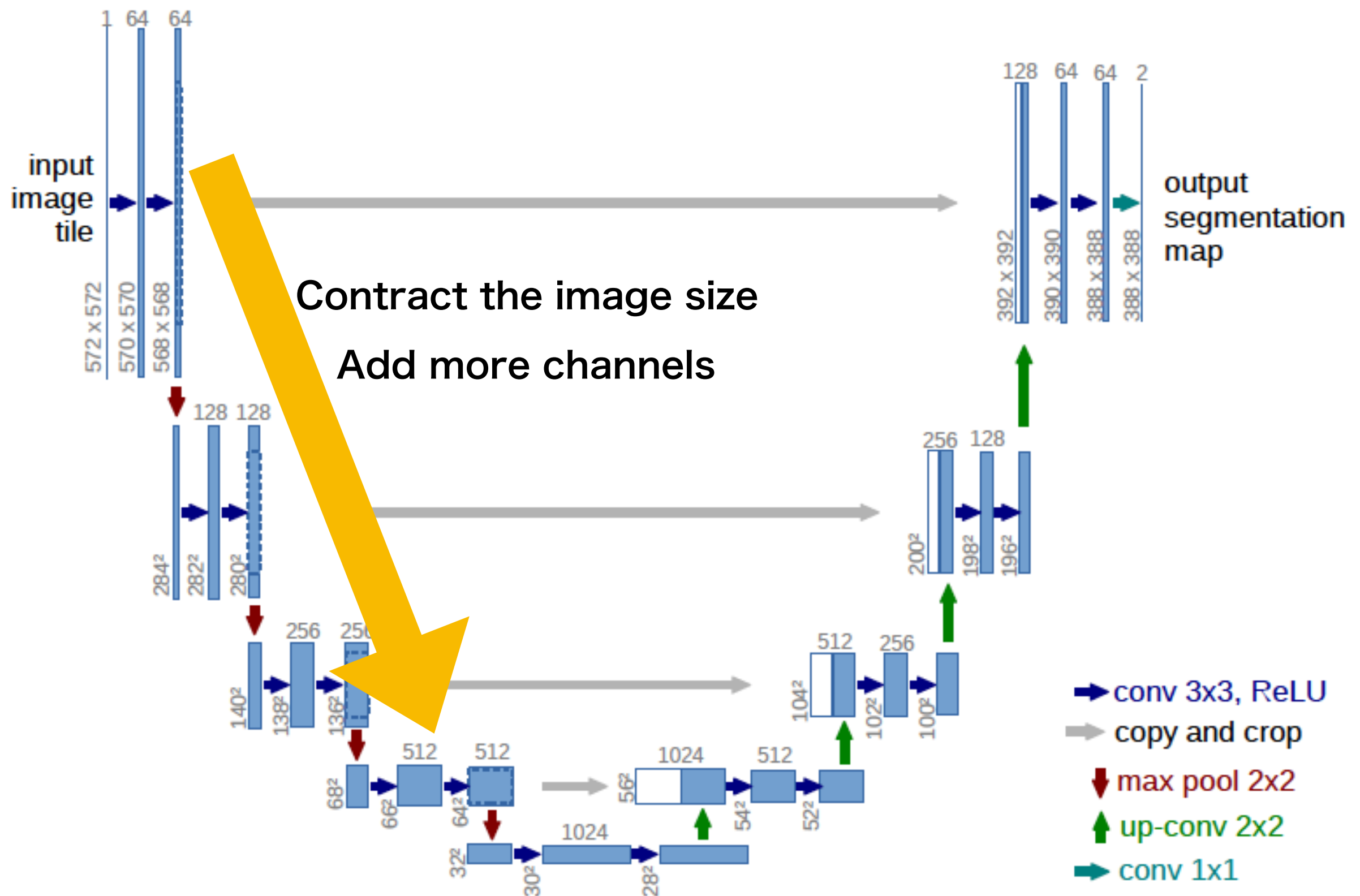
► U-Net



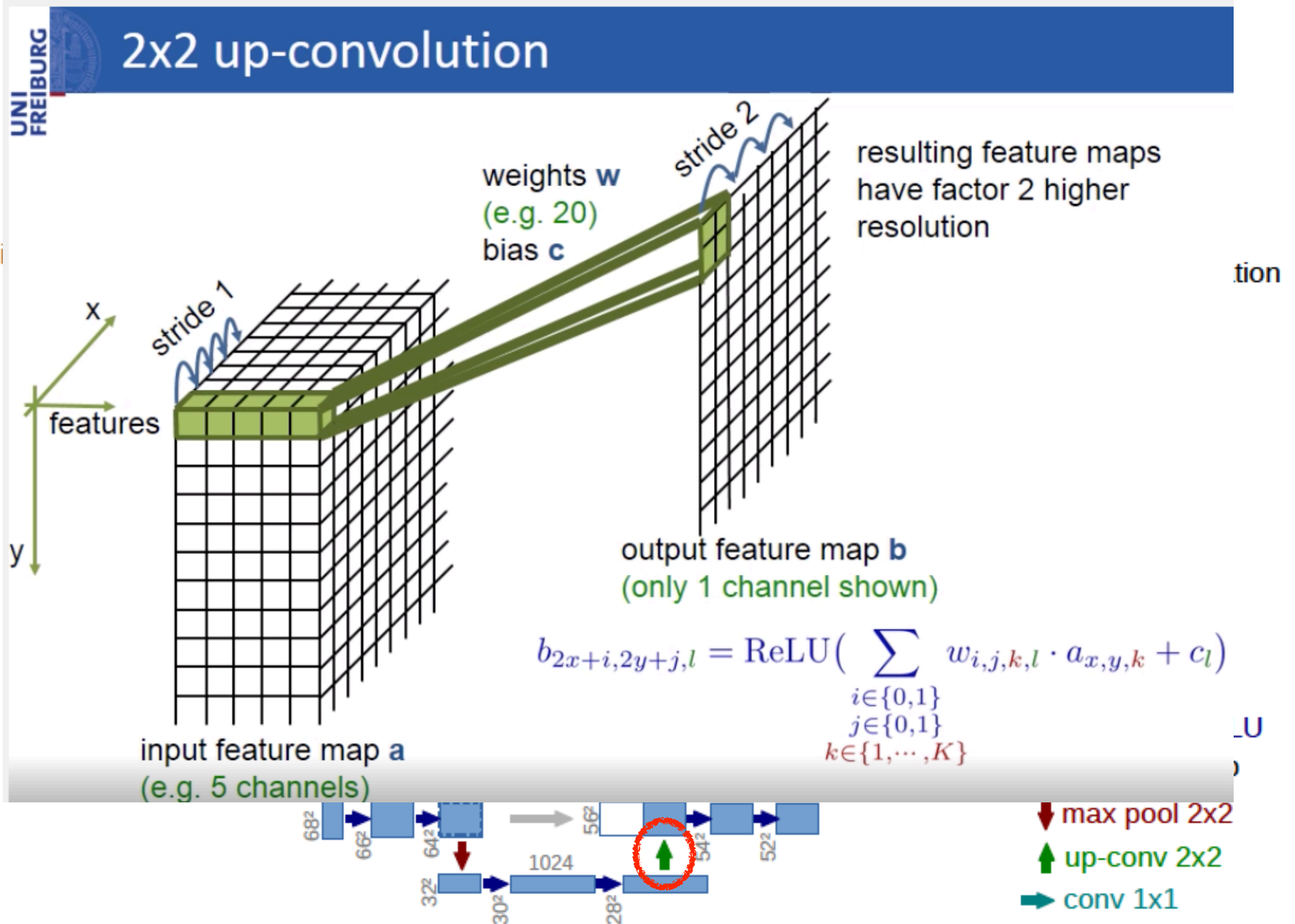
► U-Net



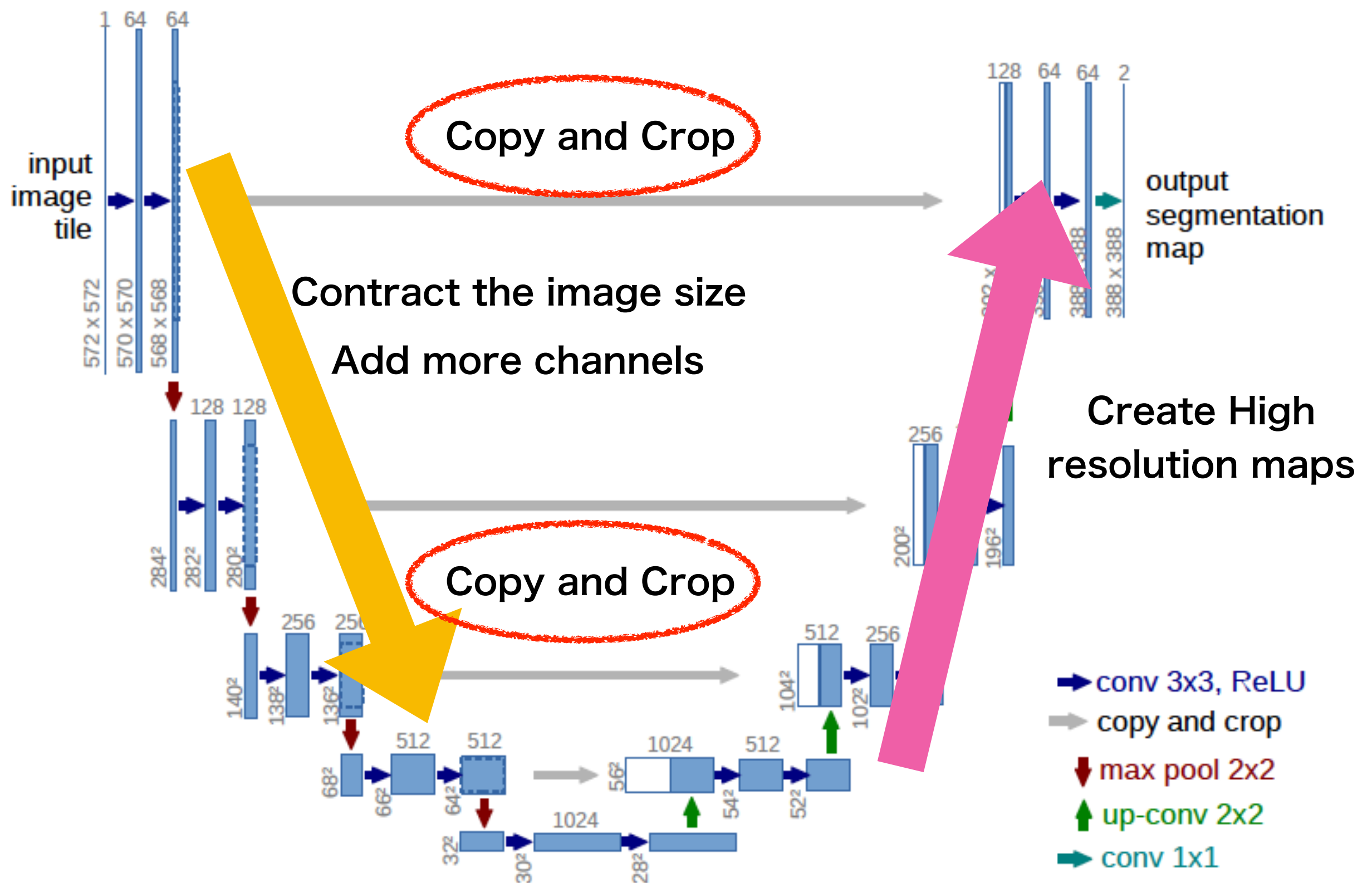
► U-Net



► U-Net



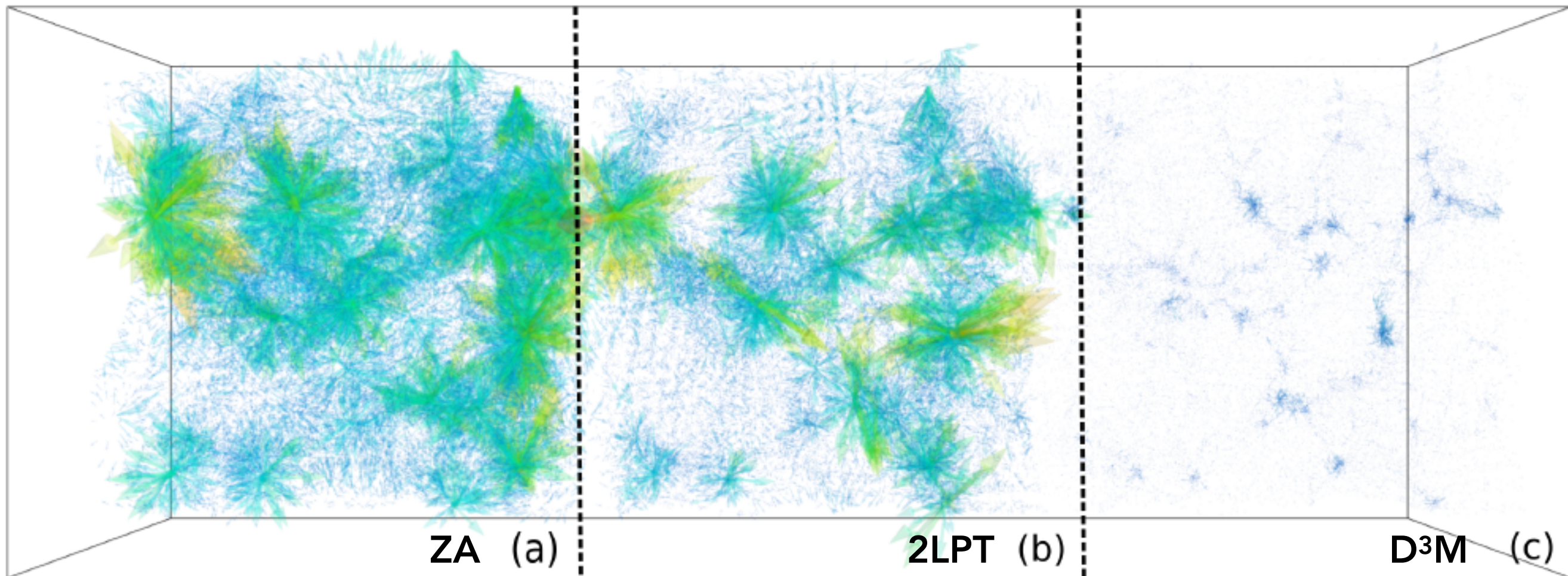
► U-Net



