EEG – GAN :
Generative adversarial networks for electroencephalograhic(EEG) brain signals

NAME : LEE. YEONSU

2020.04.01

Abstract

- Introduce **modification** to the improved training of **Wasserstein GAN**s(WGAN-GP) to stabilize training
  - Also, investigate a range of architectural choices critical for time series generation

- For evaluation,
  - Inception score
  - Frechet inception distance
  - Sliced Wasserstein distance
  - Euclidean distance

- It thus opens up a range of new generative scenarios in the neuroscientific and neurological context
  - Data augmentation of a certain class
  - EEG restoration

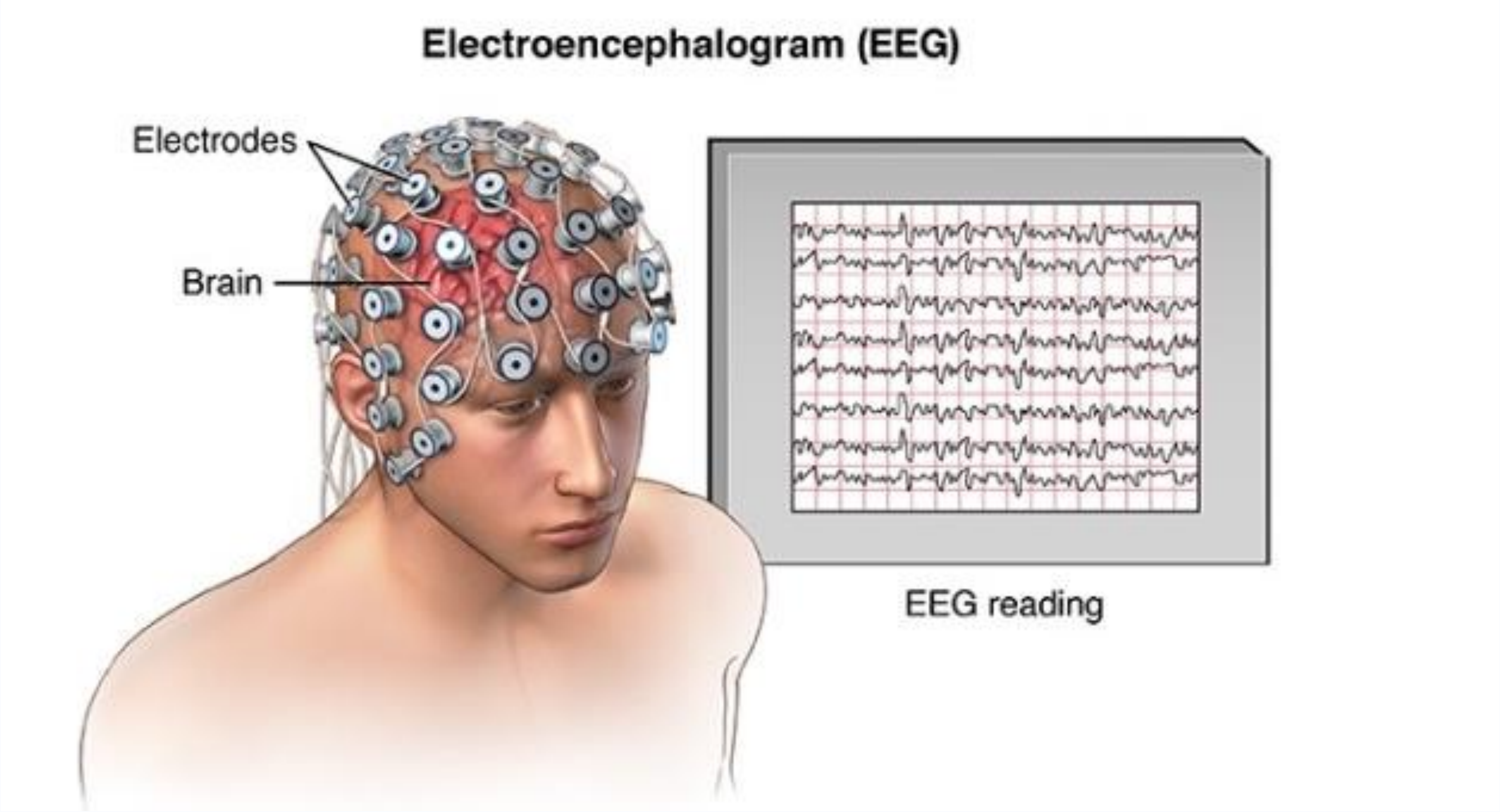
Introduction

- Vanilla GANs suffered heavily from training instability and were restricted to low resolution images.
- GANs have mainly been developed and applied to the generation of images and only a few studies investigating time series were conducted.
- No research regarding the generation of raw EEG signals with GANs has been published at this time.
- To generate naturalistic samples of EEG data, we **propose** an **improvement** to the **Wasserstein GAN** training showing increased training stability.

Data

EEG data

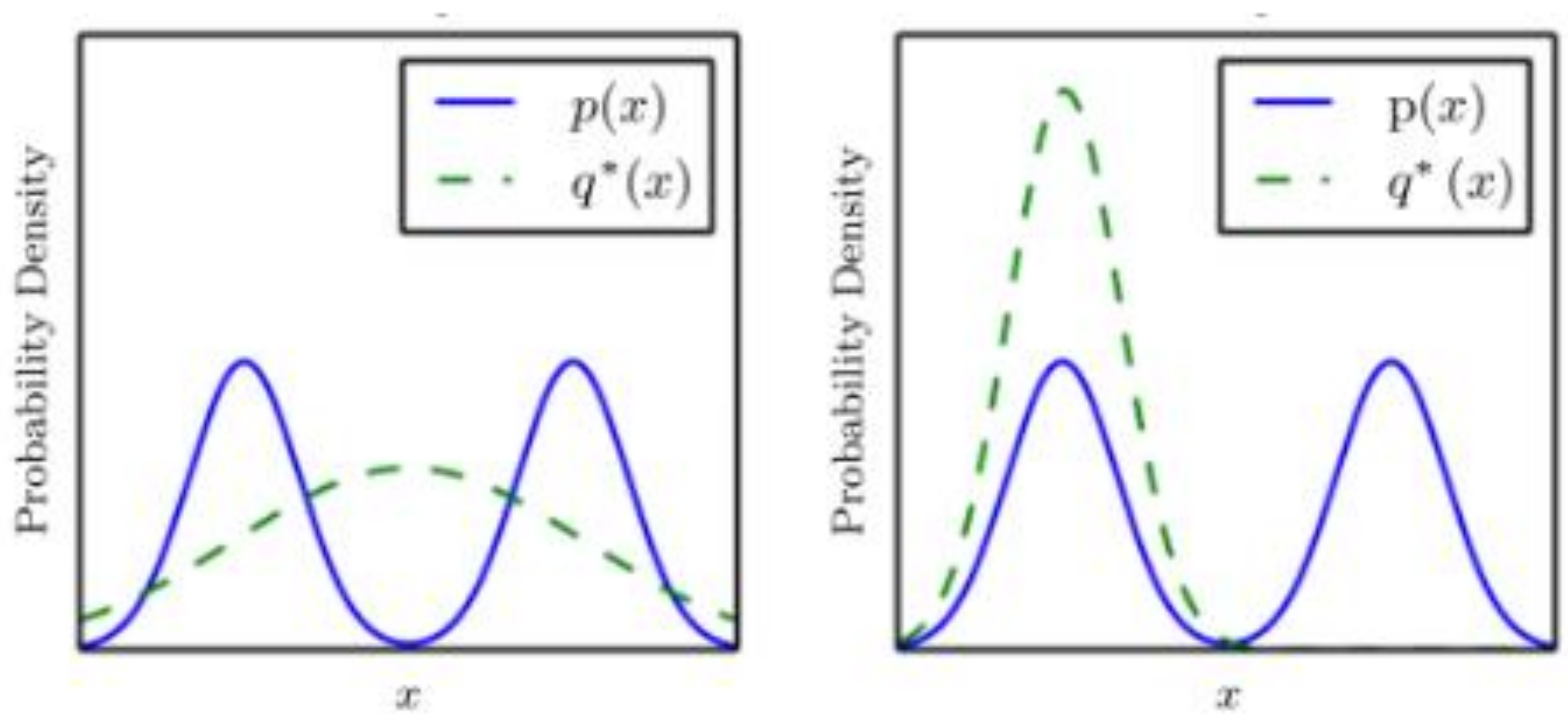
- **128 – electrode EEG system**
  - Sampling rate = 250Hz
  - Channel FCC4h (range = alpha, beta, high gamma)
  - Use Single channel ‘ FCC 4h ‘
  - Overall dataset = 438 signals
  - Training data = 286, validation = 72, test 80



