

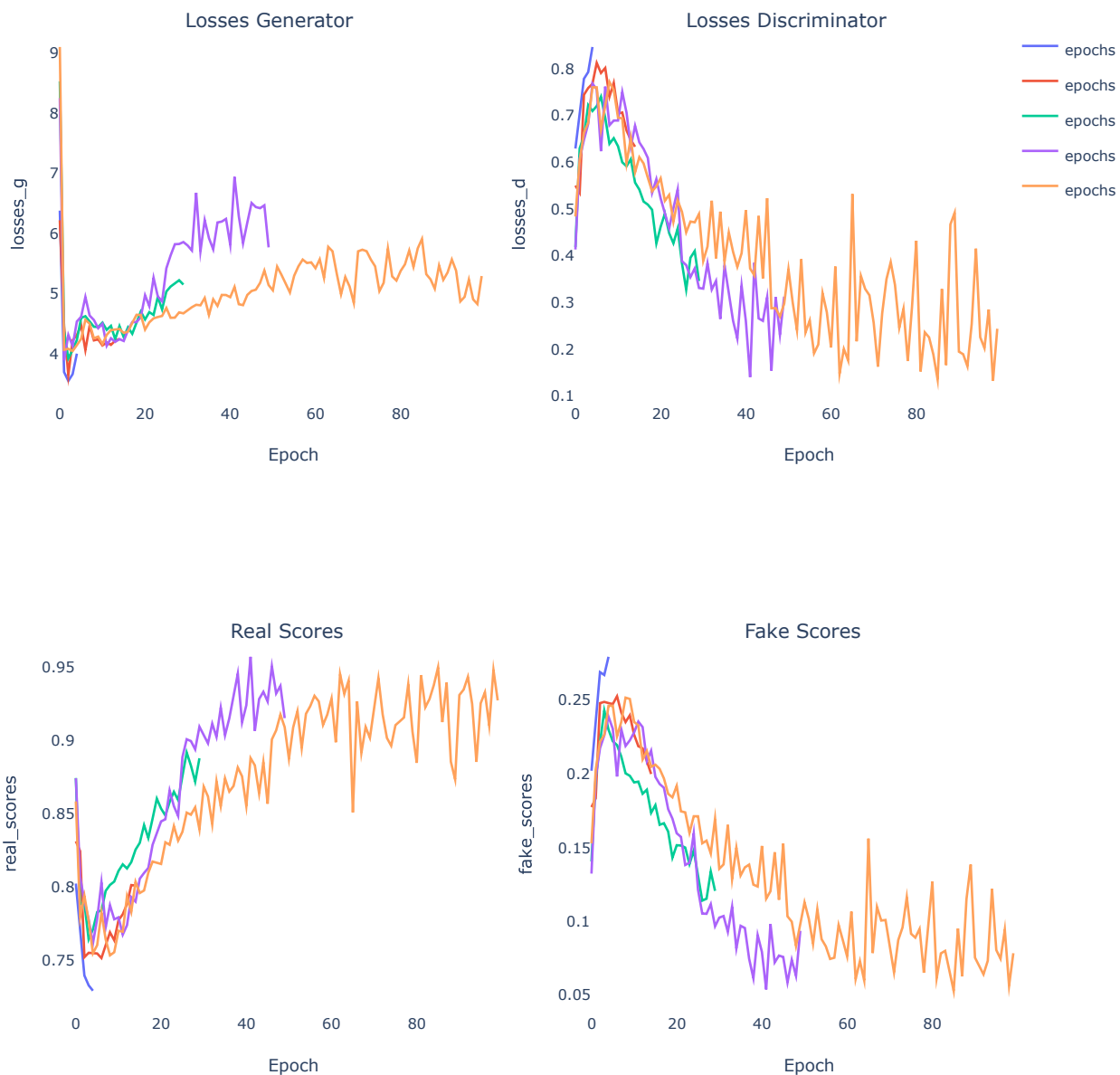
```
[7]: # Nauka jest niestabilna, nawet 0.1 real score różnicy po ~40 epokach chociaż dąży do tego samego
# 2x więcej epok -> 2x więcej czasu
plot(df, 'test_epochs', 'epochs')
#plot_png(df, 'test_epochs', 'epochs')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

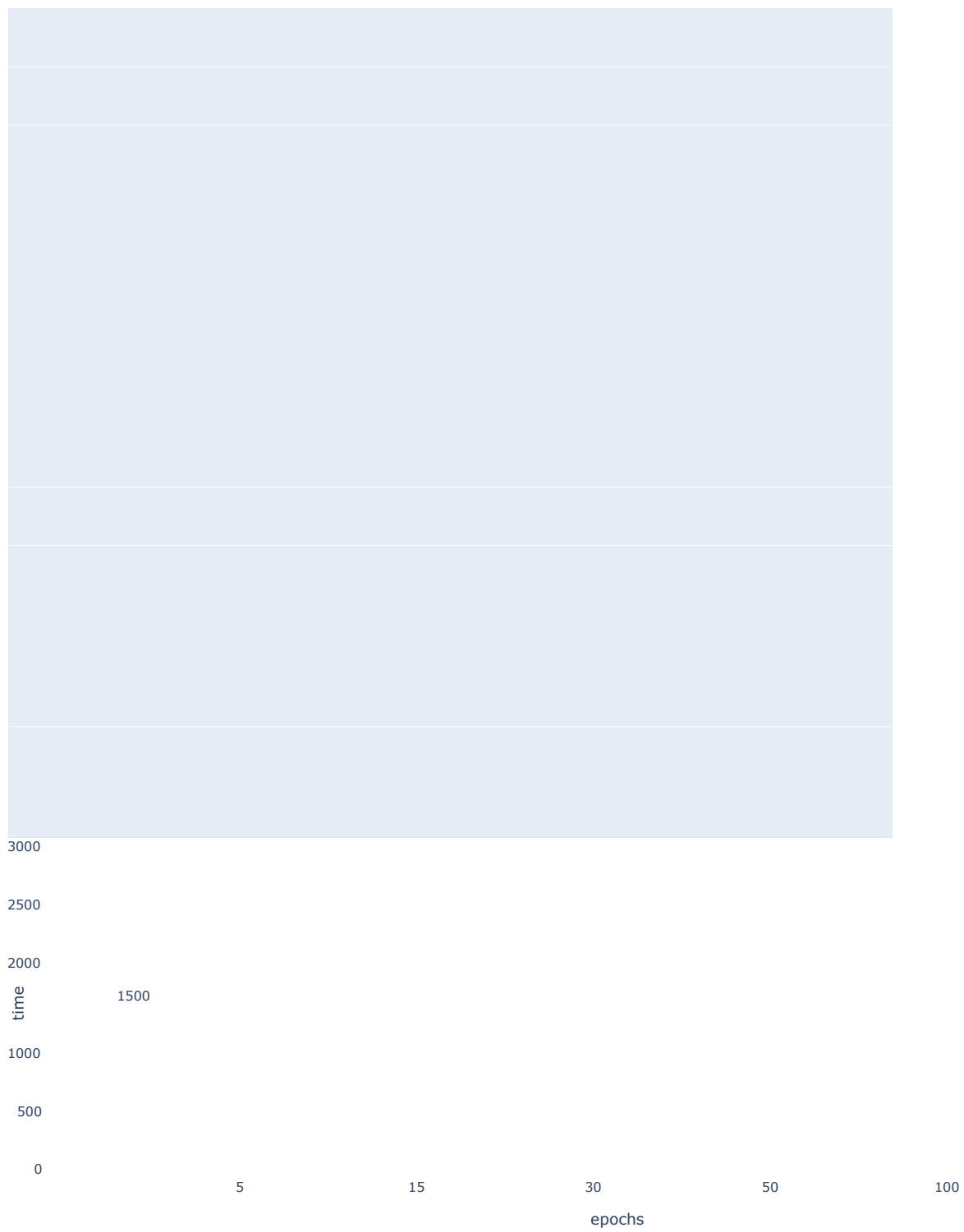
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_epochs



Time epochs