- ▶ U-Net implementation for this study
 - ▶ 2D \rightarrow 3D
 - ▶ Care must be taken for
 - ▶ periodic boundary
 - ▶ Shift and rotational symmetry of the system
 - ▶ Initial and final feature channels:
 - ▶ ~~Density (1 channel)~~ or **displacement** (3 channels)
 - ▶ At 32^3 locations
 - ▶ Total of 8.6×10^6 trainable parameters
 - ▶ Loss function

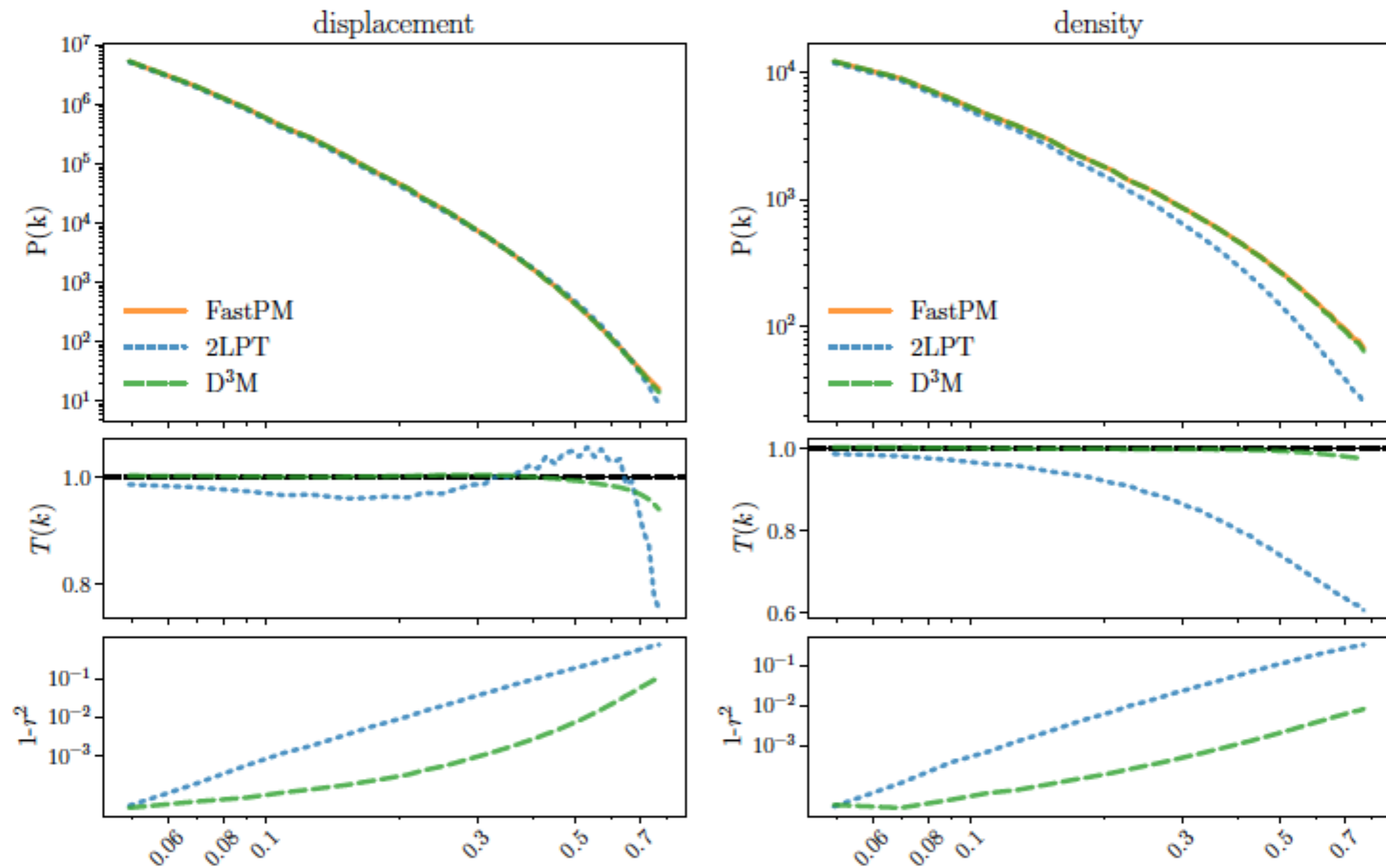
$$\mathcal{L} = \frac{1}{N} \sum_i (\hat{\Psi}_i - \Psi_{t,i})^2,$$