Method

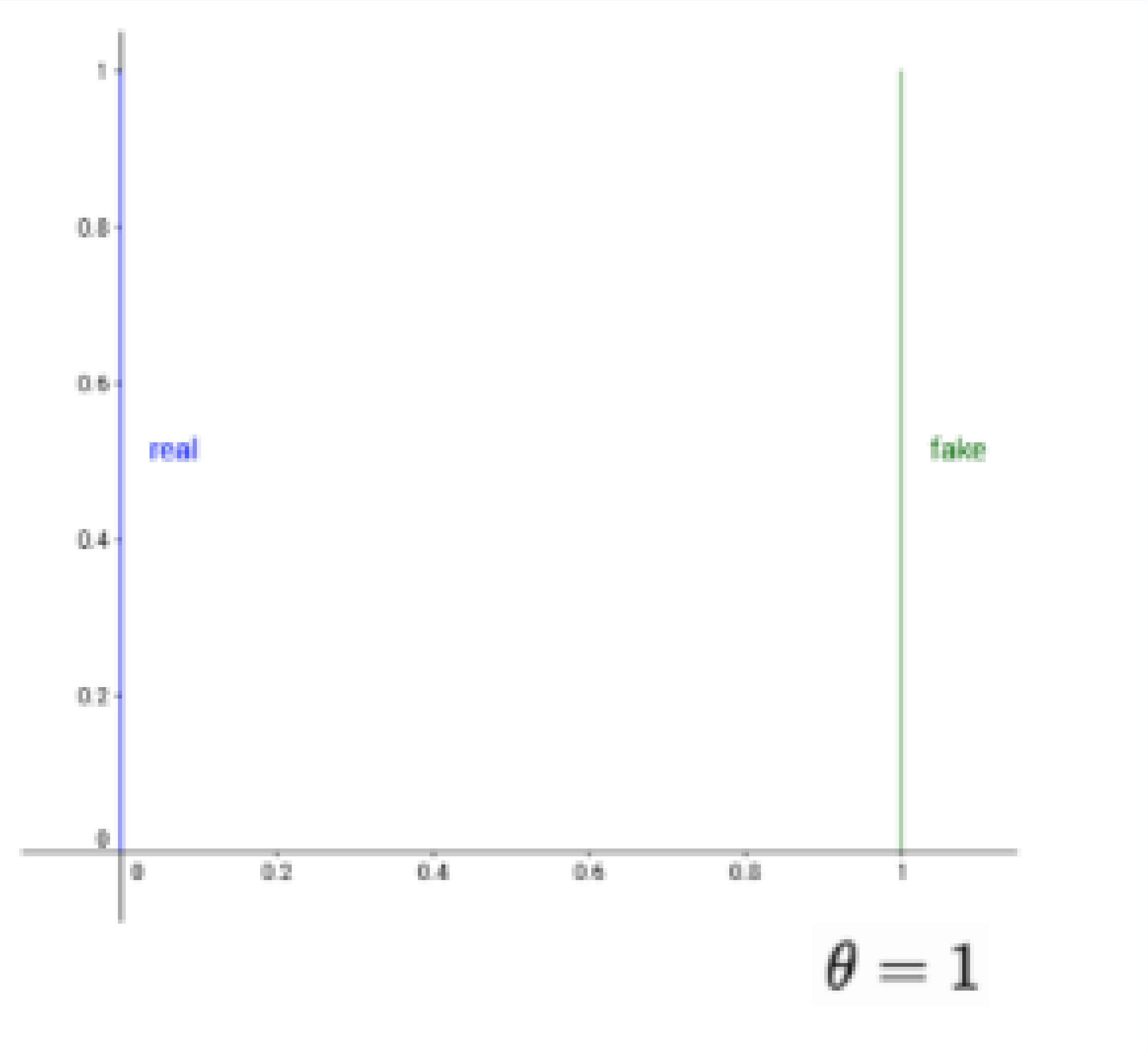
GAN background and improvement

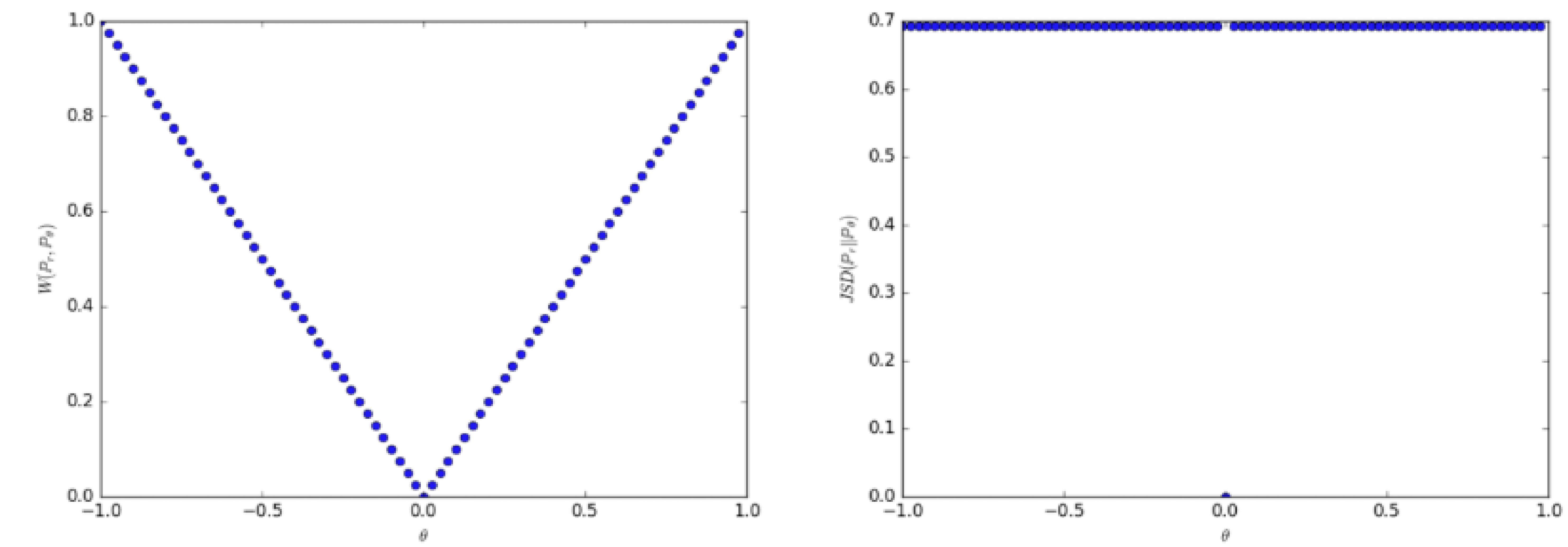
- **Vanilla GAN's draw back**
  - Vanilla GAN framework tries to minimize the Jenson-Shannon (JS) divergence between the real data distribution P_r and fake data distribution P_f
  - If the discriminator is trained to optimality this may lead to the problem of vanishing gradients for the generator. (Problem)
  - Mode collapsing problem



Method
WGAN

- **Wasserstein GAN** shows training stability.
- WGAN use wasserstein distance while GAN use Jenson-Shannon divergence.