```
[8]: # lr nie ma wpływu na czas
# najwyższy fake score przy najniższym real score jednak w analizie FID okazuje się, że daje to najlepsze obrazki
plot(df, 'test_lr', 'lr')
#plot_png(df, 'test_lr', 'lr')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

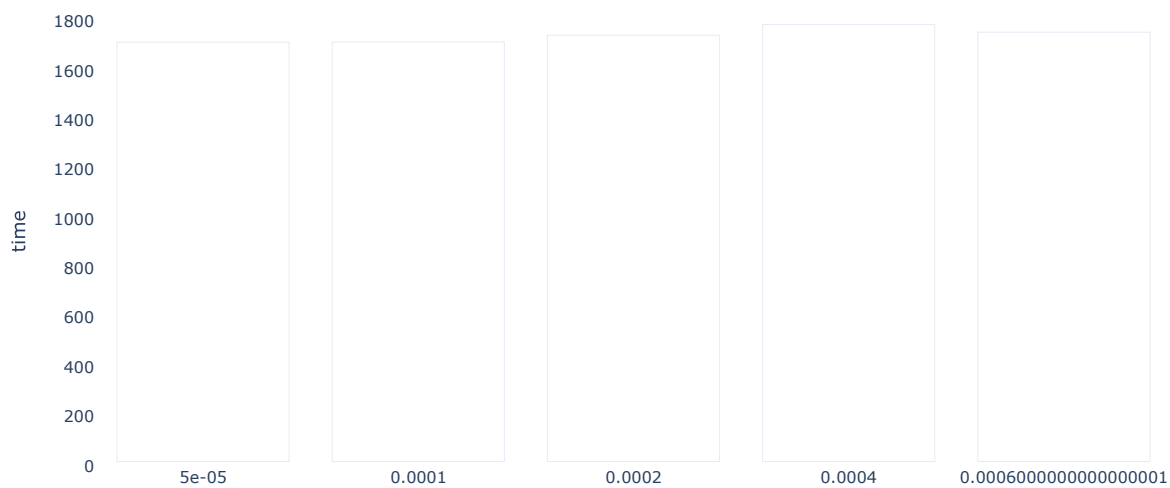
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_lr



Time Ir



Ir

```
[9]: # mały batch size -> overfitting?
# z drugiej strony najładniejsze zdjęcia właśnie z batch_size 32
# czasowo batch_size tym większy tym lepiej, ale nie można za dużego bo wtedy bez sensu ogromny batch size z małą ilością
zdf.plot(df, 'test_batch', 'batch_size')
#plot_png(df, 'test_batch', 'batch_size')
```

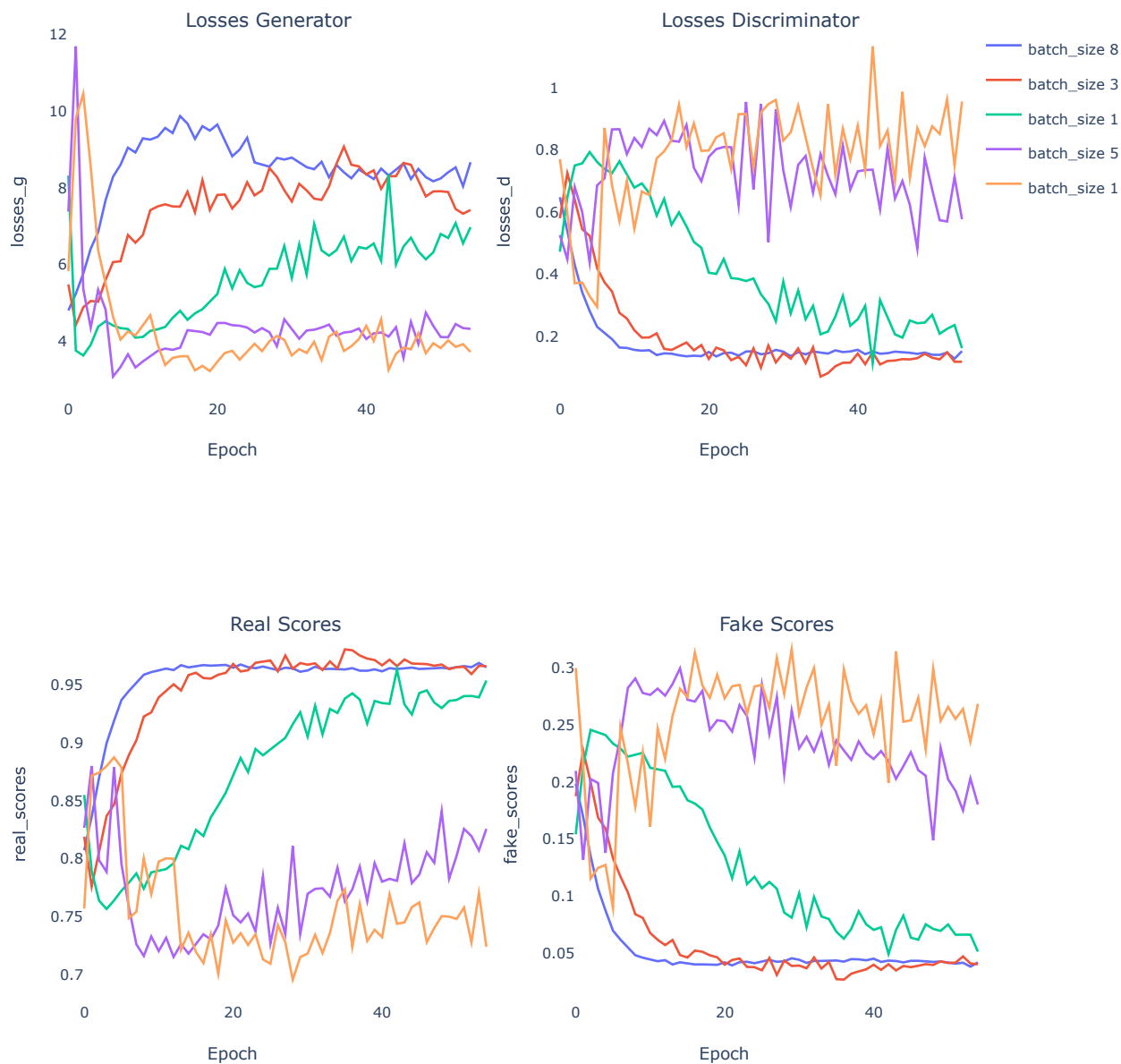
C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

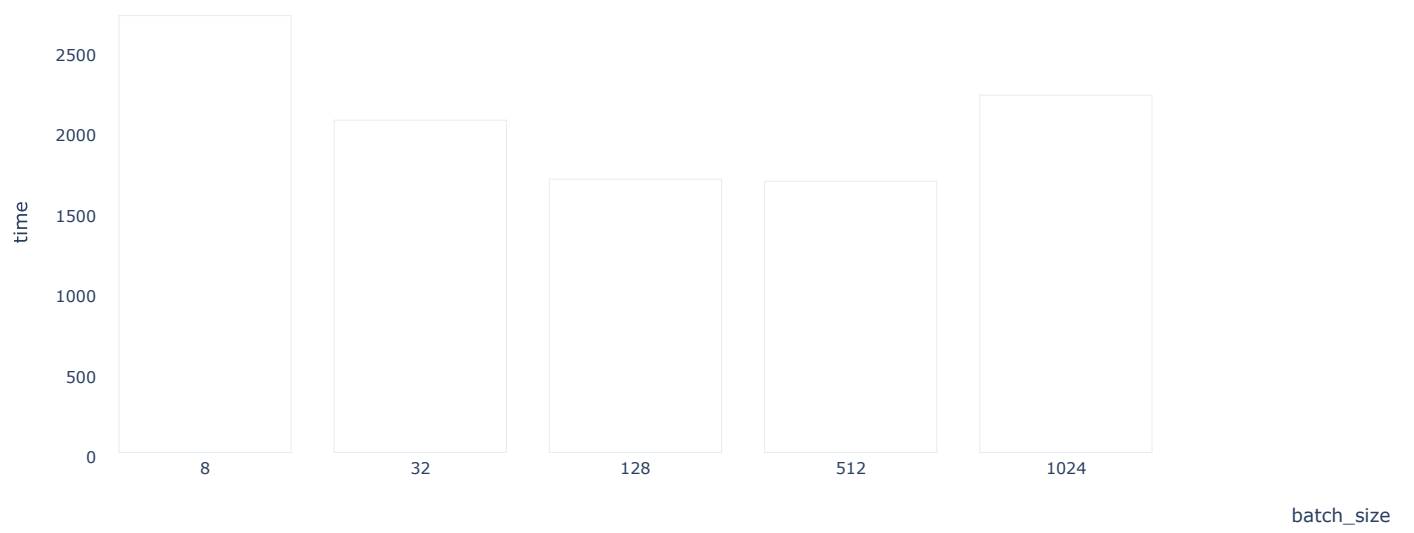
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_batch



