Exploring Deep Learning Domain Adaptation Performance: From Covariate Shift to Wasserstein and beyond

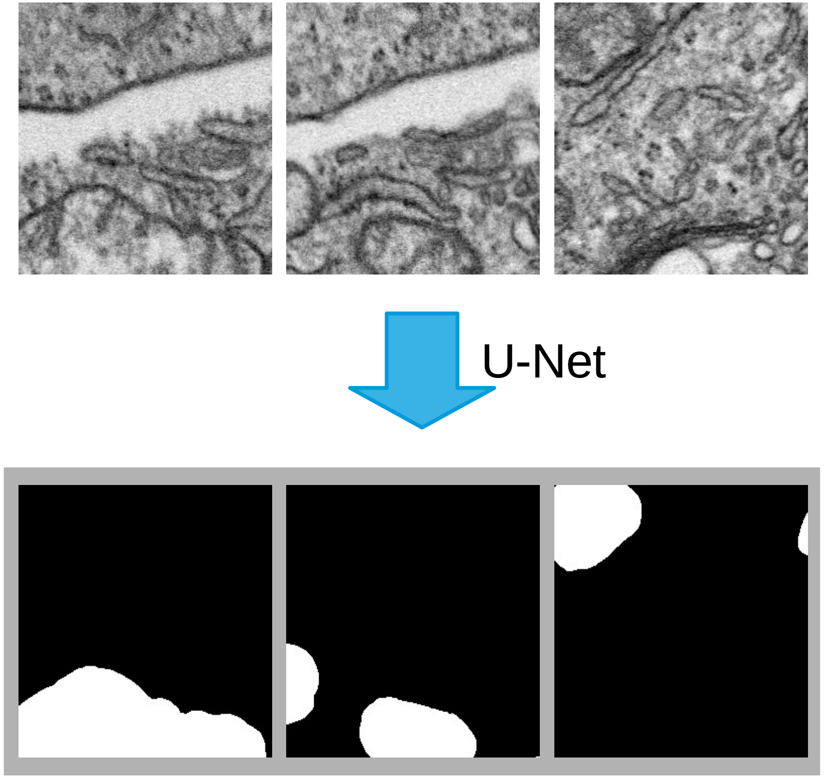
Encadrement: Étienne Baudrier, Sylvain Faisan, Alexandre Stenger