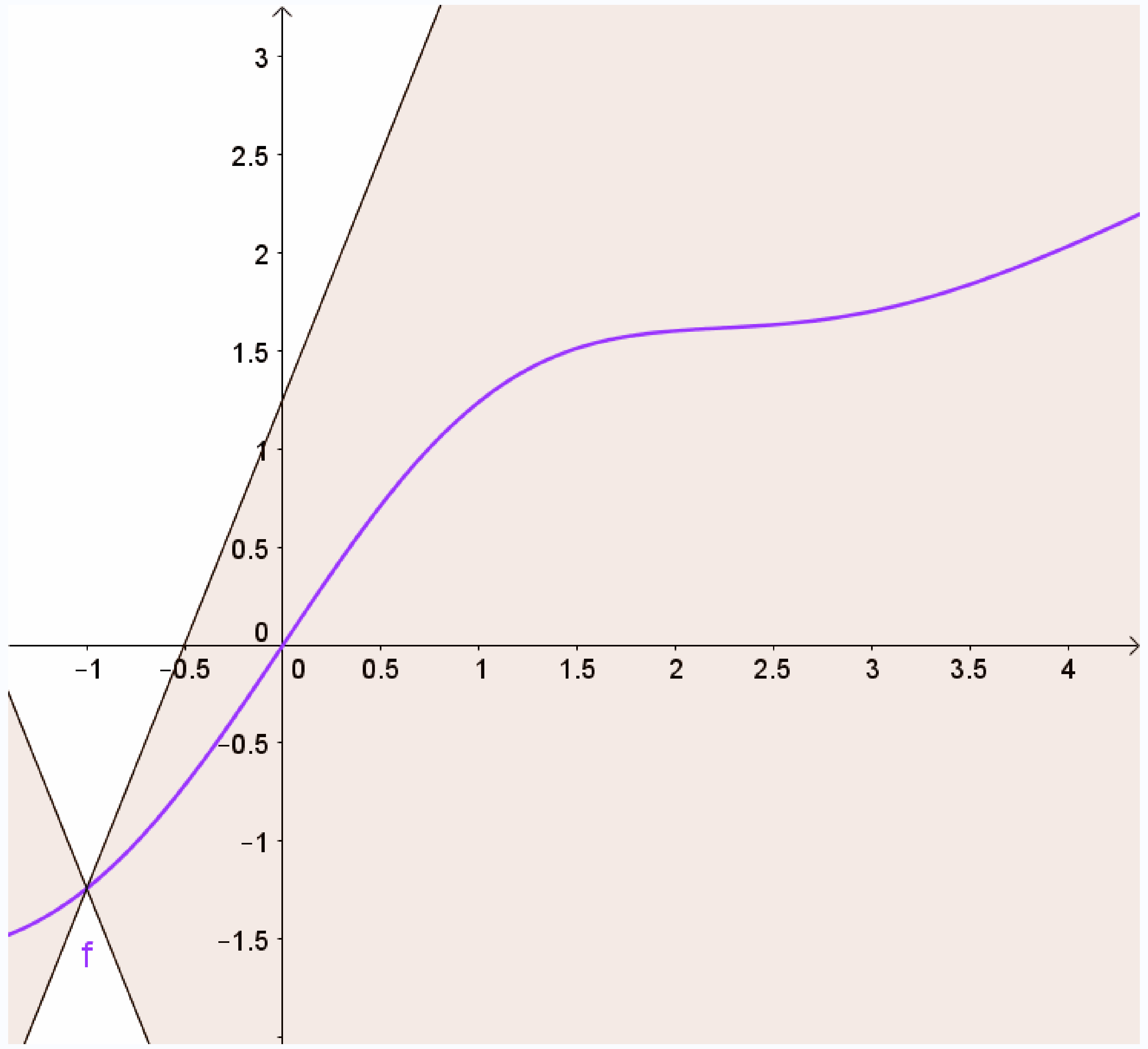


Wasserstein W와 JS를 θ에 대해 비교했다. JS

Method

WGAN - Clipping

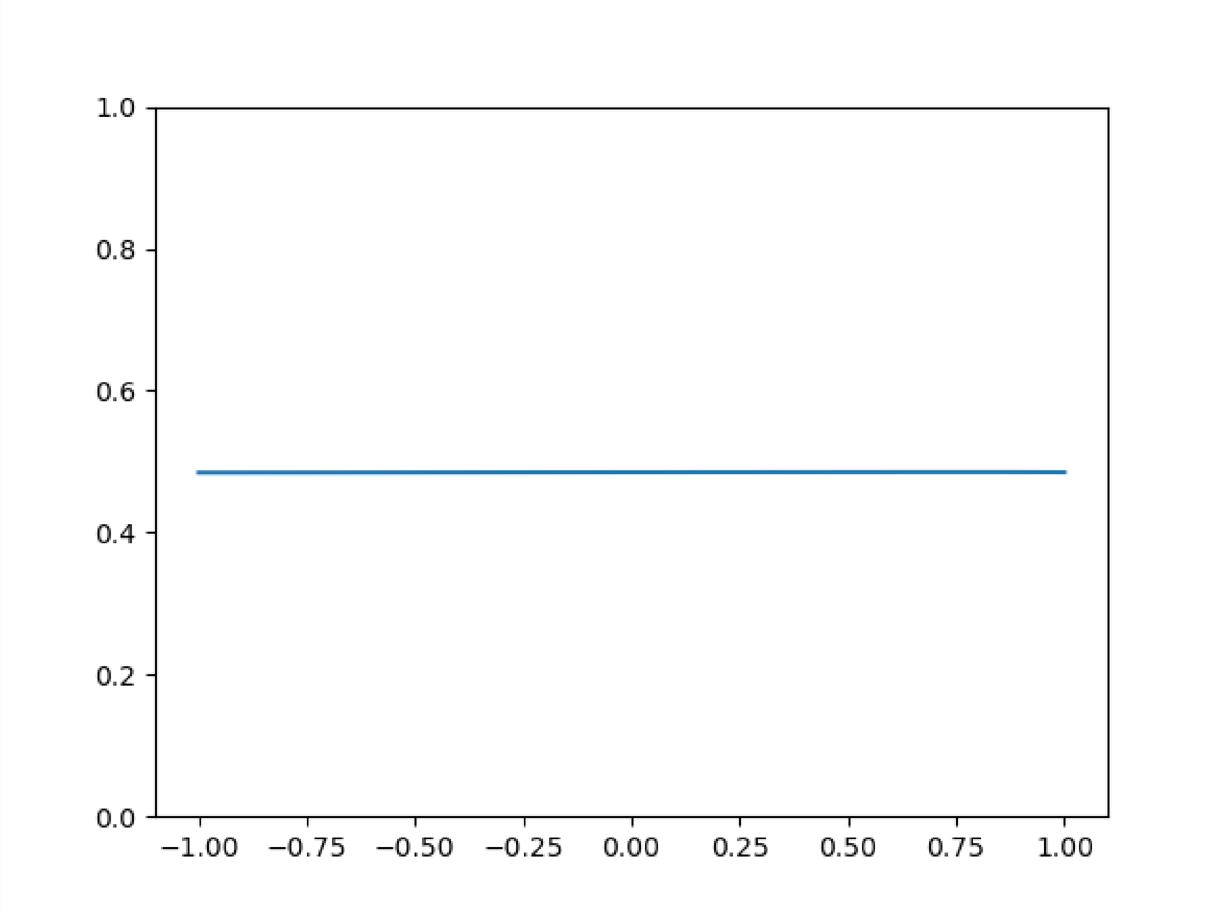
- WGAN enforce **Lipschitz continuity ($\left|\left|f\right|\right|_L \leq k$)** by clipping the weights of the discriminator to interval [-c, c]
- Limiting weights , however, leads to an undesired convergence(vanishing or exploding) of network parameters to those limits.



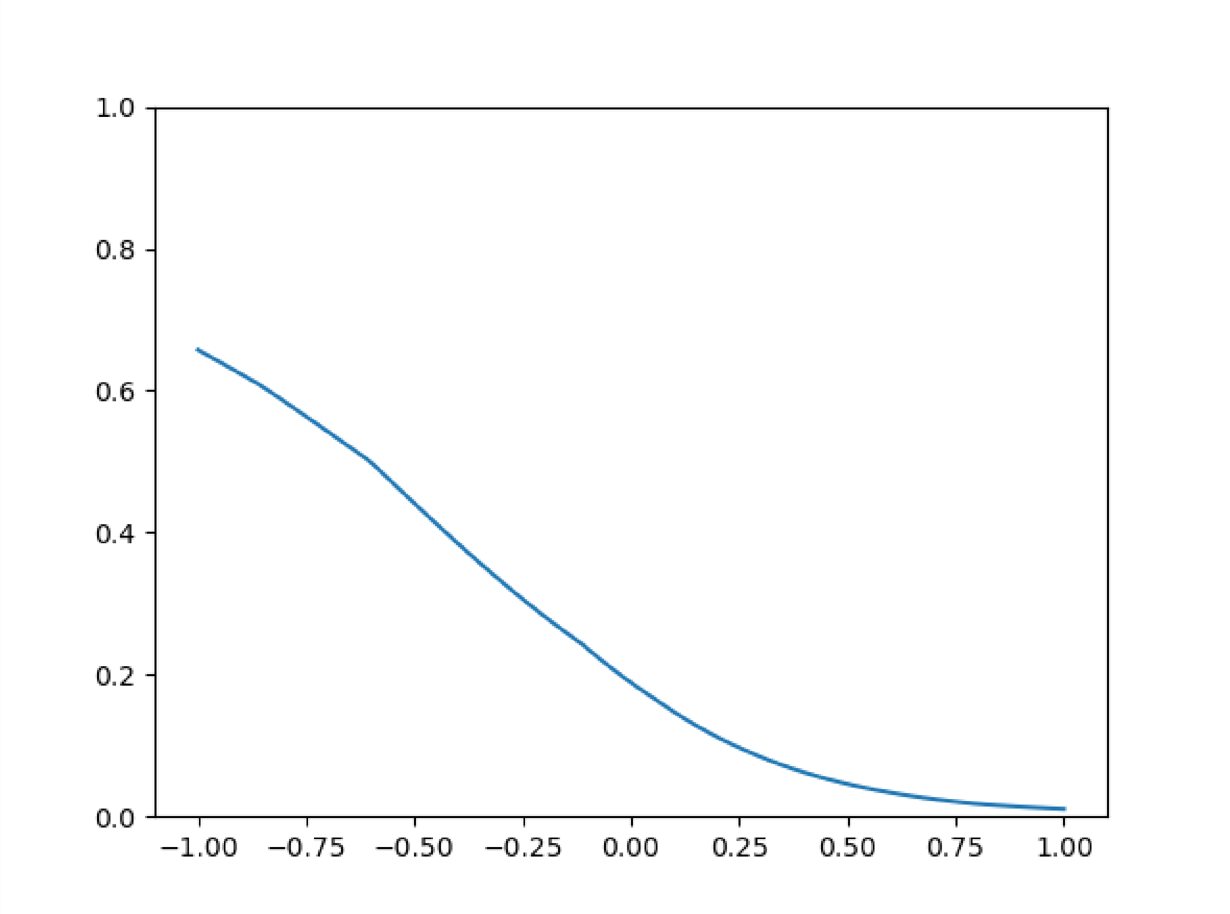
$\boldsymbol{|f(x_1) - f(x_2)| \leq k|x_1 - x_2|, k \geq 0}$ **--▷ $\boldsymbol{f}$ enforce K-Lipschitz continuous**