Time batch_size



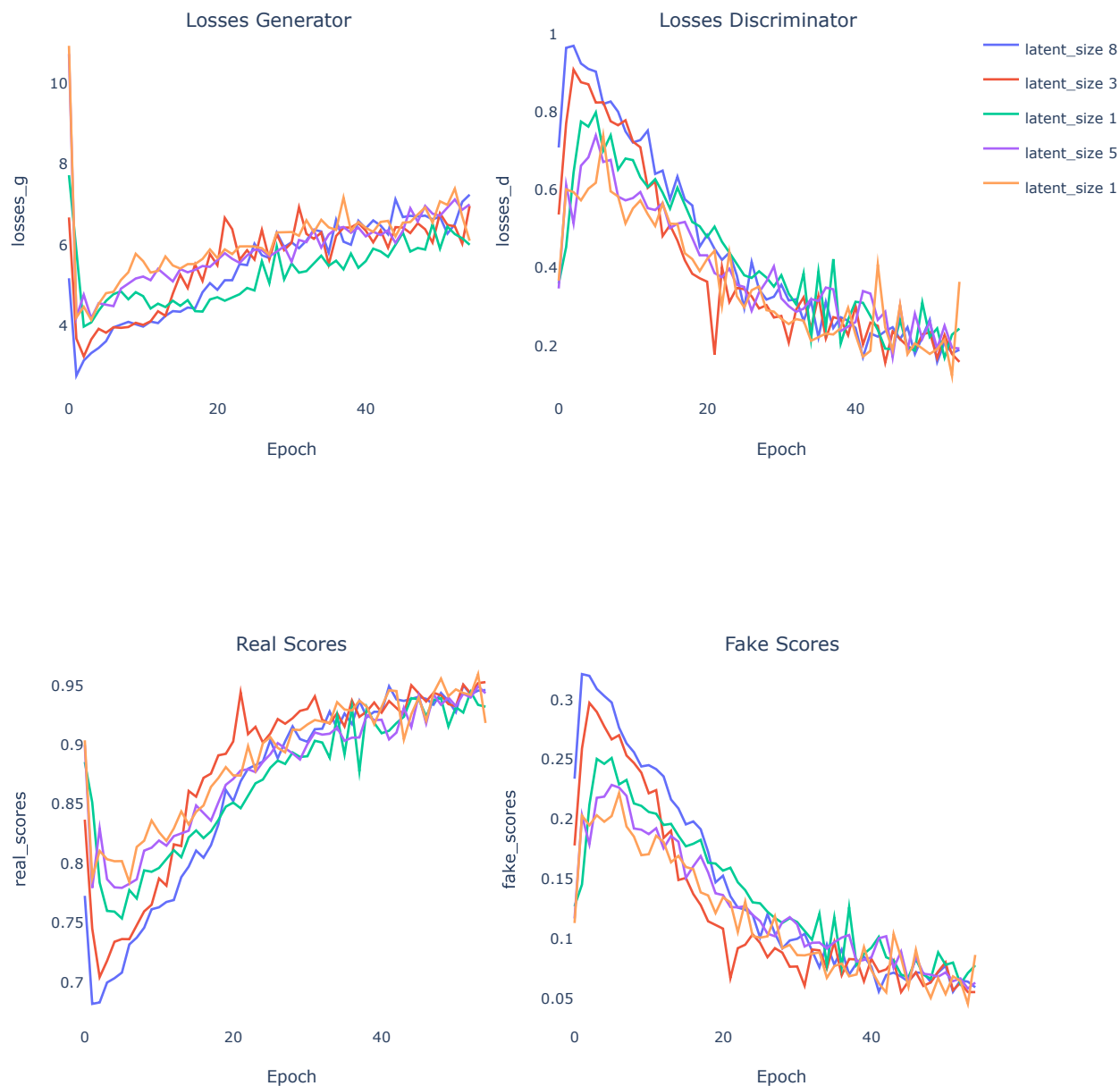
```
[10]: # wtedy latent size widać tylko w początkowych epokach
# niewielki wpływ na czas
plot(df, 'test_latent', 'latent_size')
#plot_png(df, 'test_latent', 'latent_size')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

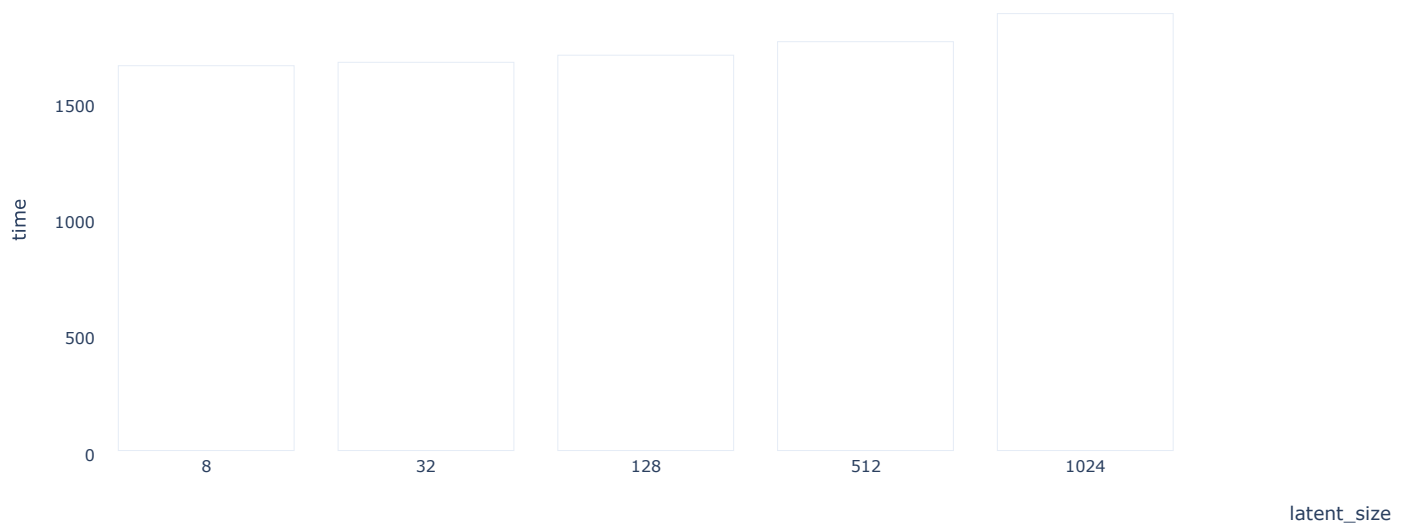
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_latent



Time latent_size