07/21/2023

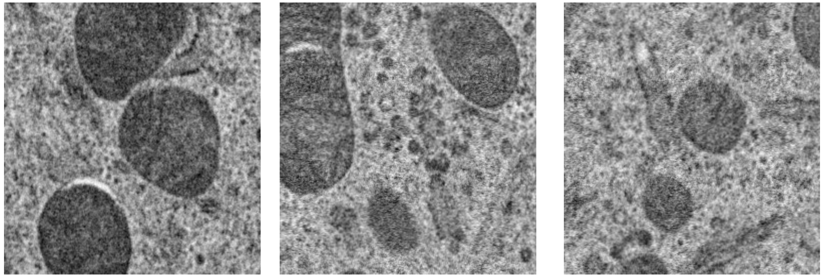
Introduction

Introduction: Covariate Shift

Source data *(chemical fixation)*

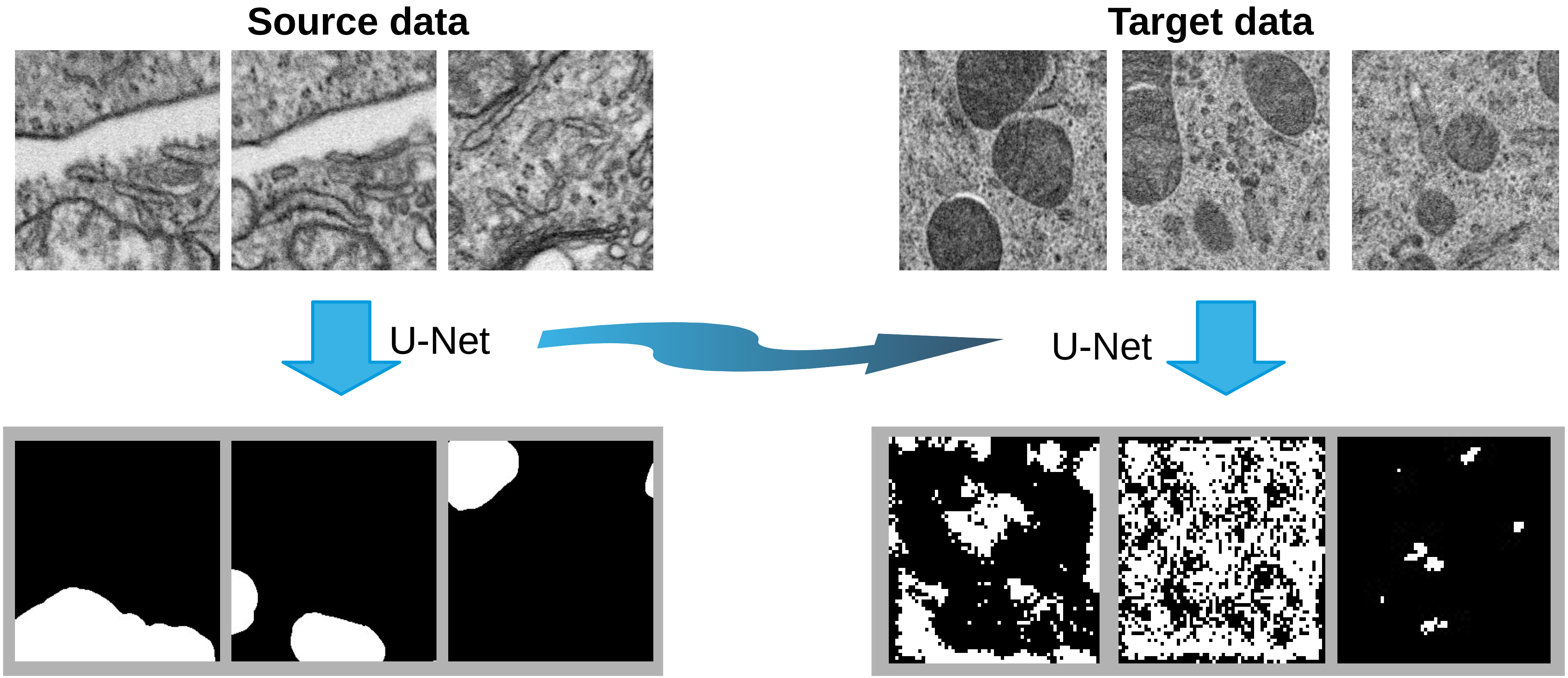


Target data *(cryo-fixation)*



Covariate shift

Introduction: Covariate Shift



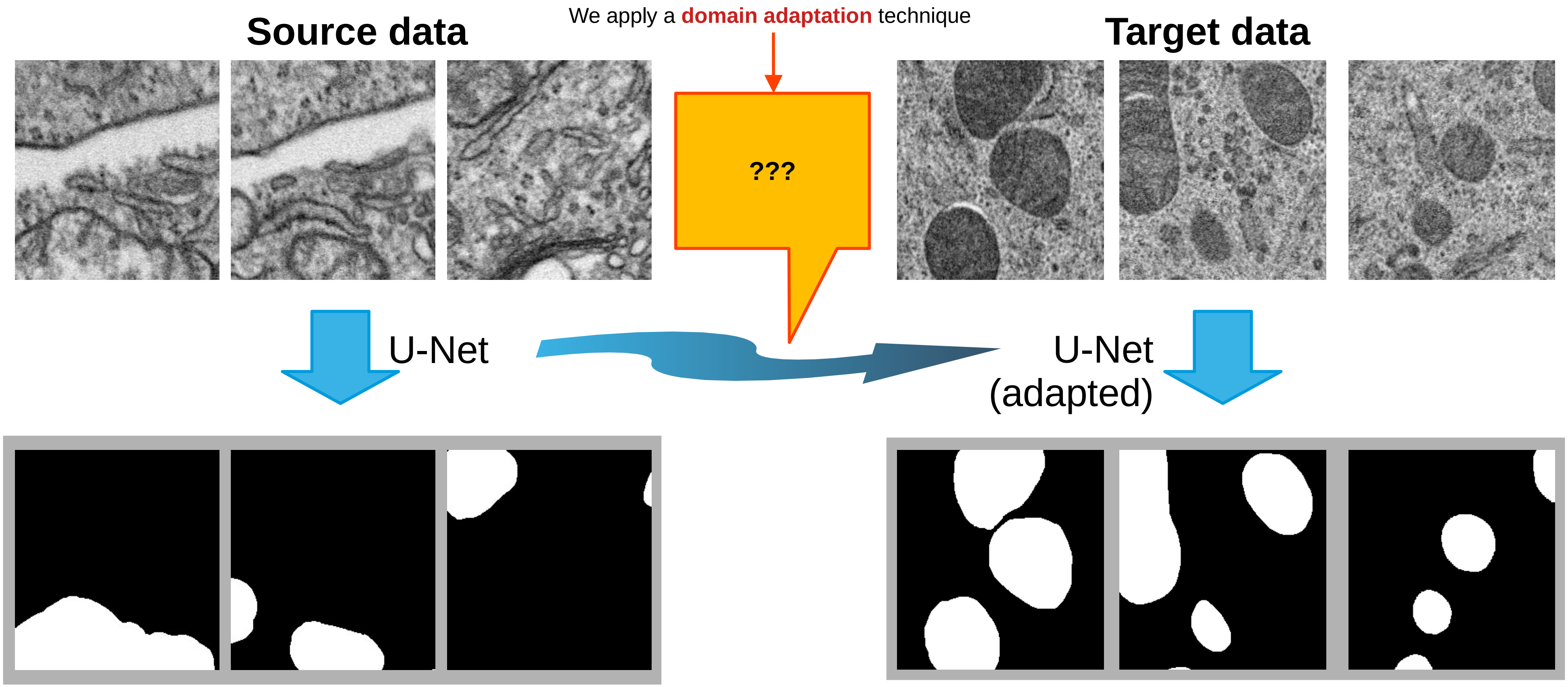
Source data

Target data

U-Net

U-Net

Introduction: Covariate Shift

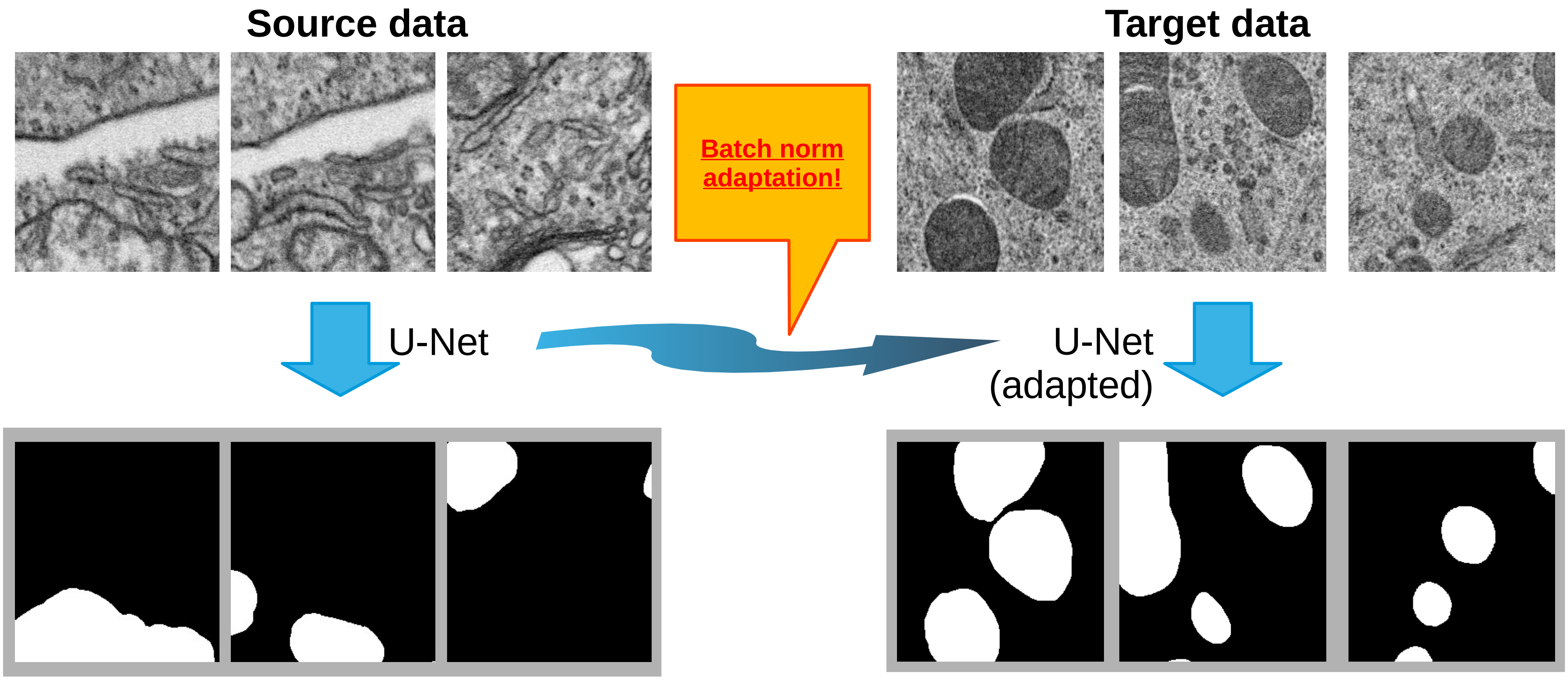


Source data

We apply a **domain adaptation** technique

Target data

Introduction: Covariate Shift



Source data

Target data

Batch norm adaptation!

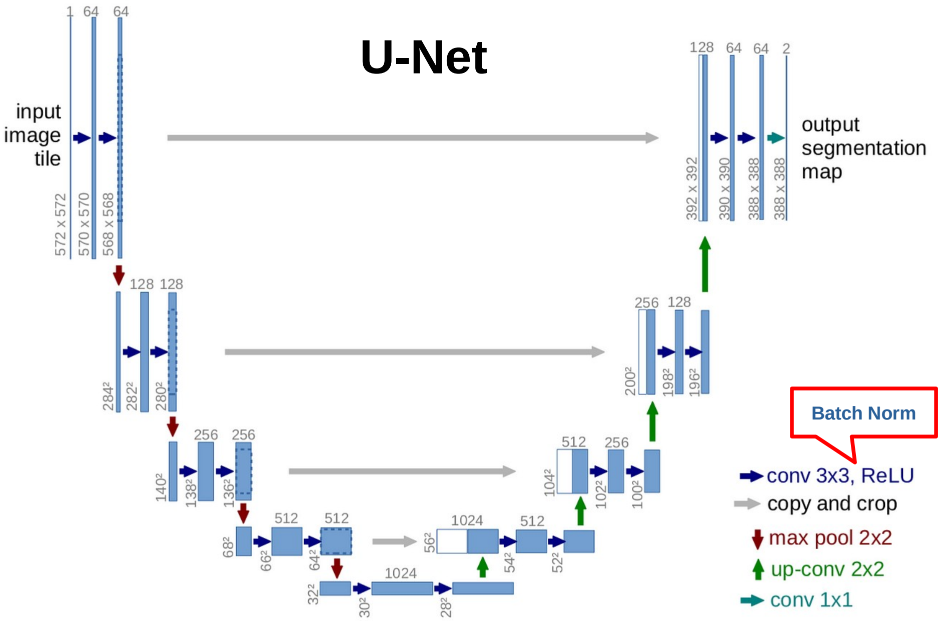
U-Net

U-Net (adapted)

Introduction: Batch Norm adaptation

What is Batch Norm?

U-Net



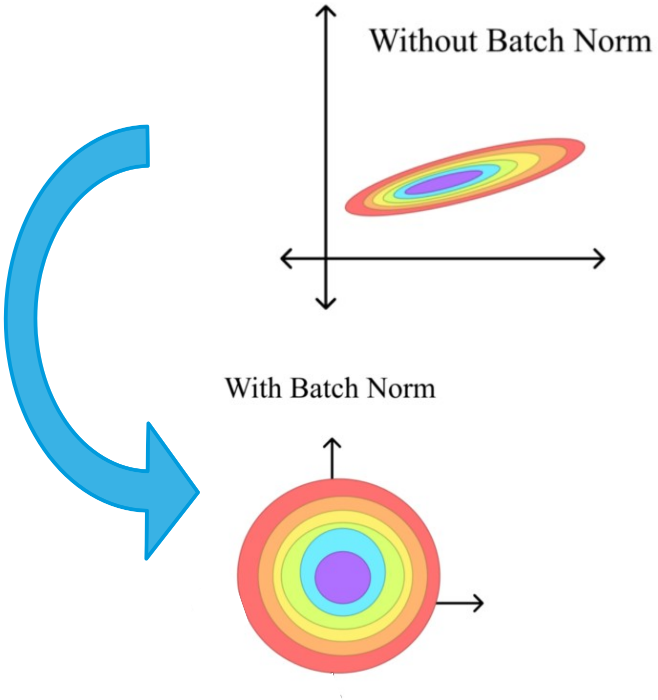
Batch Norm Layer

$$\hat{\mu} = \mathbb{E}[x]$$

$$\hat{\sigma} = \sqrt{\mathrm{Var}[x]}$$

$$y = \frac{x - \hat{\mu}}{\hat{\sigma}} * \gamma + \beta$$

γ, β learned during training



Introduction: Batch Norm adaptation

What is Batch Norm adaptation?

- we take the **target dataset** and pass it through the network, freezing all parameters but updating $\hat{\mu}$ and $\hat{\sigma}$

Introduction: The work of my équipe

FAST AND INTERPRETABLE UNSUPERVISED DOMAIN ADAPTATION FOR FIB-SEM CELL SEGMENTATION

Alexandre Stenger *† *Luc Vedrenne* *† *Patrick Schultz* ‡ *Sylvain Faisan* †
Étienne Baudrier † *Benoît Naegel* †

† ICube, UMR 7357 CNRS, University of Strasbourg, France
‡ IGMBC, UMR 7104 CNRS, U1258 INSERM, University of Strasbourg, France

ABSTRACT

Electron microscopy imaging techniques allow biologists to obtain nanoscale slice volumes of cells in which biological structures can be statistically analyzed under the condition that the tedious task of segmentation can be automated. If a learned segmentation has good results on images obtained under the same acquisition process, variations in this acquisition can drastically affect the final image's features and can lead to a drop of segmentation performances. To tackle this problem, we propose an Unsupervised Domain Adaptation (UDA) framework based on Batch Normalization. Our method tested on mitochondria segmentation shows better results than state-of-the art methods and uses few computation resources, allowing a near real-time use. Furthermore, we propose a metric based on the Wasserstein distance to evaluate the effect of this normalization. We empirically show that this measure is log-linearly correlated with the drop of performances of the adapted segmentation and sheds light on the gain obtained by our UDA framework.

Index Terms— Unsupervised domain adaptation, Cell mitochondria segmentation, FIB-SEM, quantifying domain shift.

1. INTRODUCTION

In the context of Focused Ion Beam with Scanning Electron Miscroscopy (FIB-SEM), an automatic segmentation network is able to predict segmentation mask when trained on cells imaged with a single acquisition process [1, 2]. However huge domain shifts can appear between two images depending on the acquisition protocol used. Thus, between chemical fixation (Fig. 1(a)) and cryo-fixation (Fig. 1(c)) processes, mitochondria change severely in terms of shape, contrast and texture (see segmentation masks Fig. 1(b)(d)), which leads to a drop of performance when a model trained on a database acquired with one protocol is applied to databases acquired with an other protocol. This situation leads either to call on tedious human segmentation, or to adapt the model in some way.

(a) FS1 (b) FS1 mask (c) FS2 (d) FS2 mask

Fig. 1. Example of 256 × 256 patches from our cells stacks FS1 and FS2 (a,c) as described in Section 4.1, with their mitochondria segmentation ground truth (b,d).

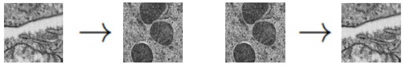
Domain Adaptation (DA) gained a broad interest with the great expansion of deep learning. In the context of semantic segmentation, most of the work still focus on adapting Cityscapes, GTA5 and SYNTHIA datasets [3]. Applying the same frameworks to biomedical images turns out to be very challenging, due to little datasets, lack of clear structure and high internal variation in this kind of data. As regards domain adaptation for cell segmentation, *Haq et al.* [4] introduced adversarial methods (based on GANs) to overcome domain shift between histological data, while *Li et al.* [5] used the same method. *Roels et al.* investigated Y-Net architecture for Electron Microscopy cell segmentation [6] along with *Franco et al.* brought in Attention Y-Net [7]. Those methods are hard to train, time consuming and energy-guzzling. On the other side, fast and easy domain adaptation based on BatchNorm statistics recomputation caught our attention. As much as this layer is able to reduce internal covariate shift [8], it can also reduce inter-domain shift [9]. The effectiveness of this method depends on the statistics estimate quality which requires a huge amount of data. However, we propose an adaptation framework based on BatchNorm recomputation and show that in some cases, it works even with small datasets, making it suitable for biomedical images. We also introduce a metric to quantify the quality of the statistics estimate, and show that it is a good indicator of the performances after adaptation. Furthermore, this framework is fast, needs few computation resources, does not need any new network training and can be run by biologists on their own.

