Contrastive Learning for Speech Enhancement

黃仁鴻

Outline

- Introduction
- Method
- Experiment
- Conclusion

Introduction

Many tasks in daily life rely on voice as the medium of information transmission.

However, all kinds of noise interference in the real environment will seriously affect the performance of the speech task.

Therefore, the speech enhancement technology that removes these noises has become an important pre-processing unit.

Introduction

And speech enhancement means that no matter what kind of noise environment, the same speech should have the same features and can be restored to the same result.

This part of the idea coincides with the self-supervised method of contrastive learning.

Contrastive learning hopes that the features between positive samples are as similar as possible, while the feature difference between negative samples is the greater the better.

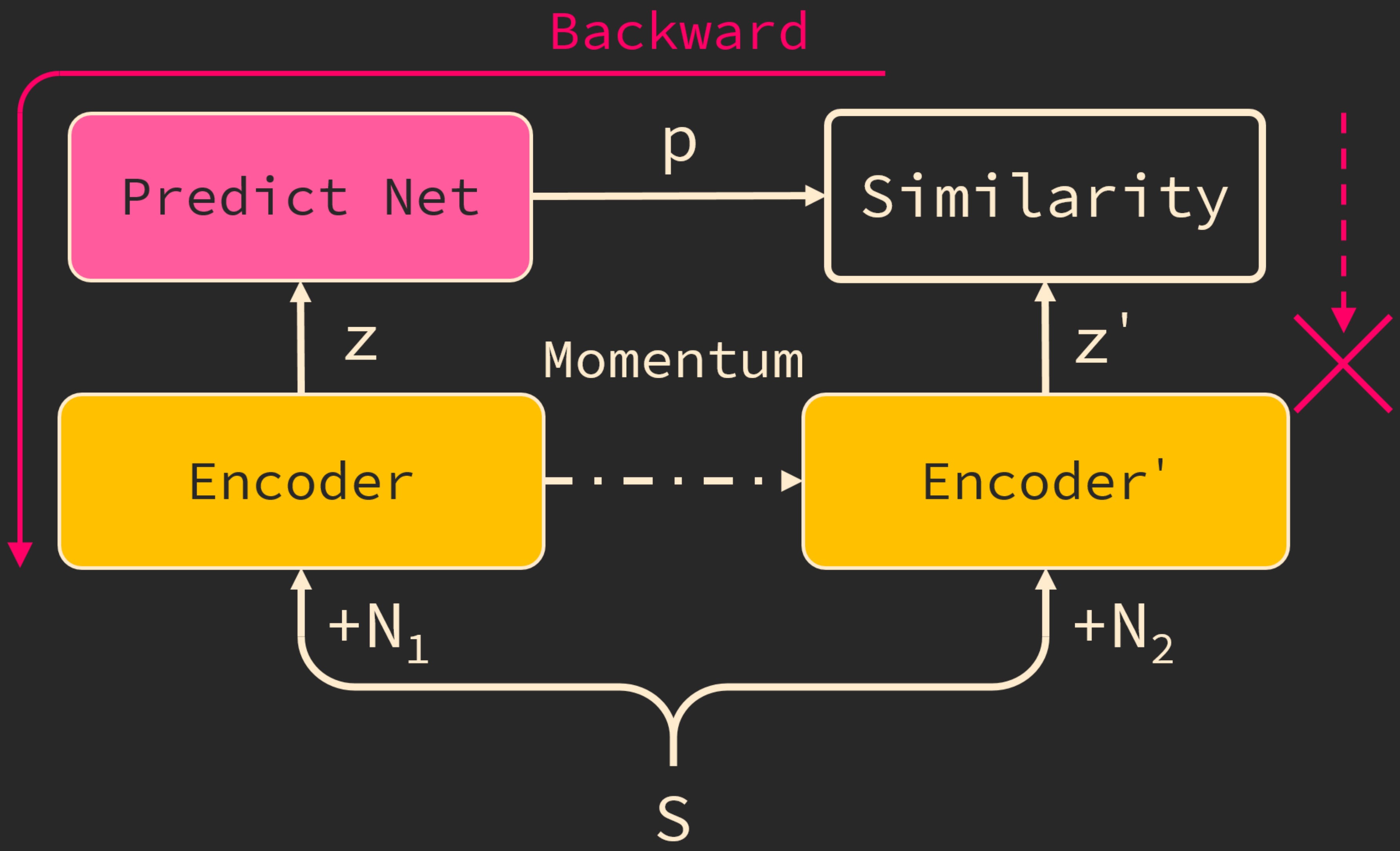
Introduction

I think learning speech features through CL method should have higher performance than common deep learning speech enhancement methods.

However, it is not easy to determine the negative sample of the frame level in the SE problem.