```
[11]: # NADAM ma lepiej zbiegać, trudno powiedzieć
# ADAMW overfitting, trudno powiedzieć
# RADAM stabilne wczesne uczenie, trudno powiedzieć
plot(df, 'test_optimizer', 'optimizers')
#plot_png(df, 'test_optimizer', 'optimizers')
```

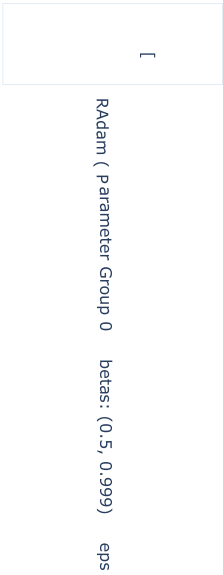
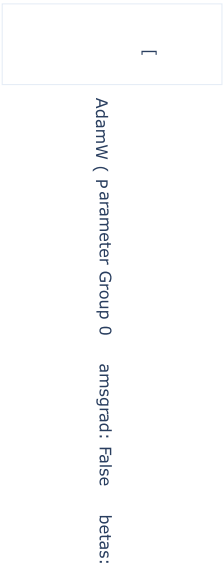
C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_optimizer





Time optimizers

time

1500
1000
500
0

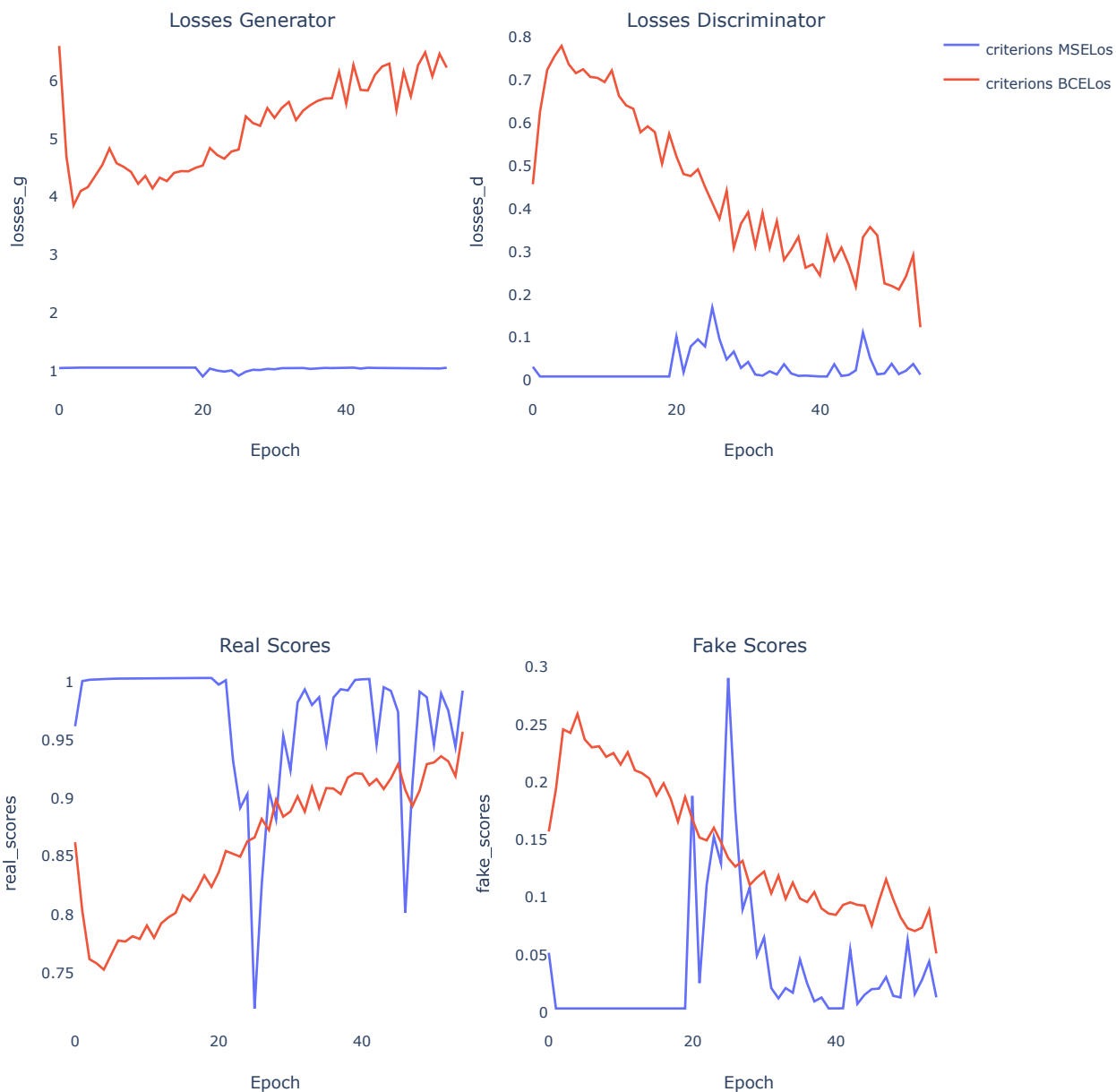
```
[12]: # samo zmienie funkcji kosztu nie pomaga
plot(df, 'test_loss', 'criteria')
#plot_png(df, 'test_loss', 'criteria')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

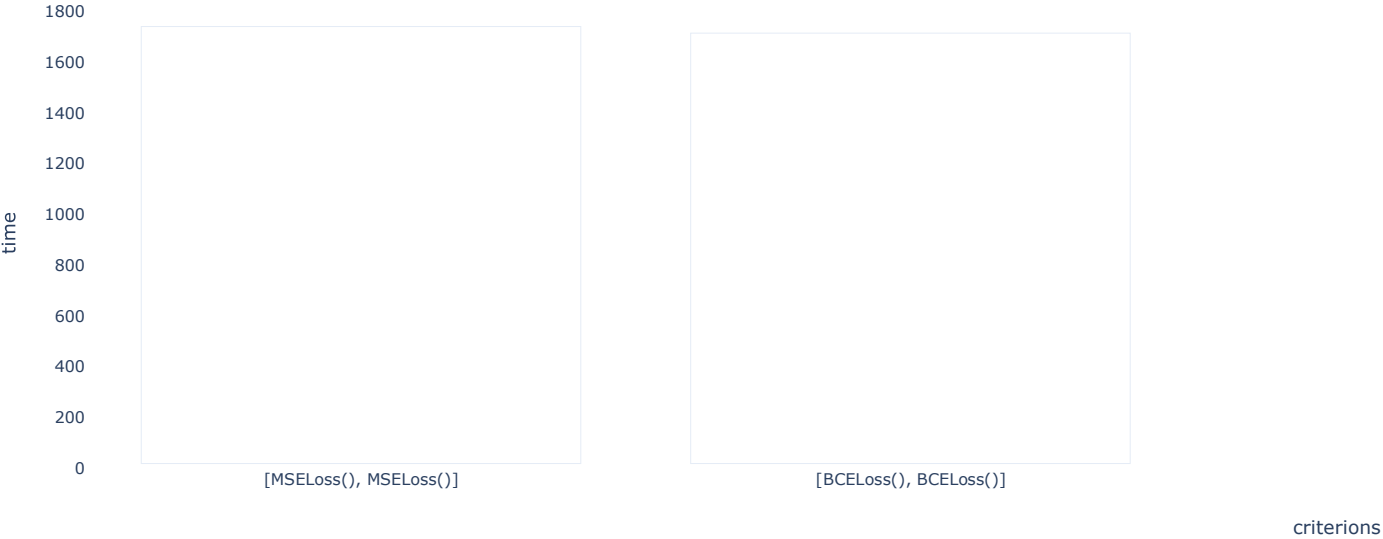
A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_loss