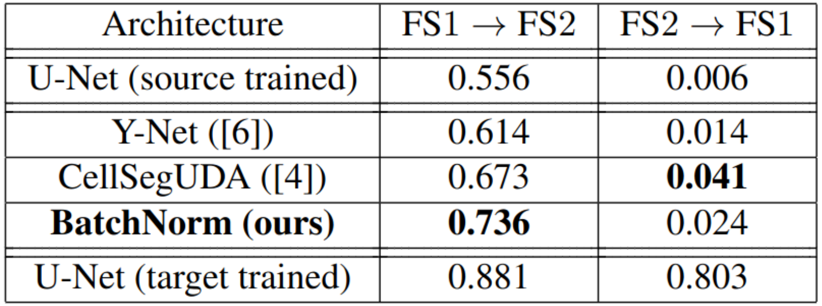
[*] Equal contribution

https://publis.icube.unistra.fr/docs/17711/ISBI_paper_559.pdf

FAST AND INTERPRETABLE UNSUPERVISED DOMAIN ADAPTATION FOR FIB-SEM CELL SEGMENTATION

Alexandre Stenger *† *Luc Vedrenne* *† *Patrick Schultz* ‡ *Sylvain Faisan* †
Étienne Baudrier † *Benoît Naegel* †

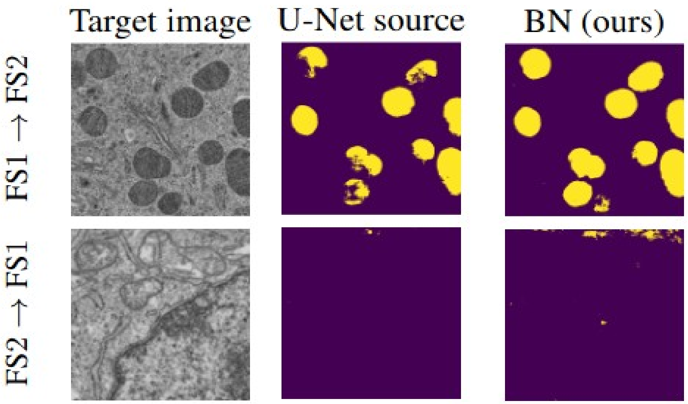




Architecture	FS1 → FS2	FS2 → FS1
U-Net (source trained)	0.556	0.006
Y-Net ([6])	0.614	0.014
CellSegUDA ([4])	0.673	**0.041**
BatchNorm (ours)	**0.736**	0.024
U-Net (target trained)	0.881	0.803

Introduction: The work of my équipe

Asymmetric behaviour of
BN (Batch Norm adaptation)



Pros:
- Simple implementation
- Extremely fast

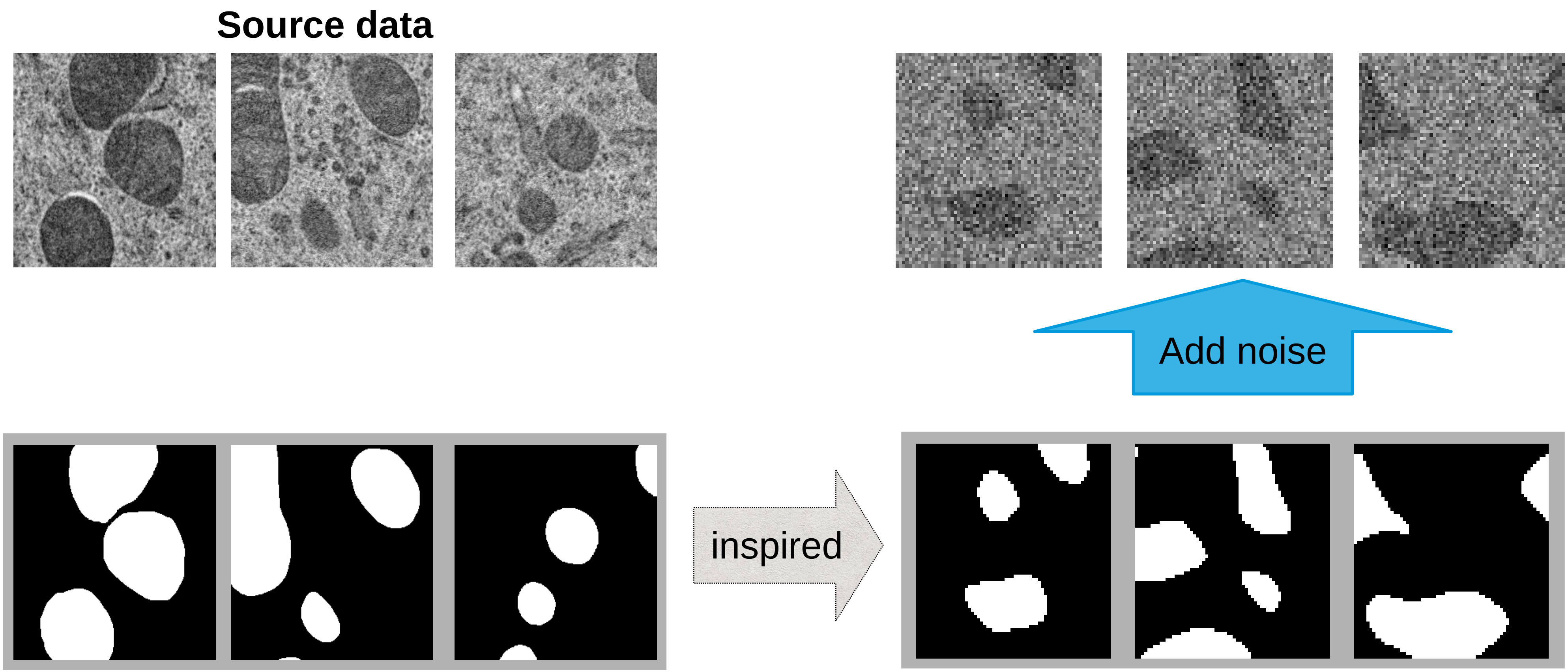
Cons:
- we don’t know when it works

My internship goal

When (and why) does Batch Norm adaptation work?

