פרויקט בראייה ממוחשבת

מגישים: אביב אסטה יניב גלבר

מטרת הפרויקט: להראות כיצד OT ליכול לעזור ללמידות.

בפרוייקט חילקנו את התכונות שיפור לשני אפשרויות משמעויתיות:

יכולות פסיביות – כאלו העוזרות למדידה של איכות למידה.

יכולות אקטיביות – כאלו ההעוזרות לשפר והאיץ למידה.

יכולות פסיביות:

נרצה להראות כי OT הוא מדד היכול לתאר loss למודל גנרטיבי.

לשם השוואה, נבדוק בניסויים הנ"ל עבור מדד FID הידוע כי יכול לתאר loss (אך כבד חישובית ולכן לא משומש כ-loss).

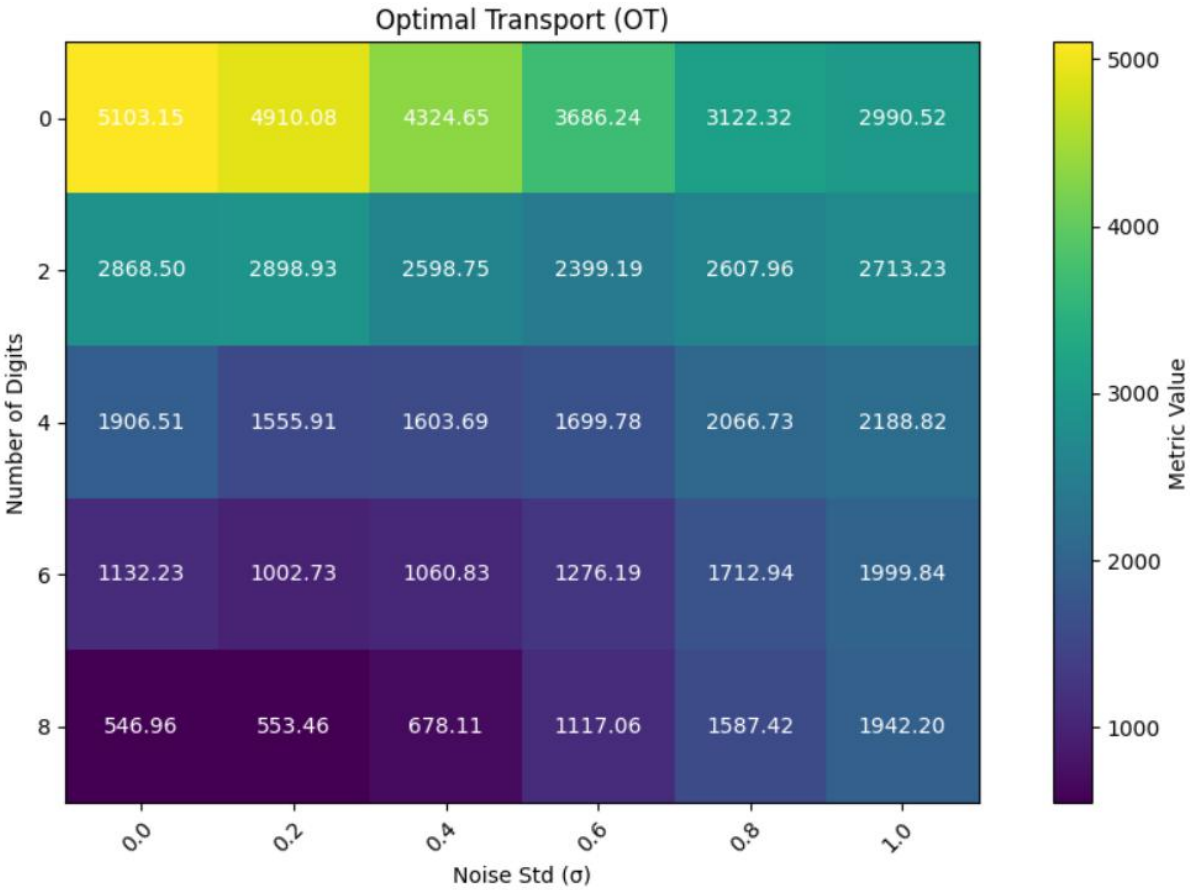
ניסוי 1 – precision/recall:

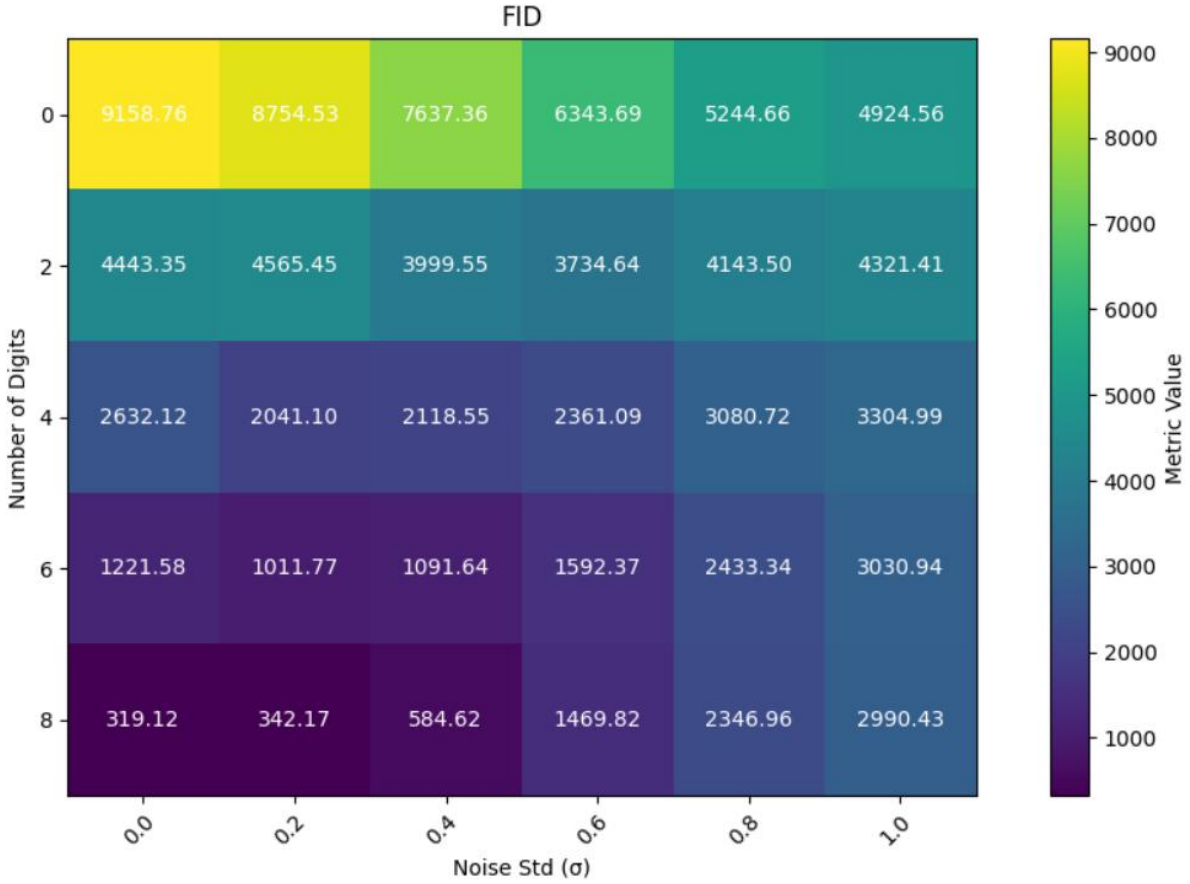
עבור קבוצה של דגימות מ-test, אם בהדרגה נוסיף רעש גאוסייאני לדגימות – הציפייה היא שה precision ירד.

יתר על כך, עבור קבוצה של דגימות מ-test, ונוריד בהדרגה את כל הדגימות עם label מסויים - הציפייה היא שה recall ירד.

כדי לבדוק כי OT מדד טוב ל- loss נרצה לראות אם עבור קבוצות אלו ככל שה precision/recall ירד, כך גם הOT בין הקבוצה הזאתי לבין test.

תוצאות הניסוי:


Optimal Transport (OT)
Number of Digits
Noise Std (σ)
Metric Value
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
5103.15
4910.08
4324.65
3686.24
3122.32
2990.52
2868.50
2898.93
2598.75
2399.19
2607.96
2713.23
1906.51
1555.91
1603.69
1699.78
2066.73
2188.82
1132.23
1002.73
1060.83
1276.19
1712.94
1999.84
546.96
553.46
678.11
1117.06
1587.42
1942.20
5000
4000
3000
2000
1000


FID
Number of Digits
Noise Std (σ)
Metric Value
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
9158.76
8754.53
7637.36
6343.69
5244.66
4924.56
4443.35
4565.45
3999.55
3734.64
4143.50
4321.41
2632.12
2041.10
2118.55
2361.09
3080.72
3304.99
1221.58
1011.77
1091.64
1592.37
2433.34
3030.94
319.12
342.17
584.62
1469.82
2346.96
2990.43
9000
8000
7000
6000
5000
4000
3000
2000
1000

מסקנות:

כפי שניתן לראות FID,OT אכן בעל heatmap מעוד דומים המעיד על כך שOT מקיים את הדרישות לנשערת ניסוי זה. יתר על כך, ניתן לראות כי באמת המדדים (FID,OT) נמוכים יותר כאשר שני האילוצים (הוספת רעש, הסרת ספרות) גדלים כפי שחזינו.

אם זאת ניתן לראות כי עבור הסרת מספרים יותר גבוה היחס מתהפך והוספת רעש מוריד את המדדים ולא מעלה אותם – ניתן להסביר תופעה זאת על כך שהוספת רעש על מספר יחיד יכול לגרום לו להיראות כמו ספרות אחרת (למשל שינוי של 0 ל6 ו8 ושינוי של 1 ל7) אשר על אף שיוריד precision, יעל את ה recallמשמעותית אשר יגרום לתופעה זו.

סה"כ ניתן להסיק כי ניסוי זה הוצלח וOT באמת מדד התופס precision/recall.