Method

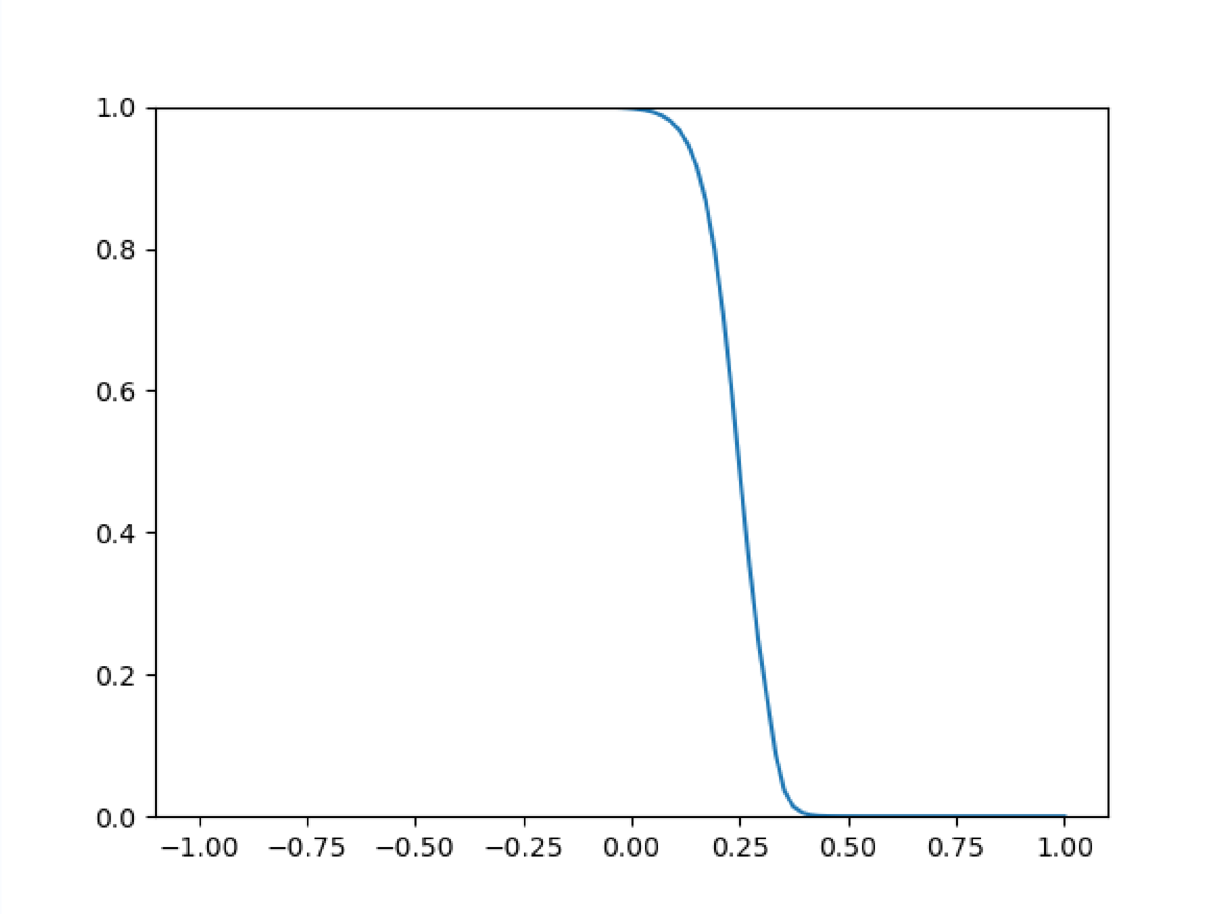
WGAN - Clipping



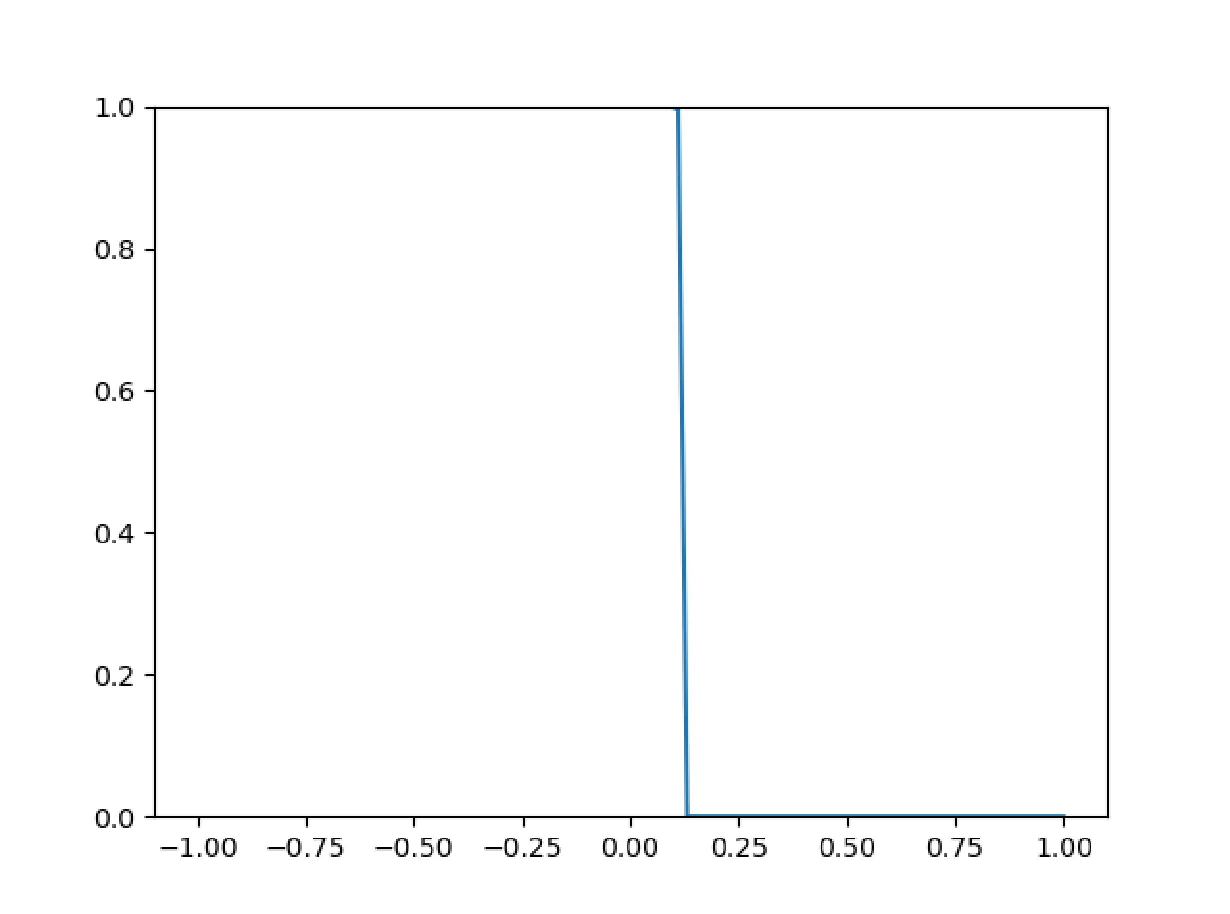
C = {-0.1,0.1}



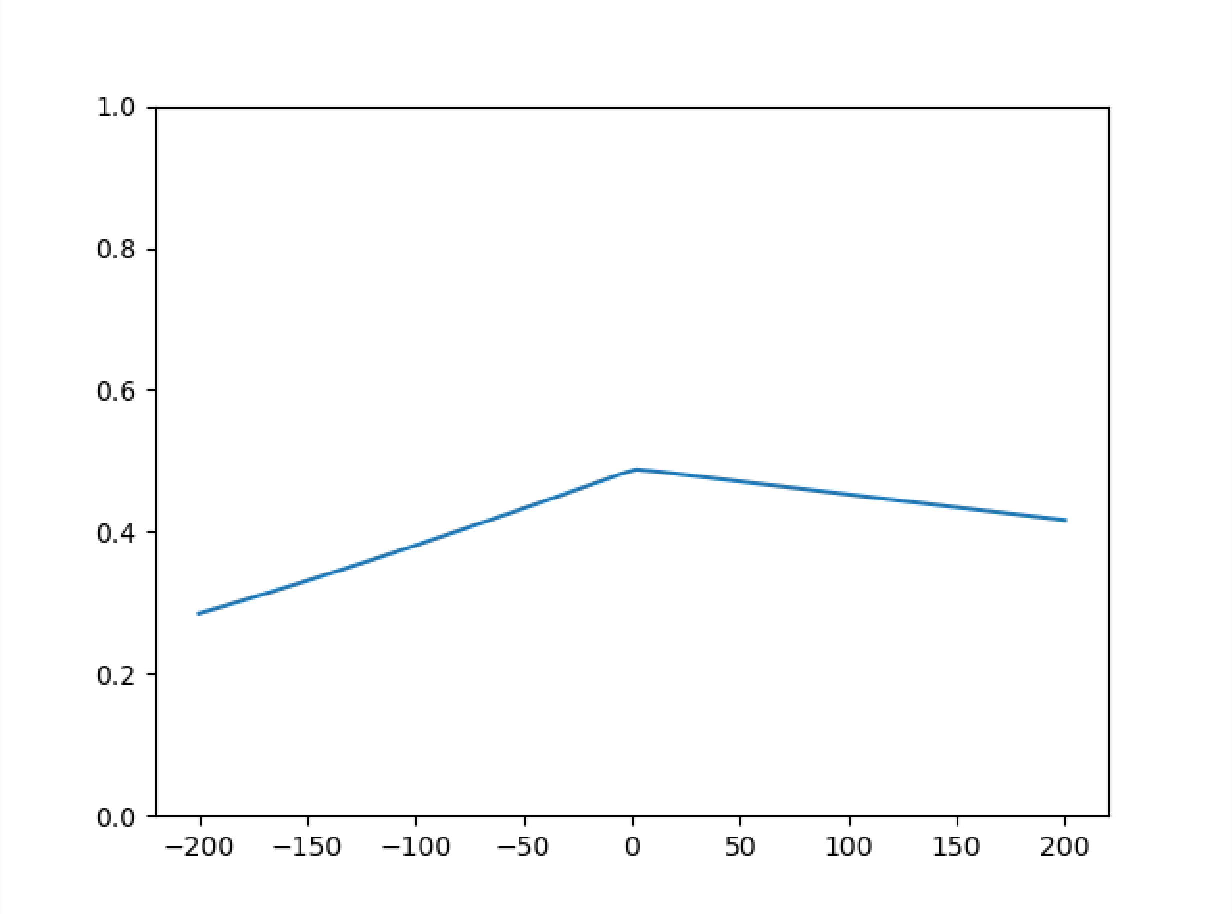
C = {-1,1}



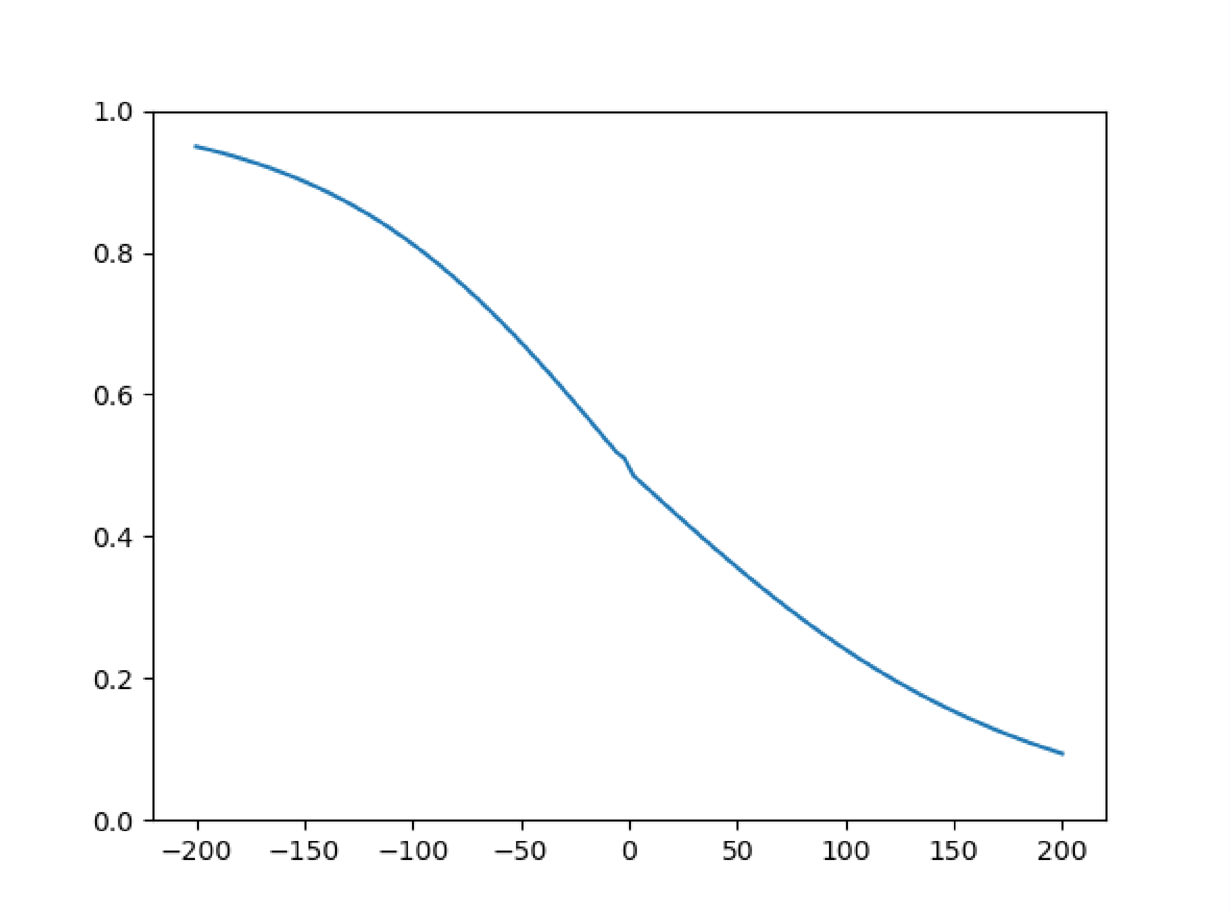
C = {-5,5}



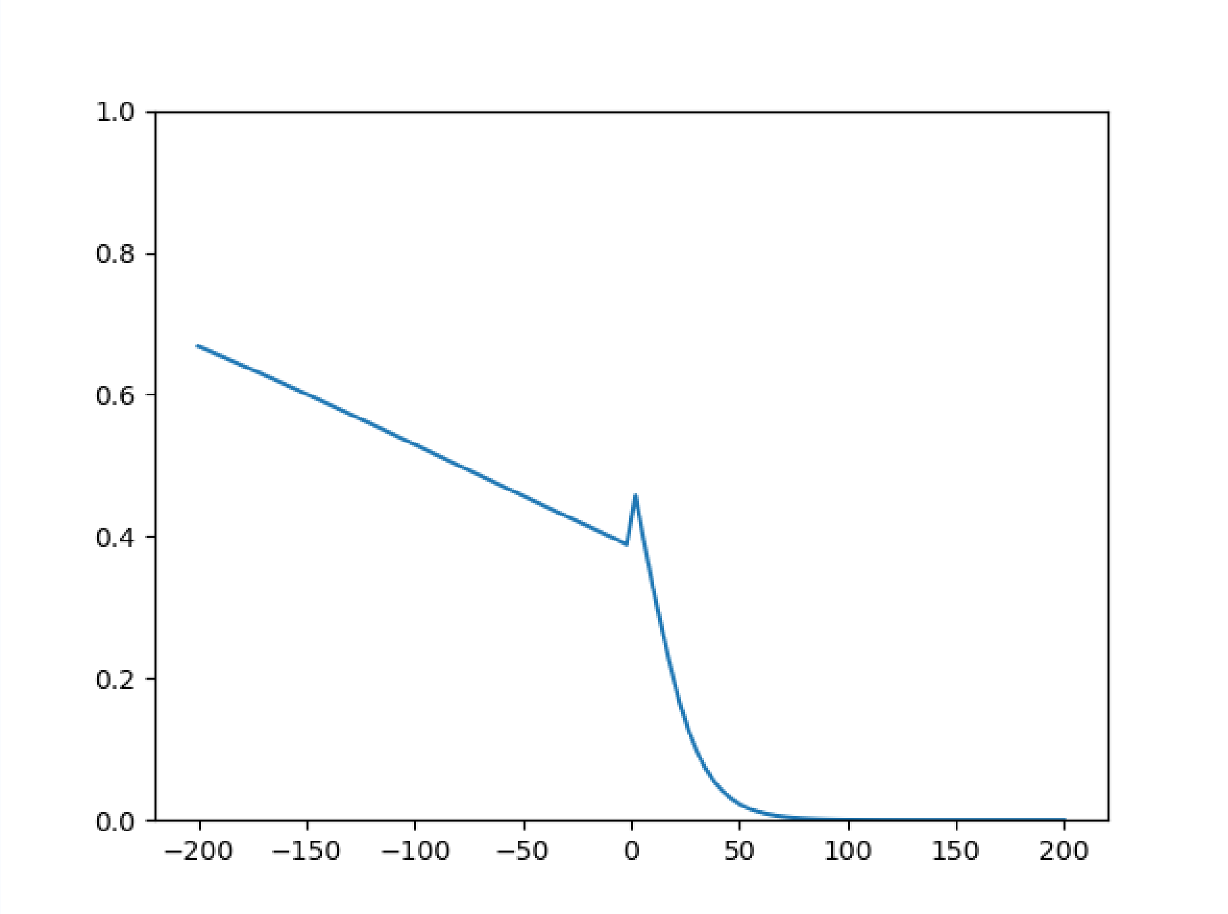
C = {-50,50}



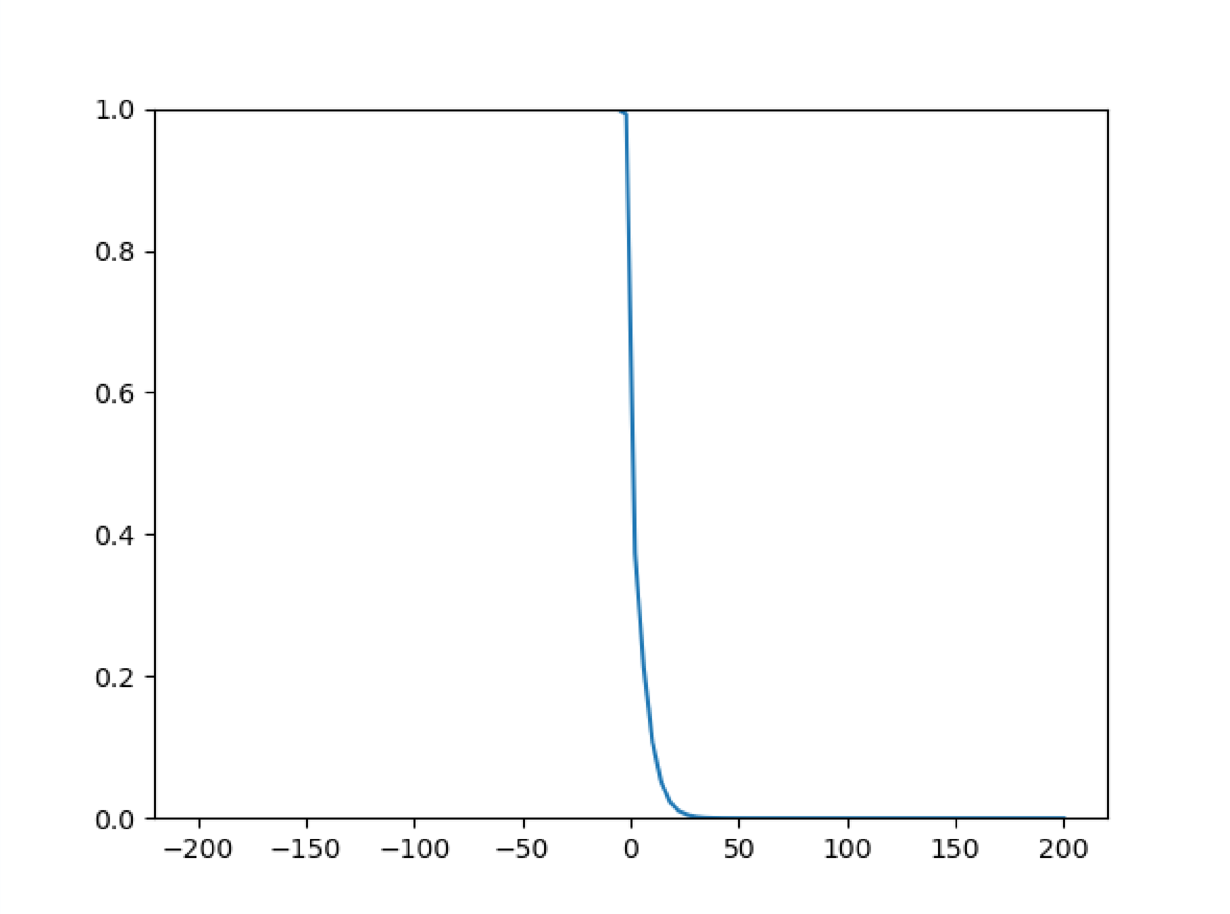
C = {-0.1,0.1}



C = {-0.2,0.2}



C = {-0.5,0.5}

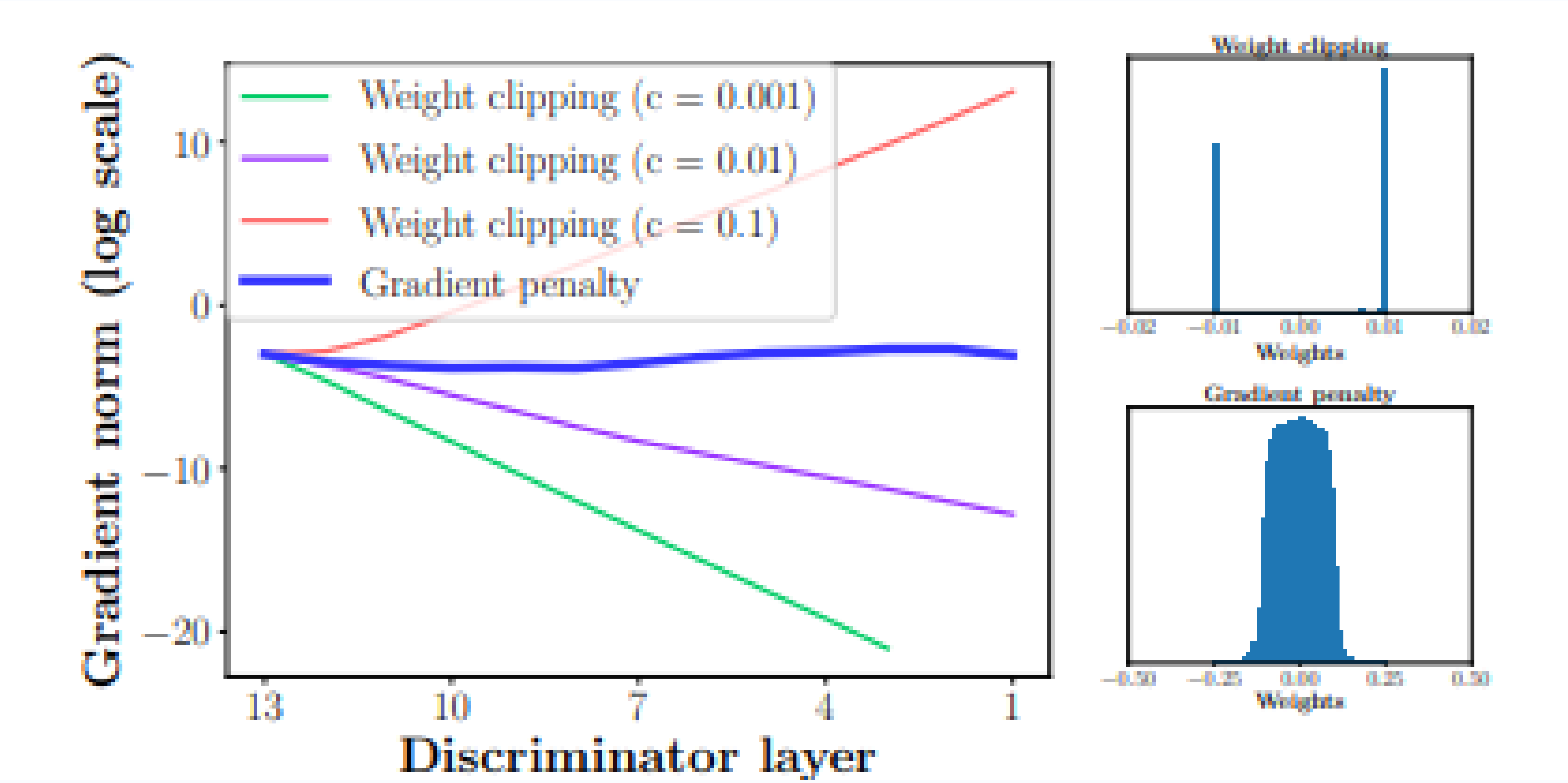


C = {-1,1}

Method

WGAN - clipping

- **WGAN** optimization process is difficult because of interactions between the weight constraint and the cost function , which result in either vanishing or exploding gradients without careful tuning of the clipping threshold c.



Method

Clipping -> GP

- Propose an alternative way to enforce the **Lipschitz constraint.**
- We consider directly constraining the gradient norm of the critic's output with respect to its input.
- WGAN GP (Gradient Penalty) gives gradient penalty in loss.

$$L_{P_2} = \underbrace{E_{\tilde{x} \sim P_g}[D(\tilde{x})] - E_{\tilde{x} \sim P_r}[D(x)]}_{\text{Original critic loss}} + \underbrace{\lambda E_{\tilde{x} \sim P_{\tilde{x}}}\left[(\|\nabla_{\tilde{x}} D(\tilde{x})\| - 1)^2\right]}_{\text{Gradient penalty}}$$

