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

Encadrement: Étienne Baudrier, Sylvain Faisan, Alexandre Stenger

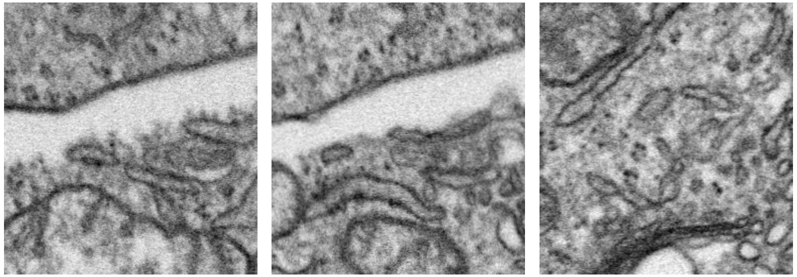
Esplorando la Performance dell’Adattazione al Dominio nell’Apprendimento Profondo: dallo Spostamento di Covarianza alla Wasserstein e oltre

Supervisione: Étienne Baudrier, Sylvain Faisan, Alexandre Stenger

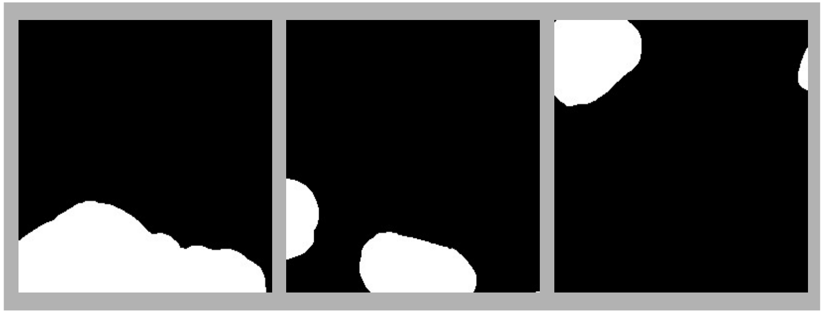
Introduzione

Introduzione: Spostamento di Covariata

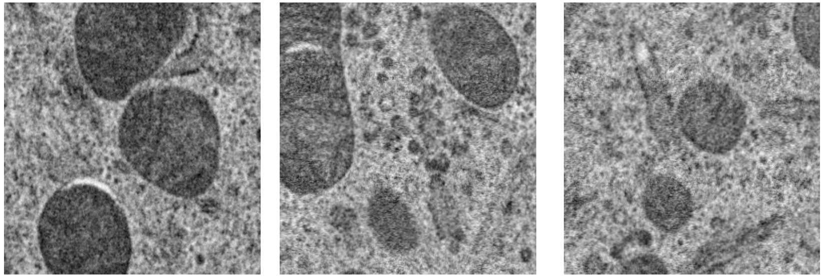
Dati sorgente *(fissione chimica)*



U-Net

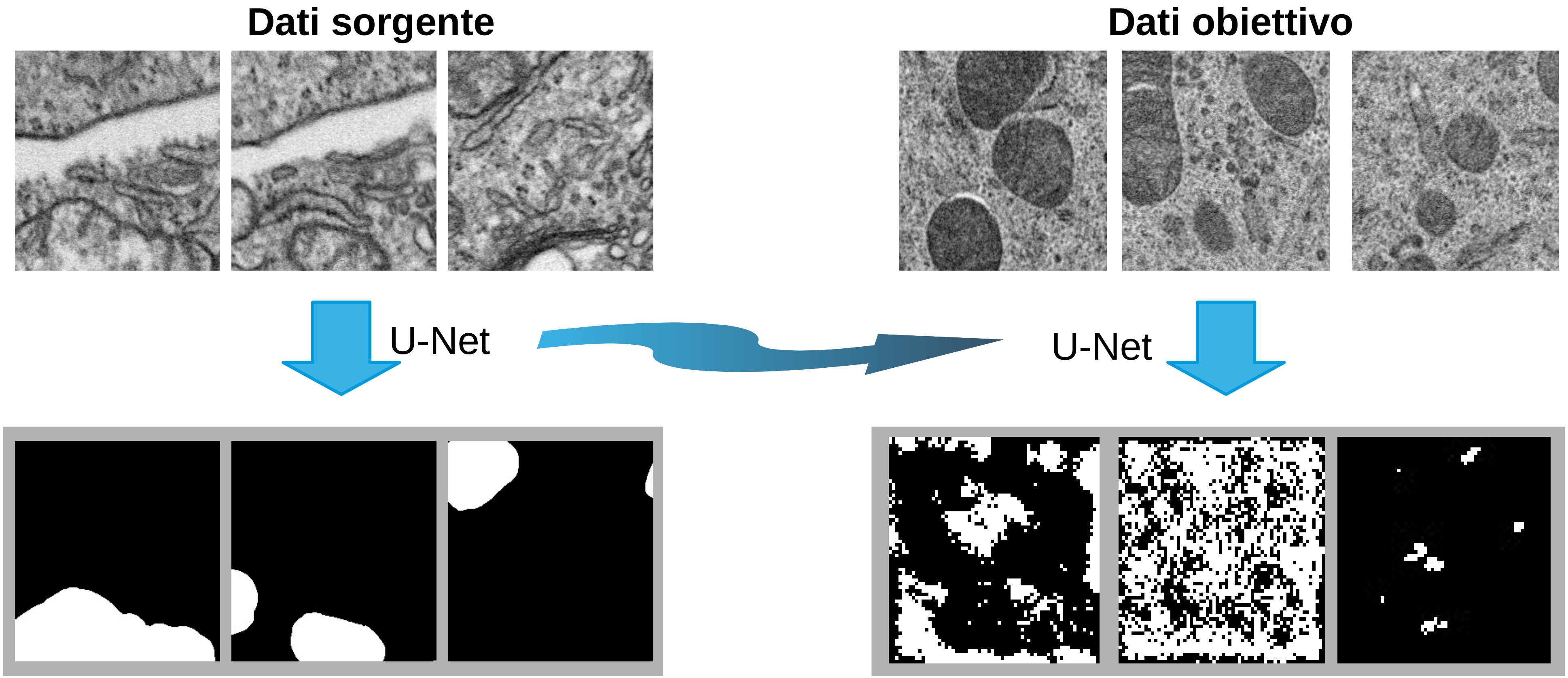


Dati obiettivo *(crio-fissione)*

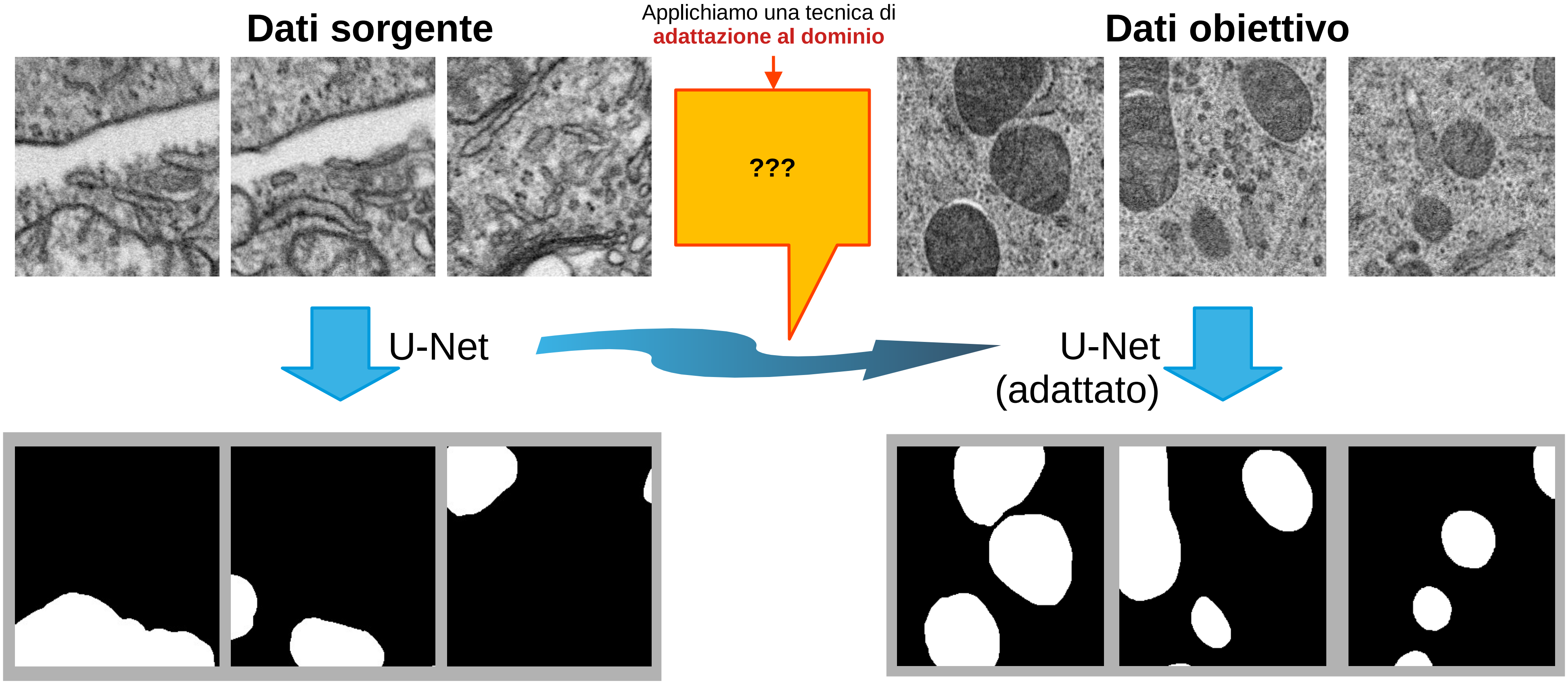


Spostamento di covariata