<u>ניסוי 2 – overfitting:</u>

לכל $\alpha \in range(0,1.1,0.1)$ נגדיר קבוצה בעלת $(1-\alpha)\cdot 1000$ דגימות מtest ו$\alpha \cdot 1000$ מtrain ועל כל קבוצה מופעל רעש גאוסיאני אשר $std = (1-\alpha)\cdot 0.07$.

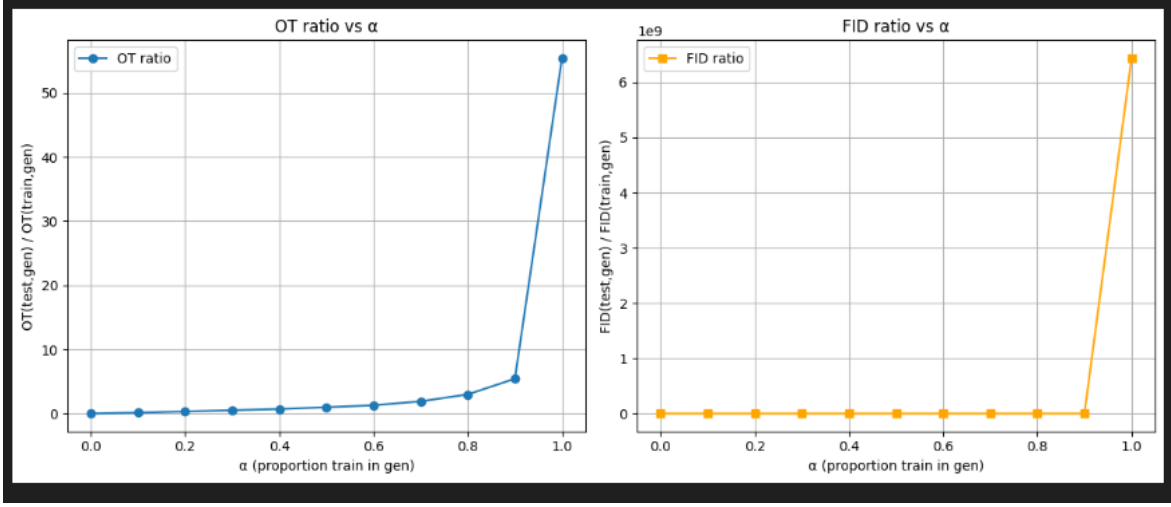
נשים לב כי נצפה שעבור α גדל גם overfitting יגדל.

כאשר יש overfitting המודל אמור להחזיר בדיוק train. אבל כשיש למידה תקינה מקבלים תוצאות שאמורות להיות כמו testאבל עם רעש כלשהו (במודל זה מחליט שהרעש גאוסייאני). לכן אנו בהדרגה אנחנו הופכים את המודל ממאומן בצורה טובה למודל בוא יש overfitting.

עבור רצף קבוצות אלו נחשב עבור שני המדדים (OT,FID) את יחס המדד עבור הקבוצות מול test חלקי המדד עבור הקבוצות מול train (אשר כפי שלמדנו בהרצאה – בודק overfitting).

תוצאות הניסוי:

```
0.0
/tmp/ipython-input-2776528633.py:20: LinAlgWarning: Matrix is singular. The result might be inaccurate or the array might not have a square root.
  covmean = scipy.linalg.sqrtm(sigma1 @ sigma2)
α=0.00, std=0.07, OT ratio=0.0183, FID ratio=0.0043
0.1
α=0.10, std=0.06, OT ratio=0.1651, FID ratio=0.0780
0.2
α=0.20, std=0.06, OT ratio=0.3556, FID ratio=0.2754
0.30000000000000004
α=0.30, std=0.05, OT ratio=0.5346, FID ratio=0.6808
0.4
α=0.40, std=0.04, OT ratio=0.7352, FID ratio=0.7017
0.5
α=0.50, std=0.04, OT ratio=1.0068, FID ratio=1.0909
0.6000000000000001
α=0.60, std=0.03, OT ratio=1.3144, FID ratio=1.0559
0.7000000000000001
α=0.70, std=0.02, OT ratio=1.9294, FID ratio=1.8200
0.8
α=0.80, std=0.01, OT ratio=3.0082, FID ratio=3.4196
0.9
α=0.90, std=0.01, OT ratio=5.4545, FID ratio=7.7282
1.0
α=1.00, std=0.00, OT ratio=55.4422, FID ratio=6441558199.9689
```



מסקנות:

כפי שניתן לראות המדד שחישבנו, הן לOT והן לFID עולה כאשר עולה ה overfitting כפי שחזינו. יתר על נראה כי אופן העלייה אצל שניהם די דומה – עלייה הדרגתית וקפיצה גדולה בחלק האחרון אם השוני הקטן שOT עם קפיצות פחות חדות (אשר מבחינתנו זה דבר יותר חיובי לגבי כי ניתן לראות באופן יותר מובהק את העלייה – המחזק את כך שOT מדד טוב). ומכך ניתן להסיק כי OT מדד אשר ניתן לתפוס דרכו overfitting.

<u>מסקנות משני הניסויים:</u>

ניתן להסיק כי OT מדד היכול לתפוס precision, recall, overfitting. ולכן – ניתן להסיק כי OT מדד היכול לתאר loss למודל.

בכך ניתן לעבור לחלק האקטיבי – בו נשתמש במדד זה במודלים.

יכולות אקטיביות – שיפור למידה:

תחת התחום של היכולות האקטיביות בדקנו שתי תכונות משמעותיות, הראשונה הייתה לבדוק האם OT מסוגל לשפר תוצאות סופיות והשנייה האם OT מסוגל להאיץ למידה.

עבור בדיקת התזה הראשונה עשינו את הדברים הבאים:

הכנו diffusion model המגיע להתכנסות אחרי 30 epochs .

בדקנו שלוש סוגי מודלים:

סוג המודלים הראשון היה הכנת diffusion model המקורי.

עבור בדיקת איכות של המודלים בעבודה דגמנו 50 תמונות ואז הצגנו אותם באופן ויזואלי וכן בעקבות הוכחת התזה עבור היכולות הפסיביות בכל סוף מודל OT יצרנו 1000 דגימות של המודל הנבדק באותו המקרה ואז עשינו:

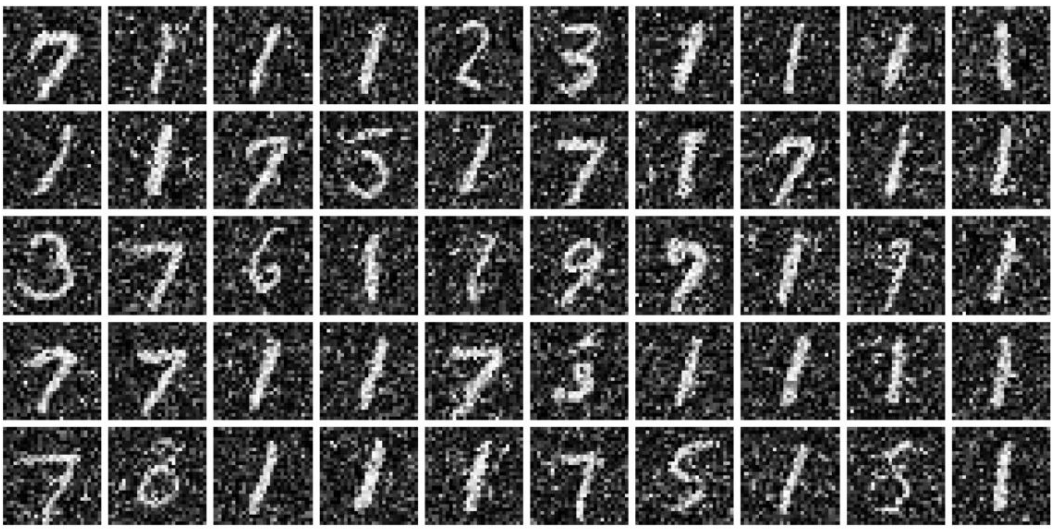
OT(gen , train)

OT(gen, test)

OT(train, test)

וכפי שכבר הראנו OT(gen , train) לעומת OT(gen, test) בודקת OVERFFITING וכן OT(gen, test) לעומת OT(train, test) בודק את איכות יצירת התמונות.

עבור מודל זה אלו התמונות להן עשינו ויזואליזציה.



וכן זה חישוב ה OTs

```
done sampling from model
tensor(2451.5420, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(2499.7644, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(387.3899, device='cuda:0', grad_fn=<SelectBackward0>)
```

כפי שניתן לראות באופן מיידי איכות יצירת התמונות לא גבוהה וכן ניתן לראות זאת גם בהשוואת OT(train, test) לעומת OT(gen, test) (בהמשך גם נראה מודלים טובים יותר וכך נראה שבאמת יש שיפור ושמודל זה אינו מאוד איכותי). לעומת זאת ניתן לראות מהתמונות המוצגות שאין coverage טוב לתמונות ולעומת זאת לפי OT(gen , train) vs OT(gen, test) אין Overffiting, זה קורה כי באמת אין Overffiting, הסיבה לחוסר הייצוג זה לא בגללOverffiting אלא בגלל שהמודל לא מצליח ללמוד את המשתנים החבויים של חלק מהספרות, אנחנו נראה אתה התופעה הזאת חוזרת לאורך כל העבודה.

לאחר מכן לאורך העבודה בדקנו שני סוגי מודלים:

סוג ראשון הוא מודלים בהם הכנסנו את ה OT לתוך שלבי האימון בפועל זה אומר שבנוסף ל Loss של ה MSE עשינו גם LOSS שמתחשב ב OT , את זה עשינו בכך שבכל שלב עשינו OT בין ה BATCH לבין קבוצה רנדומלית של 1000 תמונות מתוך ה validation (שגודלו הוא 10,000), בחרנו מספר זה לאורך כל הפרוייקט מכיוון שהוא שומר על איזון בין שהדאטא מספיק גדול כך שמייצג את ההתפלגות ובין שהוא קטן מספיק בשביל לגרום לתהליכי חישוב השונים להיות מהירים. בנוסף בחרנו כל פעם בכל המודלים השונים שאחרי כל שלב יבחרו 1000 תמונות חדשות, עשינו זאת על מנת שלא יתבצע OVERFITTING סביב 1000 דגימות וכן על מנת שאם פעם אחת יוצא דגימת תמונות לא מייצג זה לא יפגע בכל האימון אלא יגרום רק לשלב אחד להיות עם פחות COVEREGE.