Dataset Generation

Dataset generation

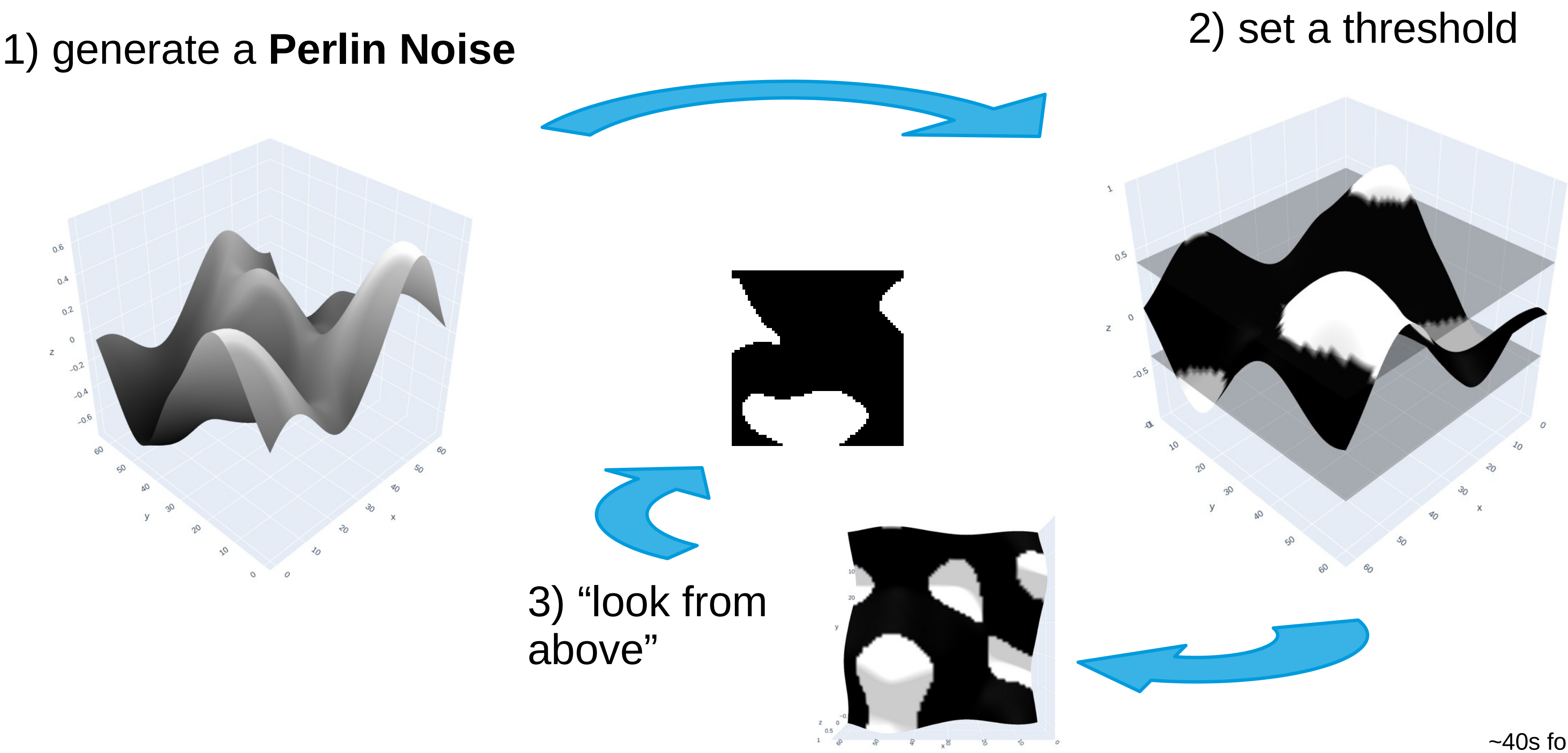


Source data

inspired

Add noise

Dataset generation



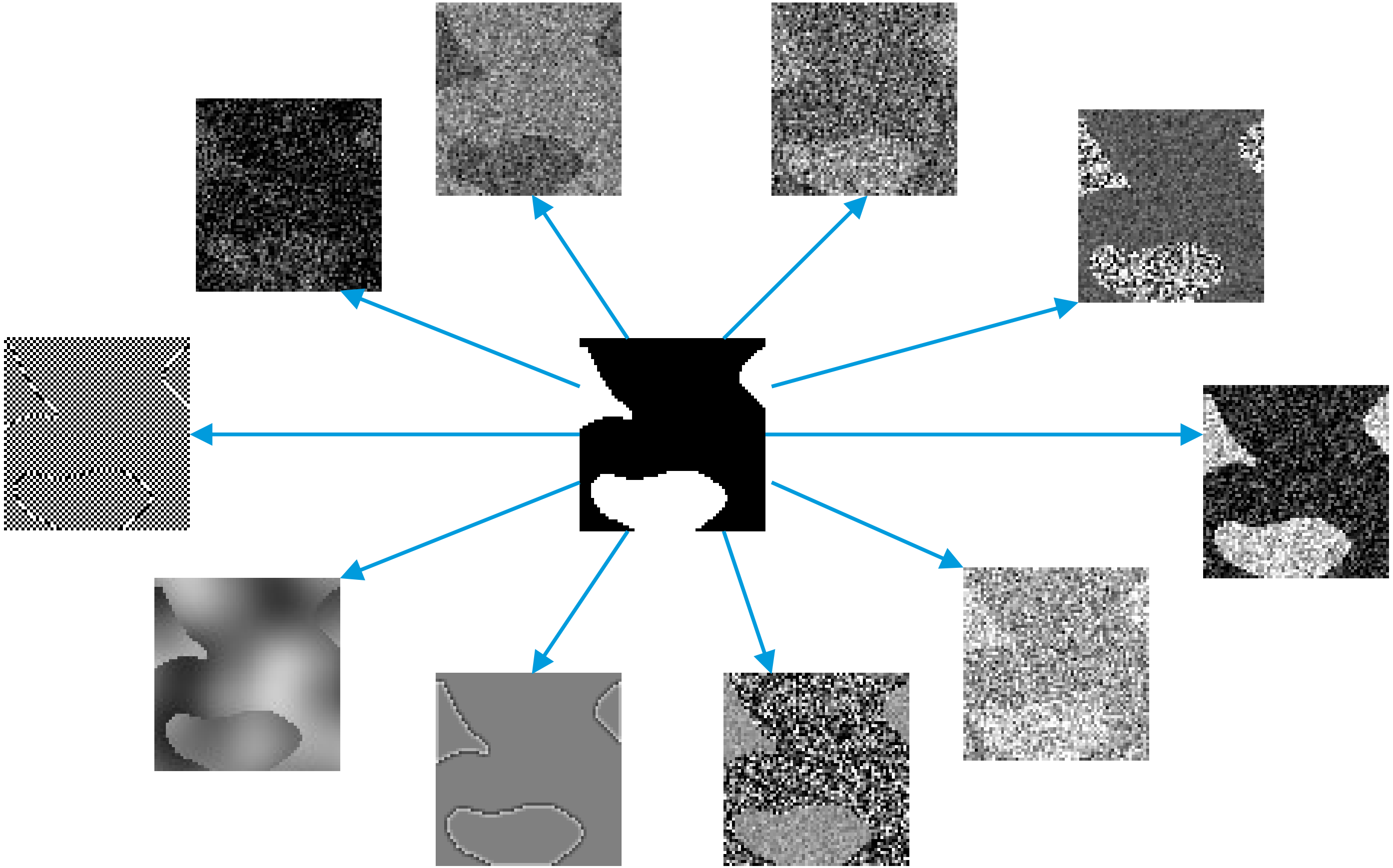
1) generate a **Perlin Noise**

2) set a threshold

3) “look from above”

~40s for 10.000 images

Dataset generation



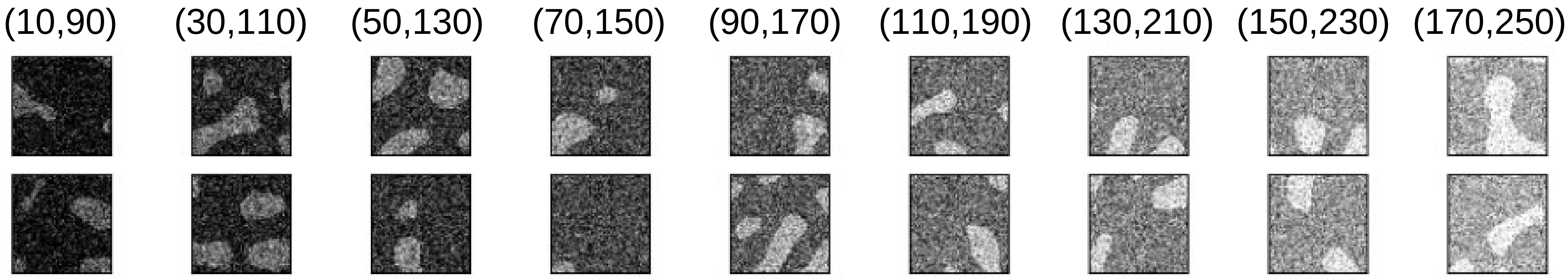
When does BN adaptation work?

when does BN adaptation work?

When BN adaptation works

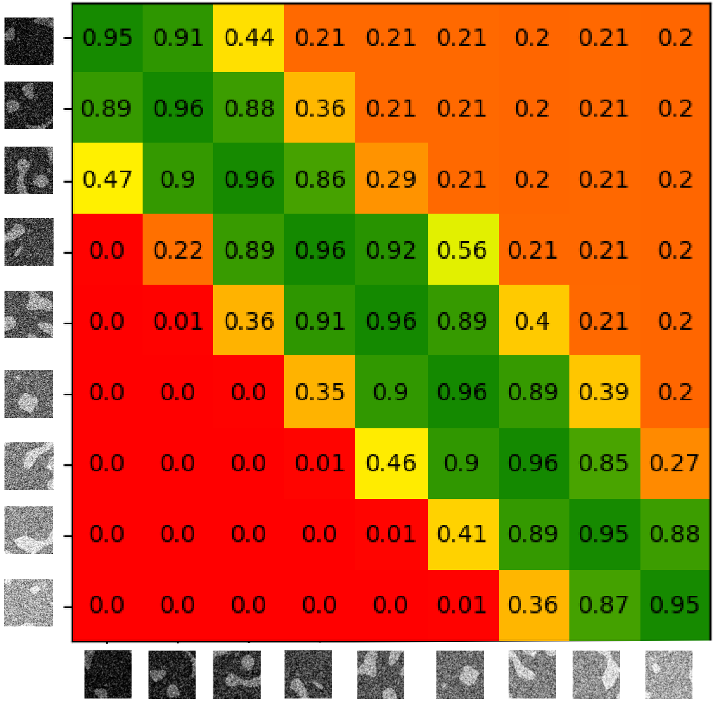
(means of mask and background white noises)

(μ_1,μ_2)



when does BN adaptation work?

U-Net WITHOUT BN adaptation

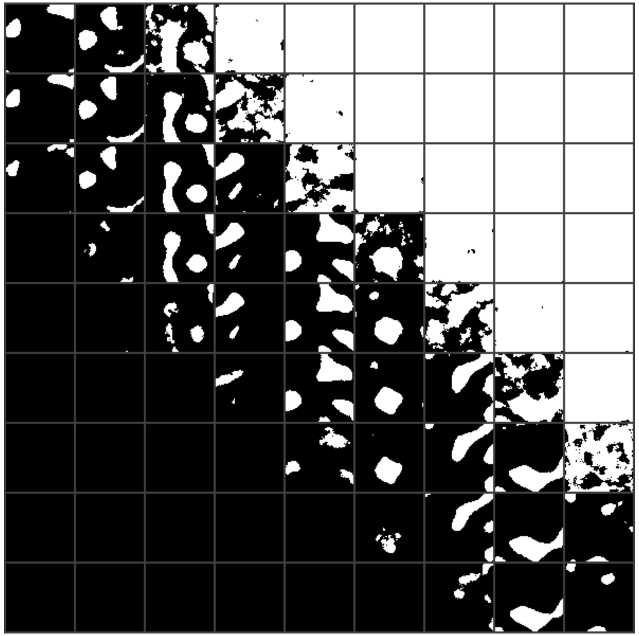


0.95	0.91	0.44	0.21	0.21	0.21	0.2	0.21	0.2
0.89	0.96	0.88	0.36	0.21	0.21	0.2	0.21	0.2
0.47	0.9	0.96	0.86	0.29	0.21	0.2	0.21	0.2
0.0	0.22	0.89	0.96	0.92	0.56	0.21	0.21	0.2
0.0	0.01	0.36	0.91	0.96	0.89	0.4	0.21	0.2
0.0	0.0	0.0	0.35	0.9	0.96	0.89	0.39	0.2
0.0	0.0	0.0	0.01	0.46	0.9	0.96	0.85	0.27
0.0	0.0	0.0	0.0	0.01	0.41	0.89	0.95	0.88
0.0	0.0	0.0	0.0	0.0	0.01	0.36	0.87	0.95

Source dataset

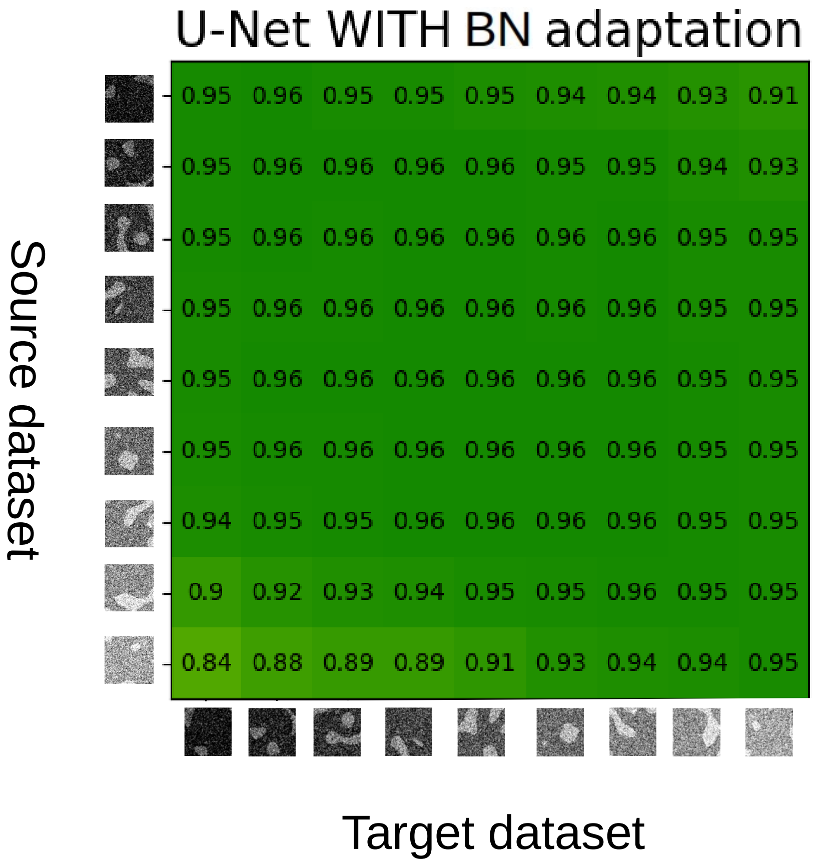
Target dataset

Mask predictions



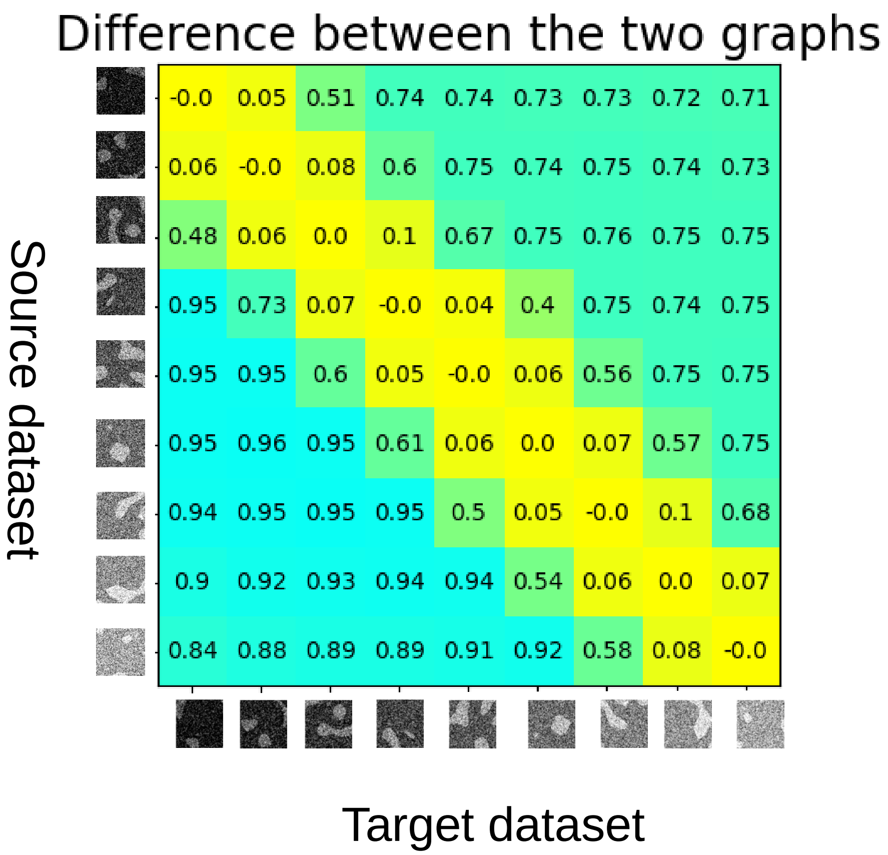
Source dark, target bright → all white
Source bright, target dark → all black

when does BN adaptation work?