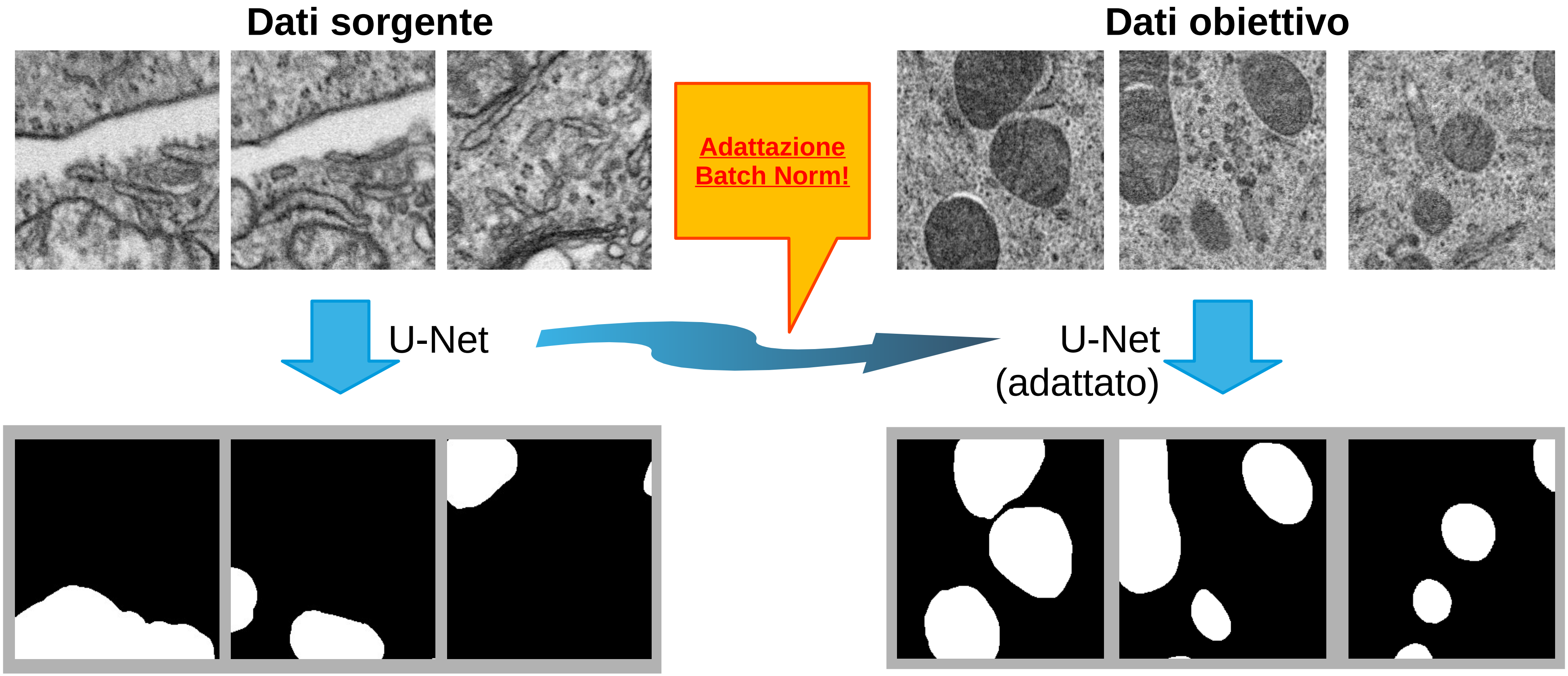
Introduzione: Spostamento di Covariata



Introduzione: Spostamento di Covariata

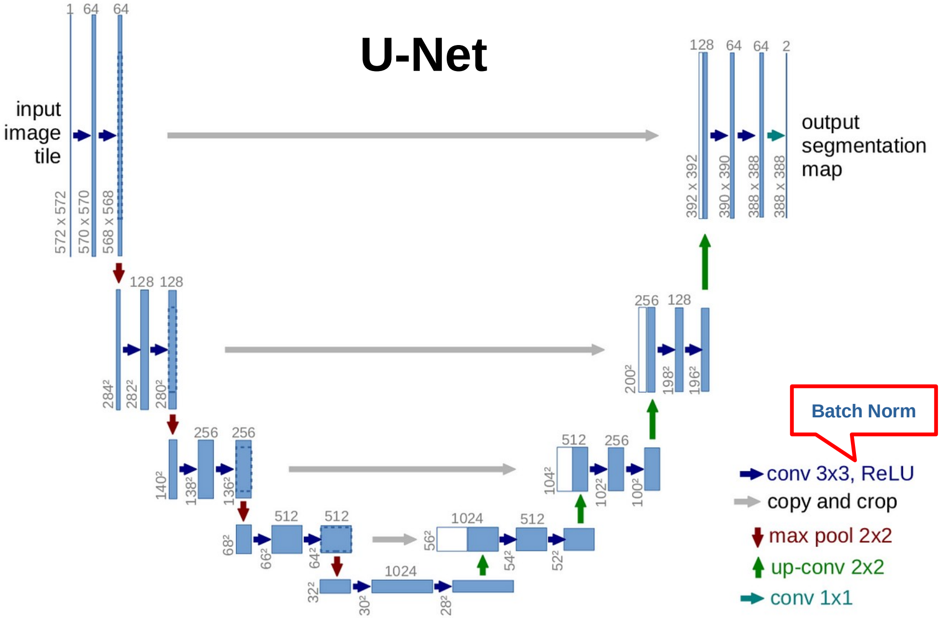


Introduzione: Spostamento di Covariata



Cos’è la Batch Norm?

U-Net



Livello Batch Norm

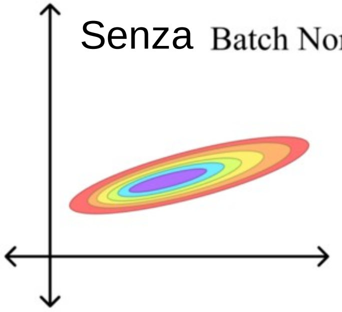
$$\hat{\mu} = \mathbb{E}[x]$$

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

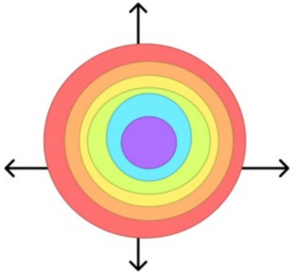
$$y = \frac{x - \hat{\mu}}{\hat{\sigma}} * \gamma + \beta$$

γ, β imparati durante l’allenamento

Senza Batch Norm



Con Batch Norm

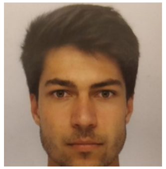


Introduzione: Adattazione Batch Norm

Cos'è l'<u>adattazione</u> Batch Norm?