בחרנו שבקירוב שני שליש מה - LOSS הכולל יהיה MSE והשליש הנותן זה loss מה OT.

סוג המודלים השניים היה מודלים בהם הלמידה היא רגילה אבל כל שלב של SAMPLING אחרי שמעבירים את הדגימות מ Xt ל X0 אז מעבירים אחרי זה ל 0 "X שמחושב בעזרת ביצוע צעד גרדיאנט על הדגימות עצמן בכל פעם.

<u>תוצאות עבור בדיקת שיפור איכות:</u>

עבור המודל עם ה OT ב training:

עם reg = 1:

```
done sampling from model
tensor(2558.0122, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(2627.3735, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(384.6354, device='cuda:0', grad_fn=<SelectBackward0>)
```

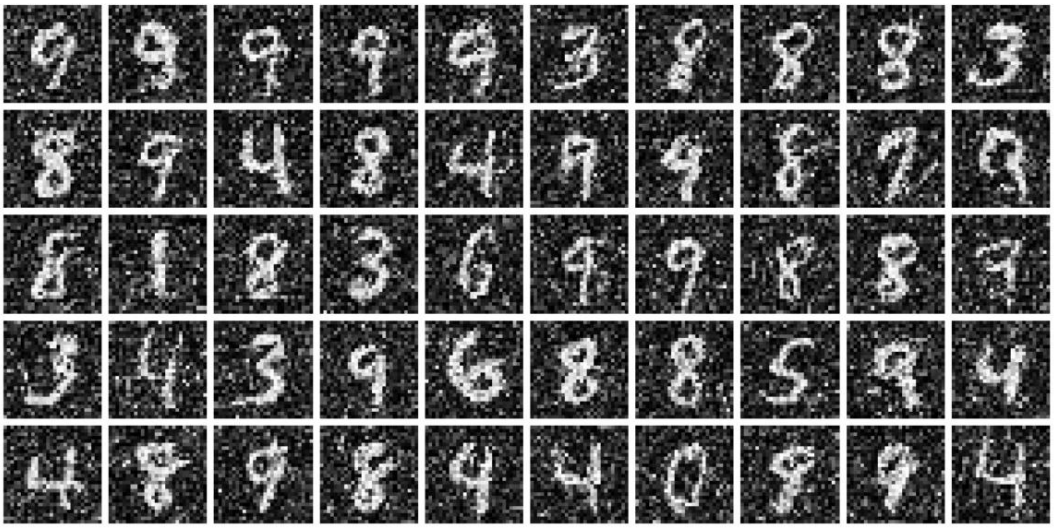
ניתן לראות שהמודל לא עוזר ואף מוריד את האיכות יצירת התמונות, נשמר כך שאין OVERFITING.

עם reg = 0.1:

```
done sampling from model
tensor(2269.9851, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(2438.6450, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(388.2331, device='cuda:0', grad_fn=<SelectBackward0>)
```

ניתן לראות שהמודל משפר מעט את איכות יצירת התמונות, נשמר כך שאין OVERFITING.

עבור reg = 0.01



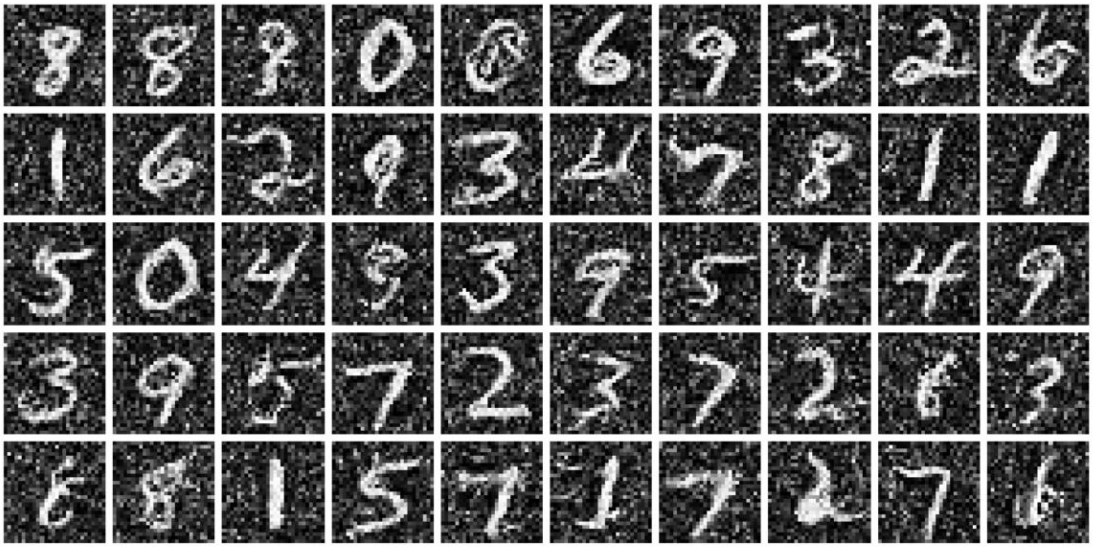
```
done sampling from model
tensor(1979.2554, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1908.3339, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(379.0163, device='cuda:0', grad_fn=<SelectBackward0>)
```

ניתן לראות שיפור משמעותי באיכות התמונות, ניתן לראות זאת גם מ הOT וגם מהתמונות, נשמר שאין OVERFITING , ניתן לשים לב גם שיש גם יצוג רחב של מספרים ולכן ניתן להסיק שגם ה COVEREGE השתפר באופן משמעותי (חוץ מזה, ניתן לציין גם שיפור של OT(gen, test) וOT(gen,train) גם מעיד על COVEREGE טוב כי אז יש פחות מספרים שה GEN יוצר אותם שהם מספרים מסוג מסויים אבל צריך להעביר אותם במרחב החבוי למספרים אחרים, מה שמוסיף הרבה ל OT).

המסקנה מכך היא ש OT אכן מסוגל לשפר את האיכות של הלמידה ושיש קשר הפוך בין גודל הרגולוזציה לרמת איכות ה SAMPLING.

<u>עבור סוג המודל השני (sampling) גם פה בדקנו עבור שלושה רגולציות:</u>

עם reg = 1:

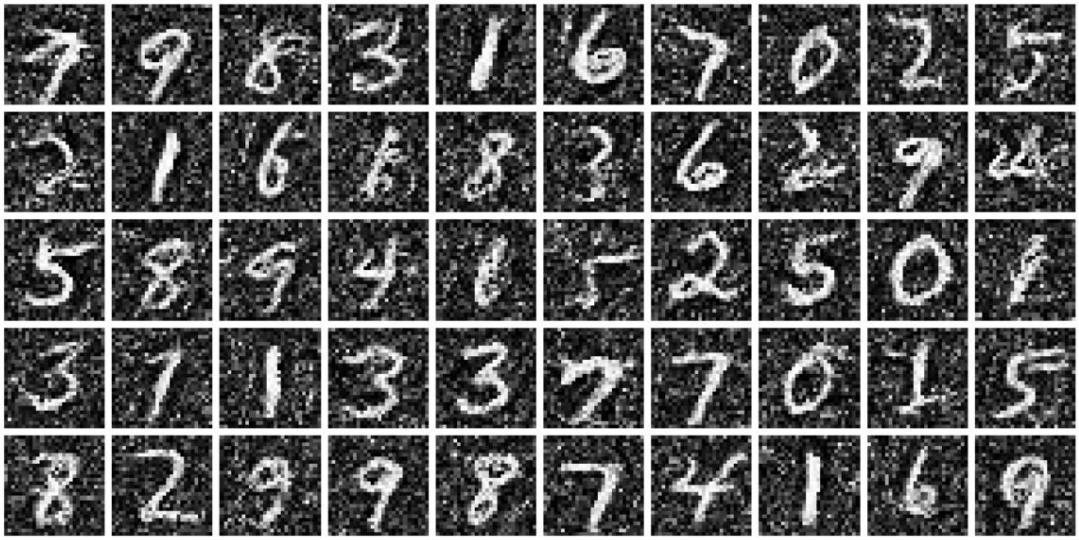


```
tensor(1305.9346, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1358.0084, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(392.6777, device='cuda:0', grad_fn=<SelectBackward0>)
```

עם reg = 0.1:

```
tensor(1331.5792, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1440.2316, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(398.1199, device='cuda:0', grad_fn=<SelectBackward0>)
```

עם reg = 0.01:



```
tensor(1392.5824, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1436.5505, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(395.7597, device='cuda:0', grad_fn=<SelectBackward0>)
```

כפי שניתן לראות, תוצאות המודל השתפרו באופן משמעותי מאשר המודל המקורי, הן בדגימות והן ב-OT. עבור הדגימות ניתן לראות כי ישנו ייצוג רחב וכיסוי טוב של הספרות. עבור הOT ניתן לראות שיפור משמעותי של OT(samples, train), OT(samples, test) והם עם ערך הרבה יותר קטן ועדיין משמרים הפרש קטן בין שניהם האומר גם שאין OVERFITTING. בנוסף ניתן לראות שלמרות שיש שיפור בערכי OT עבור regs יותר גבוהים – השוני לא משמעותי, ולכן ערכי הregs אינם חשובים מאוד עבור שיפור דרך sampling.

המסקנה מכך היא ש OT אכן מסוגל לשפר את האיכות של הלמידה.