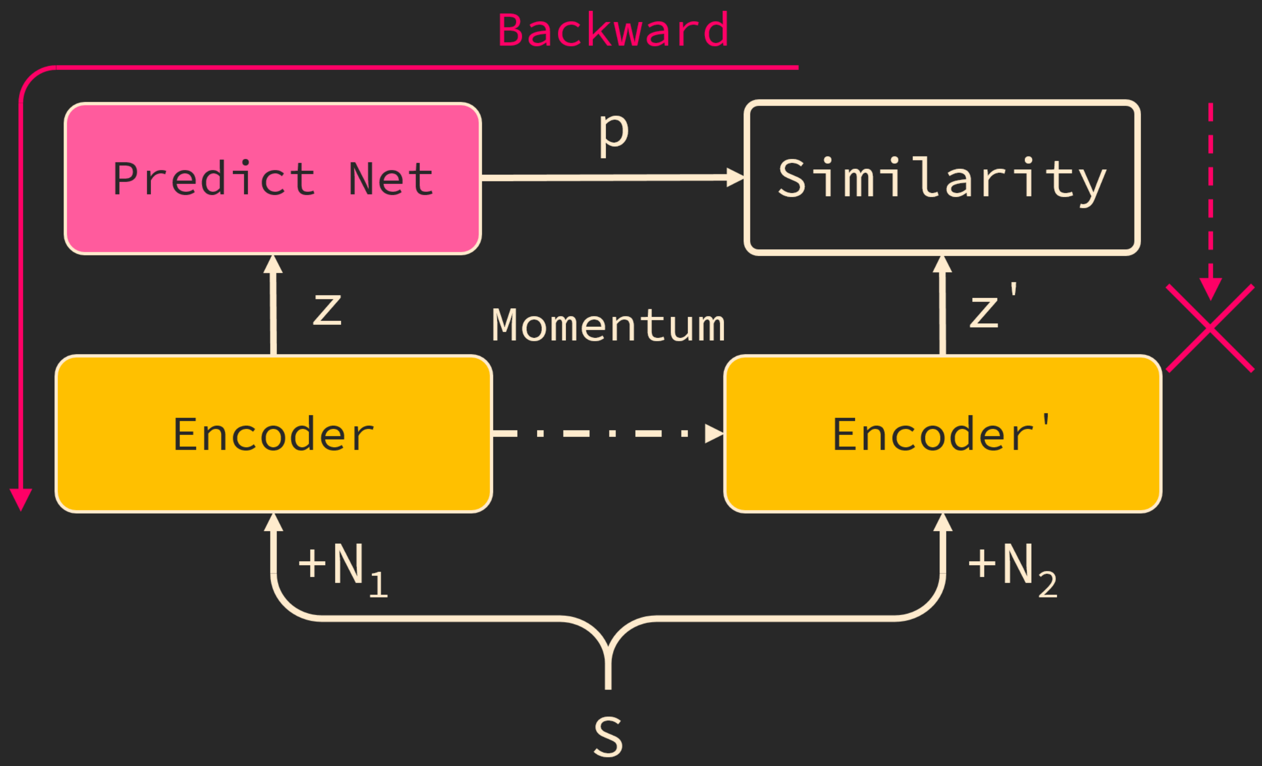
To this end, this study uses two methods, BYOL and SimSiam, which do not require negative samples, and compares them with models that do not use the CL method.