U-Net WITH BN adaptation

0.95	0.96	0.95	0.95	0.95	0.94	0.94	0.93	0.91
0.95	0.96	0.96	0.96	0.96	0.95	0.95	0.94	0.93
0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95
0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95
0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95
0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95
0.94	0.95	0.95	0.96	0.96	0.96	0.96	0.95	0.95
0.9	0.92	0.93	0.94	0.95	0.95	0.96	0.95	0.95
0.84	0.88	0.89	0.89	0.91	0.93	0.94	0.94	0.95



Difference between the two graphs

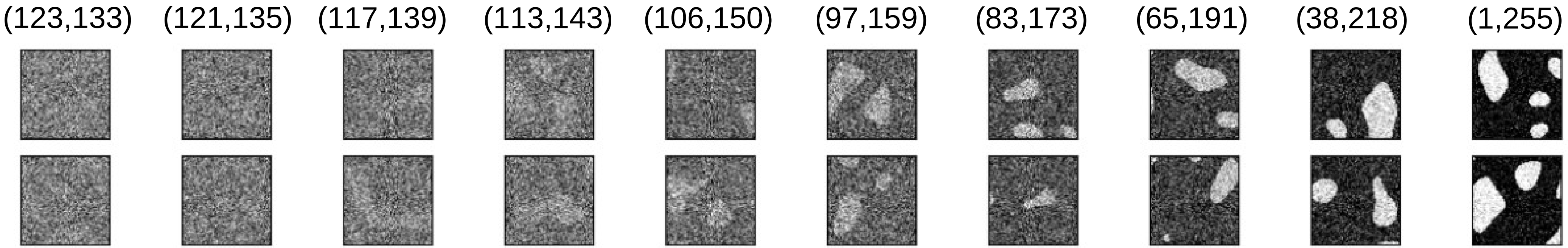
-0.0	0.05	0.51	0.74	0.74	0.73	0.73	0.72	0.71
0.06	-0.0	0.08	0.6	0.75	0.74	0.75	0.74	0.73
0.48	0.06	0.0	0.1	0.67	0.75	0.76	0.75	0.75
0.95	0.73	0.07	-0.0	0.04	0.4	0.75	0.74	0.75
0.95	0.95	0.6	0.05	-0.0	0.06	0.56	0.75	0.75
0.95	0.96	0.95	0.61	0.06	0.0	0.07	0.57	0.75
0.94	0.95	0.95	0.95	0.5	0.05	-0.0	0.1	0.68
0.9	0.92	0.93	0.94	0.94	0.54	0.06	0.0	0.07
0.84	0.88	0.89	0.89	0.91	0.92	0.58	0.08	-0.0

when does BN adaptation work?

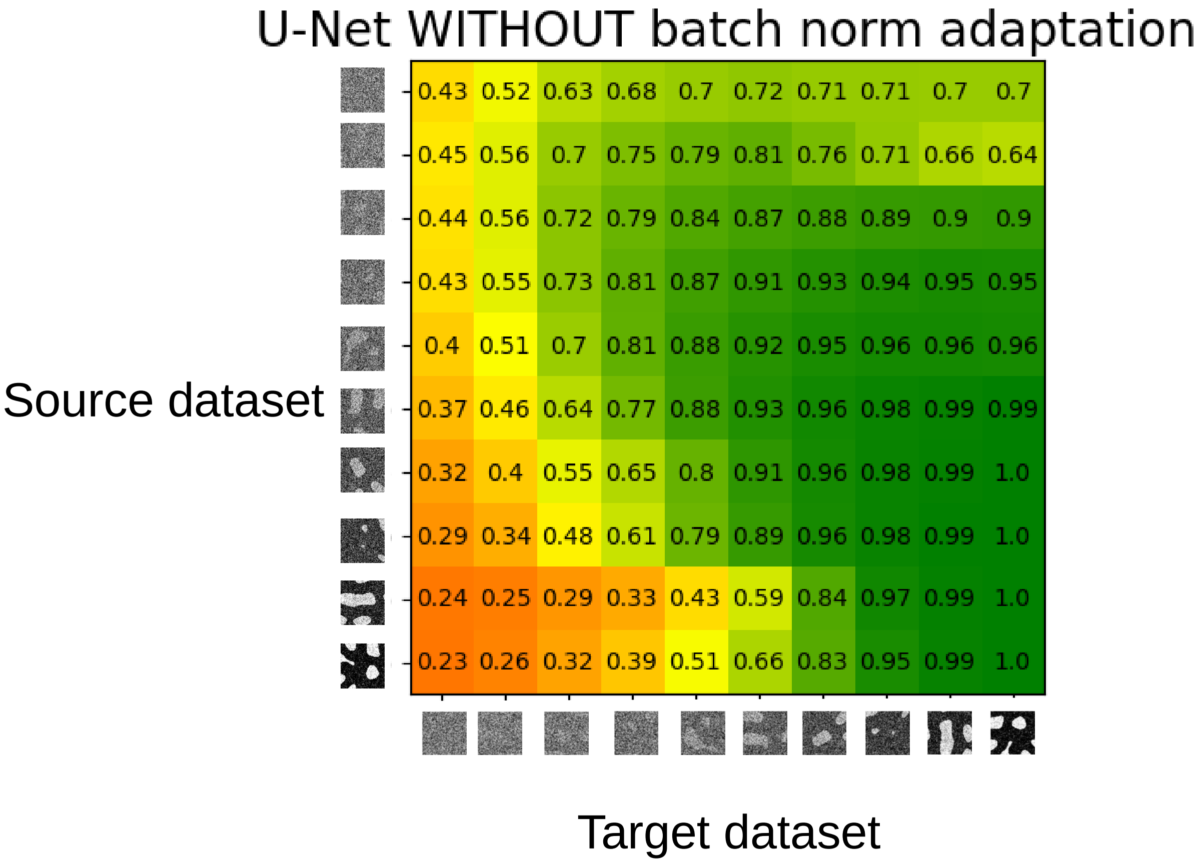
When BN adaptation ..kinda works?

(means of mask and background white noises)

(μ_1,μ_2)



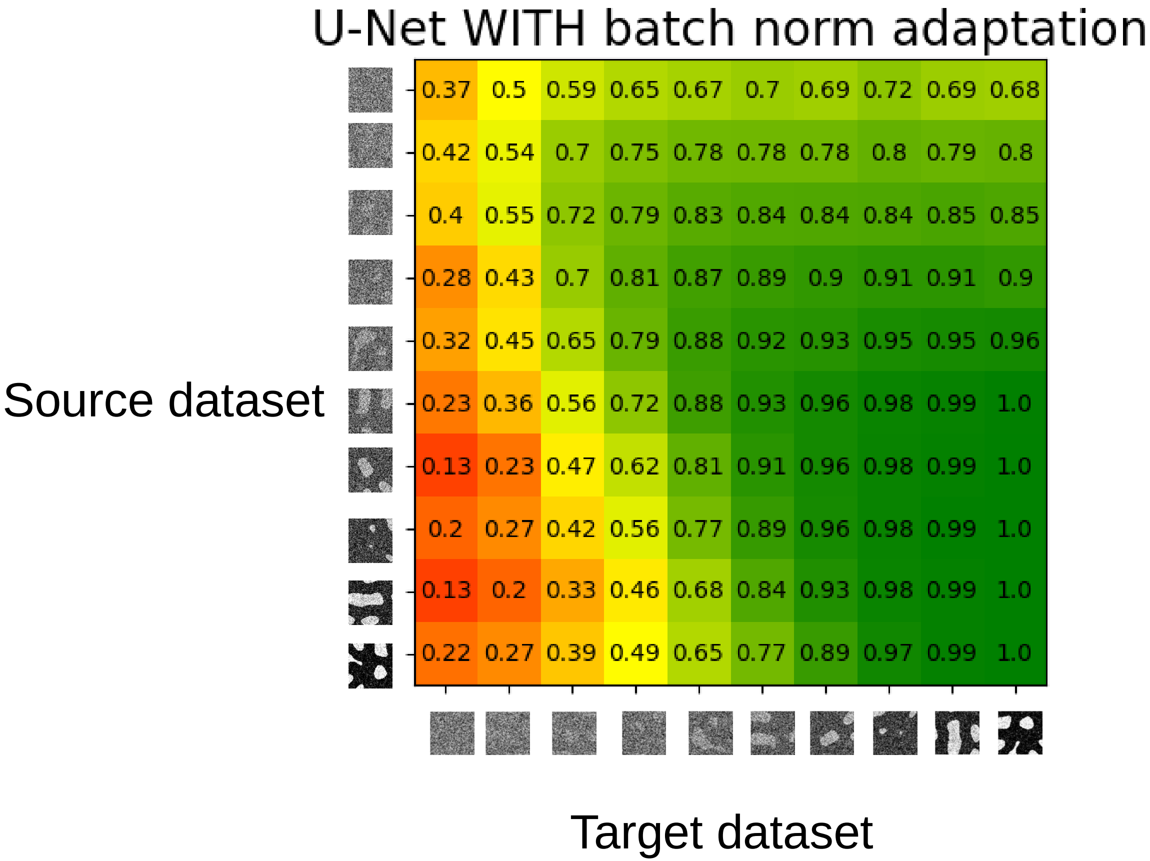
when does BN adaptation work?