יכולות אקטיביות – האצת למידה:

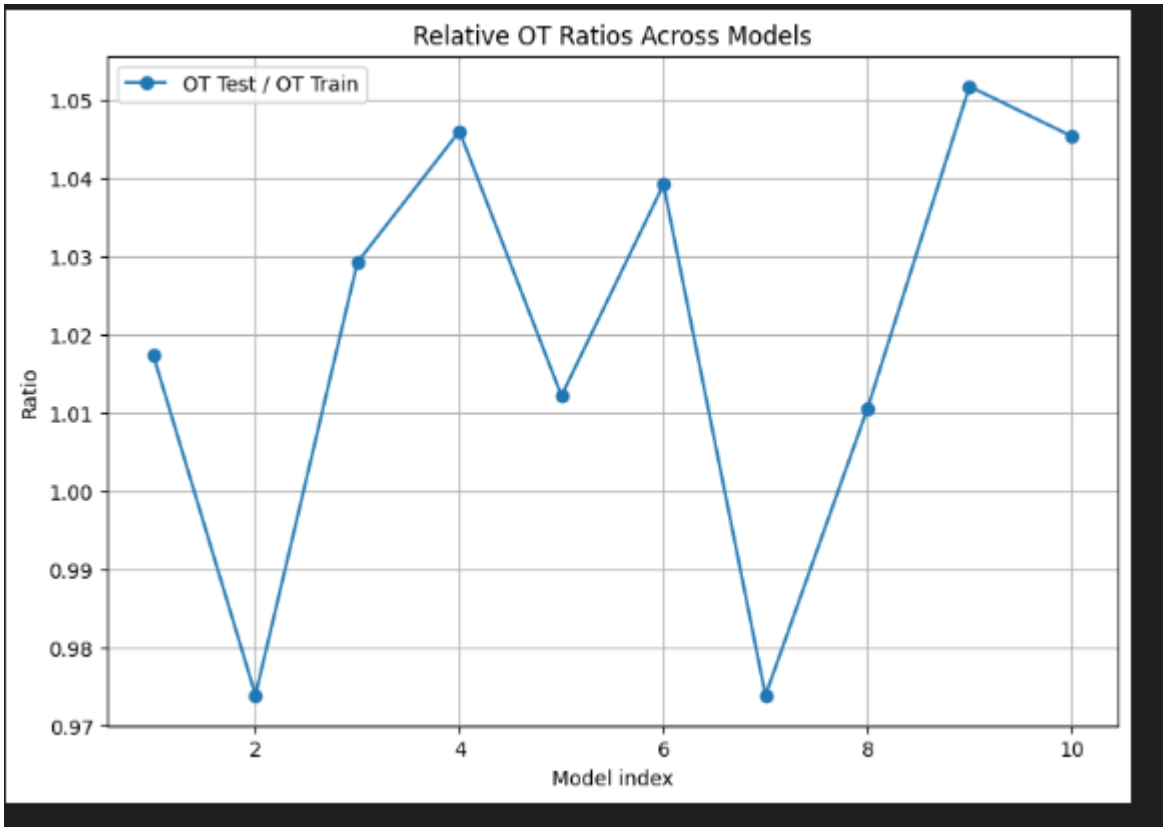
אנו רוצים לבדוק האם OT מסוגל לשפר את קצב הלמידה של diffusion models.

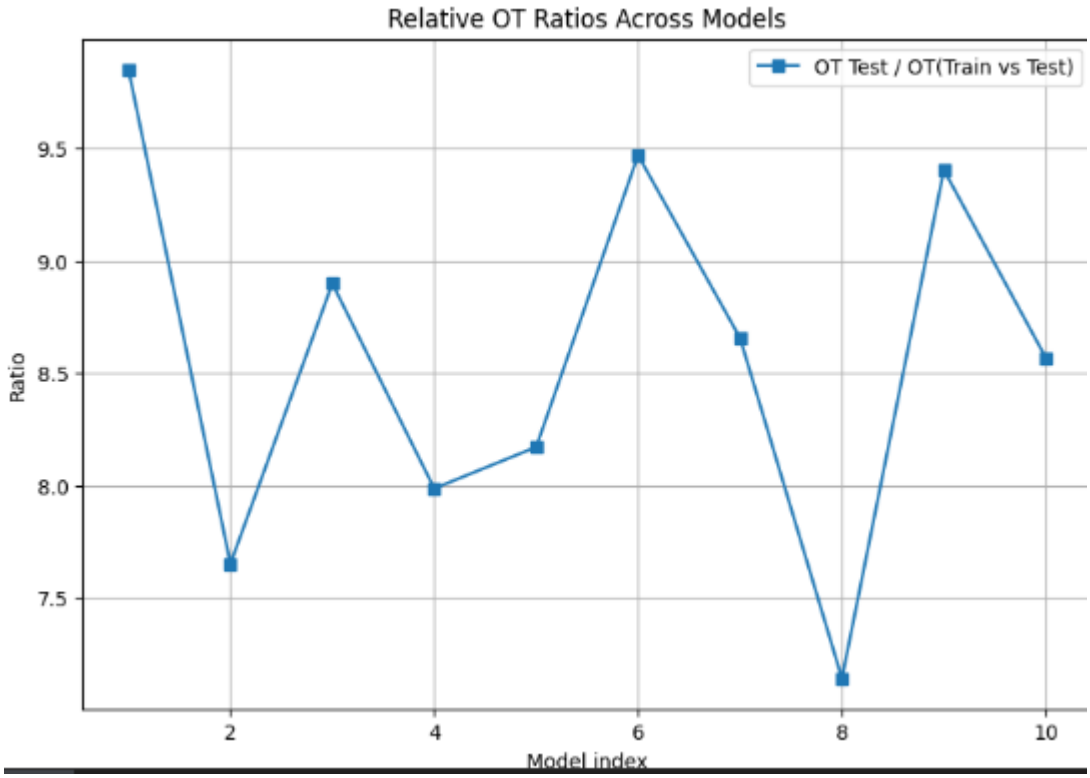
בדקנו זאת בכך שיצרנו diffusion model המתכנס ב 20 אך נתנו לו 10 epochs, ורצינו לבודק האם המודלים השונים משפרים אותו.

עבור המודל הרגיל:

```
done sampling from model
tensor(3083.1238, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(3023.6294, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(397.7182, device='cuda:0', grad_fn=<SelectBackward0>)
```

בנוסף נצרף בדיקה של OT(GEN, TEST)/OT(GEN, TRAIN) עבור ניתוח overfitting, ו OT(GEN, TEST)/OT(TEST, TRAIN) לניתוח איכות המודל (כי כדי לבדוק האצה הגרף יכול להעיד על כך):



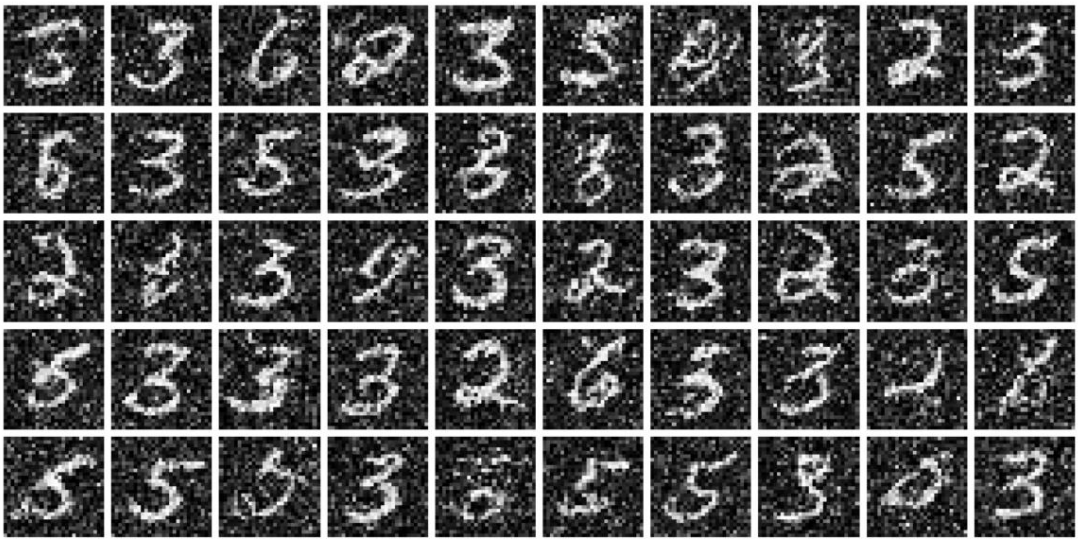


ניתן לראות שגם איכות התמונות לא טוב וגם הייצוג אינו טוב אך לחיוב אין OVERRFITING (באופן ברור כי לא נתנו למודל להתקרב בכלל להתכנסות).

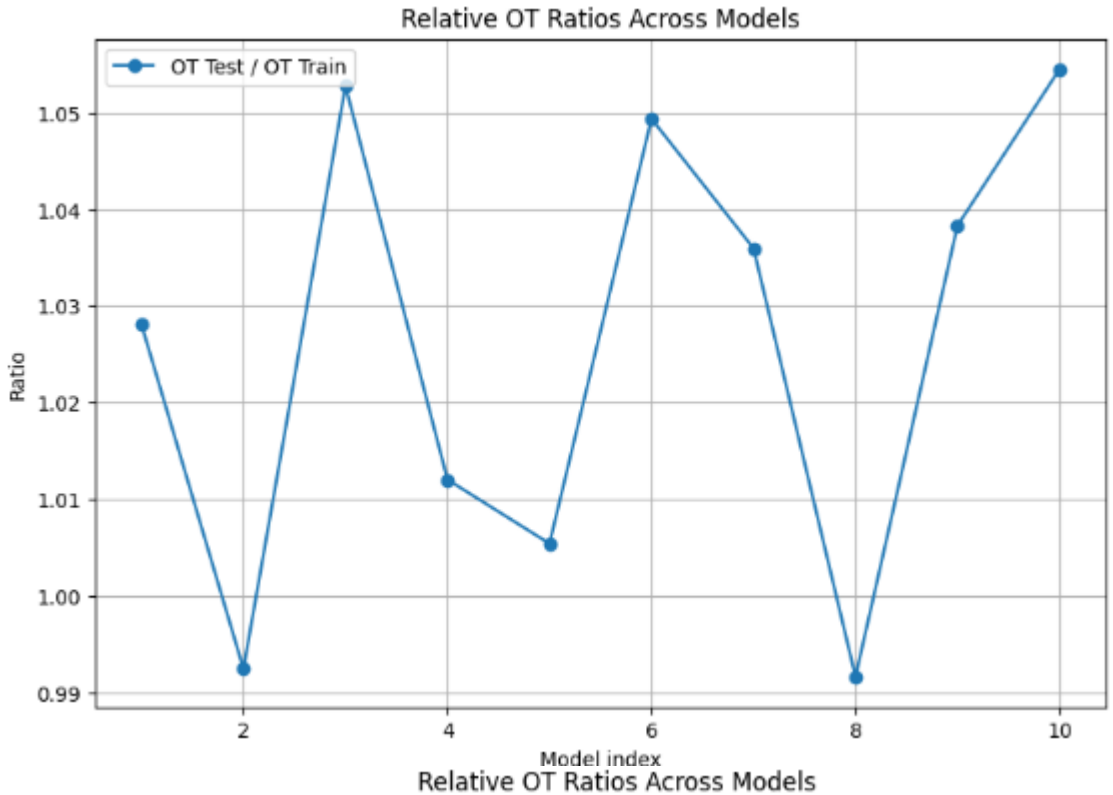
נבדוק האם שני סוגי המודלים משפרים:

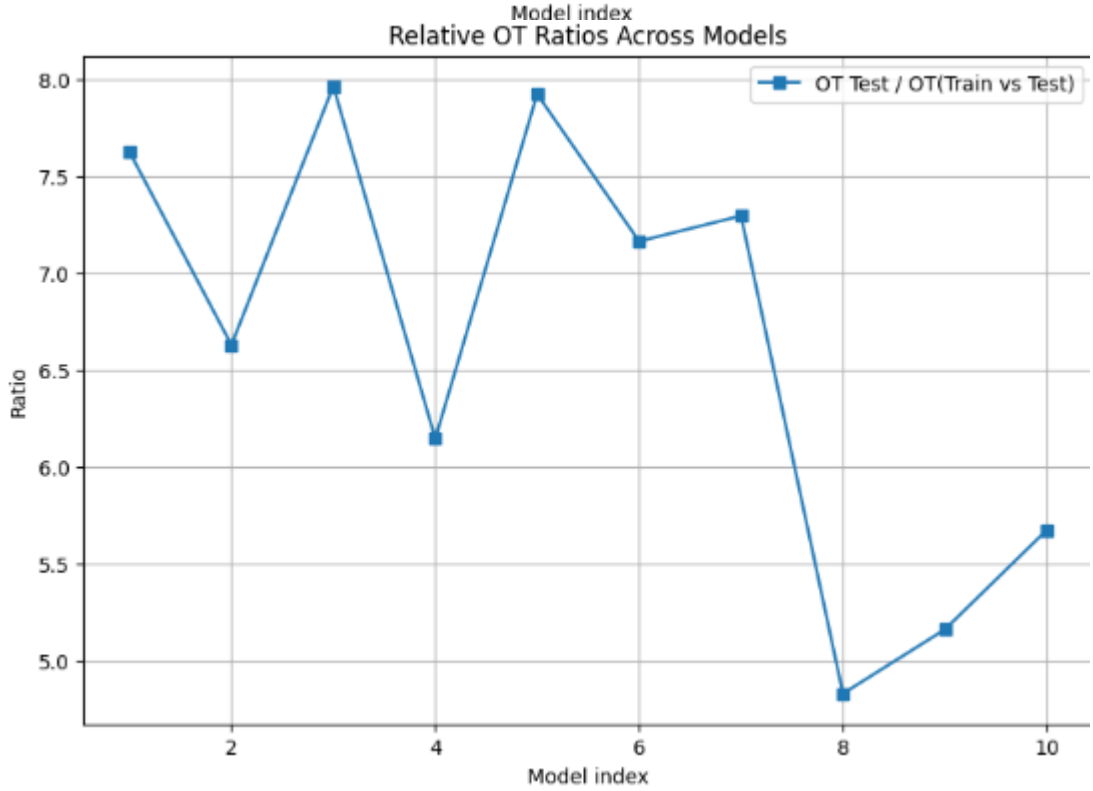
עבור training:

Reg = 1:



```
done sampling from model
tensor(1928.8655, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(2062.4819, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(401.4148, device='cuda:0', grad_fn=<SelectBackward0>)
```

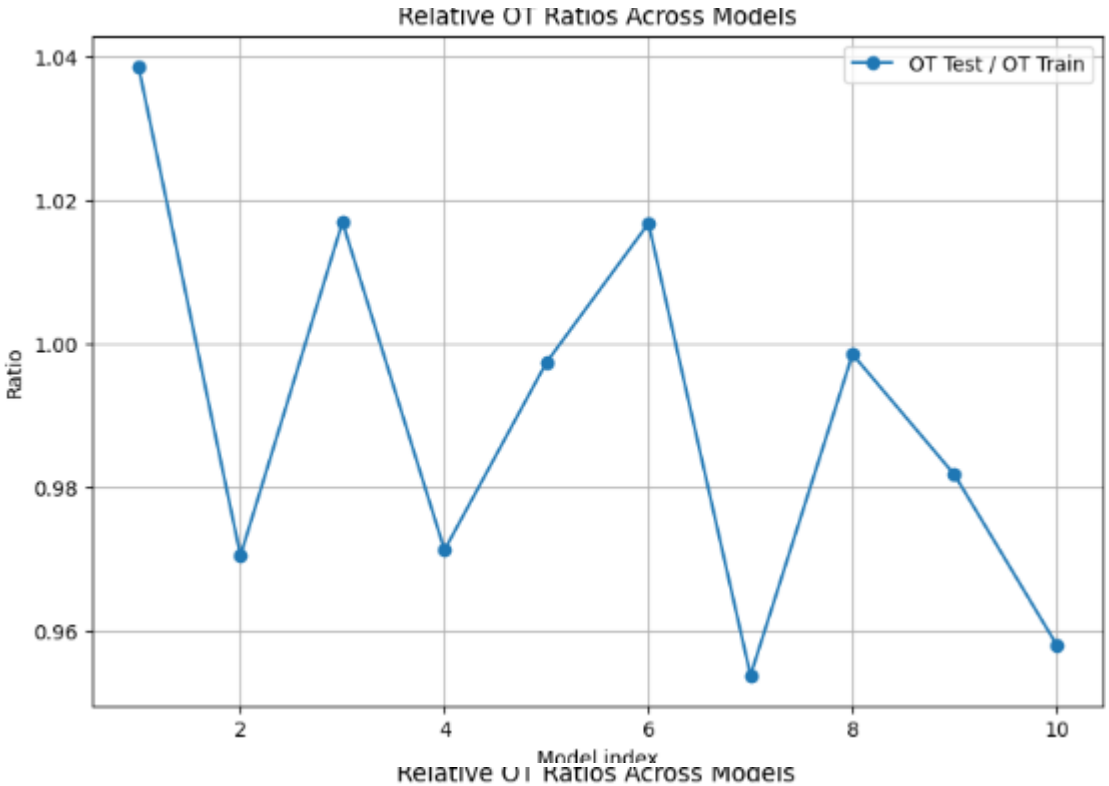


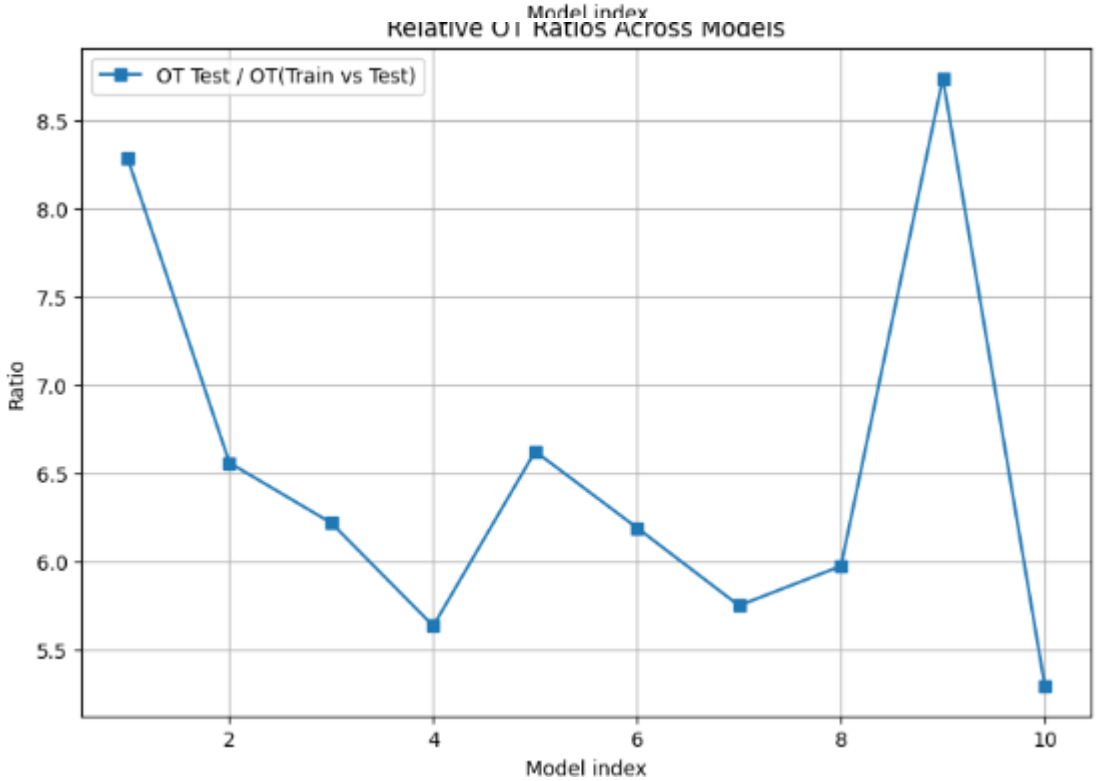


ניתן לראות שיפור משמעותי באיכות התמונות (וגם בגרף המייצג איכות תמונות) ביחס למודל הרגיל, ונשמר כך שאין OVERFITING וזה למרות שיש תוצאות טובות(זה מהיחס הקודם) ולכן ניתן להבין שאכן יש למידה מוצלחת.