BYOL


Backward
Predict Net
p
Similarity
z
Momentum
z'
Encoder
Encoder'
+N1
+N2
S

Momentum Update

$$\theta_{E'} = \tau\theta_{E'} + (1-\tau)\theta_E$$

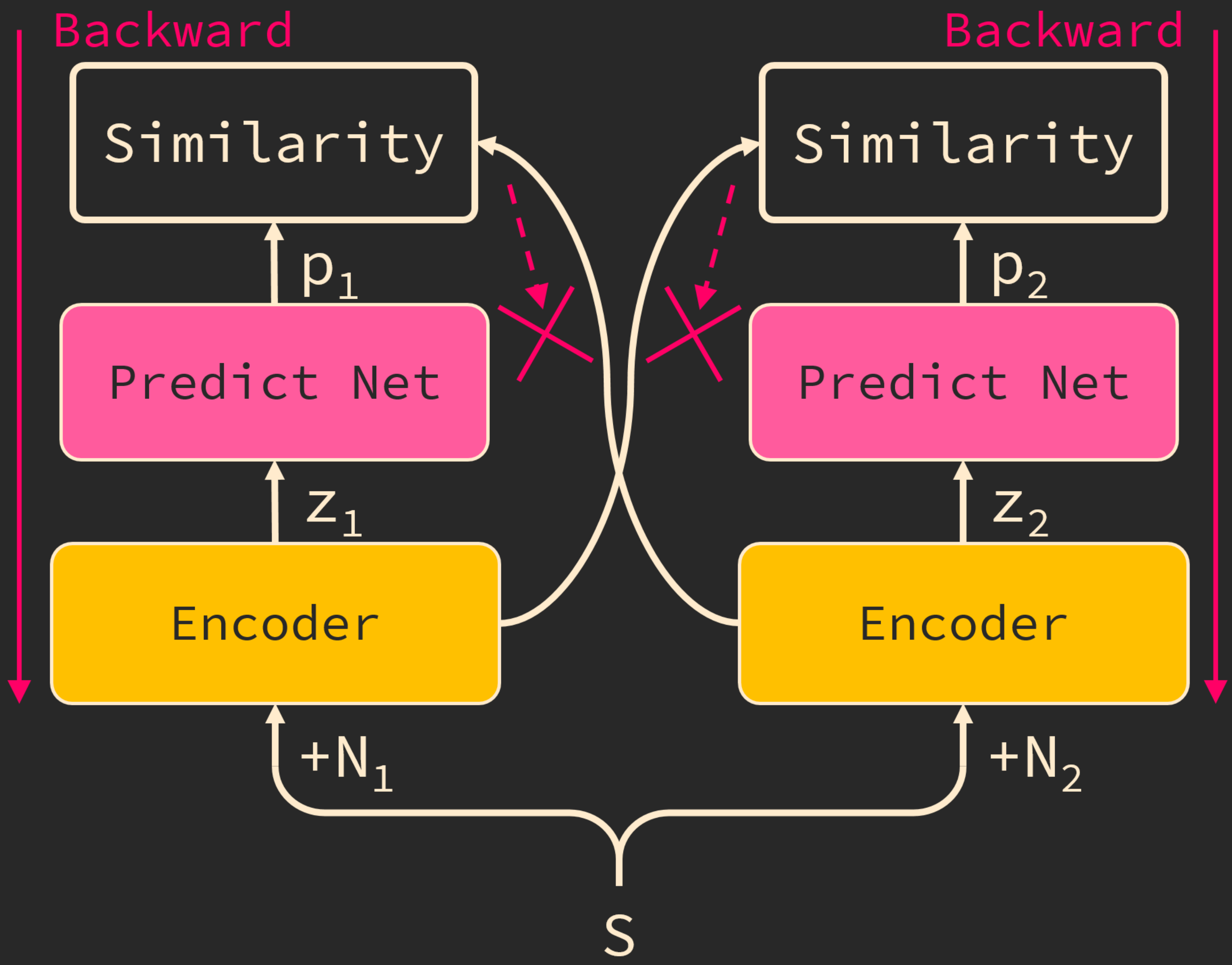


BYOL CL Loss

$$z' = E'(\mathrm{S} + \mathrm{N})$$
$$p = P\left(E(\mathrm{S} + \mathrm{N})\right)$$

$$CL\ Loss$$
$$= -\frac{Sim(p_1, z'_2) + Sim(p_2, z'_1)}{2}$$

SimSiam


Backward
Backward
Similarity
Similarity
p_1
p_2
Predict Net
Predict Net
z_1
z_2
Encoder
Encoder
$+N_1$
$+N_2$
S

SimSiam CL Loss

$$z = E(\mathrm{S} + \mathrm{N})$$
$$p = P(z)$$

$$CL\ Loss$$
$$= -\frac{Sim(p_1, z_2) + Sim(p_2, z_1)}{2}$$

Similarity

$$Sim(\vec{p}, \vec{z}) = \frac{\vec{p} \cdot SG(\vec{z})}{\|\vec{p}\|_2 \|SG(\vec{z})\|_2}$$

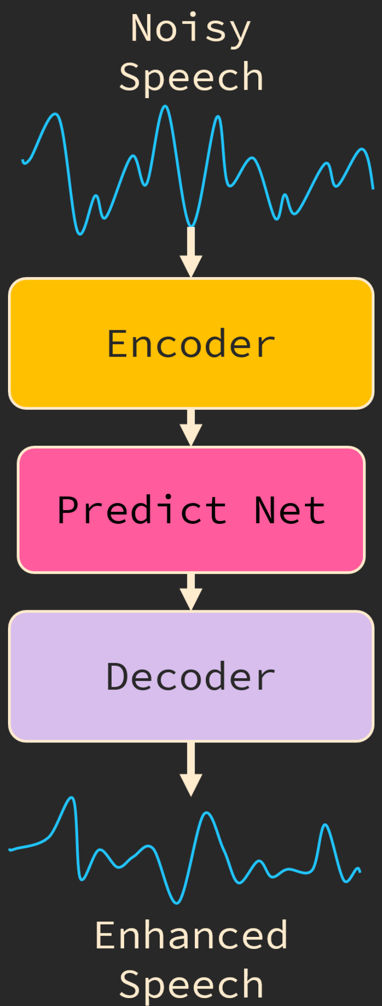
Loss

$$\hat{S} = D(p)$$

$$SE\ Loss = -\frac{SISNR(\hat{S}_1, S_1) + SISNR(\hat{S}_2, S_2)}{2}$$

$$Mix\ Loss = CL\ Loss + 0.1 * SE\ Loss$$

Model



DSB(1,32)
DSB(32,64)
DSB(64,128)
bz,1,(l 64)->bz,64,l
Conv1d(64,128,3)
Concatenate()
Conv1d(256,128,1)
Main Block(128,256,9,16,8)
Main Block(128,256,9,16,8)
Main Block(128,256,9,16,8)
Main Block(128,256,9,16,8)
Encoder

Predict Net
Main Block(128,256,9,16,8)
Main Block(128,256,9,16,8)

Conlv1d(128,64,3)
bz,64,l->bz,1,(l 64)
Decoder

Model

DSB(1,32)
DSB(32,64)
DSB(64,128)

bz,1,(l 64)->bz,64,l
Conv1d(64,128,3)

Concatenate()

Conv1d(256,128,1)

Main Block(128,256,9,16,8)

Main Block(128,256,9,16,8)

Main Block(128,256,9,16,8)

Main Block(128,256,9,16,8)

Main Block(128,256,9,16,8)

Main Block(128,256,9,16,8)

Conlv1d(128,64,3)

bz,64,l->bz,1,(l 64)

Model

Conv1d(C_i,C_o,5,group=g)