- prendiamo i **dati obiettivo** e li passiamo attraverso la rete, bloccando tutti i parametri ma aggiornando $\hat{\mu}$ e $\hat{\sigma}$

Introduzione: il lavoro della mia équipe

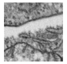
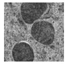
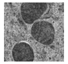
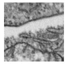





https://publis.icube.unistra.fr/docs/17711/ISBI_paper_559.pdf

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

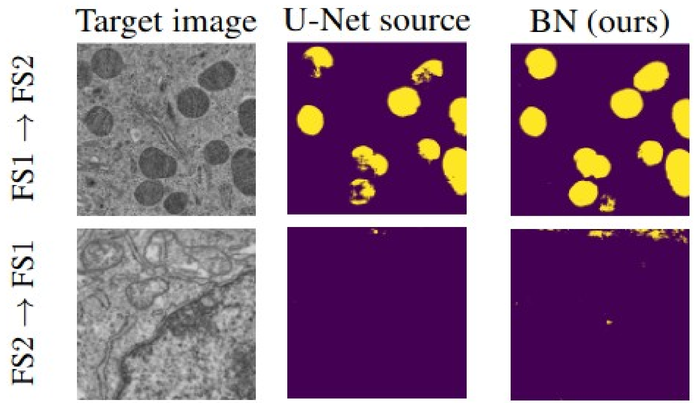
Alexandre Stenger *† *Luc Vedrenne* *† *Patrick Schultz* ‡ *Sylvain Faisan* †
Étienne Baudrier † *Benoît Naegel* †

 →   → 

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

Introduzione: il lavoro della mia équipe

Comportamento asimmetrico della BN
(adattazione Batch Norm)



Pro:
- Facile da implementare
- Estremamente veloce

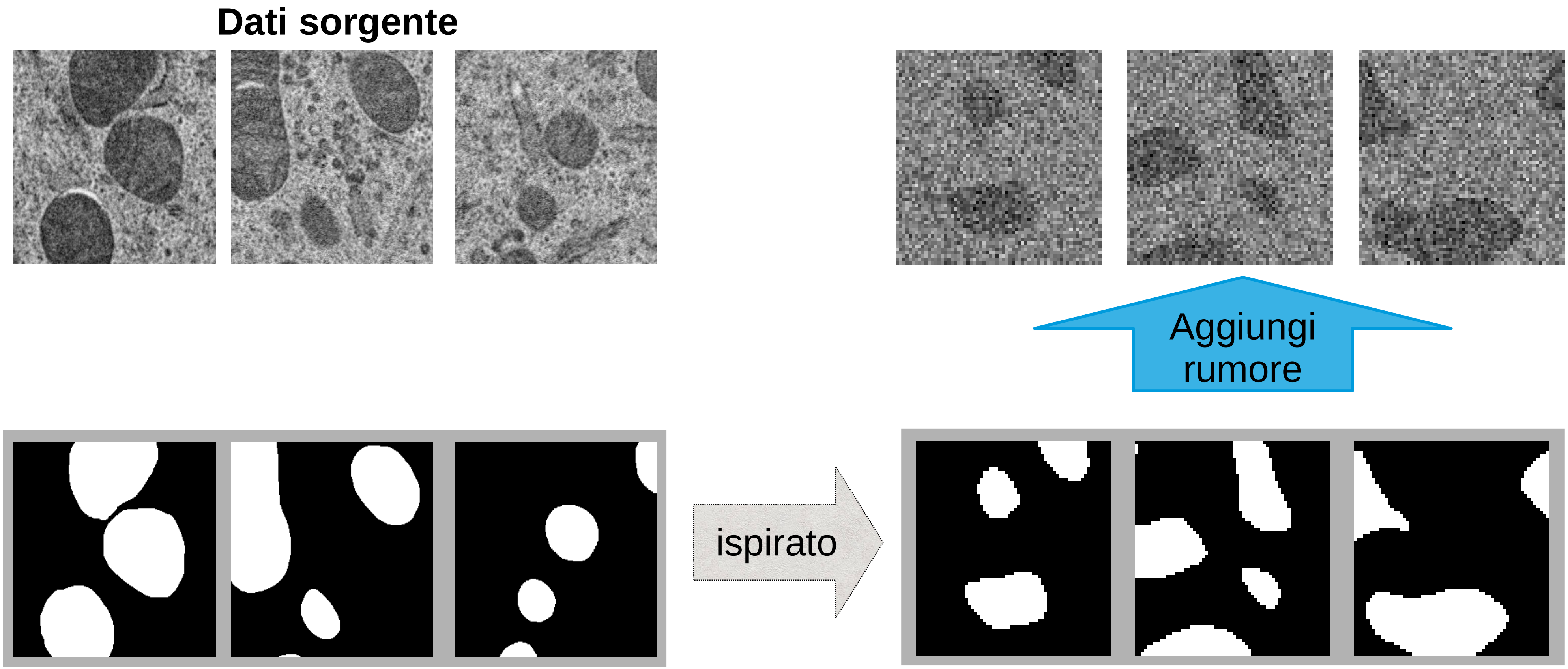
Contro:
- Non sappiamo quando funziona

L’obiettivo del mio stage

Perché (e quando) funziona l’adattazione Batch Norm?