► Results (error vector)



Average relative displacement error: 9.3% (2LPT) vs 2.8% (D³M)

► Results (2pt correlation function)

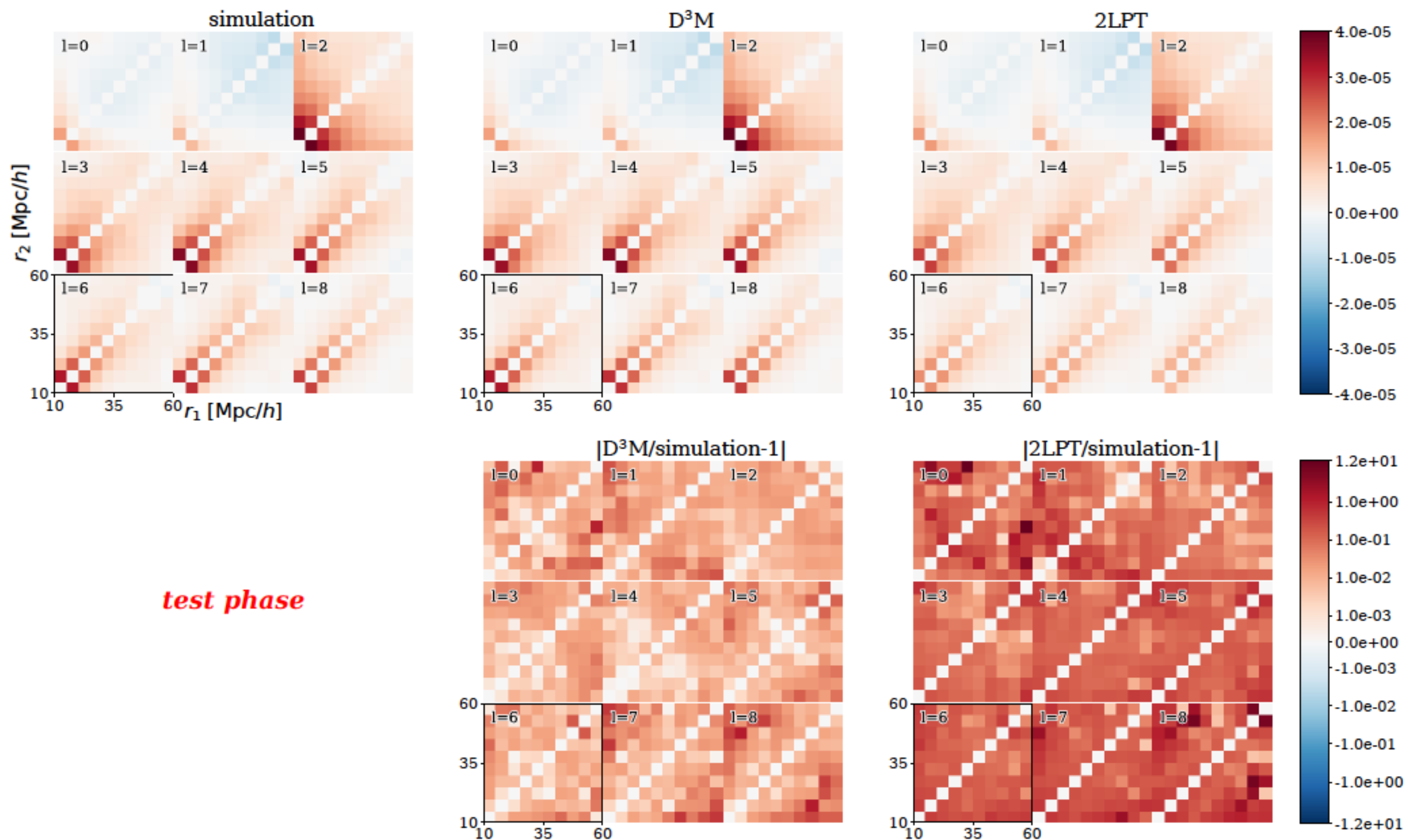


(a) 2-point analysis

$$T(k) = \sqrt{\frac{P_{\text{pred}}(k)}{P_{\text{true}}(k)}}$$

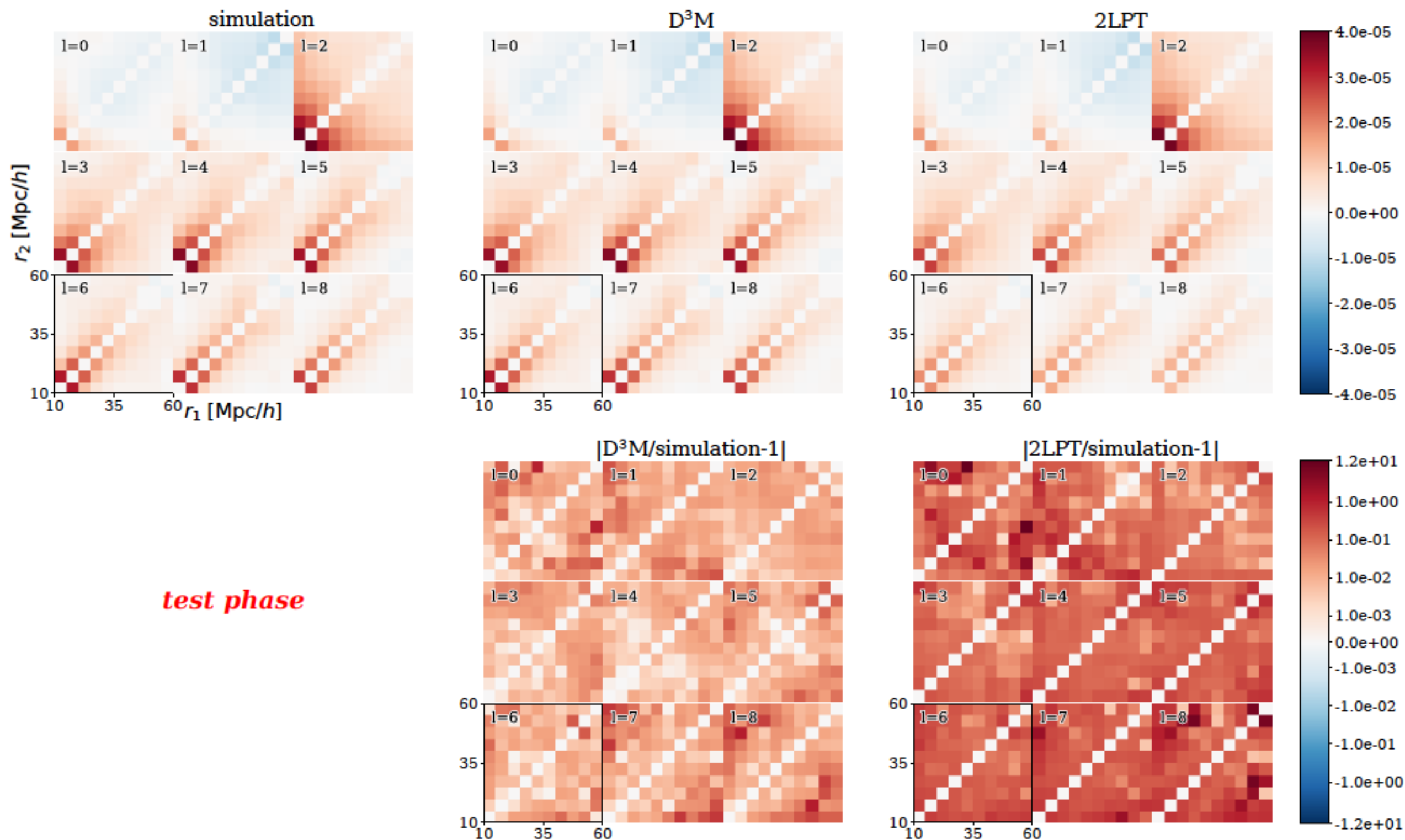