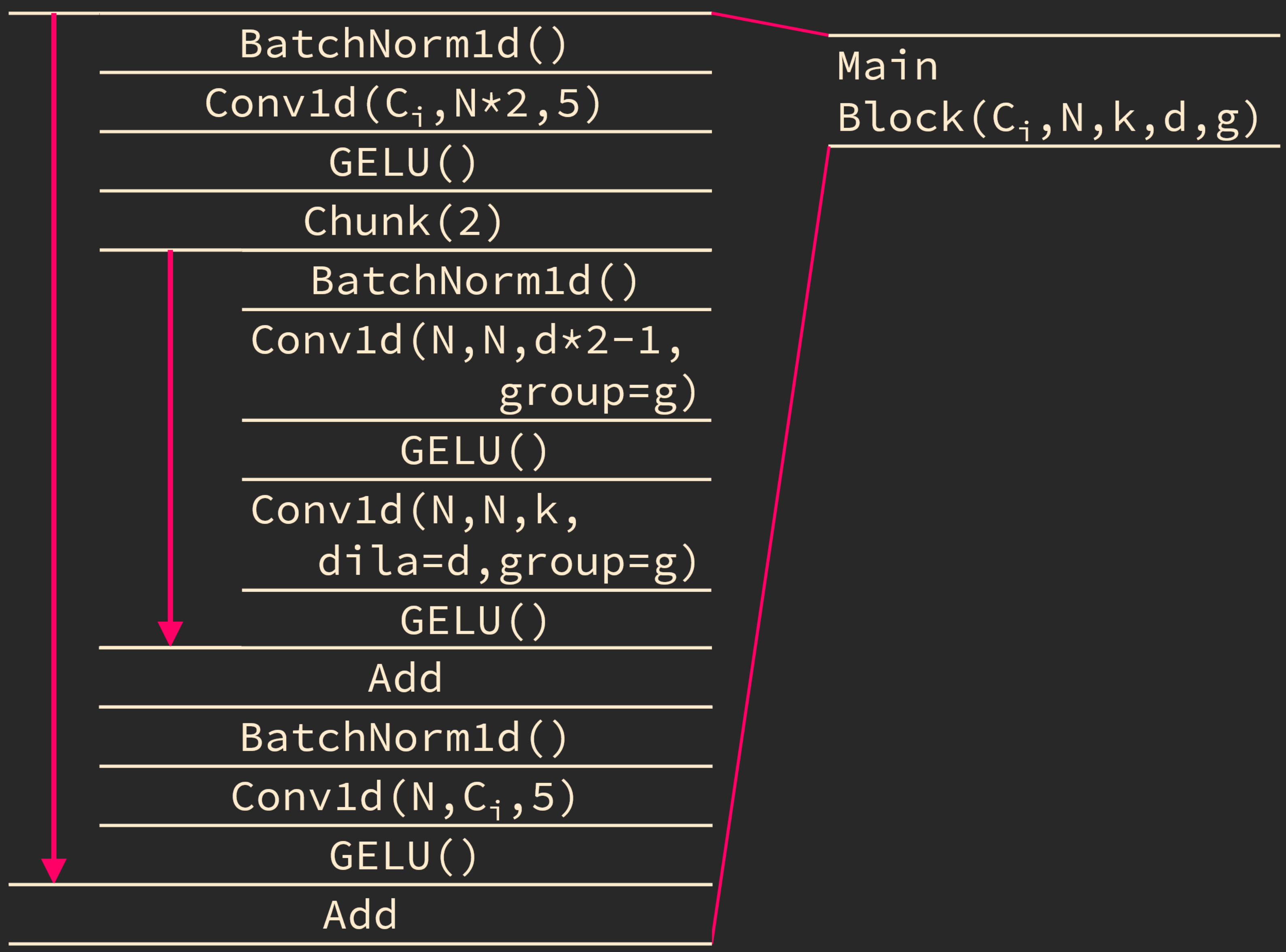
GELU()

Maxpool1d(4,4)

BatchNorm1d()

Down Sample Block(C_i,C_o,g)

Model


BatchNorm1d()
Conv1d(C_i,N*2,5)
GELU()
Chunk(2)
BatchNorm1d()
Conv1d(N,N,d*2-1,
group=g)
GELU()
Conv1d(N,N,k,
dila=d,group=g)
GELU()
Add
BatchNorm1d()
Conv1d(N,C_i,5)
GELU()
Add
Main
Block(C_i,N,k,d,g)

Experiment

	Normal	BYOL	SimSiam
-	使用 SE loss	使用 Mix loss	
Round	每 50 個 epoch 就更換一次 loss (Mix loss 與 SE loss 交替)		
Pretrain	前 50 個 epoch 使用 Mix loss，之後都使用 SE loss		
Round (100 step)	每 100 個 epoch 就更換一次 loss (Mix loss 與 SE loss 交替)		
Few	將 train data 與 test data 交換		