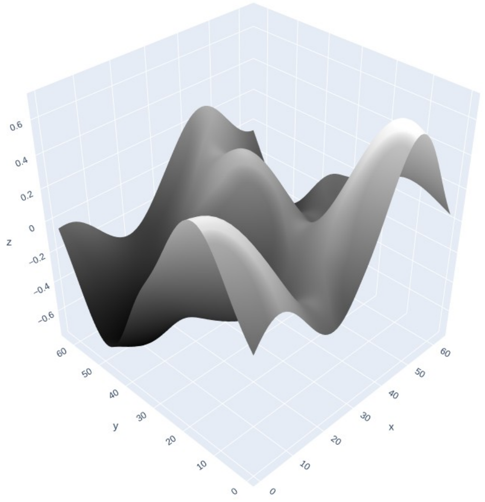
Generazione dei dati

Generazione dei dati

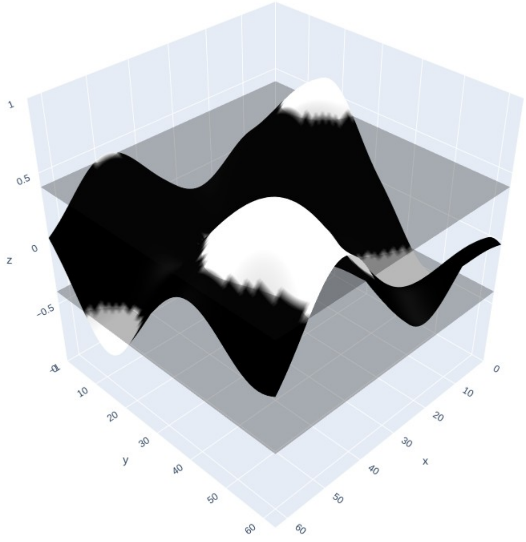


Generazione dei dati

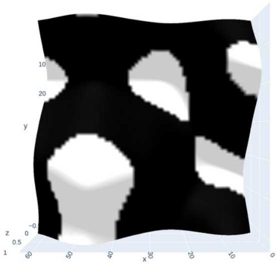
1) Generi un **Rumore di Perlin**



2) definisci una soglia

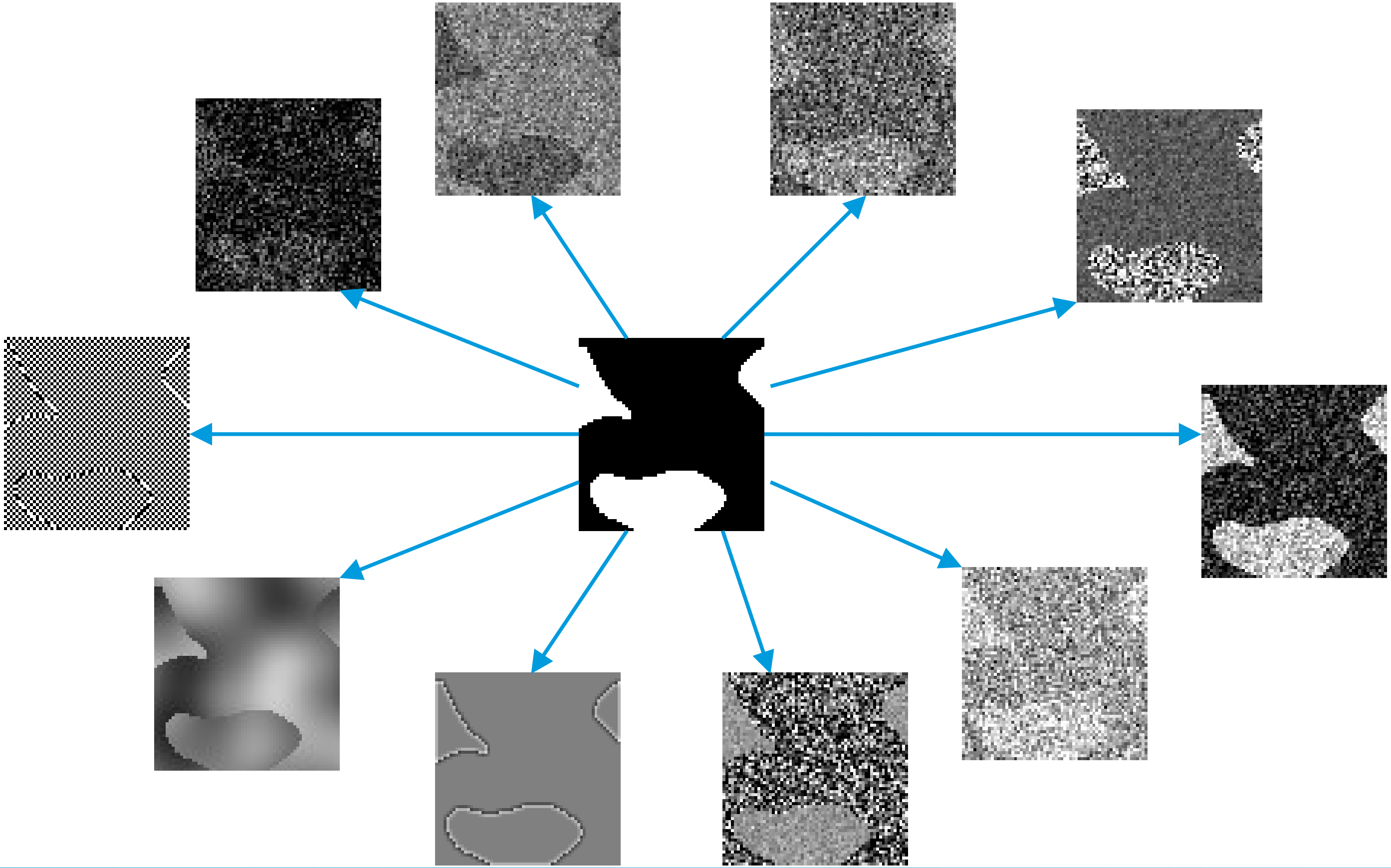


3) “guardi da sopra”



~40s per 10.000 immagini

Generazione dei dati



When does BN adaptation work?

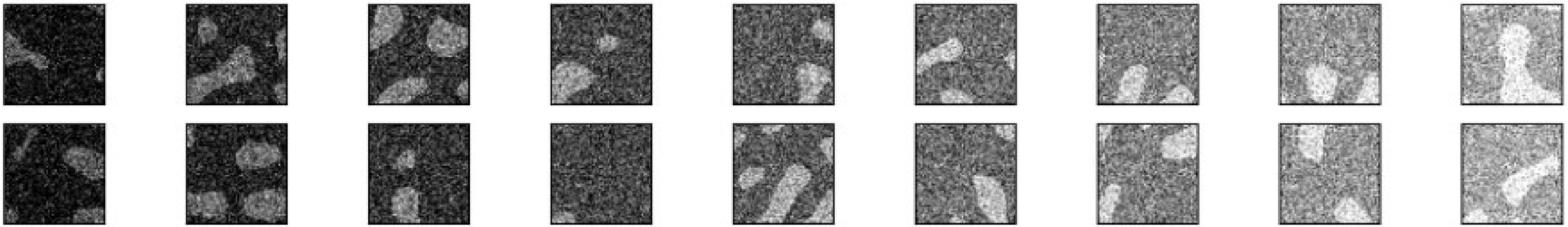
when does BN adaptation work?

When BN adaptation works

(means of mask and background white noises)

(μ_1,μ_2)

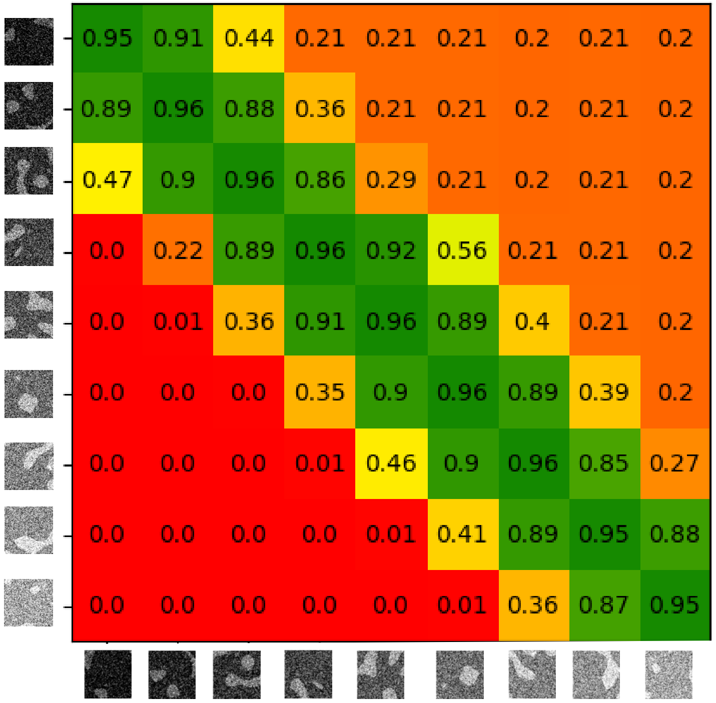
(10,90) (30,110) (50,130) (70,150) (90,170) (110,190) (130,210) (150,230) (170,250)



when does BN adaptation work?