Time criterions



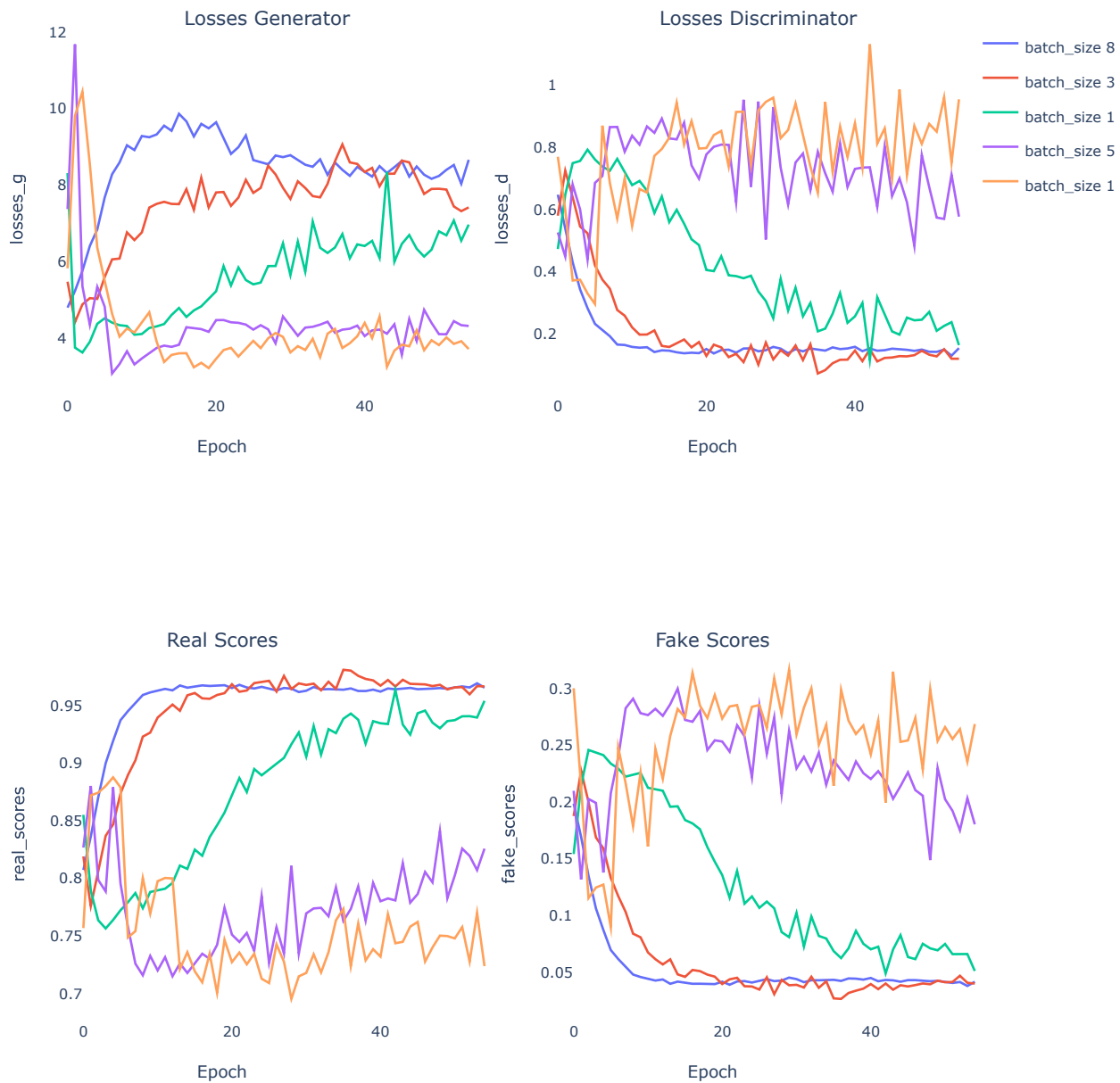
```
[13]: # za duży batch size nie fituje, za mały mała różnorodność
# tym większy batch tym szybciej uczy. Nagły skok 1024 tłumaczę, że robił duży batch na mały ogonek który mu został
plot(df, 'test_batch', 'batch_size')
#plot_png(df, 'test_batch', 'batch_size')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

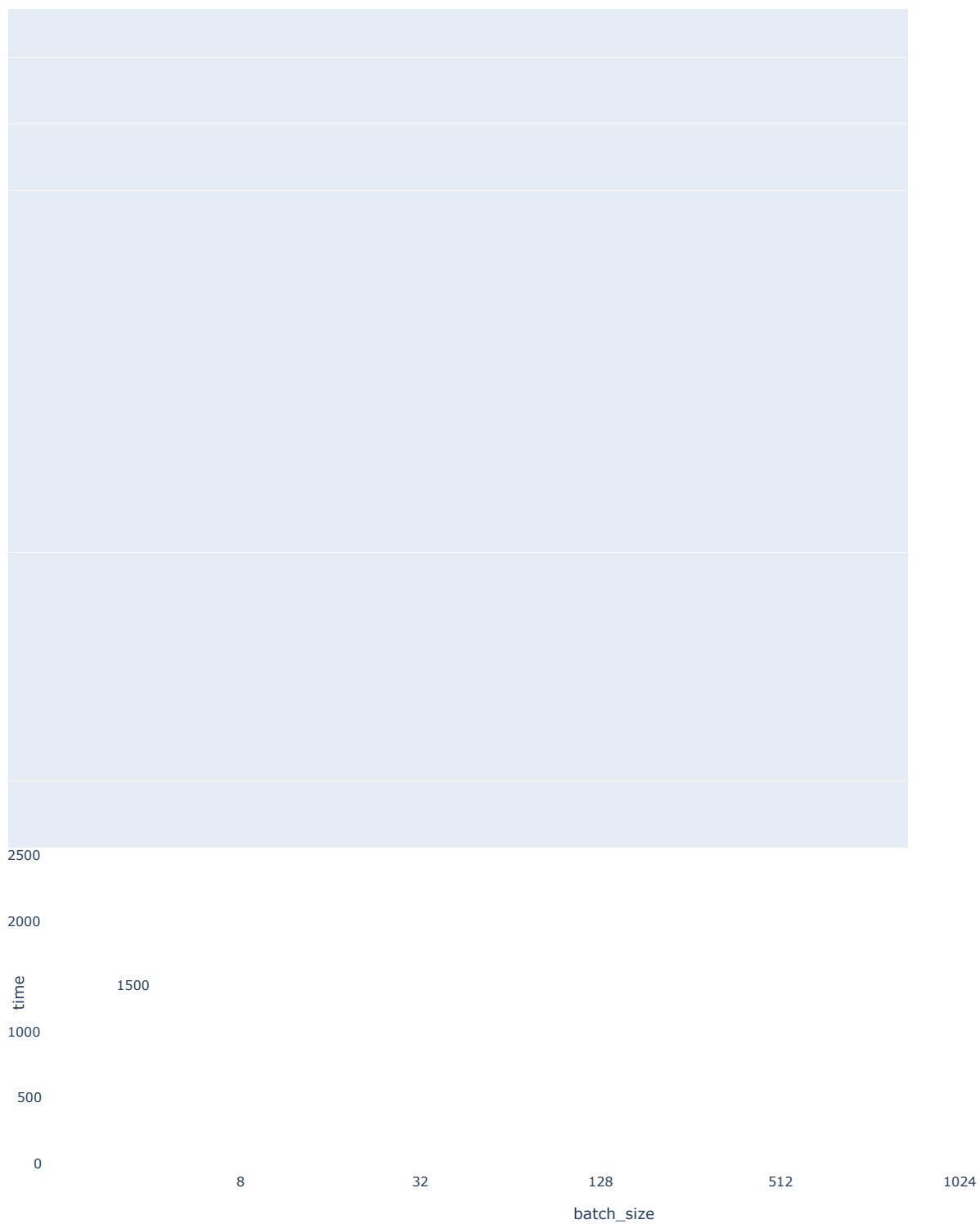
A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_batch



Time batch_size



```
[14]: # Lsgan szybsza architektura,
# nie nromowanie wag poprawia (?) uczenie. Myślę tak bo zazwyczaj poprawiwa się dyskryminator i dlatego generator ma gorsze
# wyniki i to tu widać
plot(df, 'test_models', 'model_name', ['MSE lsgan norm', 'MSE lsgan no norm',
'MSE basic norm', 'MSE basic no norm',
'BCELOSS basic norm', 'BCELOSS basic no norm'])
#plot_png(df, 'test_loss', 'criteria')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