U-Net WITHOUT batch norm adaptation

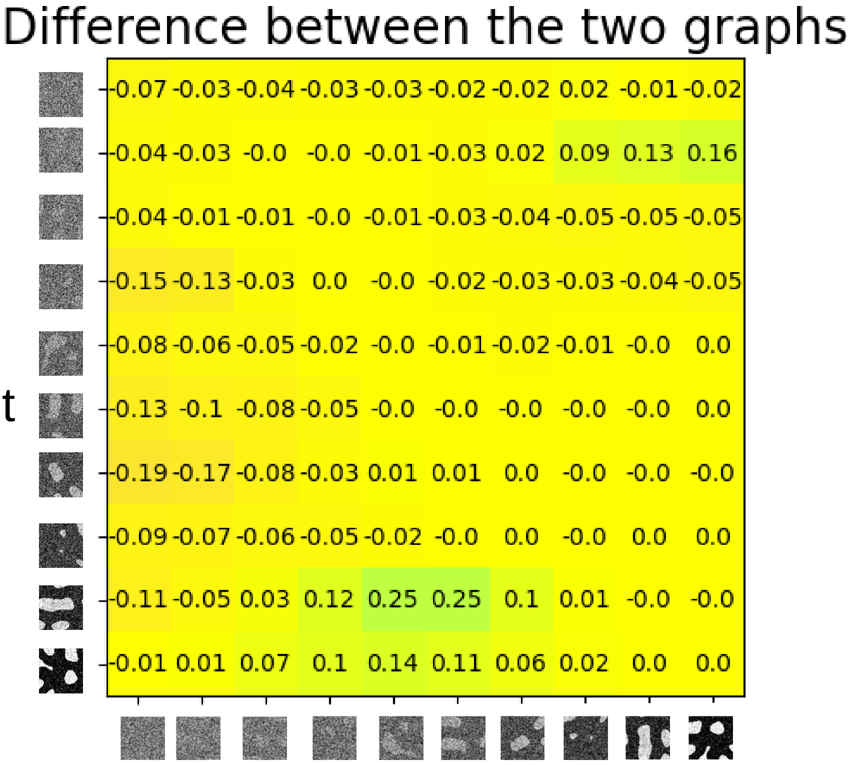
0.43	0.52	0.63	0.68	0.7	0.72	0.71	0.71	0.7	0.7
0.45	0.56	0.7	0.75	0.79	0.81	0.76	0.71	0.66	0.64
0.44	0.56	0.72	0.79	0.84	0.87	0.88	0.89	0.9	0.9
0.43	0.55	0.73	0.81	0.87	0.91	0.93	0.94	0.95	0.95
0.4	0.51	0.7	0.81	0.88	0.92	0.95	0.96	0.96	0.96
0.37	0.46	0.64	0.77	0.88	0.93	0.96	0.98	0.99	0.99
0.32	0.4	0.55	0.65	0.8	0.91	0.96	0.98	0.99	1.0
0.29	0.34	0.48	0.61	0.79	0.89	0.96	0.98	0.99	1.0
0.24	0.25	0.29	0.33	0.43	0.59	0.84	0.97	0.99	1.0
0.23	0.26	0.32	0.39	0.51	0.66	0.83	0.95	0.99	1.0

when does BN adaptation work?



0.37	0.5	0.59	0.65	0.67	0.7	0.69	0.72	0.69	0.68
0.42	0.54	0.7	0.75	0.78	0.78	0.78	0.8	0.79	0.8
0.4	0.55	0.72	0.79	0.83	0.84	0.84	0.84	0.85	0.85
0.28	0.43	0.7	0.81	0.87	0.89	0.9	0.91	0.91	0.9
0.32	0.45	0.65	0.79	0.88	0.92	0.93	0.95	0.95	0.96
0.23	0.36	0.56	0.72	0.88	0.93	0.96	0.98	0.99	1.0
0.13	0.23	0.47	0.62	0.81	0.91	0.96	0.98	0.99	1.0
0.2	0.27	0.42	0.56	0.77	0.89	0.96	0.98	0.99	1.0
0.13	0.2	0.33	0.46	0.68	0.84	0.93	0.98	0.99	1.0
0.22	0.27	0.39	0.49	0.65	0.77	0.89	0.97	0.99	1.0

when does BN adaptation work?



Difference between the two graphs

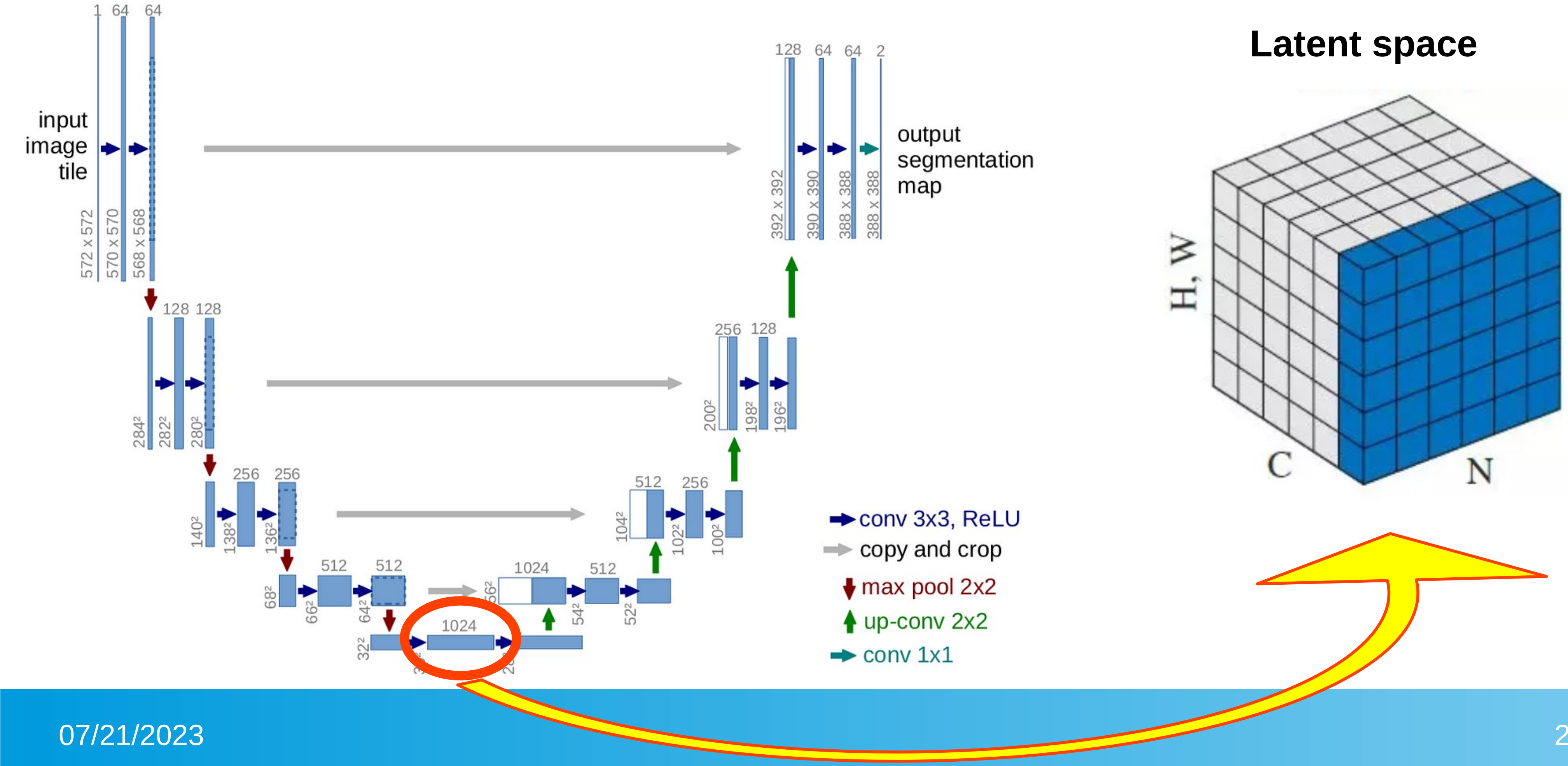
-0.07	-0.03	-0.04	-0.03	-0.03	-0.02	-0.02	0.02	-0.01	-0.02
-0.04	-0.03	-0.0	-0.0	-0.01	-0.03	0.02	0.09	0.13	0.16
-0.04	-0.01	-0.01	-0.0	-0.01	-0.03	-0.04	-0.05	-0.05	-0.05
-0.15	-0.13	-0.03	0.0	-0.0	-0.02	-0.03	-0.03	-0.04	-0.05
-0.08	-0.06	-0.05	-0.02	-0.0	-0.01	-0.02	-0.01	-0.0	0.0
-0.13	-0.1	-0.08	-0.05	-0.0	-0.0	-0.0	-0.0	-0.0	0.0
-0.19	-0.17	-0.08	-0.03	0.01	0.01	0.0	-0.0	-0.0	-0.0
-0.09	-0.07	-0.06	-0.05	-0.02	-0.0	0.0	-0.0	0.0	0.0
-0.11	-0.05	0.03	0.12	0.25	0.25	0.1	0.01	-0.0	-0.0
-0.01	0.01	0.07	0.1	0.14	0.11	0.06	0.02	0.0	0.0

Source dataset

Target dataset

When should we use BN adaptation?

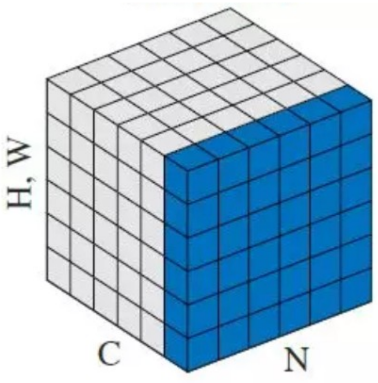
When should we use BN adaptation?



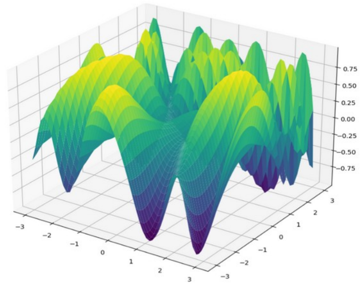
When should we use BN adaptation?