Data

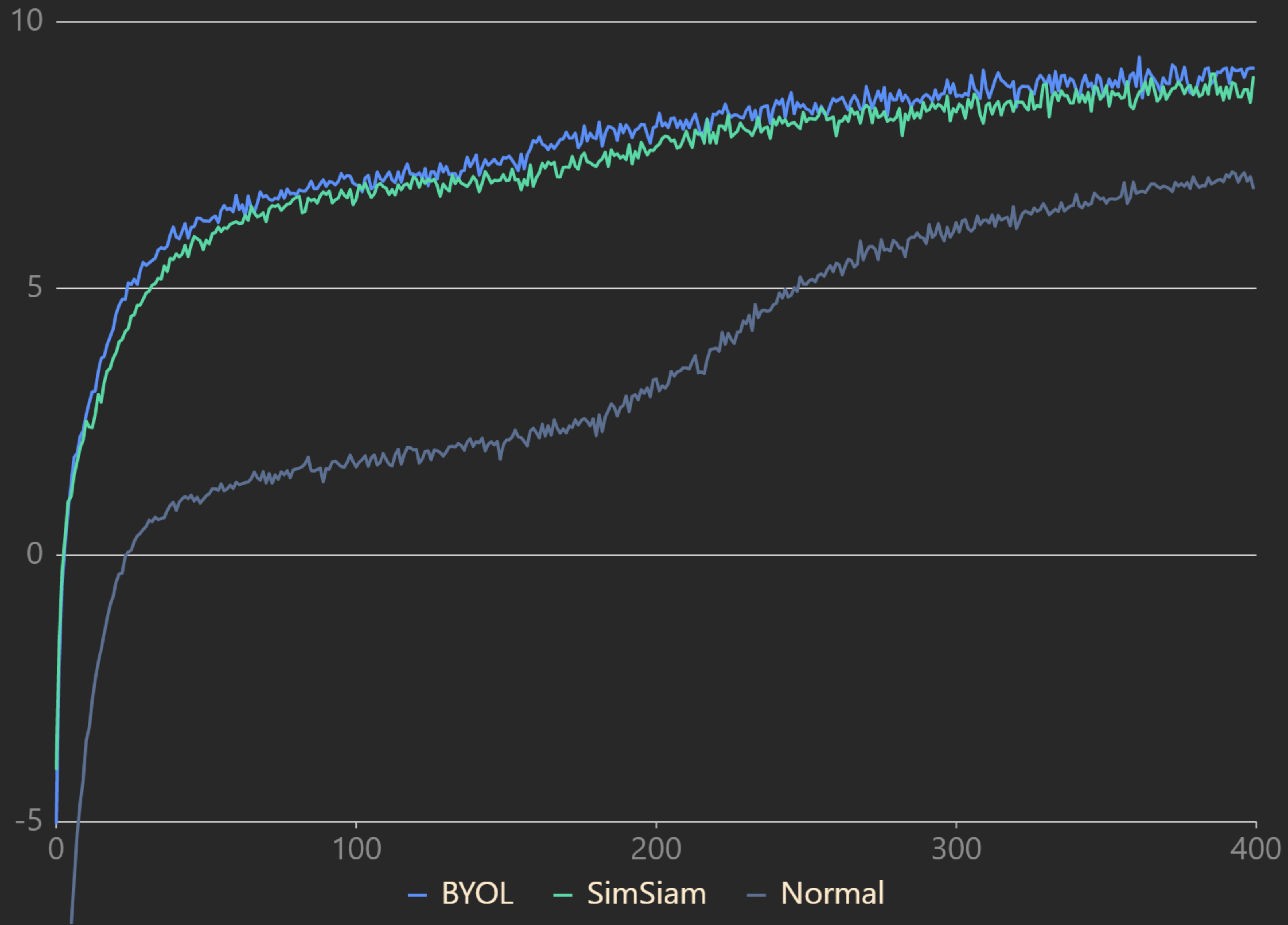
	Train	Test
Speech	TIMIT(4120)	TIMIT(500)
Noise	Nonspeech(75)	Nonspeech(25)
SNR(dB)	-10, -5, 0, 5, 10	-7.5, -2.5, 2.5, 7.5

Hyperparameter

Optimizer:SGD	lr	momentum	weight decay
	0.05	0.9	0.0001
Batch Size	N_1+N_2 = 128+128		
BYOL τ	0.99		