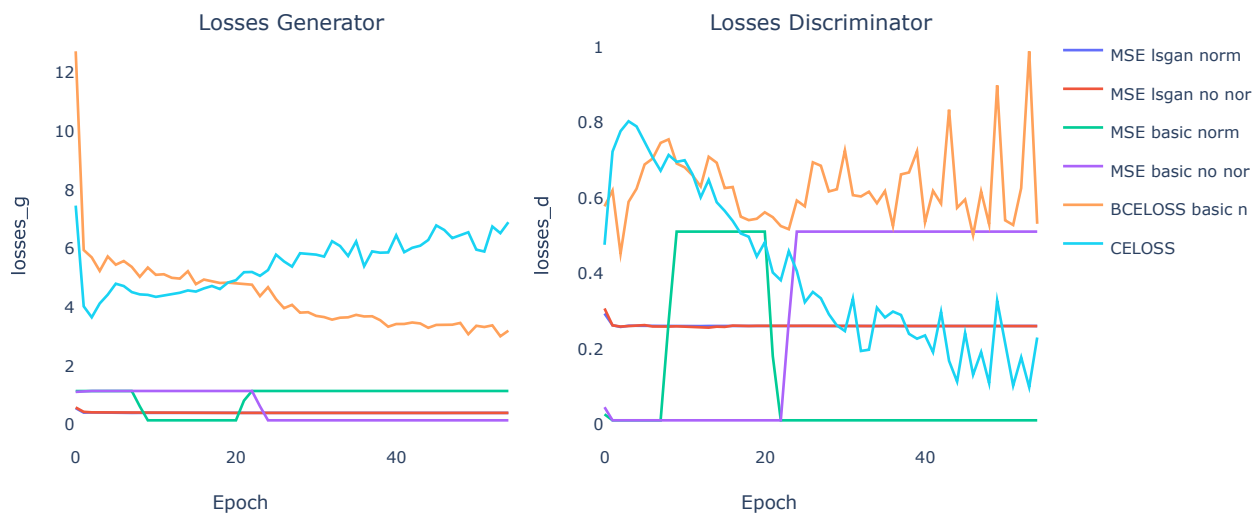
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

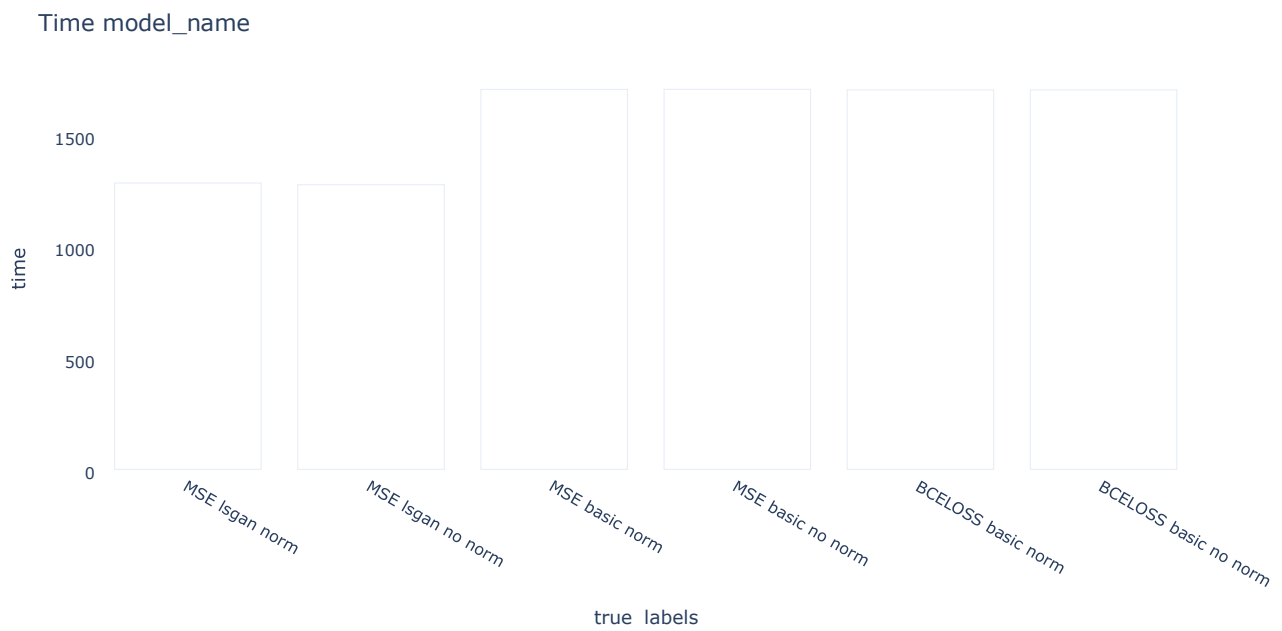
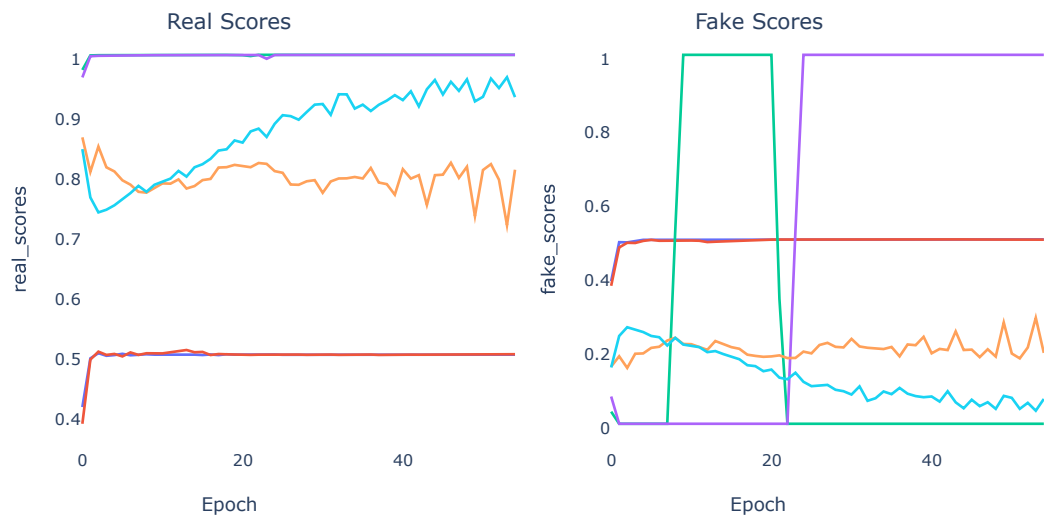
C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:65: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_models






```
[16]: # 0.5 leakrealu bardziej roztrzęsione? #
      należy testować na nieobciążonym komputerze
      plot(df, 'test_arch_trash', 'model_name', ['G:R, D:LR 0.2', 'G:R, D:R', 'G:R, D:LR 0.5',
      'G:LR 0.2, D:LR 0.2', 'G:LR 0.2, D:R', 'G:LR 0.2, D:LR 0.5',
      'G:LR 0.5, D:LR 0.2', 'G:LR 0.5, D:R', 'G:LR 0.5, D:LR 0.5'])
      #plot_png(df, 'test_loss', 'criteria')
```

C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:20: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