$$r(k) = \frac{P_{\text{pred} \times \text{true}}(k)}{\sqrt{P_{\text{pred}}(k)P_{\text{true}}(k)}},$$

► Results (3pt correlation function)



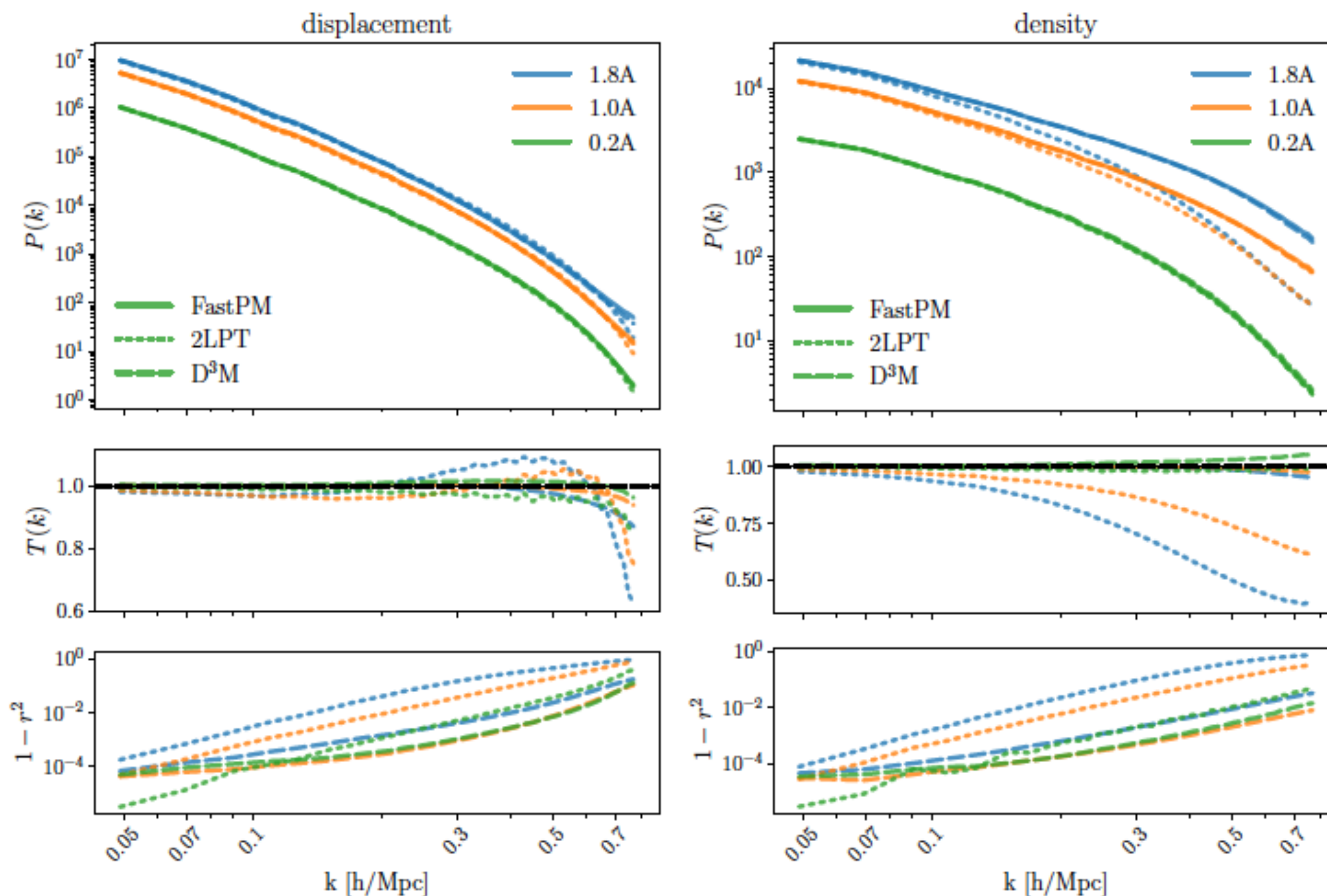
$$\zeta_\ell(r_1, r_2) = (2\ell + 1) \int d\theta P_\ell(\cos \theta) \zeta(r_1, r_2, \theta)$$

► Results (3pt correlation function)