Reg = 0.1:

```
done sampling from model
tensor(2134.6733, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(2259.7412, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(378.1490, device='cuda:0', grad_fn=<SelectBackward0>)
```

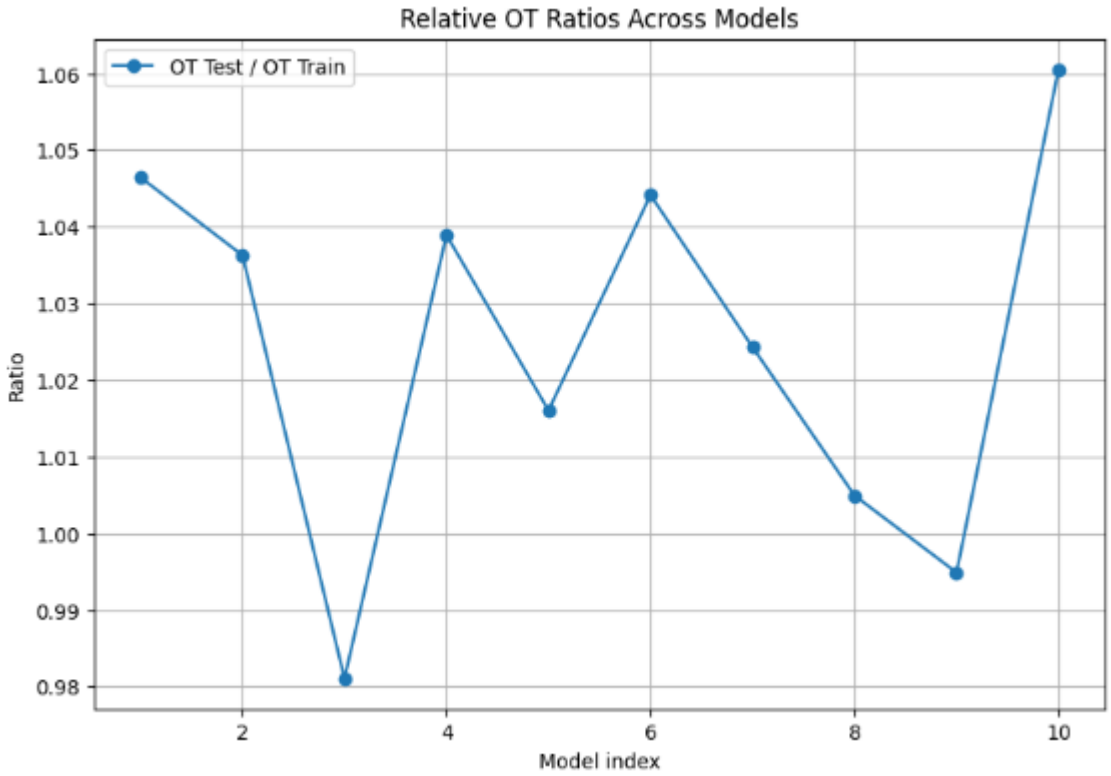


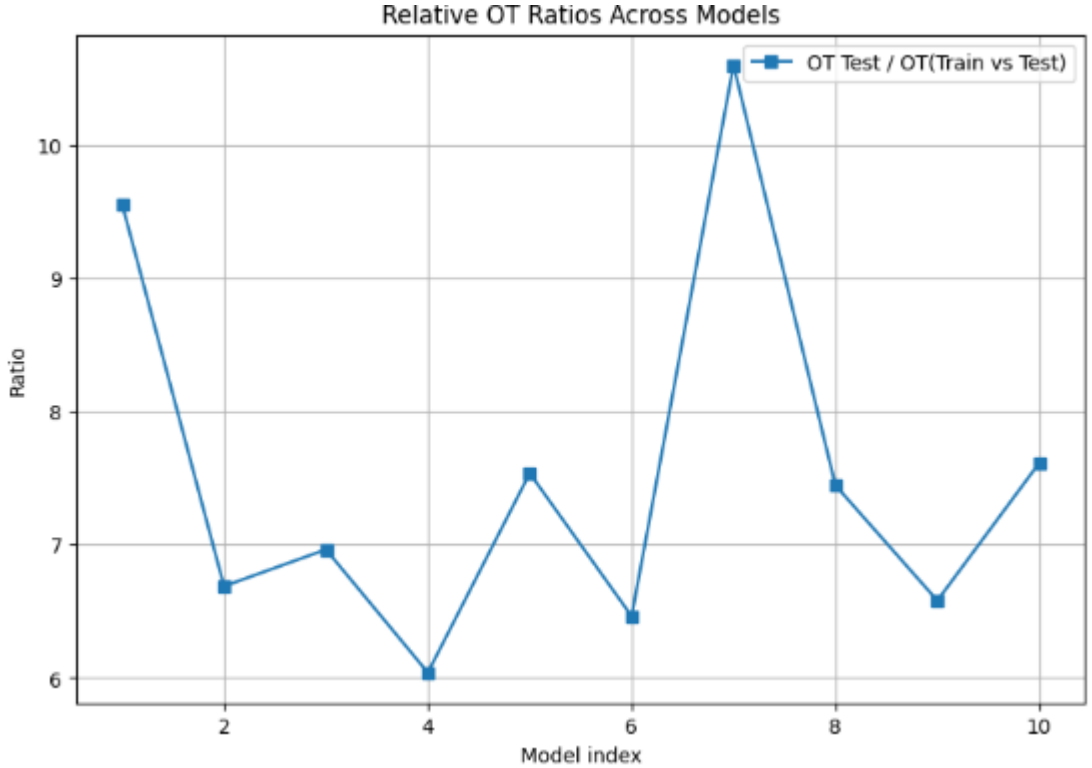


ניתן לראות שיפור משמעותי באיכות התמונות (וגם בגרף המייצג איכות תמונות) ביחס למודל הרגיל, ונשמר כך שאין OVERFITING וזה למרות שיש תוצאות טובות(זה מהיחס הקודם) ולכן ניתן להבין שאכן יש למידה מוצלחת.

Reg = 0.01:

```
tensor(2741.4238, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(2822.8252, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(387.3942, device='cuda:0', grad_fn=<SelectBackward0>)
```





ניתן לראות שיפור באיכות התמונות (וגם בגרף המייצג איכות תמונות) ביחס למודל הרגיל, ונשמר כך שאין OVERFITING וזה למרות שיש תוצאות יחסית טובות(זה מהיחס הקודם) ולכן ניתן להבין שאכן יש למידה יחסית מוצלחת.

סה"כ ניתן לראות כי הוספת OT ב train אכן מאיצה את שיפור המודל בכך שהמודלים הנ"ל באותו מספר epochs יותר טובים (וחלקם גם באופן משמעותי) מאשר המודל ללא שימוש בOT

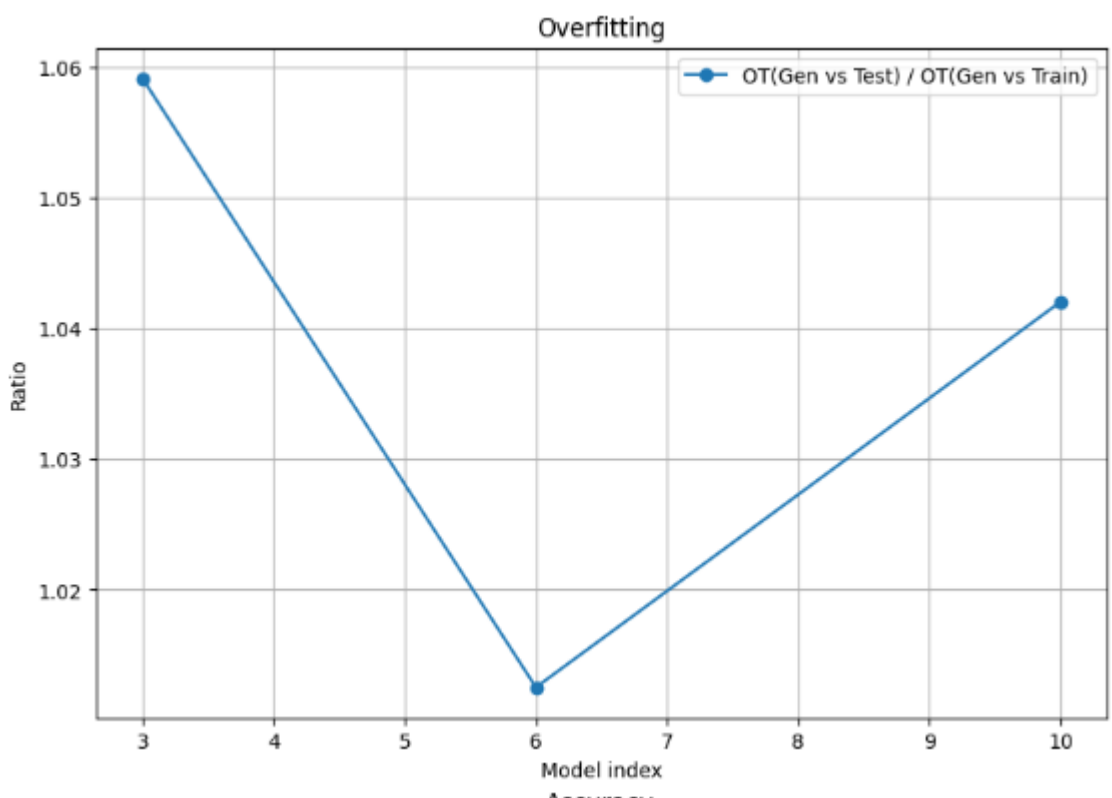
המסקנה מכך היא ש OT אכן מסוגל לשפר את האיכות של הלמידה ושיש קשר עולה בין גודל הרגולוזציה לרמת איכות ה SAMPLING.