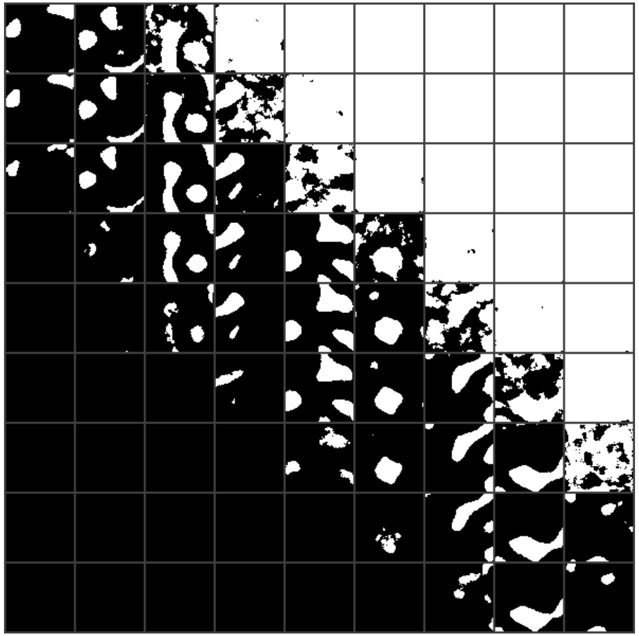
U-Net WITHOUT bn adaptation

Source dataset



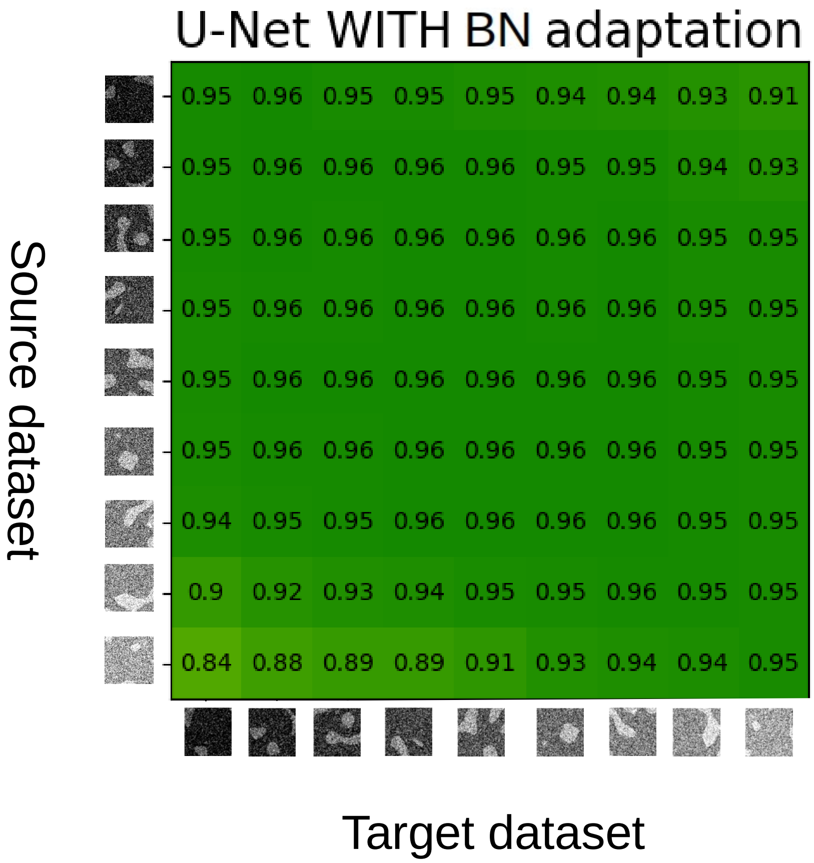
Target dataset

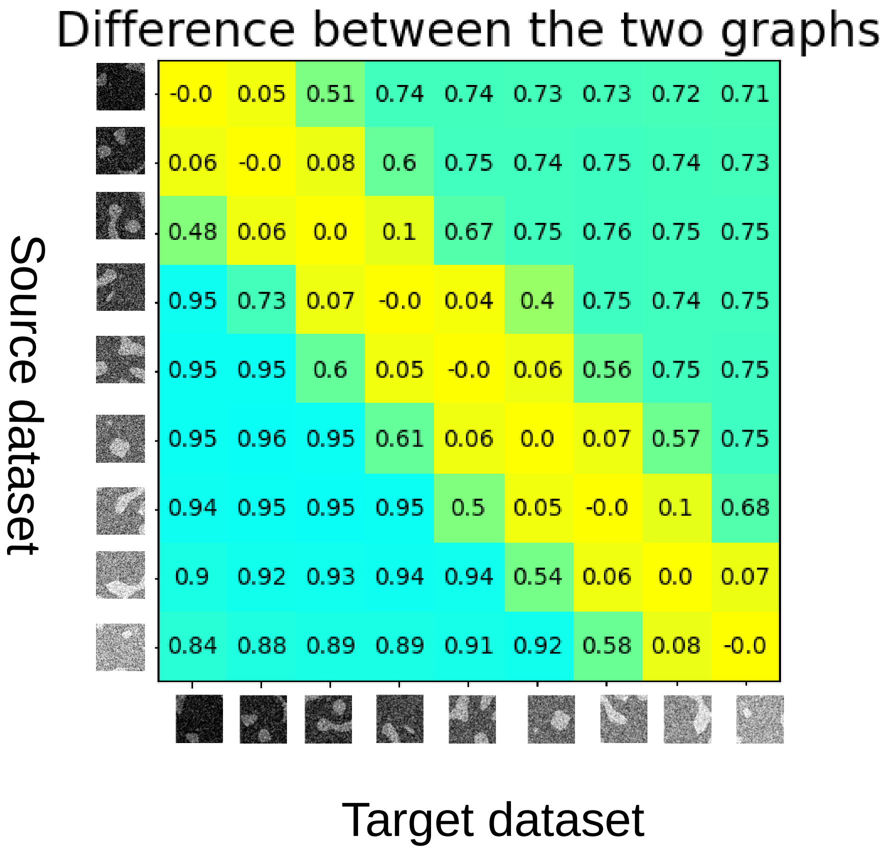
Mask predictions



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

when does BN adaptation work?



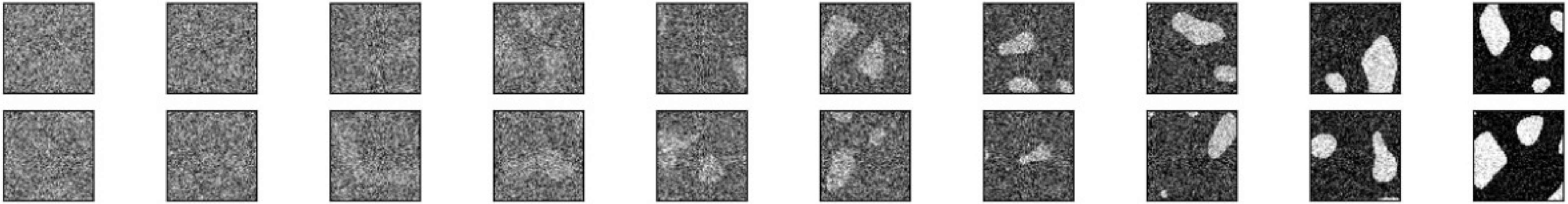


When BN adaptation ..kinda works?

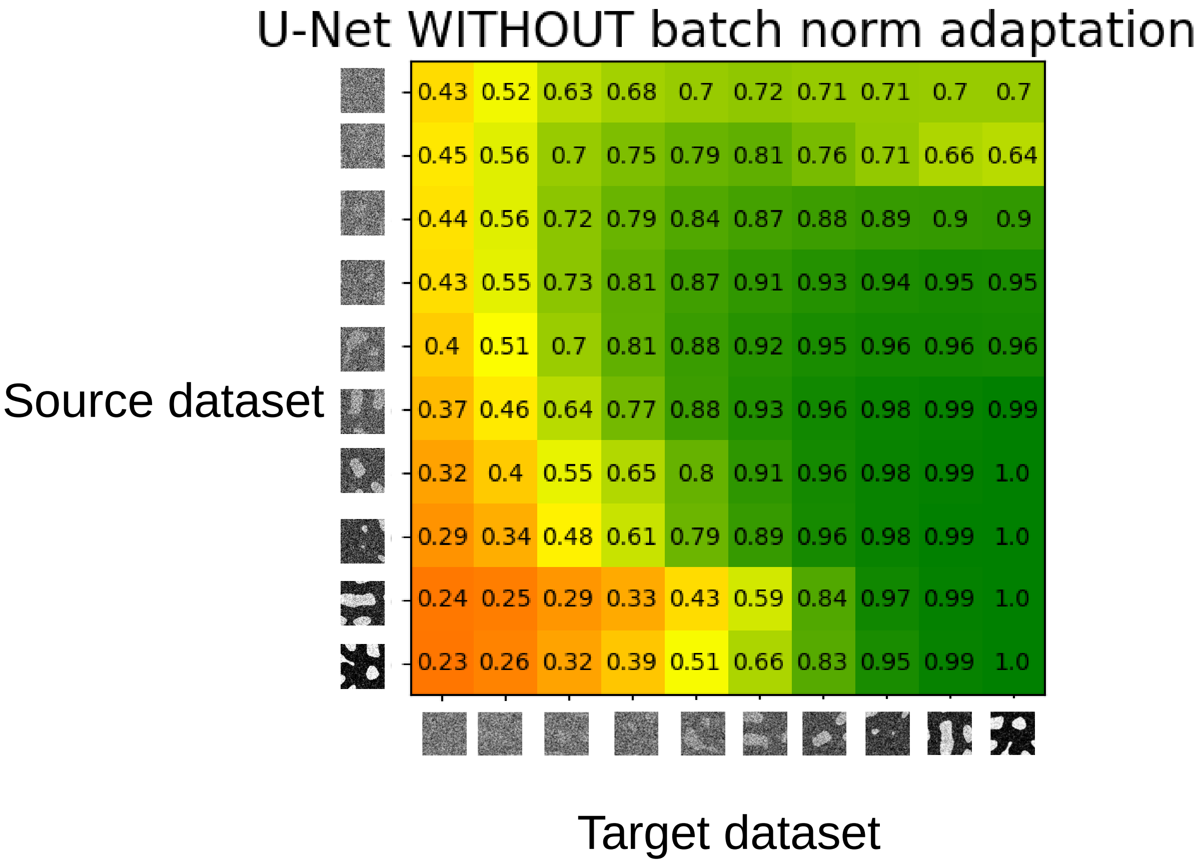
(means of mask and background white noises)

(μ_1,μ_2)