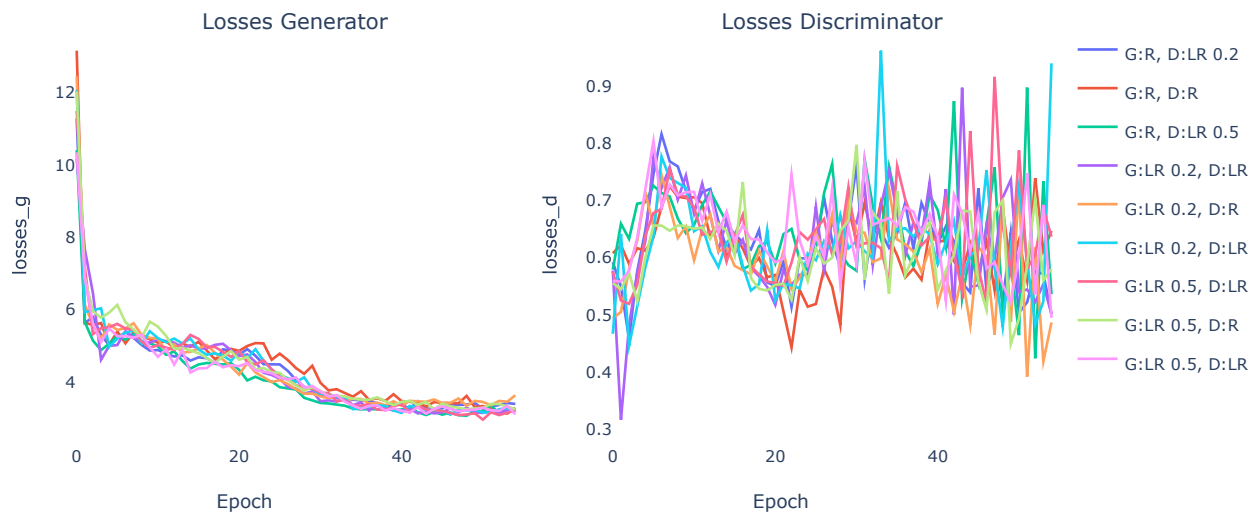
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

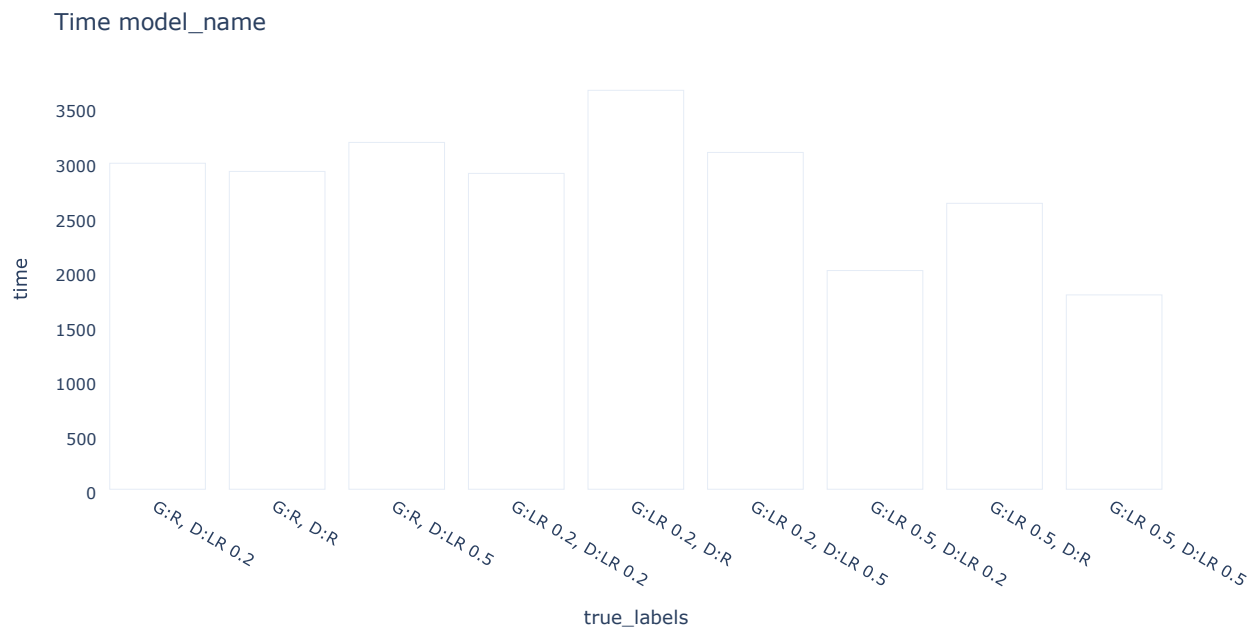
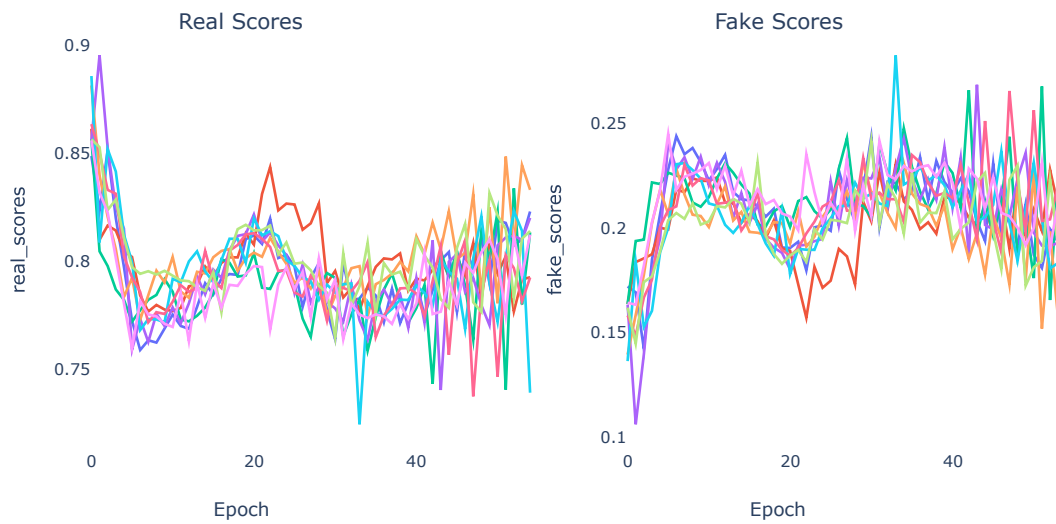
C:\Users\karol\AppData\Local\Temp\ipykernel_17256\857538823.py:65: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Exp Title: test_arch_trash





In []:

Najmniejsze wartości bezwzględne FID:
 lrs4: 7972.9016168492335
 models2: 8189.5090676333075
 models3: 8328.41189501782
 models0: 1.587665380337061e+57
 epochs1: 5.492464018463346e+57

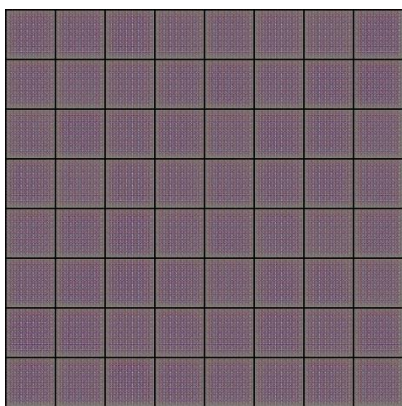
Największe wartości bezwzględne FID:
 arch8: -4.204442255154379e+93
 lrs3: 5.957012032801552e+83
 epochs2: -1.6905645028648165e+79
 batches0: -8.105446246612134e+77
 epochs0: 1.5378636851831057e+77

Podane są indeksy eksperymentó

LRS4 wygląda ładnie tak samo jak wiele innych ocenionych tragicznie przez FID

MODELS2 i MODELS 3 wyglądają mniej więcej tak jak na następnej stronie. Może to być błąd w kodzie?

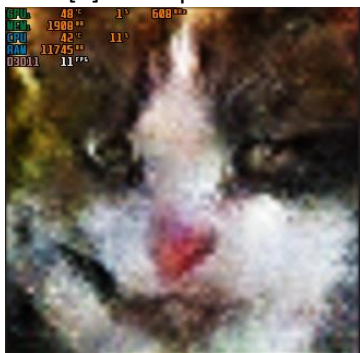
To jest naprawdę taka rozkład?