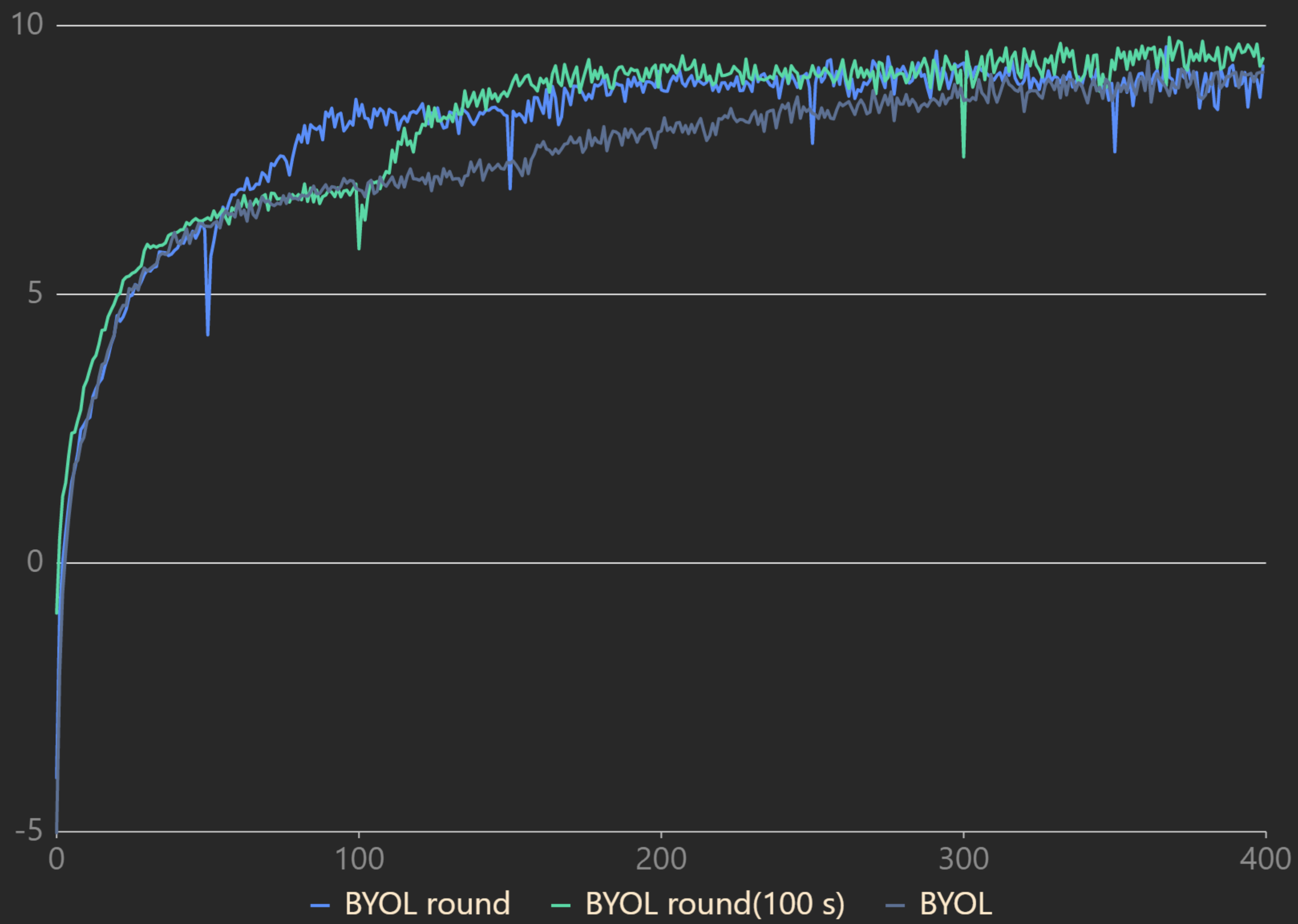
Experiment

CL vs Normal

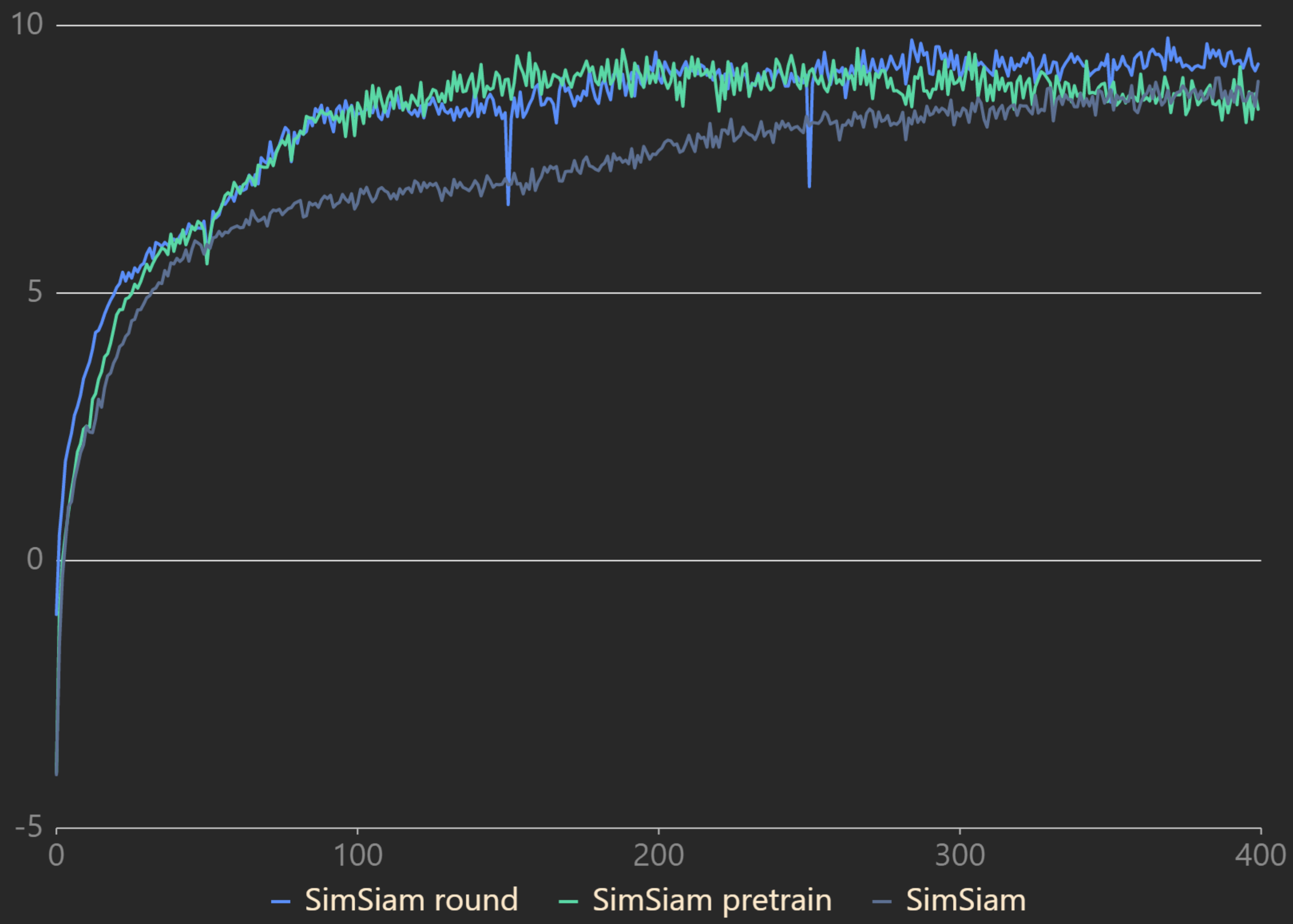


Experiment

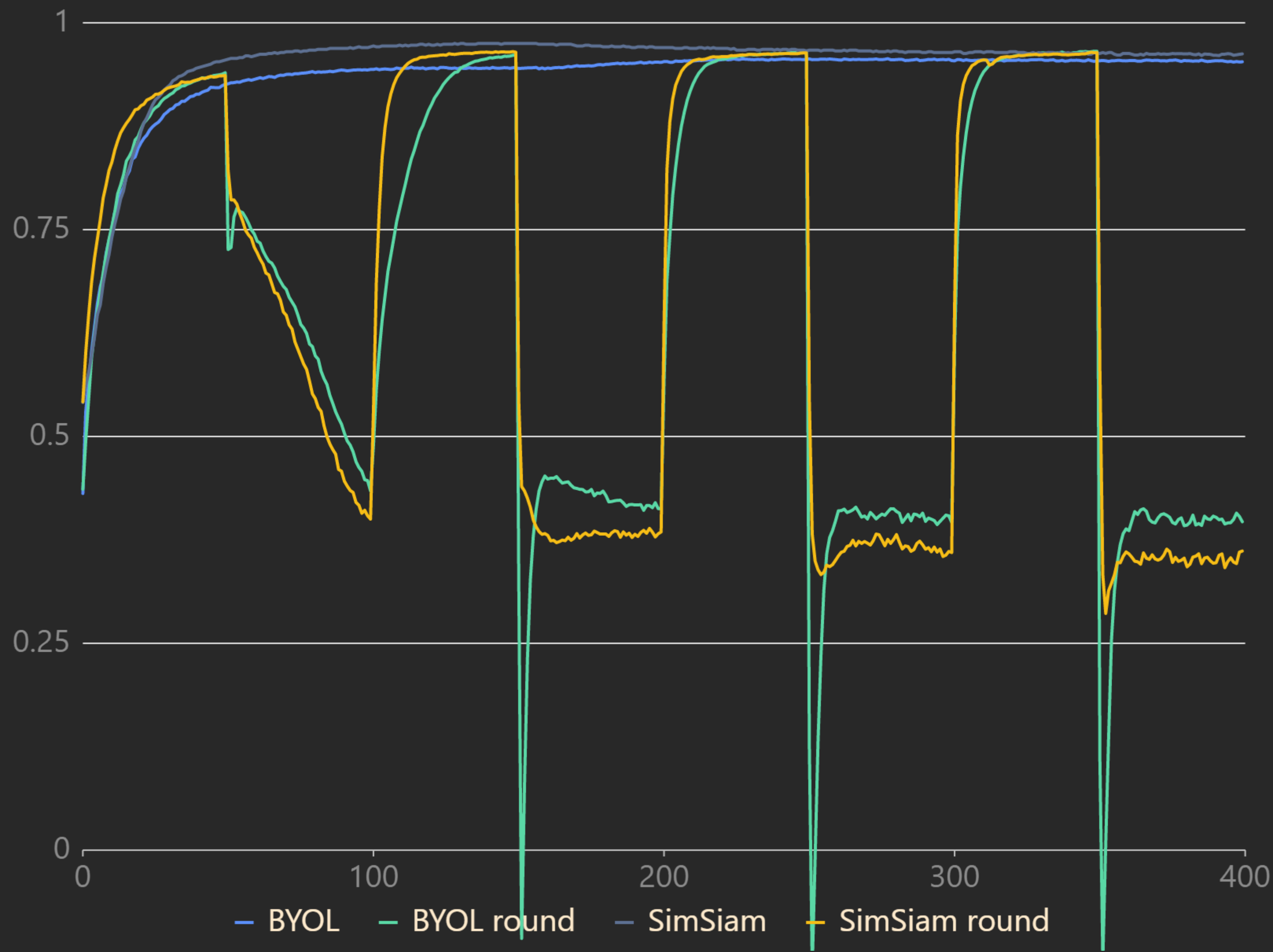
BYOL



SimSiam


10
5
0
-5
0
100
200
300
400
SimSiam round
SimSiam pretrain
SimSiam

Train Similarity



Evaluation Metrics

Model	Evaluation Metrics		
	PESQ	STOI	SI-SNR
Noisy	1.813	0.764	0.001
Normal	2.273	0.814	7.146
BYOL	2.392	0.844	9.174
BYOL round	2.461	0.858	9.378
BYOL round(100 s)	2.474	0.861	9.526
SimSiam	2.374	0.84	8.884
SimSiam round	2.472	0.861	9.529

PESQ

Model	SNR: -7.5	-2.5	2.5	7.5
	PESQ			
Noisy	1.337	1.644	1.971	2.3
Normal	1.826	2.138	2.438	2.688
BYOL	1.875	2.253	2.59	2.851
BYOL round	1.904	2.3	2.671	2.97
BYOL round(100 s)	1.913	2.308	2.683	2.991
SimSiam	1.873	2.24	2.563	2.82
SimSiam round	1.937	2.317	2.672	2.962