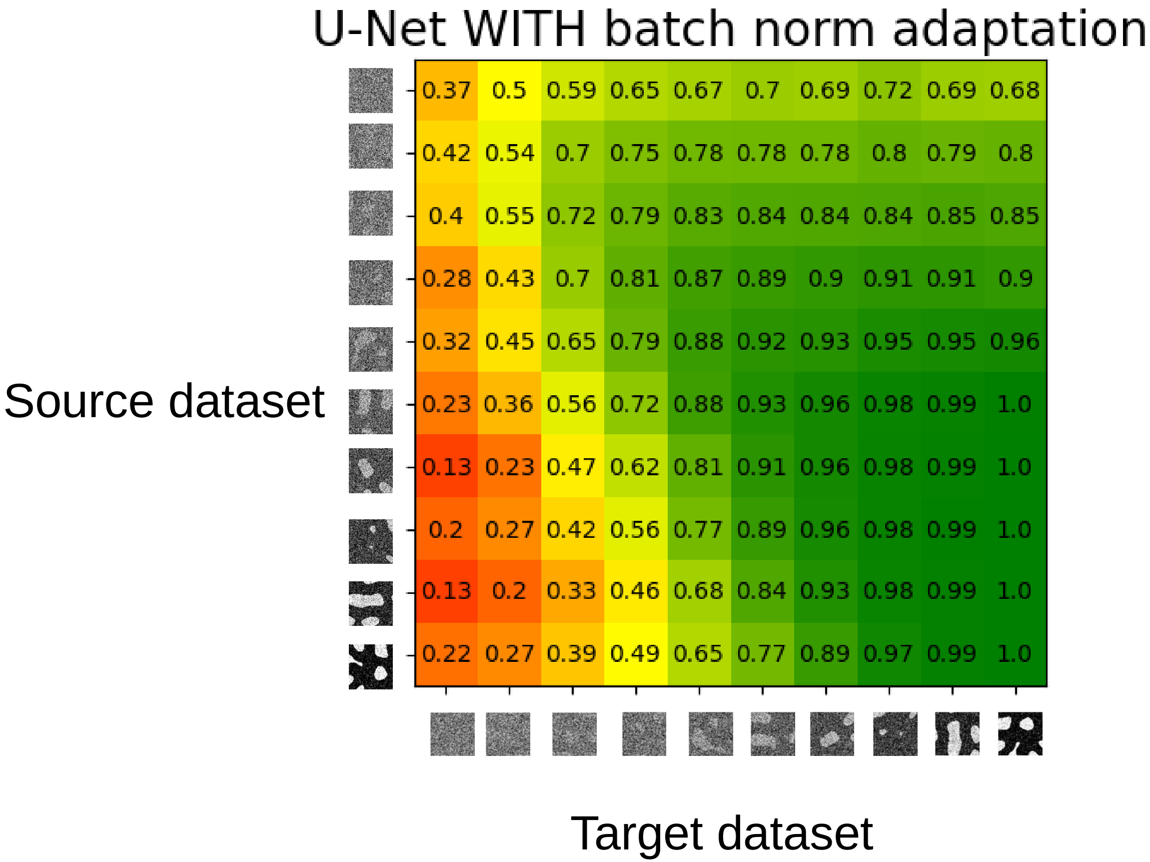
(123,133) (121,135) (117,139) (113,143) (106,150) (97,159) (83,173) (65,191) (38,218) (1,255)



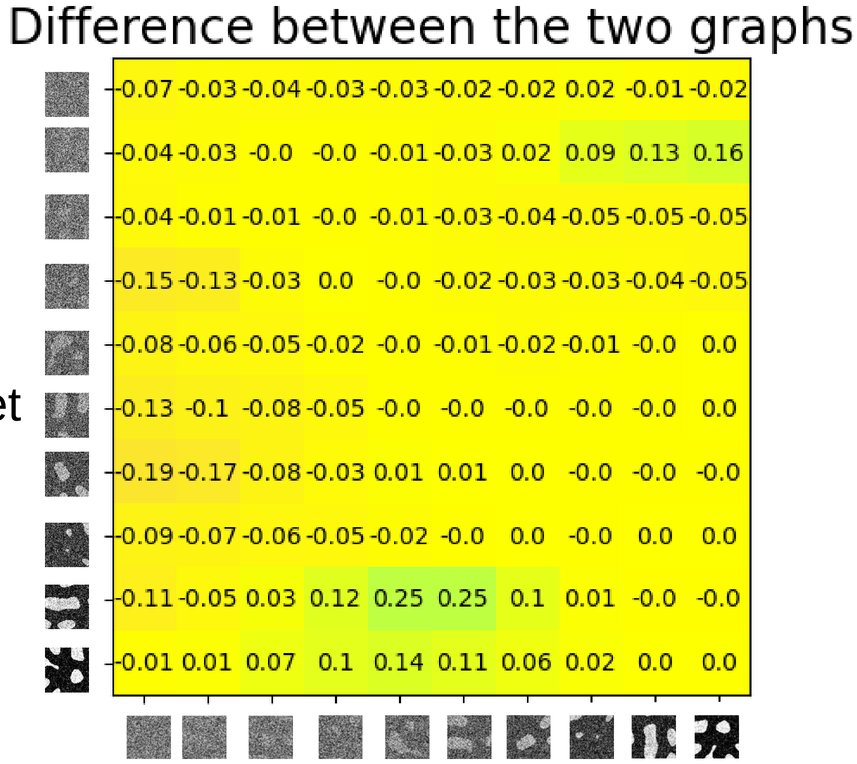
when does BN adaptation work?



when does BN adaptation work?



when does BN adaptation work?

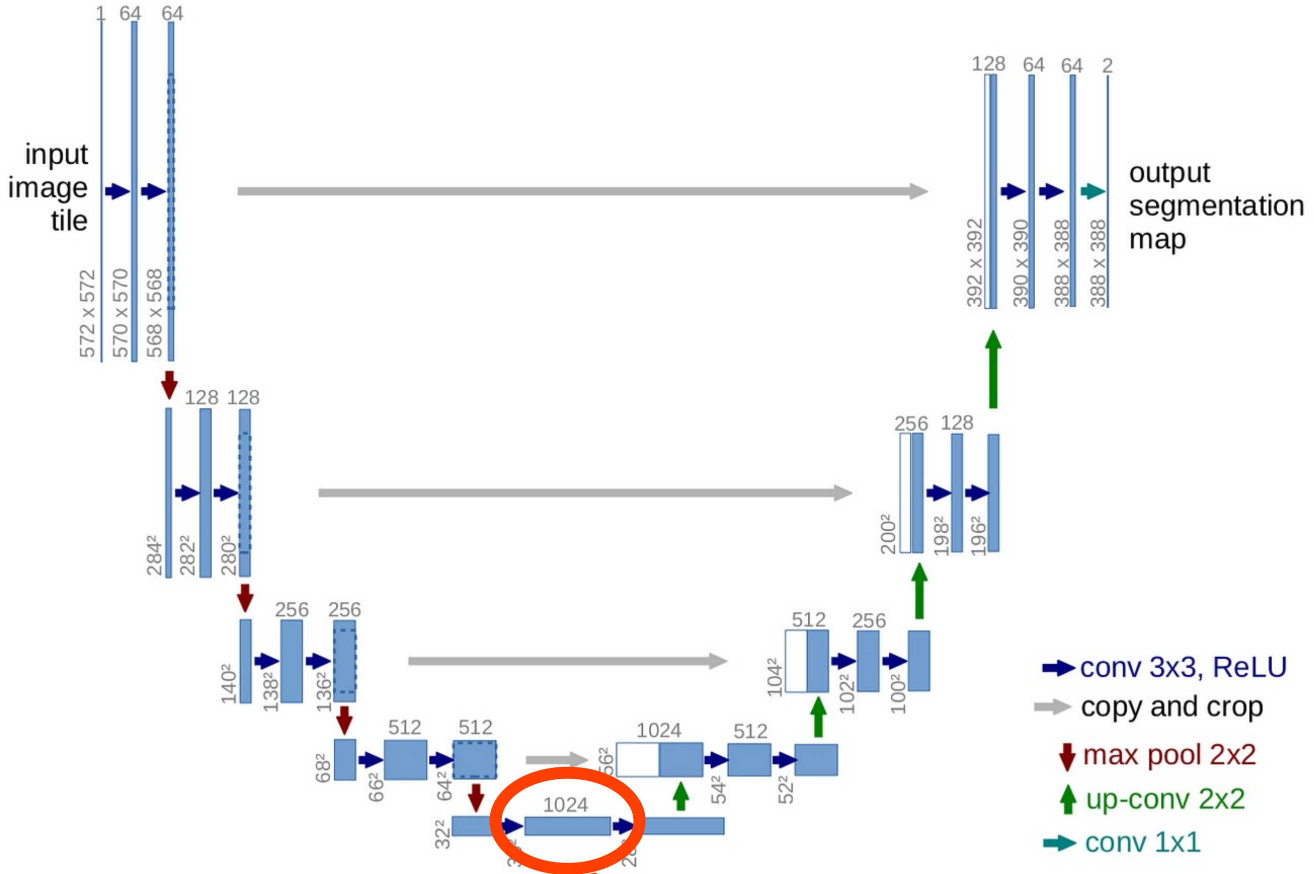


Source dataset

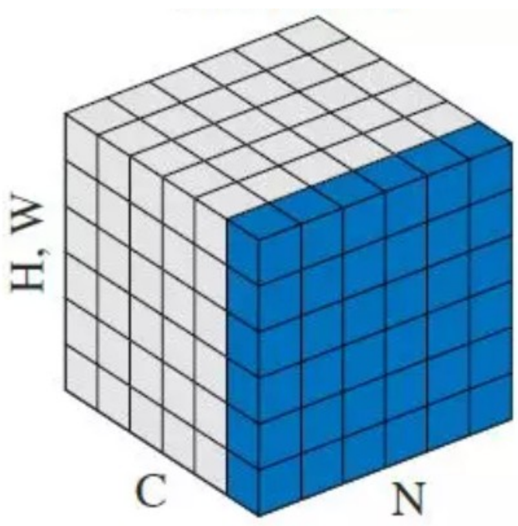
Target dataset

When should we use BN adaptation?