$$P_1 = \lambda E_{\tilde{x} \sim P_{\tilde{x}}}\left[max(0, \|\nabla_{\tilde{x}} D(\tilde{x})\| - 1)^2\right]$$

$$P_2 = \lambda E_{\tilde{x} \sim P_{\tilde{x}}}\left[(\|\nabla_{\tilde{x}} D(\tilde{x})\| - 1)^2\right]$$

Method
Proposed method

- We will not use the two-sided penalty P2
- They did not state a specific reason to choose the two-sided penalty over the one-sided penalty, but preferred it from empirical results.
- The resulting loss function for the critic Generative adversarial networks for brain signals then becomes: **Proposed Loss function**
- Instead of only weighting the penalty term with λ, we also scale it by the current critic difference $\widetilde{W}(P_r, P_{theta})$

$$L_c = \underbrace{E_{\widetilde{x} \sim P_{theta}}[D(\widetilde{x})] - E_{\widetilde{x} \sim P_r}[D(x)]}_{-\widetilde{W}} + \boxed{max(0, \widetilde{W}(P_r, P_{theta}) \cdot P_1}$$

Training and architecture choices

Network architecture

- We start at a resolution 24 time samples and increase the resolution by factor **2** over **6** steps to arrive at 768 samples.
- Factor 2 introduced the least frequency artifacts and led to the best results.
- Use Upsampling : cubic interpolation, linear interpolation, nearest-neighbor upsampling