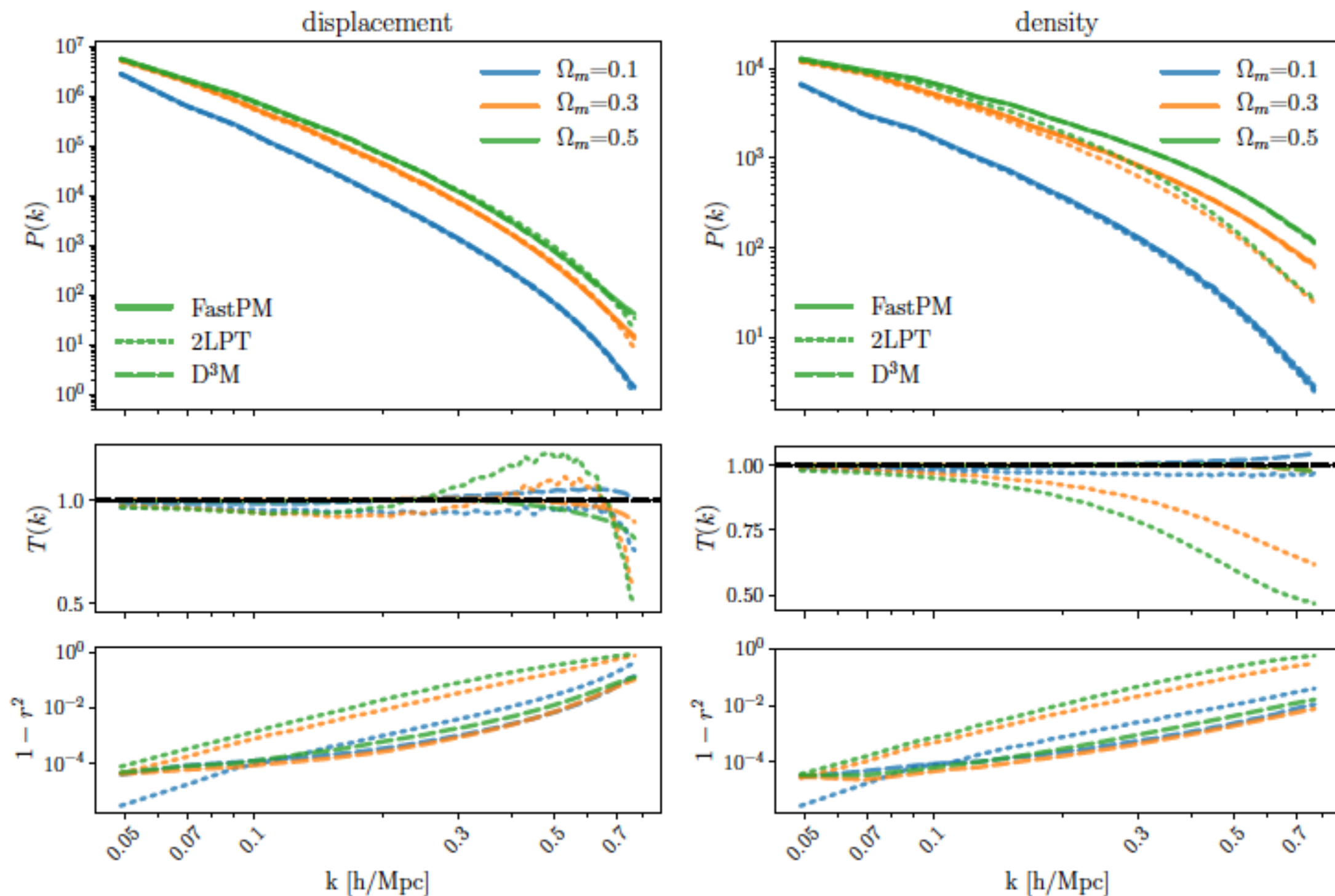
$$\zeta_\ell(r_1, r_2) = (2\ell + 1) \int d\theta P_\ell(\cos \theta) \zeta(r_1, r_2, \theta)$$

► Results (2ptCF, A_s varied cosmology)



(a) Varying Primordial Amplitude of Scalar Perturbations A_s

► Results (2ptCF, Ω_m varied cosmology)



(b) Varying matter density parameter Ω_m

[3] CosmoFlow: Using Deep Learning to Learn the Universe at Scale

- ▶ ML for **backward** modeling of the cosmic large scale structure
- ▶ DL application with high scalability
 - ▶ Efficient implementation of many element-wise operations for **threading** to improve training performance on Intel Xeon Phi processors
 - ▶ Also utilize Cray PE Machine Learning Plugin for efficient scaling to multiple **nodes**
- ▶ First large-scale science application of the TensorFlow framework at supercomputer scale with fully synchronous training
 - ▶ 8192 nodes, 77% parallel efficiency, 3.5 Pflop/s

► Goal

- 3D cosmology datasets (large-scale structure) —> cosmological parameters
- Significant update from their previous paper (with only cosmologists; arXiv: 1711.02033)
 - x3 improved constraint on (Ω_m , σ_8) compared to the standard $P(k)$ analysis

► Dataset

- $L=512\text{Mpc}/h$, $N=512^3$
- 12,632 simulations with COLA
 - 150 as validation data, 50 as test data
- Density values in 8 sub-volumes at 128^3 voxels (101,056 sub-volumes in total)
- Simulations with 3 varied cosmological parameters (Ω_m , σ_8 , n_s)