When should we use BN adaptation?



Latent space

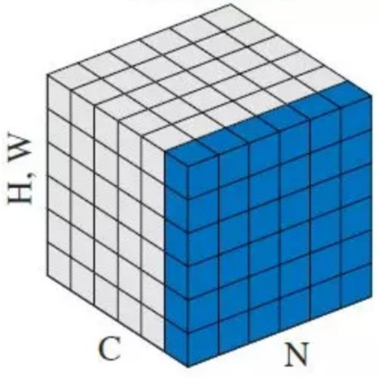


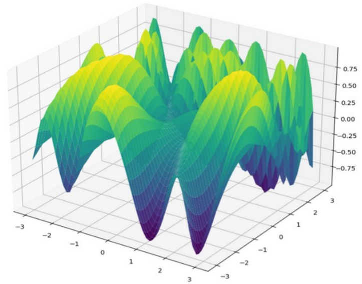
When should we use BN adaptation?

Latent spaces

1024-dimensional distributions

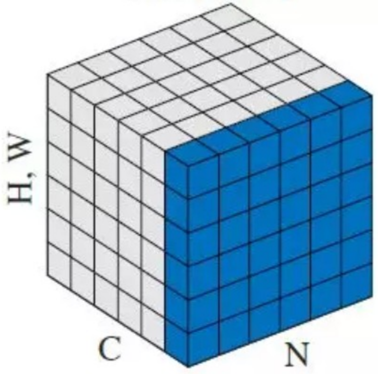
Source dataset → U-Net

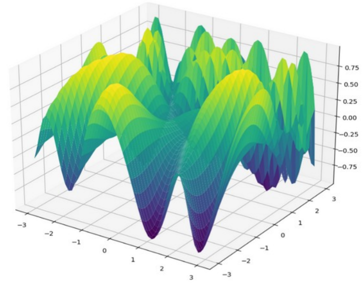




distance

Target dataset → U-Net (adapted)

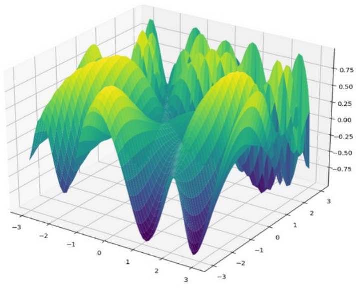




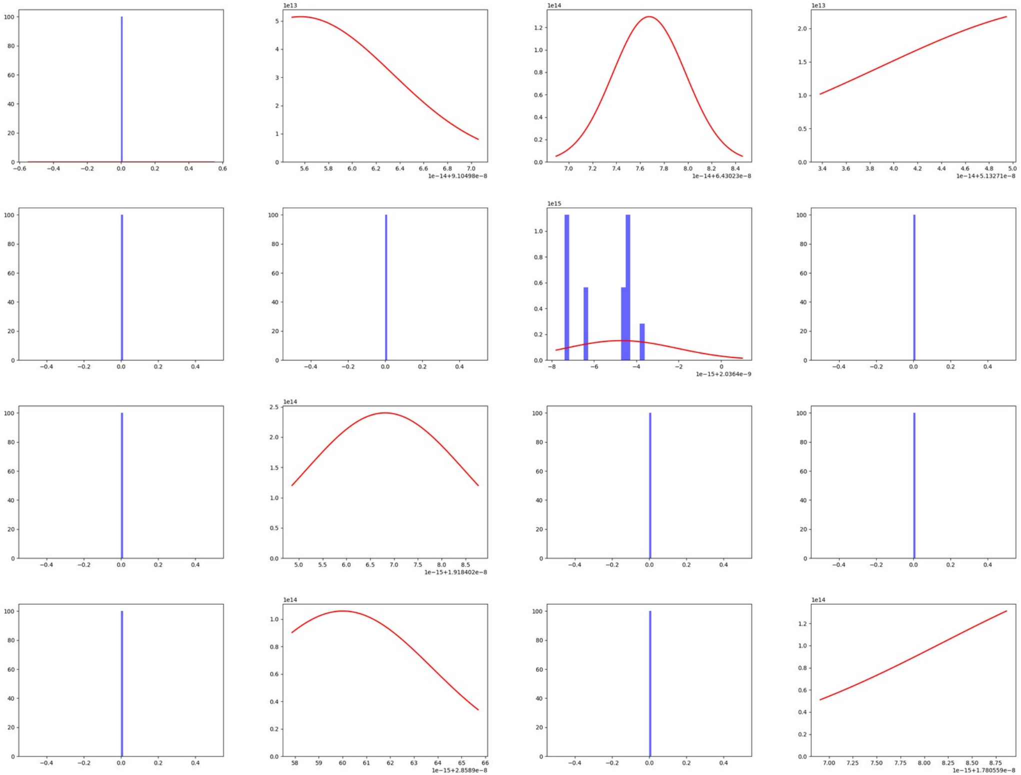
Predict BN adaptation improvement

When should we use BN adaptation?

1024-dimensional distributions



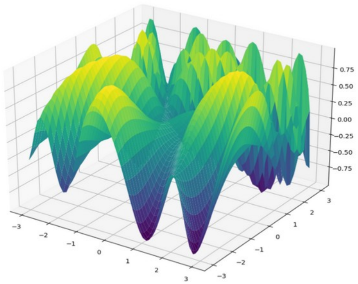
Projection per-dimension



What is going on???

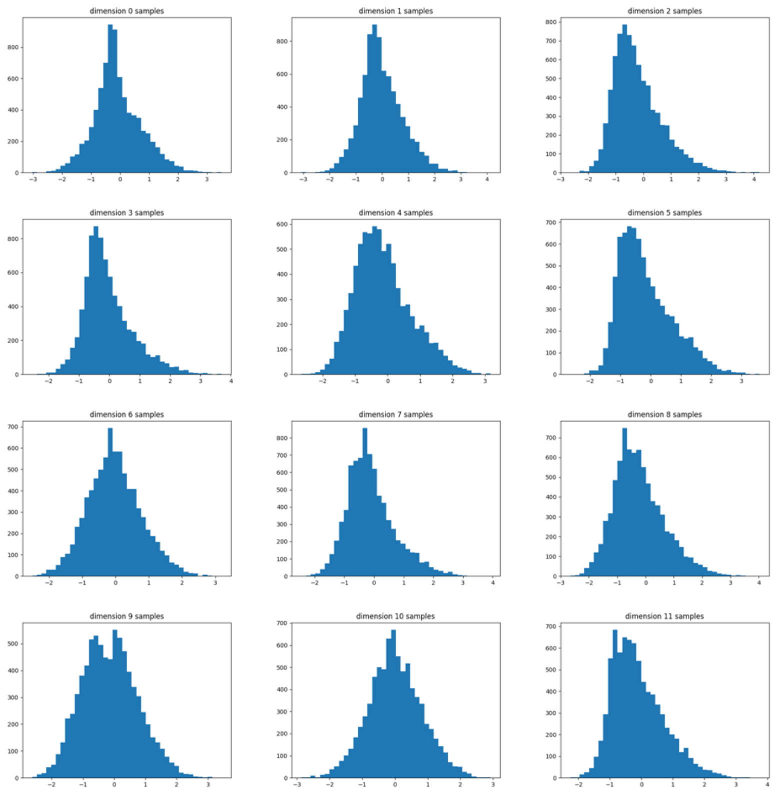
When should we use BN adaptation?

1024-dimensional distributions



Removed weight decay + Before ReLU

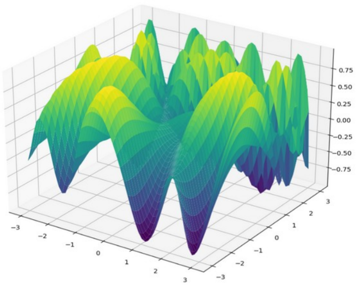
Projection per-dimension



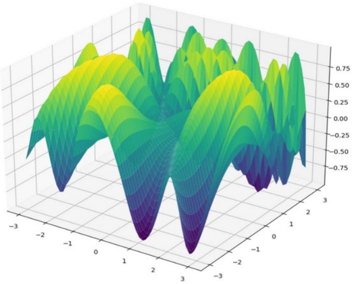
We can work with this!