  - NN upsampling introduces strong high-frequency artifacts
  - CUB, LIN lead to much weaker artifacts

Table 1. Network architecture

(a) Generator

Layer	Act./Norm.	Output shape
Latent vector	-	200 x 1
Linear	LReLU	50 x 12
Upsample	-	50 x 24
Conv 9	LReLU/PN	50 x 24
Conv 9	LReLU/PN	50 x 24
Upsample	-	50 x 48
Conv 9	LReLU/PN	50 x 48
Conv 9	LReLU/PN	50 x 48
Upsample	-	50 x 96
Conv 9	LReLU/PN	50 x 96
Conv 9	LReLU/PN	50 x 96
Upsample	-	50 x 192
Conv 9	LReLU/PN	50 x 192
Conv 9	LReLU/PN	50 x 192
Upsample	-	50 x 384
Conv 9	LReLU/PN	50 x 384
Conv 9	LReLU/PN	50 x 384
Upsample	-	50 x 768
Conv 9	LReLU/PN	50 x 768
Conv 9	LReLU/PN	50 x 768
Conv 1	-	1 x 768

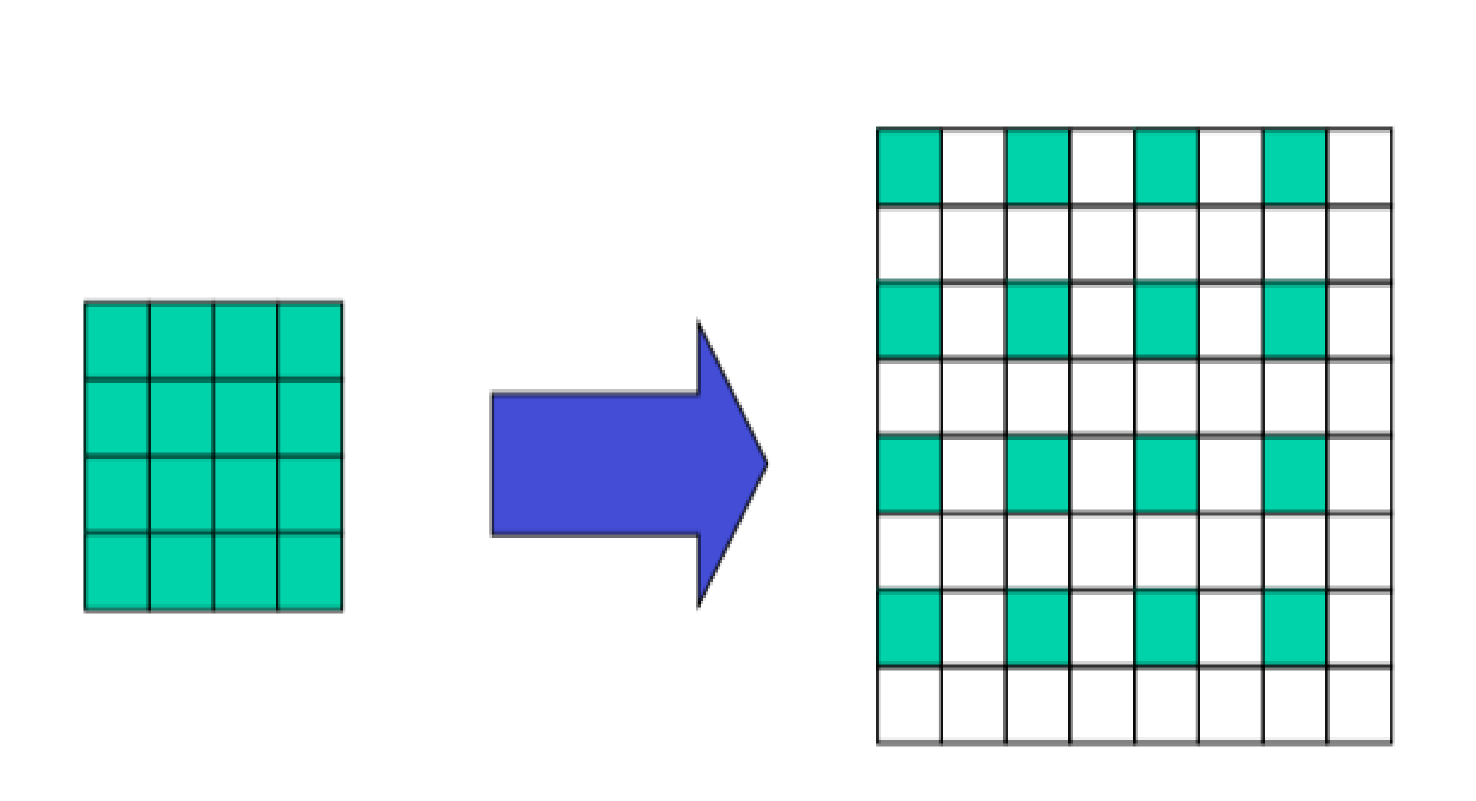
(b) Critic

Layer	Act.	Output shape
Input signal	-	1 x 768
Conv 1	LReLU	50 x 768
Conv 9	LReLU	50 x 768
Conv 9	LReLU	50 x 768
Downsample	-	50 x 384
Conv 9	LReLU	50 x 384
Conv 9	LReLU	50 x 384
Downsample	-	50 x 192
Conv 9	LReLU	50 x 192
Conv 9	LReLU	50 x 192
Downsample	-	50 x 96
Conv 9	LReLU	50 x 96
Conv 9	LReLU	50 x 96
Downsample	-	50 x 48
Conv 9	LReLU	50 x 48
Conv 9	LReLU	50 x 48
Downsample	-	50 x 24
Conv 9	LReLU	50 x 24
Conv 9	LReLU	50 x 24
Downsample	-	50 x 12
Linear	-	1 x 1