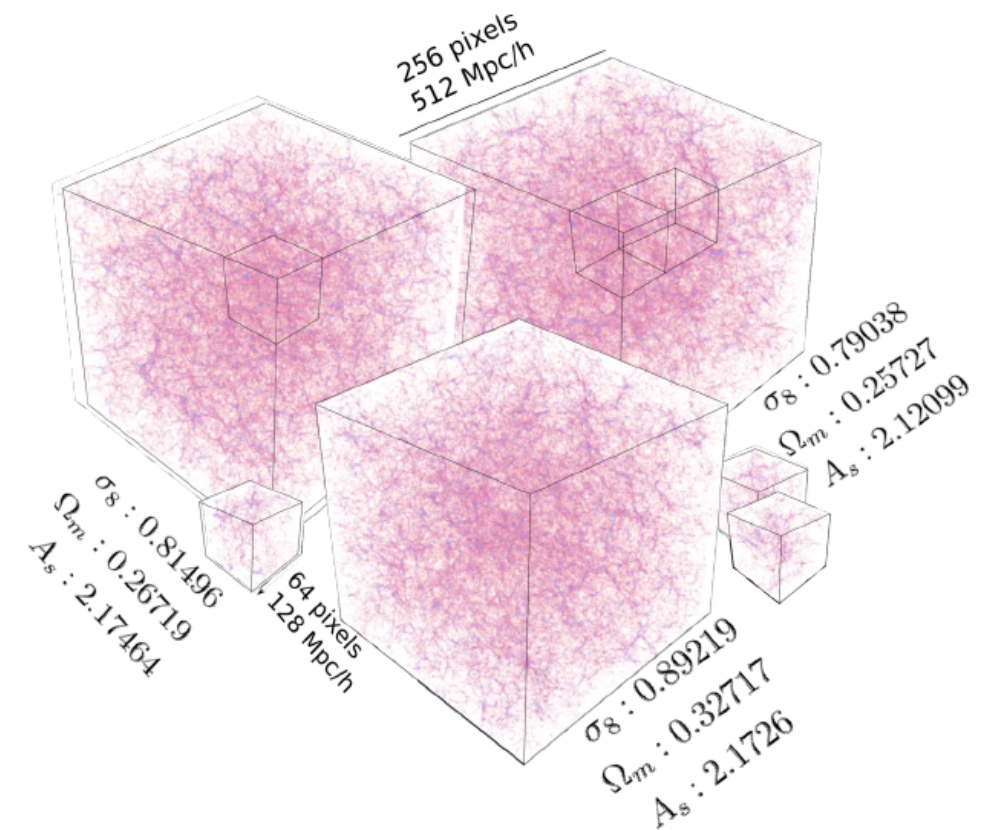


Figure 1. Dark matter distribution in three cubes produced using different sets of parameters. Each cube is divided into small sub-cubes for training and prediction. Note that although cubes in this figure are produced using very different cosmological parameters in our constrained sampled set, the effect is not visually discernible.

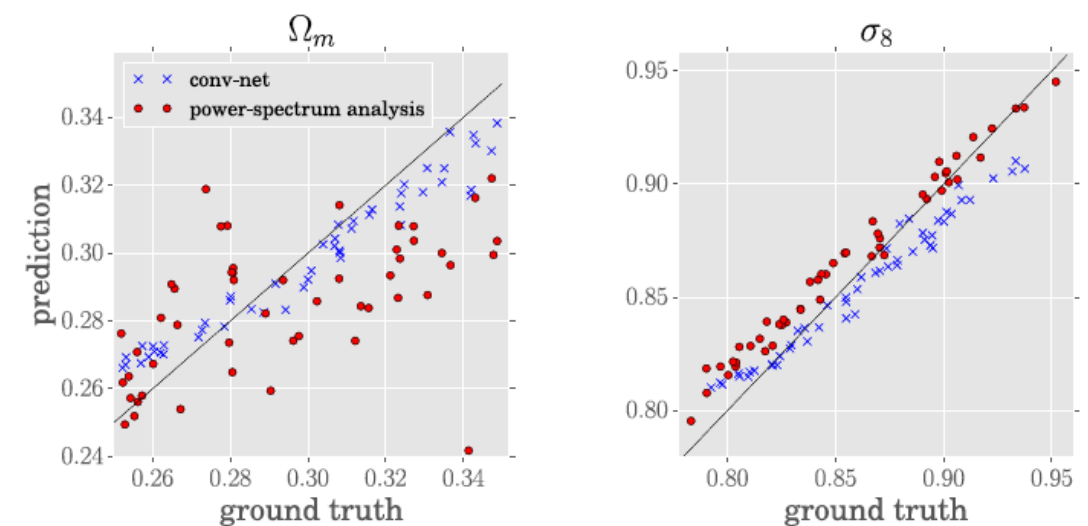


Figure 2. Prediction and ground truth of Ω_m and σ_8 using 3D conv-net and analysis of the power-spectrum on 50 test cube instances.

► ML architecture and training

>7,000,000 parameters

69.33 Gflop / mimi-batch (mini-batch size = 1)

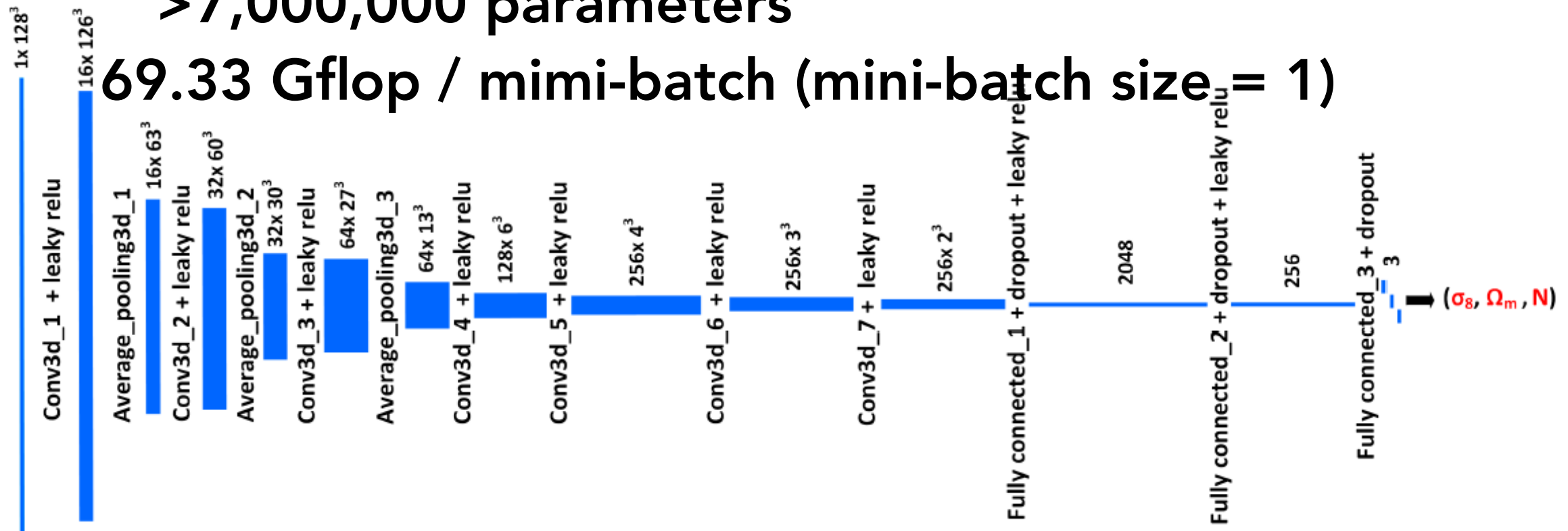


Fig. 2. CosmoFlow network topology, showing the different network layers and data sizes at each layer.