When should we use BN adaptation?

1024-dimensional distributions



distance



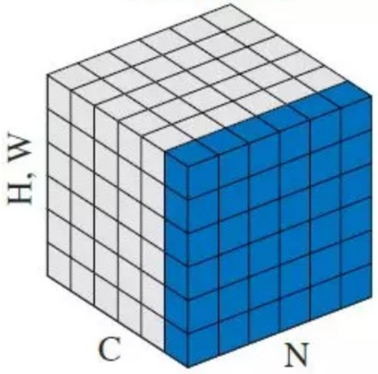
	WASSERSTEIN	SLICED WASSERSTEIN	BHATTACHARYYA
Accounts for	full distribution	projections	projections
Assumptions	Gaussianity	-	-
Sparsity of data	causes less issues	causes issues	causes issues
Computations	Rounding errors	[?]	manageable

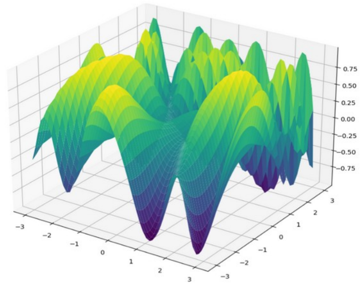
When should we use BN adaptation?

Latent spaces

1024-dimensional distributions

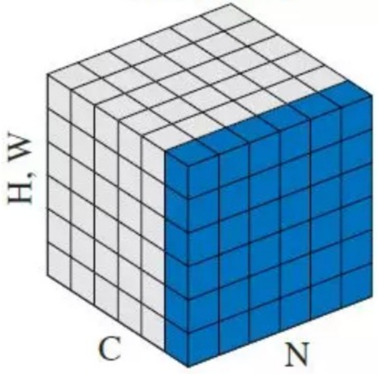
Source dataset → U-Net

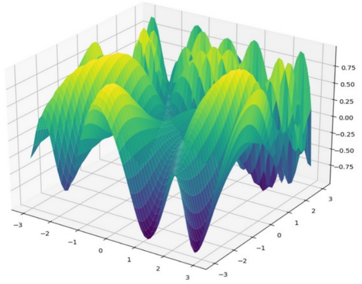




Distance (Bhattacharyya / Wasserstein / …?)

Target dataset → U-Net (adapted)

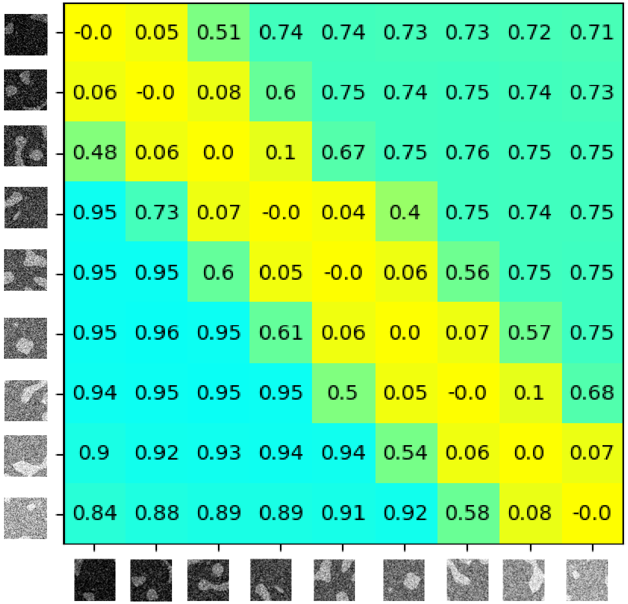




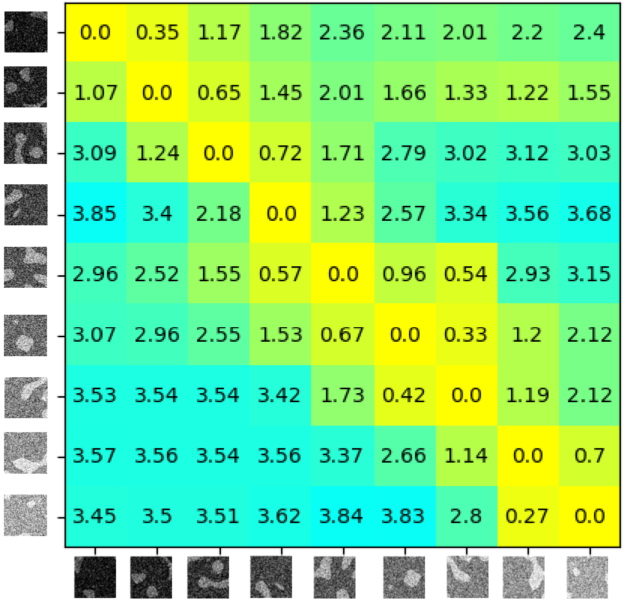
Predict BN adaptation improvement

When should we use BN adaptation?

BN adaptation improvement



Bhattacharyya



Close! But not there yet ...

Synthesis and future ideas

Synthesis

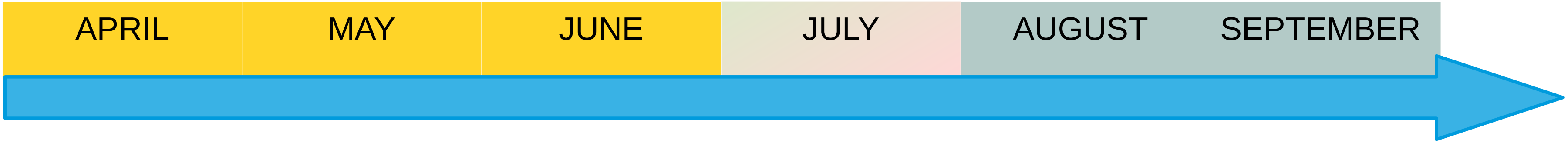
- BN adaptation works well with **brightness shifts**

- 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



What is the plan for my next 2 months?

What I'm working on

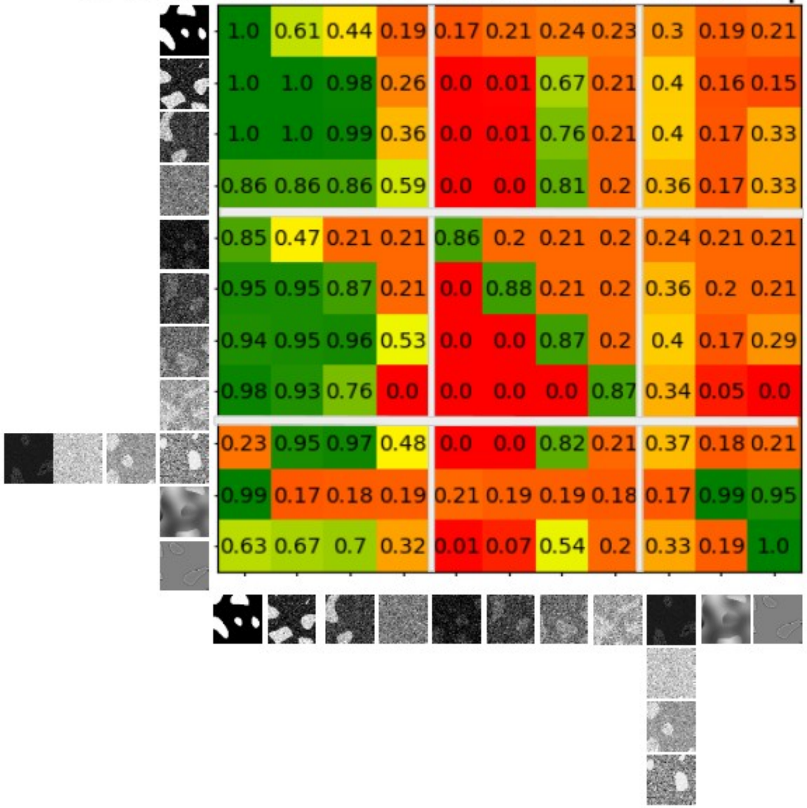
- keep experimenting with different datasets combinations
- test on all latent spaces of UNet
- test different distance measures (e.g. sliced Wasserstein)
- test for different image transformations (cropping, affine transformations, deformations, brighness, contrast, hue, ...)

Thanks for your attention!

- extra slides -

EXTRA

U-Net WITHOUT batch norm adaptation



U-Net WITH batch norm adaptation