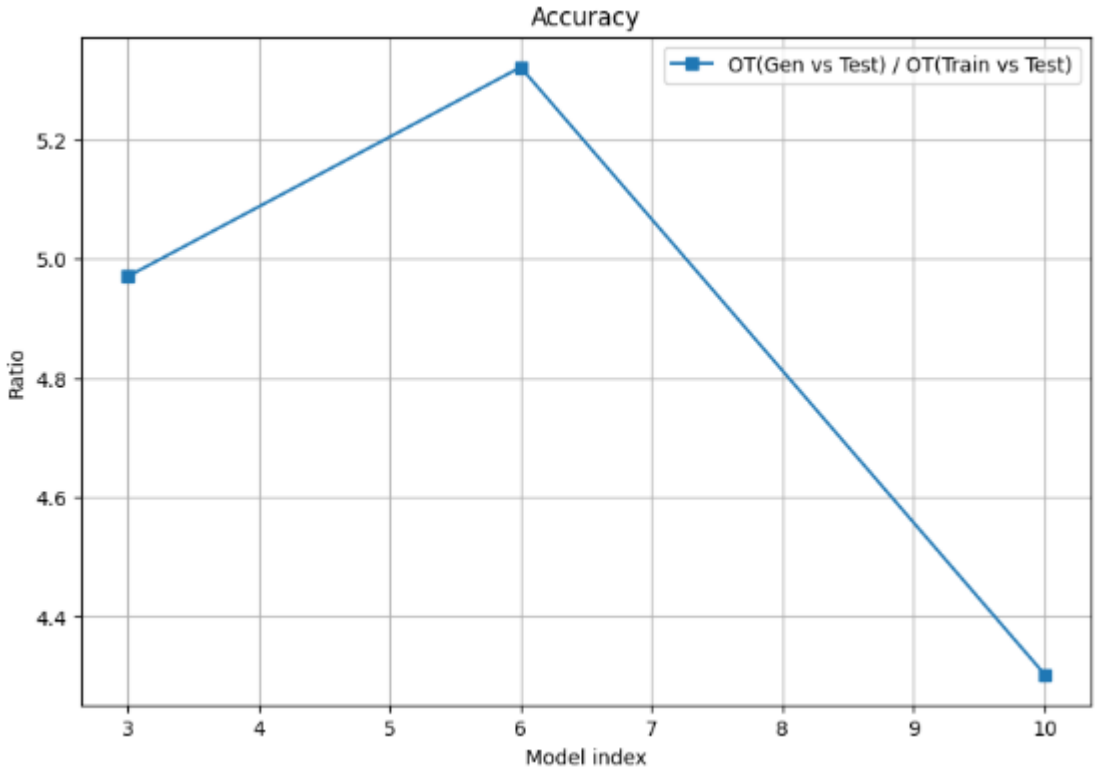
<u>עבור sampling:</u>

מכיוון שבאופן טבעי לקיחת דגימות יהיה יותר כבד עבור sampling בגרף האיכות, overfitting ישנם את המדדים רק עבור הepochs 3,6,10.

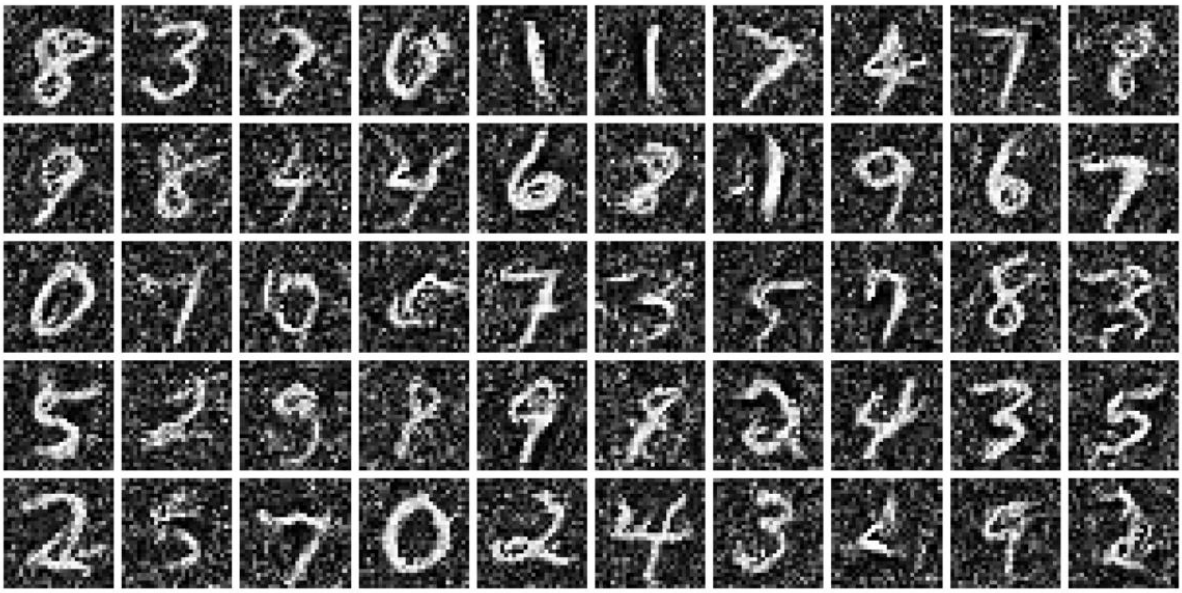
Reg = 1:

```
tensor(1553.7400, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1562.2300, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(384.8263, device='cuda:0', grad_fn=<SelectBackward0>)
```

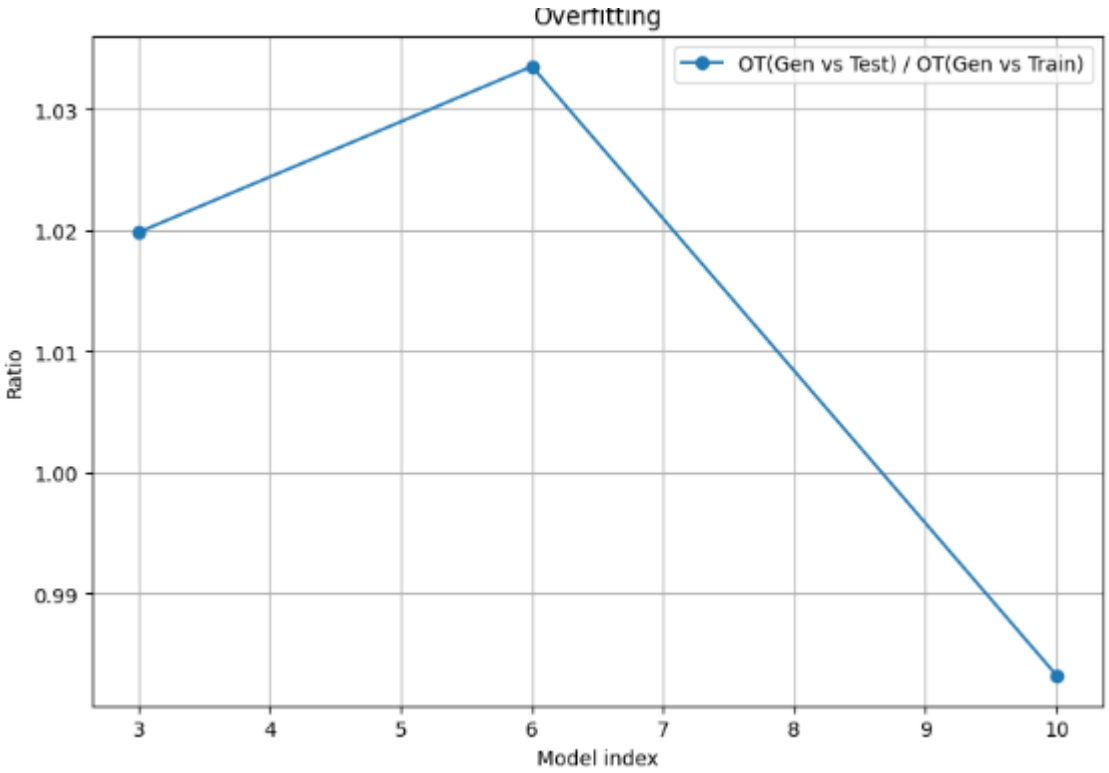

Overfitting
OT(Gen vs Test) / OT(Gen vs Train)
Ratio
1.06
1.05
1.04
1.03
1.02
3
4
5
6
7
8
9
10
Model index

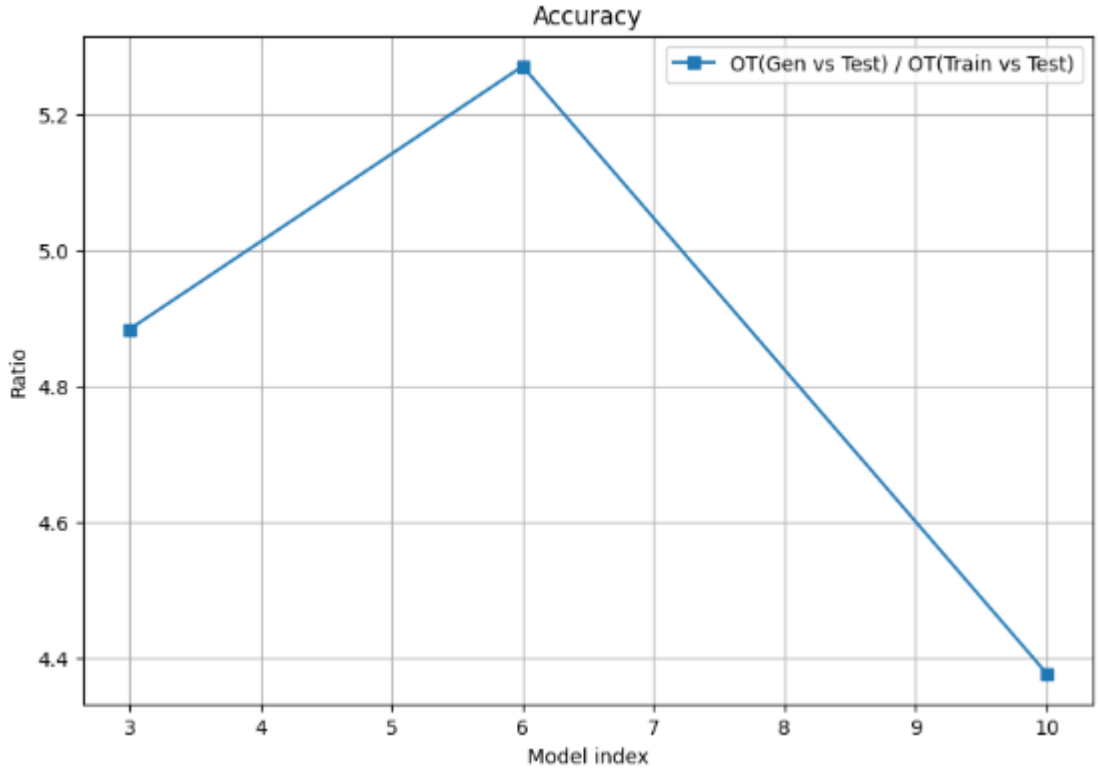

Accuracy
OT(Gen vs Test) / OT(Train vs Test)
Ratio
5.2
5.0
4.8
4.6
4.4
3
4
5
6
7
8
9
10
Model index