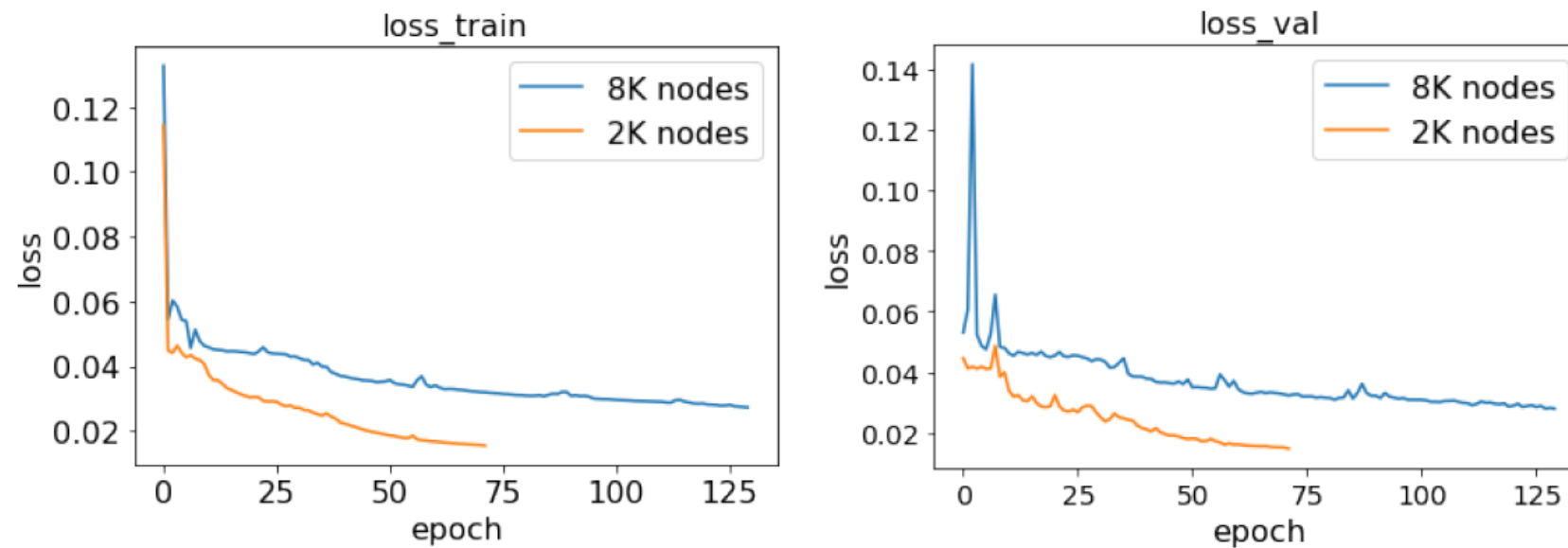


Fig. 5. Loss function for the training dataset (left) and validation dataset (right), for 2048 and 8192 node runs. Note that the datasets used in the two runs are not same and of different size, so the loss function and epoch time cannot be directly compared.

► Results

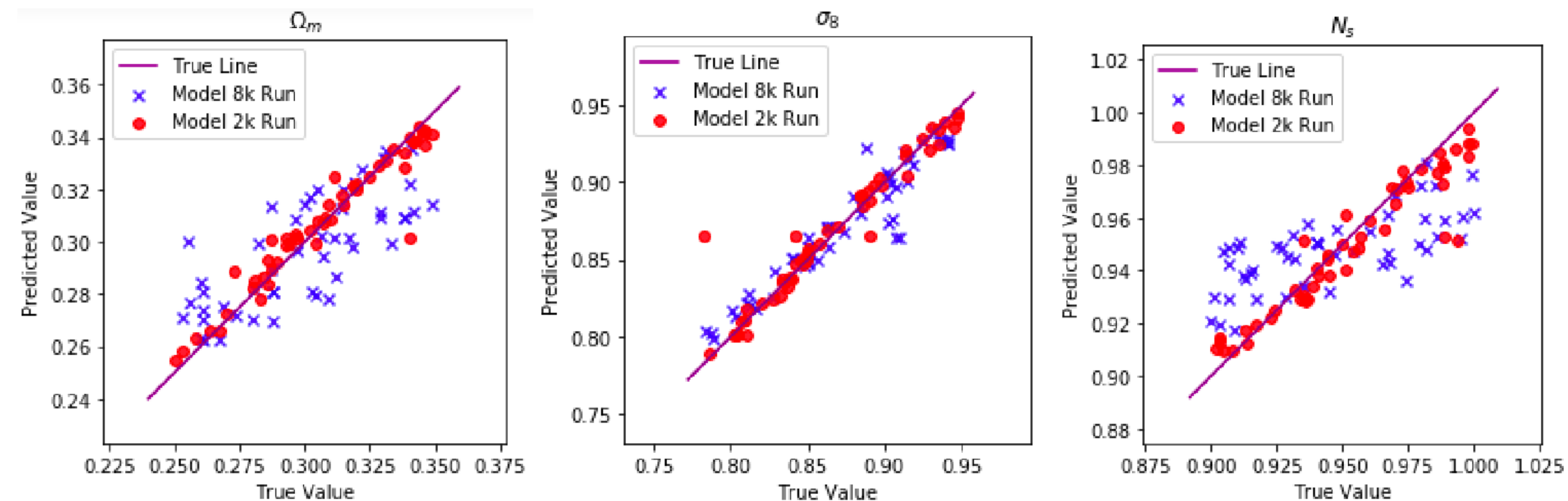


Fig. 6. Estimates of Ω_M , σ_8 and n_s from the 2048- and 8192-node runs.