Source dataset → U-Net

Latent spaces

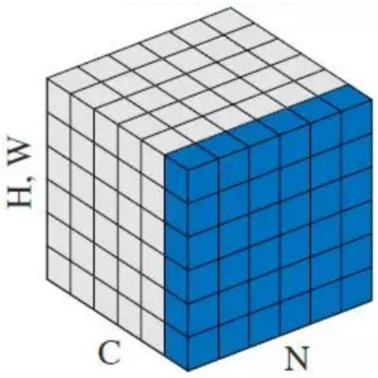


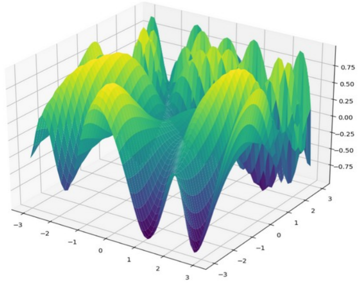
1024-dimensional distributions



distance

Target dataset → U-Net (adapted)

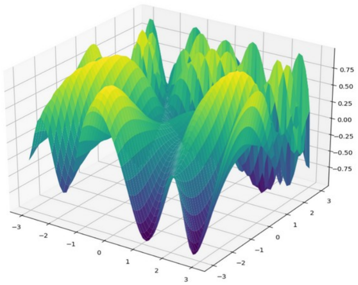




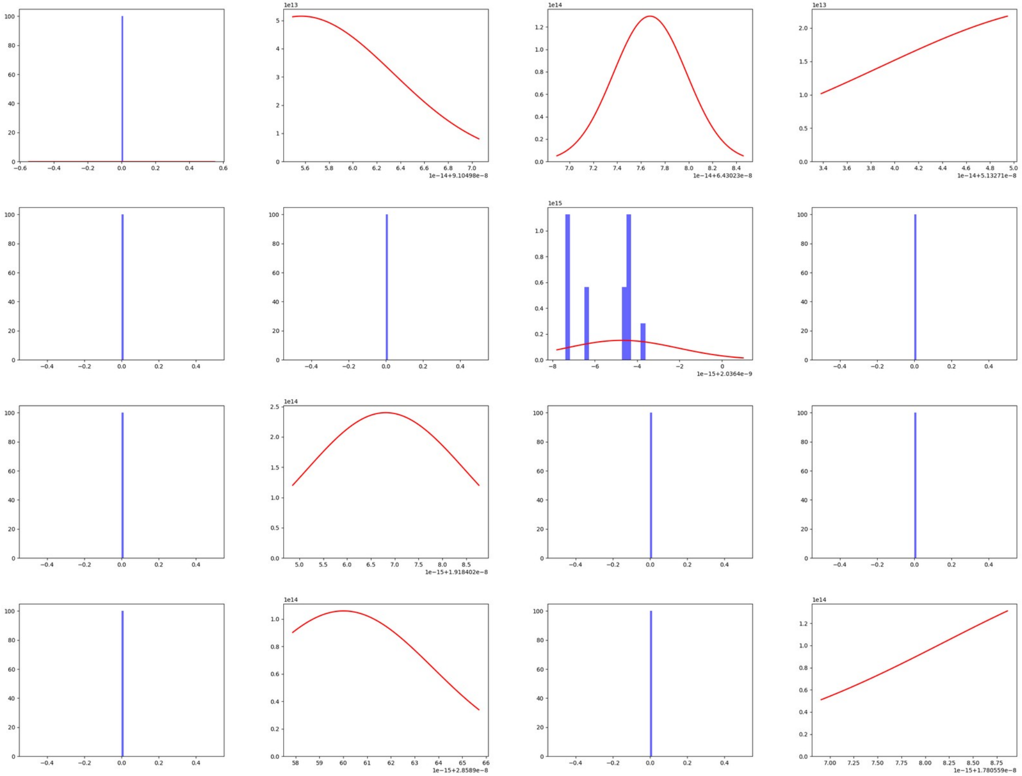
Predict BN adaptation improvement

When should we use BN adaptation?

1024-dimensional distributions



Projection per-dimension

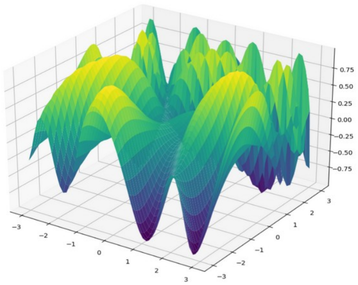


What is going on???

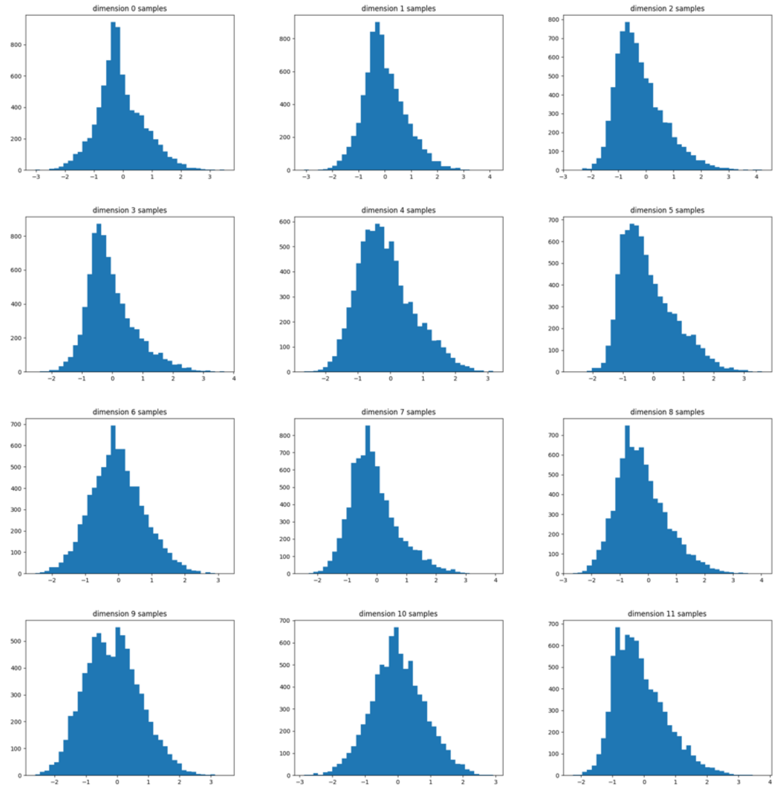
When should we use BN adaptation?

Removed weight decay + Before ReLU

1024-dimensional distributions



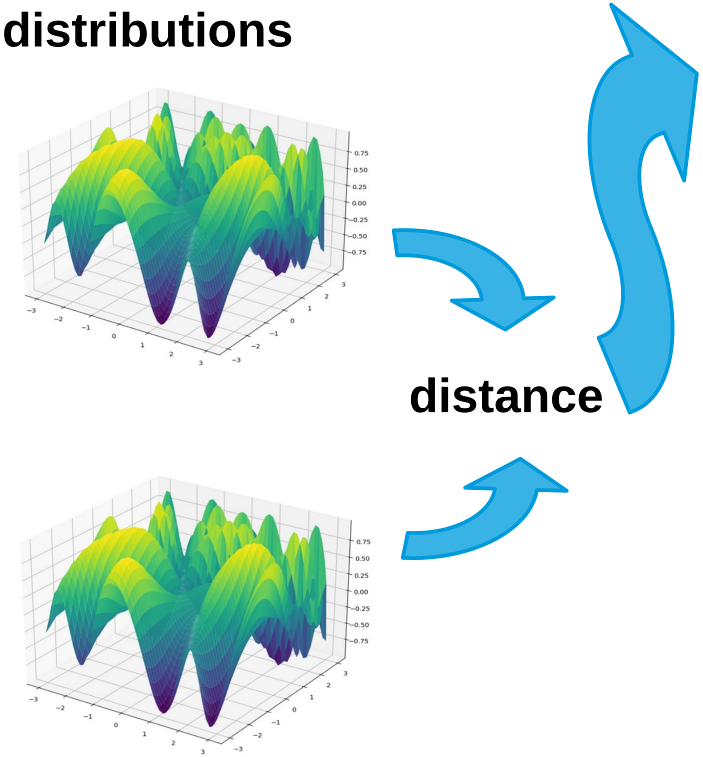
Projection per-dimension



We can work with this!

When should we use BN adaptation?

1024-dimensional distributions



distance

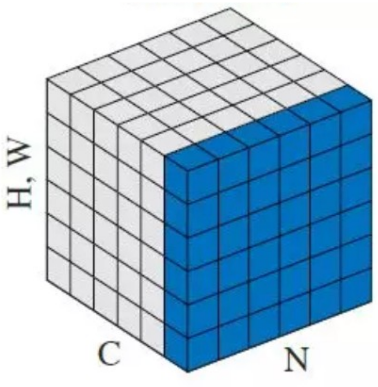
	WASSERSTEIN	SLICED WASSERSTEIN	BHATTACHARYYA
Accounts for	full distribution	projections	projections
Assumptions	Gaussianity	-	-
Sparsity of data	causes less issues	causes issues	causes issues
Computations	Rounding errors	[?]	manageable

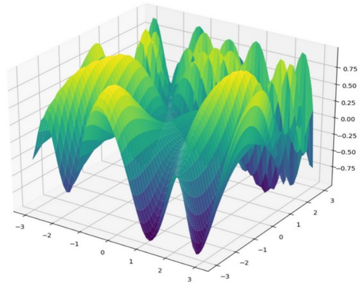
When should we use BN adaptation?

Latent spaces

1024-dimensional distributions

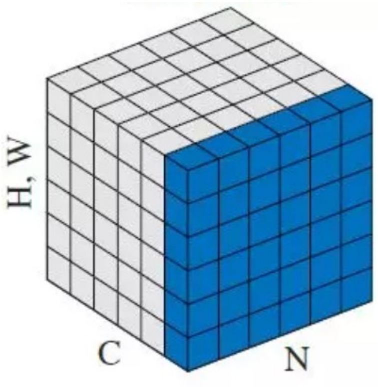
Source dataset → U-Net

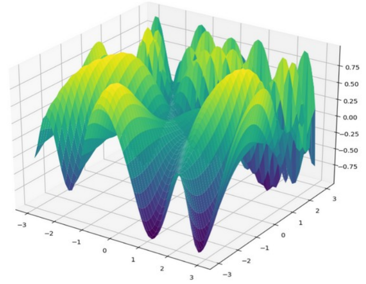




Distance (Bhattacharyya / Wasserstein / …?)

Target dataset → U-Net (adapted)

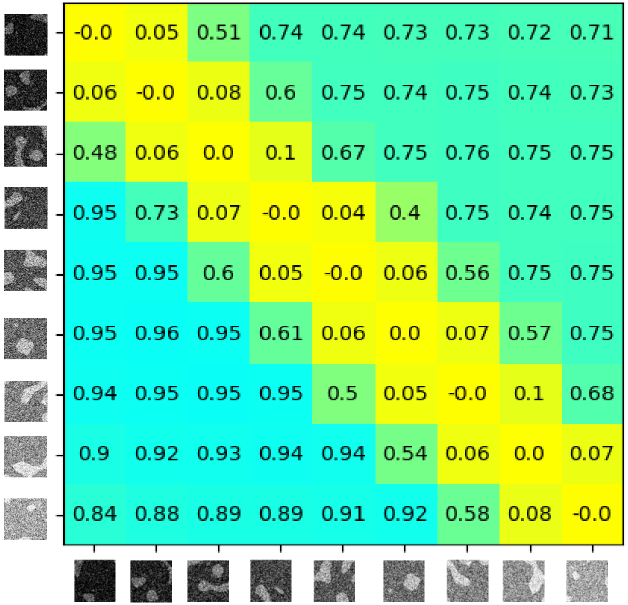




Predict BN adaptation improvement

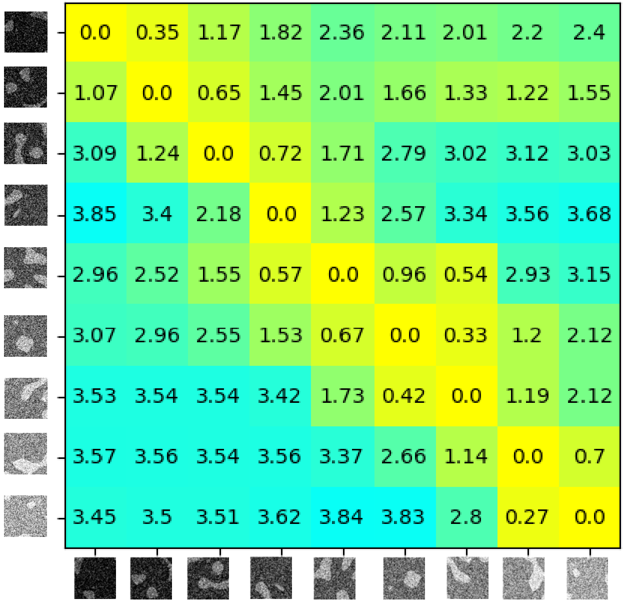
When should we use BN adaptation?