:Reg = 0.1



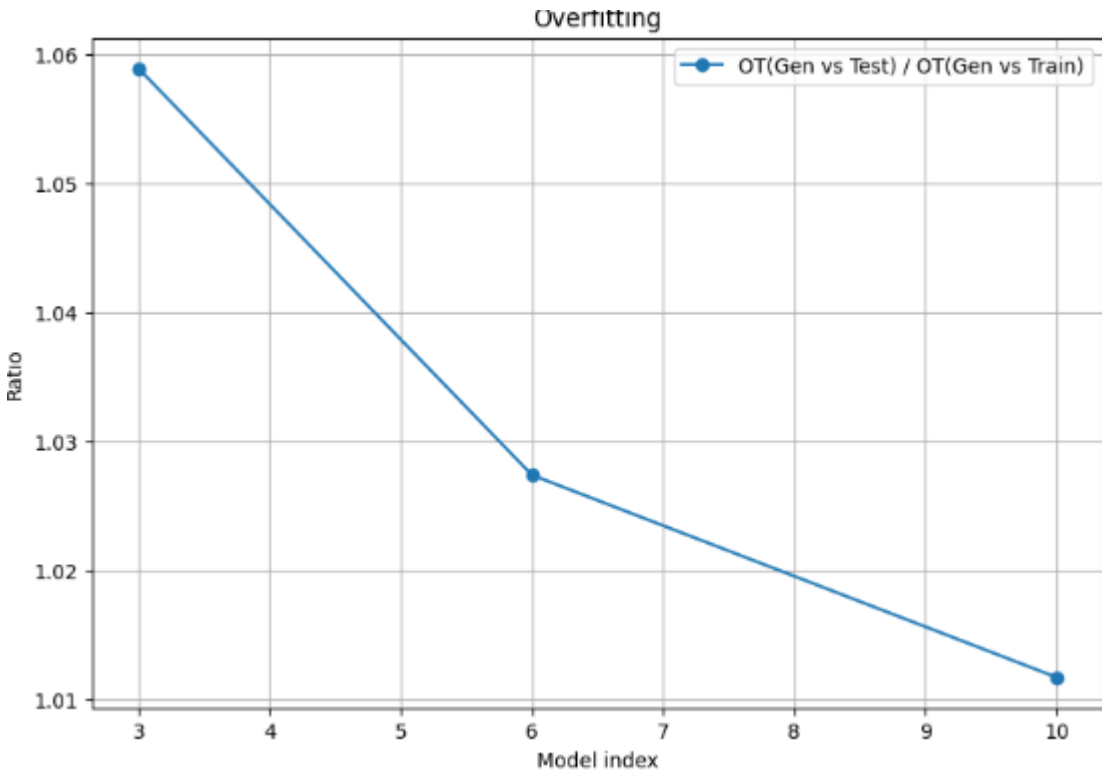
```
tensor(1544.6418, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1591.8997, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(393.6196, device='cuda:0', grad_fn=<SelectBackward0>)
```

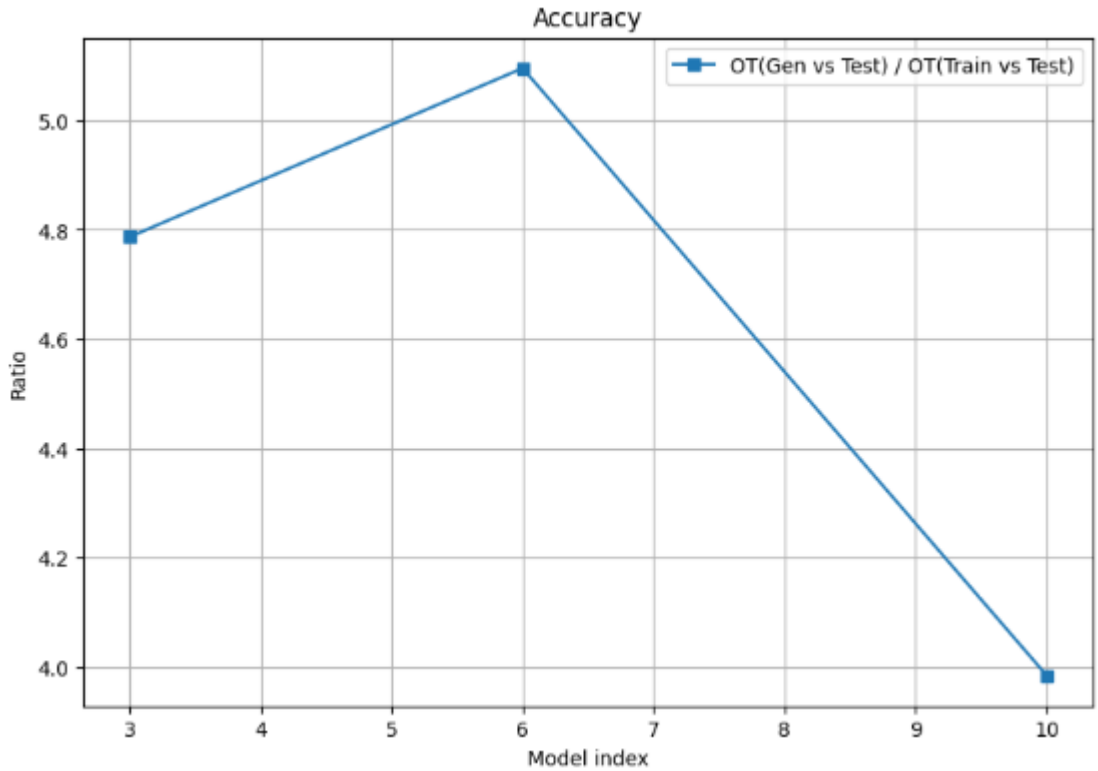

Overfitting
OT(Gen vs Test) / OT(Gen vs Train)
1.03
1.02
1.01
1.00
0.99
Ratio
3
4
5
6
7
8
9
10
Model index


Accuracy
OT(Gen vs Test) / OT(Train vs Test)
5.2
5.0
4.8
4.6
4.4
Ratio
3
4
5
6
7
8
9
10
Model index

:Reg = 0.01

```
tensor(1600.5725, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(1610.4231, device='cuda:0', grad_fn=<SelectBackward0>)
tensor(377.4041, device='cuda:0', grad_fn=<SelectBackward0>)
```





כמו שהוסק עבור הבדיקה האקטיבית, ניתן לראות כי תוצאות המודל השתפרו באופן משמעותי מאשר המודל המקורי, הן בדגימות והן ב-OT. עבור הדגימות ניתן לראות כי ישנו ייצוג רחב וכיסוי טוב של הספרות. עבור הOT ניתן לראות שיפור משמעותי של OT(samples, train), OT(samples, test) והם עם ערך הרבה יותר קטן ועדיין משמרים הפרש קטן בין שניהם האומר גם שאין OVERFITTING. בנוסף ניתן לראות שהשוני של תוצאות המודלים עבור regs שונים לא משמעותי, ולכן ערכי הregs אינם חשובים מאוד עבור שיפור דרך sampling.

מסקנות:

כל המודלים שיפרו ביצועים ביחס למודל המקורי זאת בניגוד לבדיקה של השיפור איכות בהתכנסות.

בנוסף גם המודל ה training הכי טוב הפעם הוא עם ה reg = 1 לא ה reg = 0.01.

דבר שנשמר זה שמודל ה sampling גם משפר הכי טוב את איכות התמונות בהתכנסות וגם עוזר הכי הרבה ללמידה.

לסיכום, הראנו שOT גם יכול לשפר את קצב הלמידה וגם מסוגל לשפר את איכות התוצאות הסופיות (כאשר כבר המודל מגיע להתכנסות), גם למדנו שעבור קצב שיפור הלמידה כל המודלים הכוללים OT משפרים אך בסופי רק ה TRAINING עם 0.01 משפר מבין המודלים עם ה OT בזמן האימון וזה בניגוד לקצב השיפור ששם התוצאות הכי טובות הן עבור 1 , הסיבה שניתן לחשוב שזה קורה היא ש OT עם ה 1 יותר קל ללמידה לעומת 0.01 כי הוא מזיז יותר את ההתפלגות כשהיא כללית יותר (כשעדיין אין למידה טובה) לכיוון התפלגות הנכונה אך בקרבת התכנסות, כאשר ההתפלגות כבר קרובה להתפלגות האמיתית הוא גורם לתמונות מסוג מסויים להשתנות לכיוון של תמונות מסוג אחר שלא בצדק, וזה בגלל הרגולוזציה הגבוהה יחסית, הגורמת לחישוב של ה OT לפזר את החישוב של המרחקים בין דוגמא להרבה דוגמאות שונות ולא מרחק רק לדוגמה ספציפית.
דבר זה ככל הנראה גורם לכך שבקרבת ההתכנסות תמונות מה BATCH "מצ'ופרות" על להיות בין לבין סוגי מספרים, זה קורה בעקבות כך שOT עם REG 1 מחשב מרחק מכל סוגי המספרים – ולכן נקודת שיווי המשקל כנראה לא תהיה כאשר התכונות הן חד משמעיות בfeature-space עבור תכונות של מספר מסויים.

שימוש מעניין יכול להיות התחלה ב reg = 1 בחצי הראשון של הלמידה ואז החלפה ל reg = 0.01 זאת על מנת להרוויח למידה מהירה וגם התכנסות מדוייקת.

בנוסף המודל עם ה SAMPLING תמיד משפר באופן הכי משמעותי את התוצאות, גם עבור קצב הלמידה וגם עבור התוצאות הסופיות.