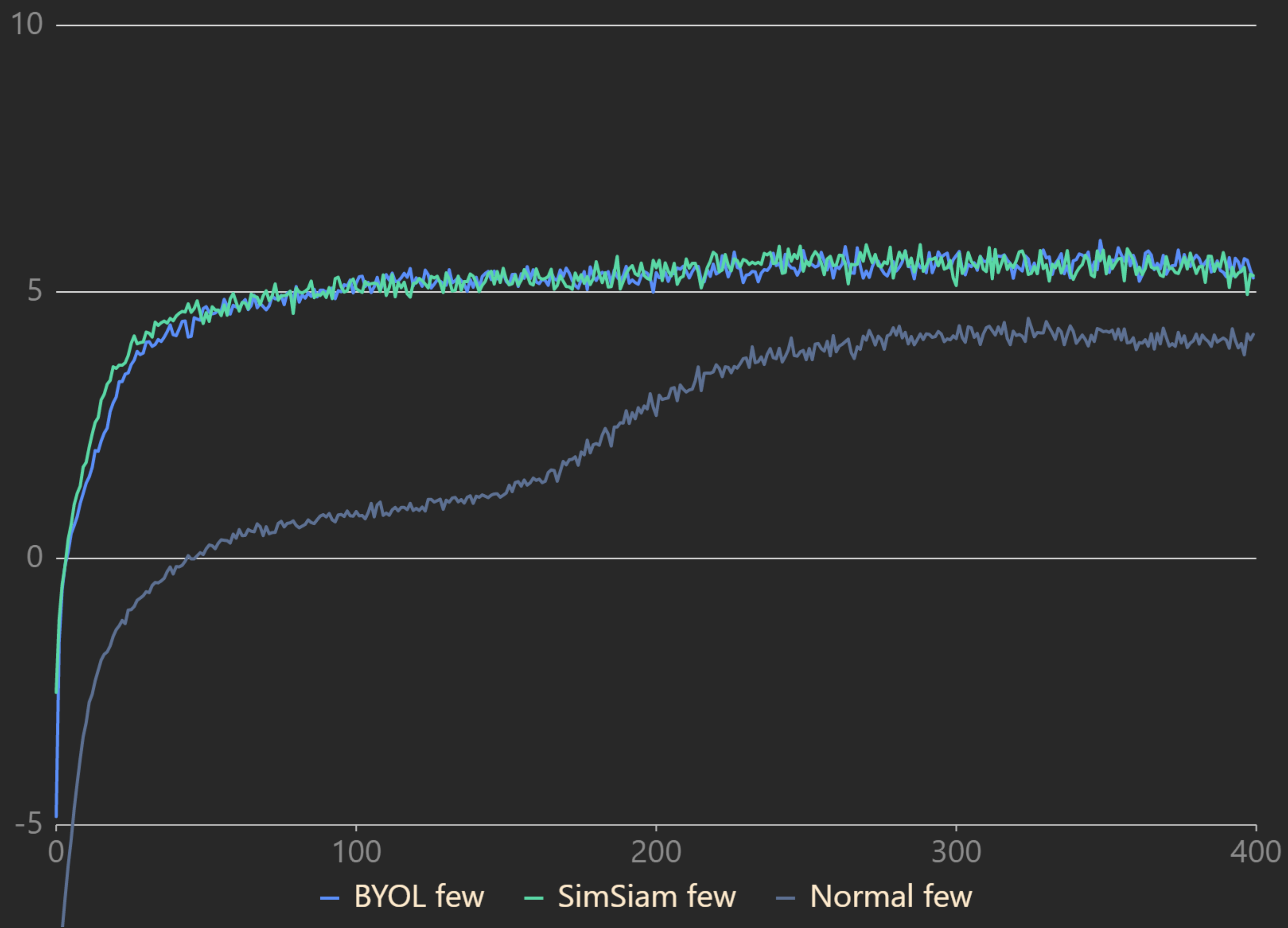
STOI

Model	SNR: -7.5	-2.5	2.5	7.5
	STOI			
Noisy	0.643	0.728	0.809	0.878
Normal	0.702	0.793	0.859	0.904
BYOL	0.734	0.826	0.889	0.928
BYOL round	0.746	0.841	0.904	0.942
BYOL round(100 s)	0.75	0.844	0.906	0.944
SimSiam	0.729	0.822	0.885	0.926
SimSiam round	0.753	0.845	0.905	0.942

SI-SNR

Model	SNR: -7.5	-2.5	2.5	7.5
	SI-SNR			
Noisy	-7.497	-2.498	2.503	7.498
Normal	2.611	6.065	8.972	10.935
BYOL	3.677	7.785	11.281	13.951
BYOL round	3.396	7.772	11.615	14.728
BYOL round(100 s)	3.457	7.859	11.784	15.004
SimSiam	3.544	7.508	10.913	13.572
SimSiam round	3.583	7.935	11.751	14.847

Few Data



Conclusion

- 在訓練前期利用 CL Loss 對中間特徵進行約束能夠加速模型收斂。
- 中後期使用 CL Loss 會降低模型的收斂速度與效能。
- 使用 CL Loss 能夠抑制 Overfitting 的問題。
- 與 SimSiam 相比，BYOL 的 CL Loss 需要更長一點的時間收斂。

Todo

- 測試不同比例混和的 Mix Loss 效果。
- 使用複數的噪音跟語音混和進行訓練。
- 研究 Mix Loss 的自適應混合權重。
- 區分噪音種類進行訓練。

Reference

- Bootstrap your own latent: A new approach to self-supervised learning. CoRR, abs/2006.07733, 2020.
- Exploring simple siamese representation learning. CoRR, abs/2011.10566, 2020.