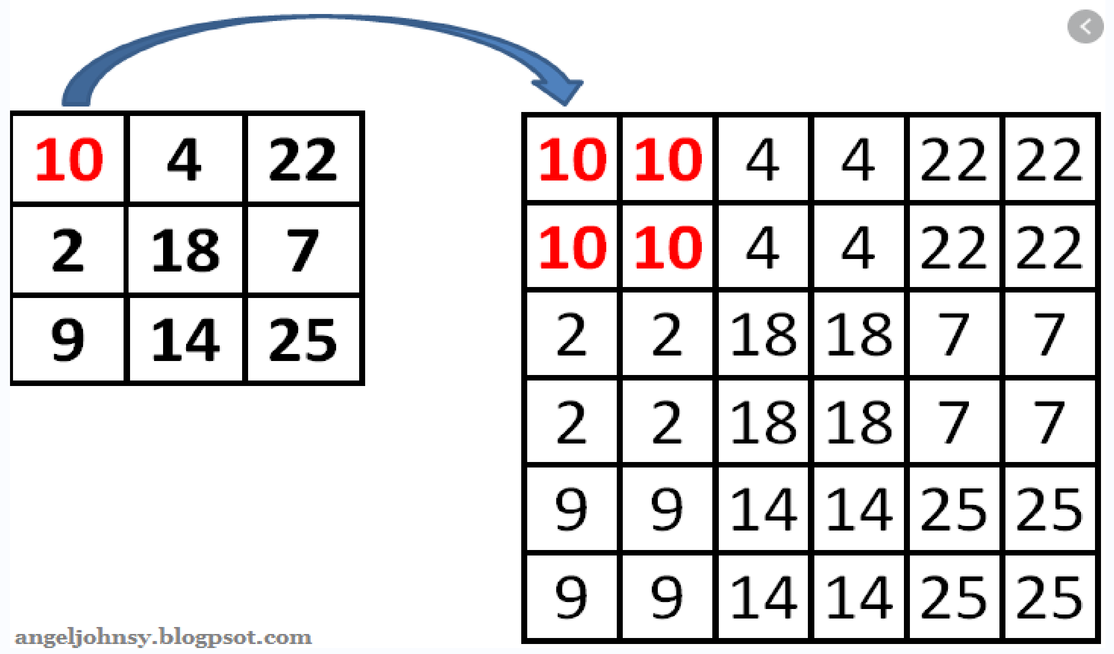
Training and architecture choices

Network architecture

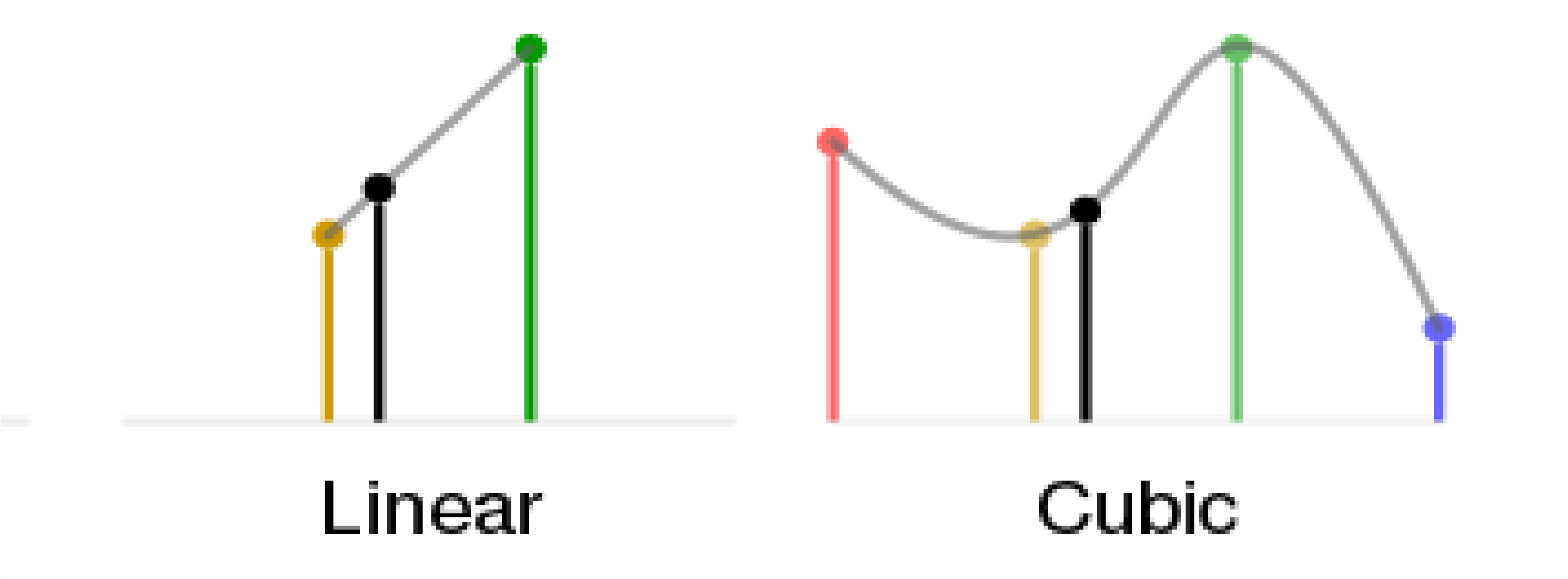
- Upsampling Interpolation



upsampling



NN upsampling

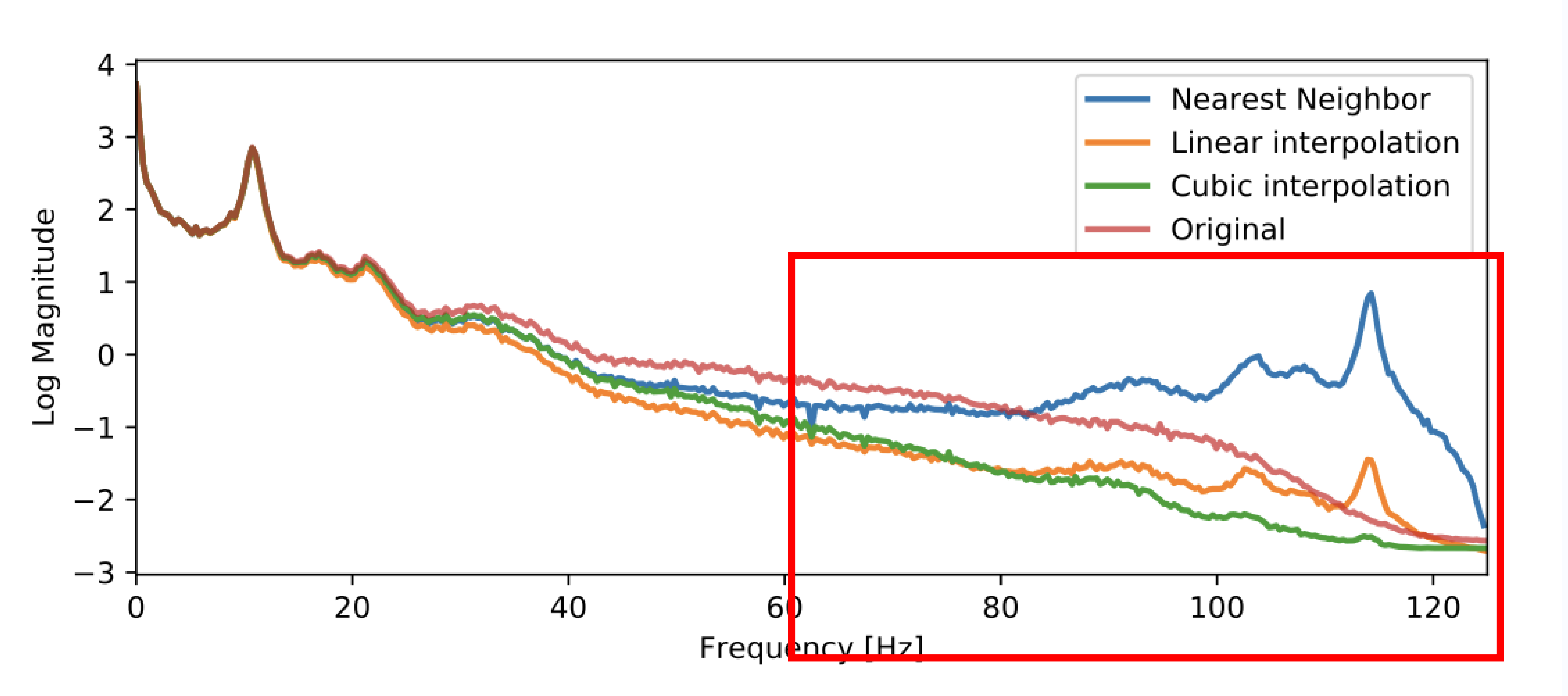


interpolation

Training and architecture choices

Network architecture

- We start at a resolution 24 time samples and increase the resolution by factor **2** over **6** steps to arrive at 768 samples.
- Factor 2 introduced the least frequency artifacts and led to the best results.
- Use Upsampling : cubic interpolation, linear interpolation, nearest-neighbor upsampling
  - **NN upsampling introduces strong high-frequency artifacts**
  - **CUB, LIN lead to much weaker artifacts**