Obraz, gdy szum wejściowy same zera:

Następuje coś ciekawego, bo gdy zmienia się jeden piksel to dla dodanych liczb 'wyłania się jedno zdjęcie kota, a dla ujemnych inne zdjęcie kota np. (jedenk piksle odpowiada za dwa zdjęcia kota?)

szum[0] >> 0 np. 0.5 lub 100000



Szum[0] << 0 np. -0.5 lub -100000

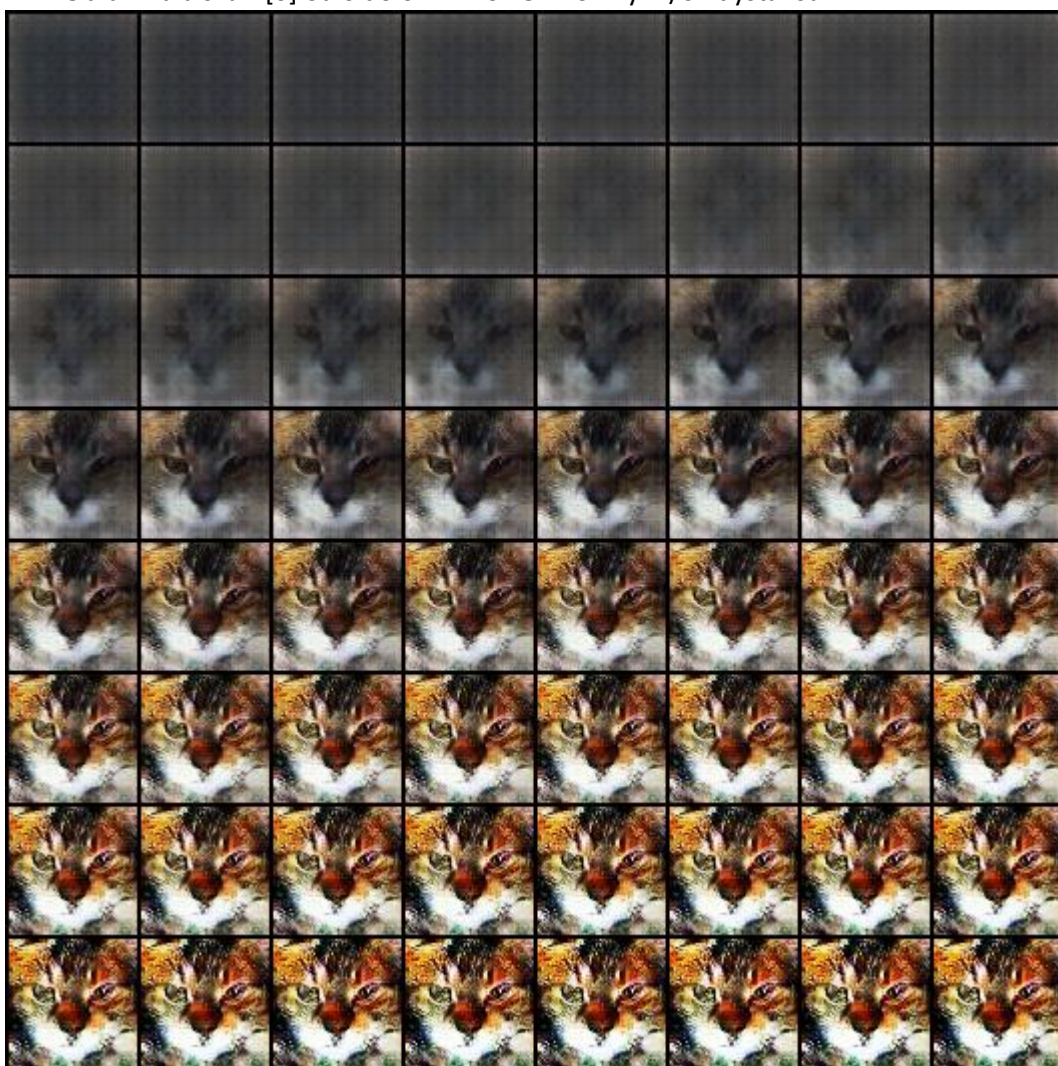


Następuje 'wyłanianie', bo np. dla bardzo bliskich 0 wartości mamy kota (np. szum[0] – 0.0007)

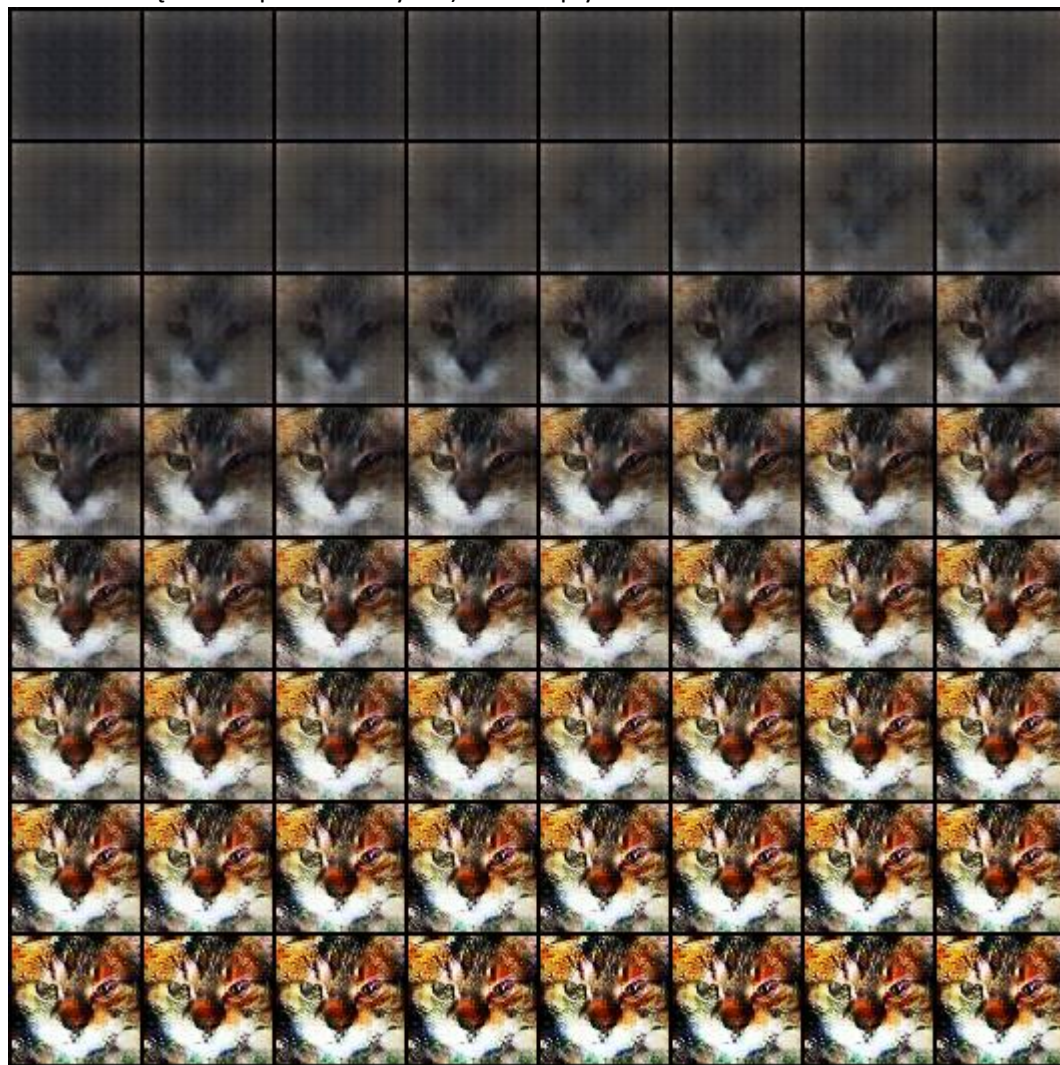


Chciałem zrobić badanie jak kot się pojawia i znalazłem coś dziwnego nie wiem czy to błąd w kodzie, ale nie znalazłem takiego

Obrazki dla szum[0] od 0 do 0.2 z krokiem równym /64 dystansu:



Natomiast dla dla szum[0] od 0 do 100000 z krokiem równym /64 dystansu mamy bardzo podobny obrazek!!!
Spodziewałbym się tylko 1 kwadratu niewyraźnego, a mamy praktycznie ten sam obrazek (czasami różnice są delikatne ale zawsze są bardzo podobne wyniki) Batch wpływa?



Batch normalization to jest odpowiedź niestety:<