BN adaptation improvement



-0.0	0.05	0.51	0.74	0.74	0.73	0.73	0.72	0.71
0.06	-0.0	0.08	0.6	0.75	0.74	0.75	0.74	0.73
0.48	0.06	0.0	0.1	0.67	0.75	0.76	0.75	0.75
0.95	0.73	0.07	-0.0	0.04	0.4	0.75	0.74	0.75
0.95	0.95	0.6	0.05	-0.0	0.06	0.56	0.75	0.75
0.95	0.96	0.95	0.61	0.06	0.0	0.07	0.57	0.75
0.94	0.95	0.95	0.95	0.5	0.05	-0.0	0.1	0.68
0.9	0.92	0.93	0.94	0.94	0.54	0.06	0.0	0.07
0.84	0.88	0.89	0.89	0.91	0.92	0.58	0.08	-0.0

Bhattacharyya



0.0	0.35	1.17	1.82	2.36	2.11	2.01	2.2	2.4
1.07	0.0	0.65	1.45	2.01	1.66	1.33	1.22	1.55
3.09	1.24	0.0	0.72	1.71	2.79	3.02	3.12	3.03
3.85	3.4	2.18	0.0	1.23	2.57	3.34	3.56	3.68
2.96	2.52	1.55	0.57	0.0	0.96	0.54	2.93	3.15
3.07	2.96	2.55	1.53	0.67	0.0	0.33	1.2	2.12
3.53	3.54	3.54	3.42	1.73	0.42	0.0	1.19	2.12
3.57	3.56	3.54	3.56	3.37	2.66	1.14	0.0	0.7
3.45	3.5	3.51	3.62	3.84	3.83	2.8	0.27	0.0

Close! But not there yet ...

Synthesis and future ideas

Synthesis

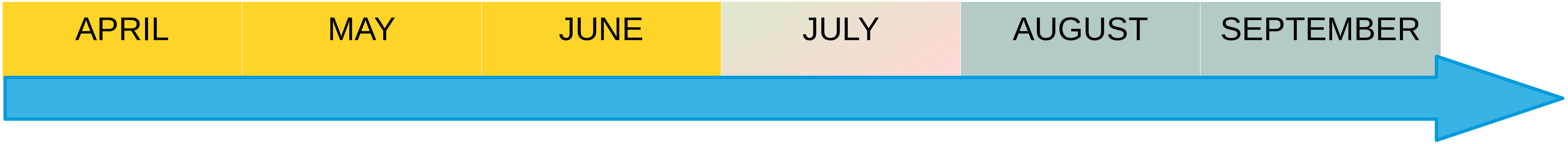
- BN adaptation works well with **brightness shifts**

- it can **correct irregular training**

- We need **more experiments** to find a good predictor for the improvement!

- in some cases it **does not improve** the performance

Future ideas



APRIL MAY JUNE JULY AUGUST SEPTEMBER

What is the plan for my next 2 months?

What I’m working on

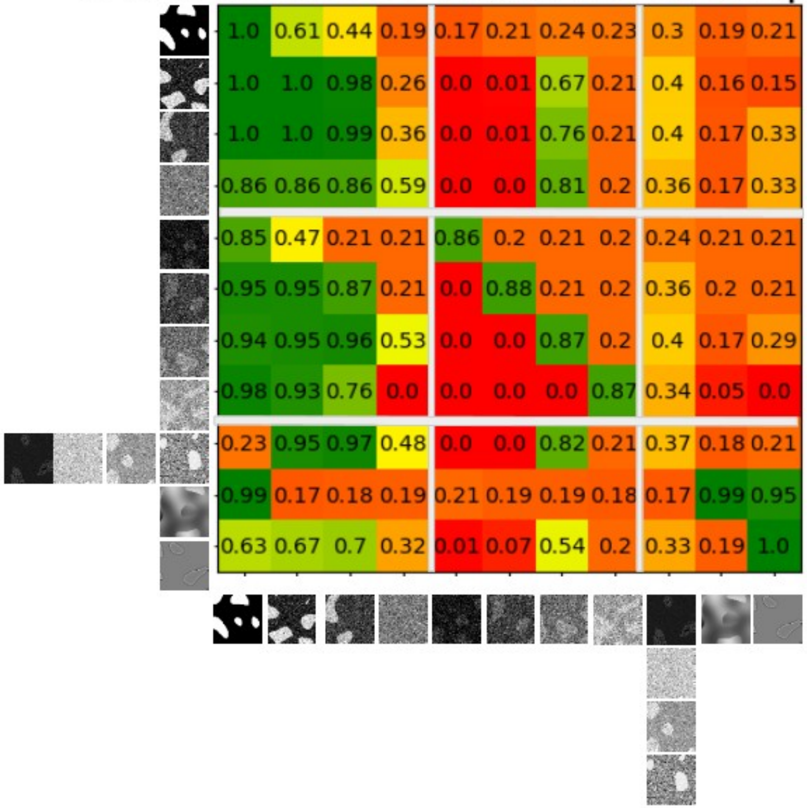
- keep experimenting with different datasets combinations
- test on all latent spaces of UNet
- test different distance measures (e.g. sliced Wasserstein)
- test for different image transformations (cropping, affine transformations, deformations, brighness, contrast, hue, ...)

Thanks for your attention!

- extra slides -

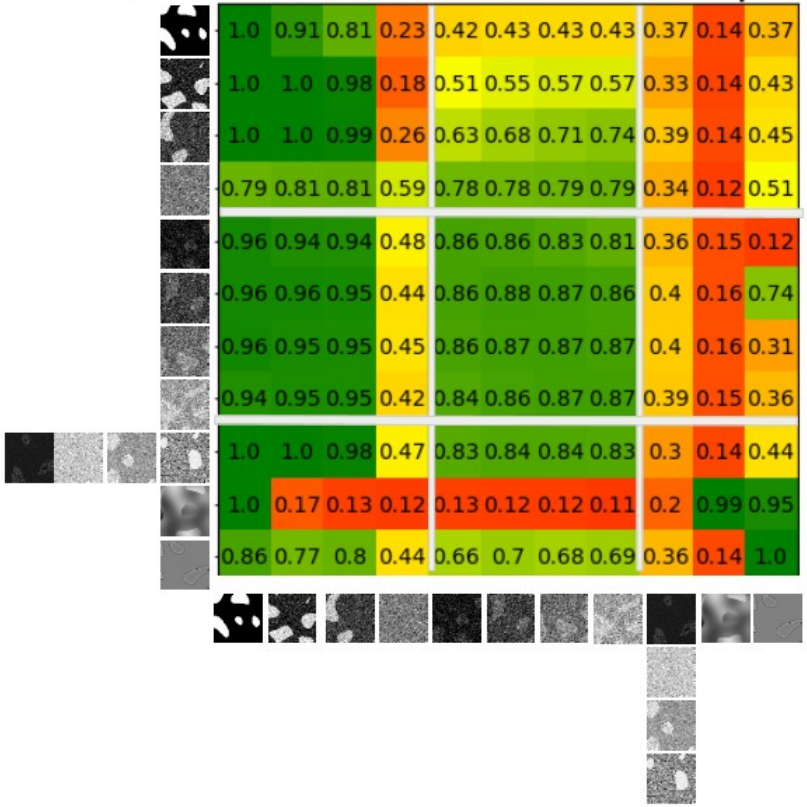
EXTRA

U-Net WITHOUT batch norm adaptation



1.0	0.61	0.44	0.19	0.17	0.21	0.24	0.23	0.3	0.19	0.21
1.0	1.0	0.98	0.26	0.0	0.01	0.67	0.21	0.4	0.16	0.15
1.0	1.0	0.99	0.36	0.0	0.01	0.76	0.21	0.4	0.17	0.33
0.86	0.86	0.86	0.59	0.0	0.0	0.81	0.2	0.36	0.17	0.33
0.85	0.47	0.21	0.21	0.86	0.2	0.21	0.2	0.24	0.21	0.21
0.95	0.95	0.87	0.21	0.0	0.88	0.21	0.2	0.36	0.2	0.21
0.94	0.95	0.96	0.53	0.0	0.0	0.87	0.2	0.4	0.17	0.29
0.98	0.93	0.76	0.0	0.0	0.0	0.0	0.87	0.34	0.05	0.0
0.23	0.95	0.97	0.48	0.0	0.0	0.82	0.21	0.37	0.18	0.21
0.99	0.17	0.18	0.19	0.21	0.19	0.19	0.18	0.17	0.99	0.95
0.63	0.67	0.7	0.32	0.01	0.07	0.54	0.2	0.33	0.19	1.0

U-Net WITH batch norm adaptation



1.0	0.91	0.81	0.23	0.42	0.43	0.43	0.43	0.37	0.14	0.37
1.0	1.0	0.98	0.18	0.51	0.55	0.57	0.57	0.33	0.14	0.43
1.0	1.0	0.99	0.26	0.63	0.68	0.71	0.74	0.39	0.14	0.45
0.79	0.81	0.81	0.59	0.78	0.78	0.79	0.79	0.34	0.12	0.51
0.96	0.94	0.94	0.48	0.86	0.86	0.83	0.81	0.36	0.15	0.12
0.96	0.96	0.95	0.44	0.86	0.88	0.87	0.86	0.4	0.16	0.74
0.96	0.95	0.95	0.45	0.86	0.87	0.87	0.87	0.4	0.16	0.31
0.94	0.95	0.95	0.42	0.84	0.86	0.87	0.87	0.39	0.15	0.36
1.0	1.0	0.98	0.47	0.83	0.84	0.84	0.83	0.3	0.14	0.44
1.0	0.17	0.13	0.12	0.13	0.12	0.12	0.11	0.2	0.99	0.95
0.86	0.77	0.8	0.44	0.66	0.7	0.68	0.69	0.36	0.14	1.0