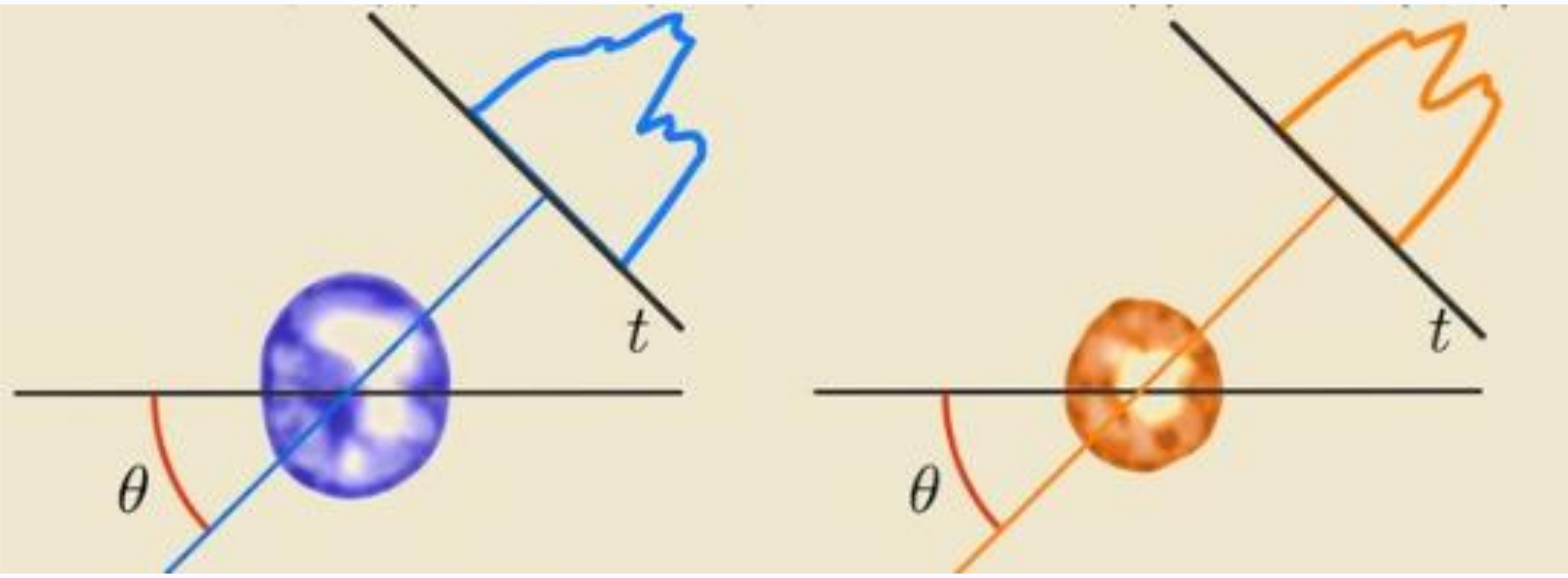


Evaluation metrics

- INCEPTION SCORE
- FRECHET INCEPTION DISTANCE

  > real data & fake data feature space distance

  > $FID^2 = \|m_f - m_r\|^2 + Tr(C_f + C_r - 2(C_f C_r)^{\frac{1}{2}})$
- EUCLIDEAN DISTANCE (??? 왜 마이너스지 ???)
- SLICED WASSERSTEIN DISTANCE



Results

- WGAN-GP model collapsed (IS gave no strong evidence for the collapse of the mode but the others)
- Different architectures performed best for different metrics.

- CONV-LIN performed best for IS
- AVG-NN performed best for FID
- AVG-NN performed best again for ED
- CONV-CUB performed best for SWD

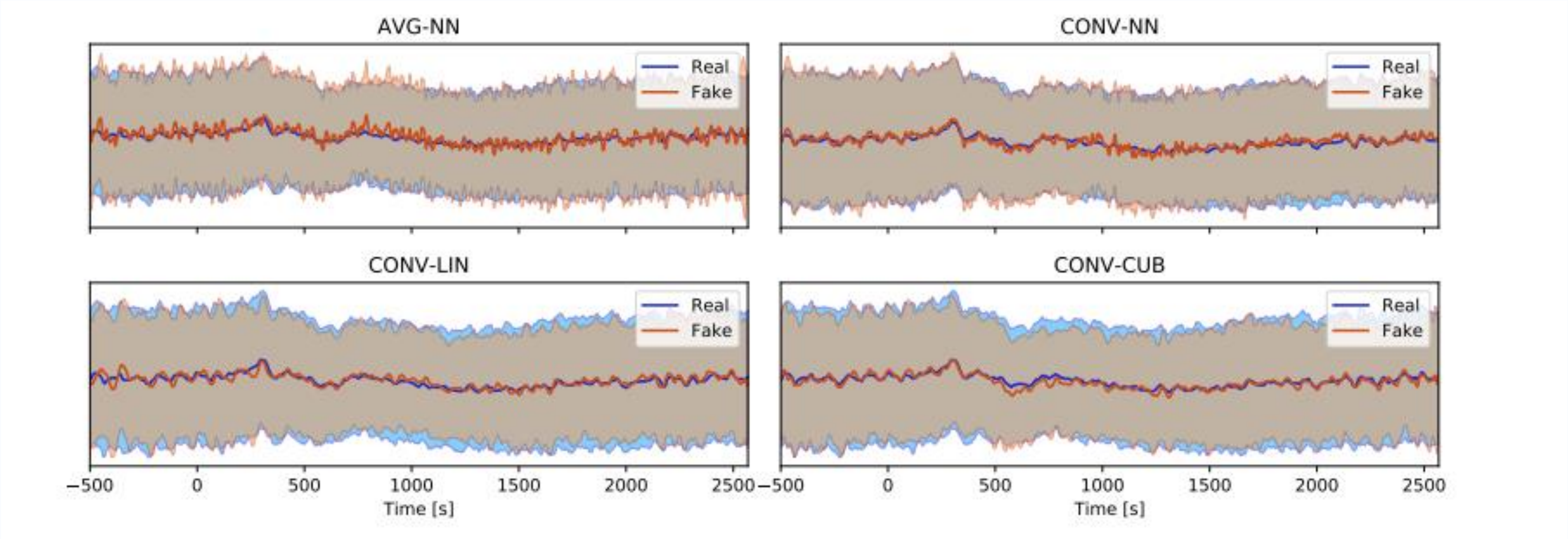
- AVG = average pooling
- NN = nearest-neighbor upsampling
- LIN = linear interpolation
- CUB = cubic interpolation

#	Model	IS	FID	ED_{min}	SWD
1	AVG-NN	1.361	**9.523**	**-0.056**	*0.102*
2	CONV-NN	1.297	16.755	-0.121	0.084
3	CONV-LIN	**1.363**	11.854	-0.252	0.086
4	CONV-CUB	*1.292*	*33.765*	*-0.375*	**0.078**
5	WGAN-GP CONV-CUB	1.281	120.854	+0.034	0.309
	Real	1.555	0.	4.653	0.
	Noise	1.049	614.782	+1.061	0.155

Visual inspection

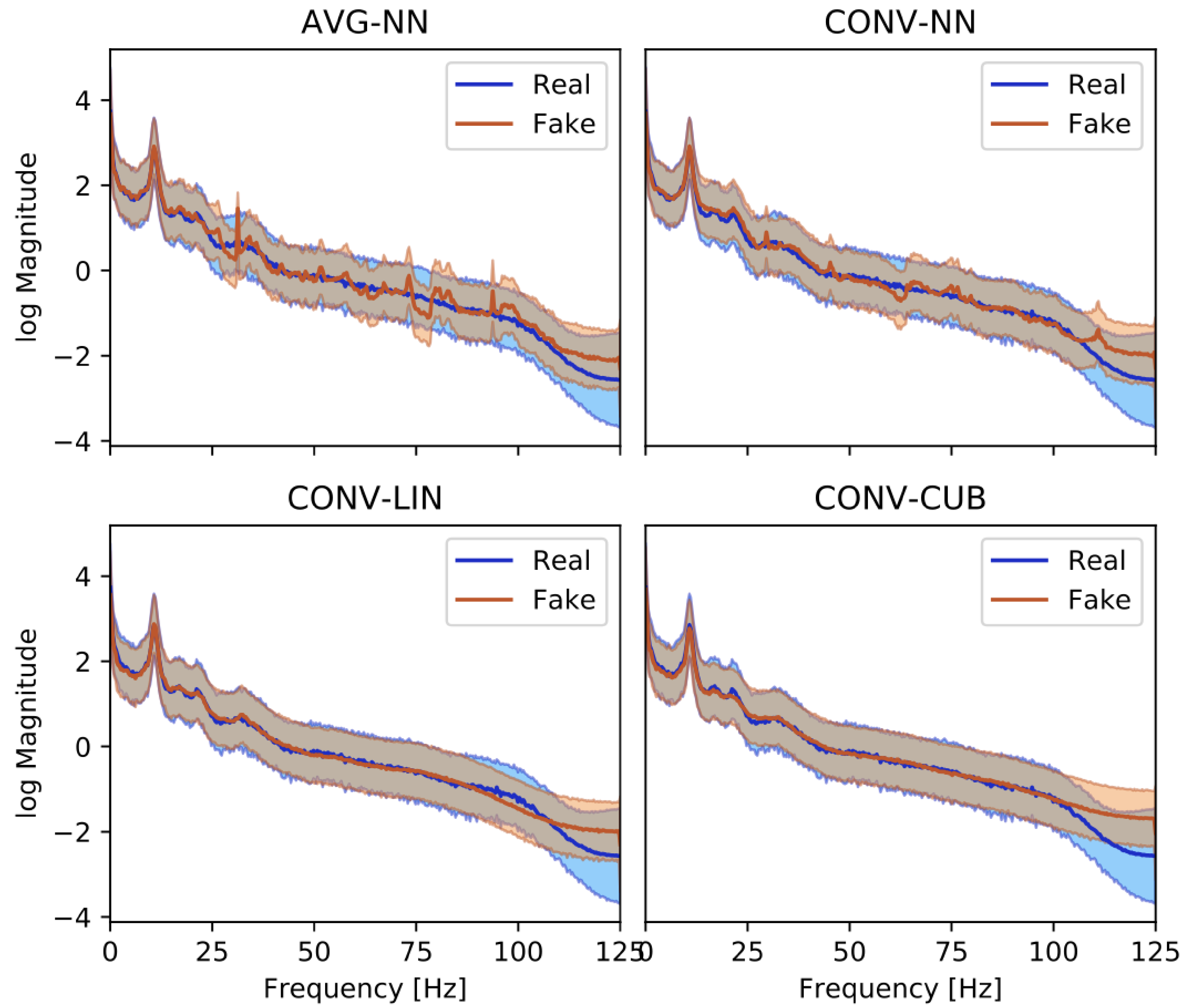
TIME SAMPLES

- AVG-NN shows a clear deviation of the generated sample distributions from real data
- CONV-CUB shows a very good fit.



Visual inspection
FREQUENCY SPECTRA

- CONV-LIN and CONV-CUB show a good fit.
- CONV-LIN better fits low frequencies, whereas CON-LIN shows better fits in high frequencies.
- No model managed to properly fit frequencies higher than 100Hz



Conclusion

- Inception score(IS) did not give meaningful information about the quality of signals generated by a model.
- Also, Frechet inception distances (FID) did not necessarily produce signals with spatial and spectral properties similar to the real input samples.
- The model expressing the most natural looking spatial and spectral distributions had the best sliced Wasserstein distance (SWD).
- Overall, no single metric gave sufficient information about the quality of a model
- Combination of FID, SWD and ED gave a good idea about its possible overall properties

Future works

- Training not only single channel, also multi-channel EEG recordings.
- Understand the impact of different design choices such as convolution size and up-down sampling.

Thank you.