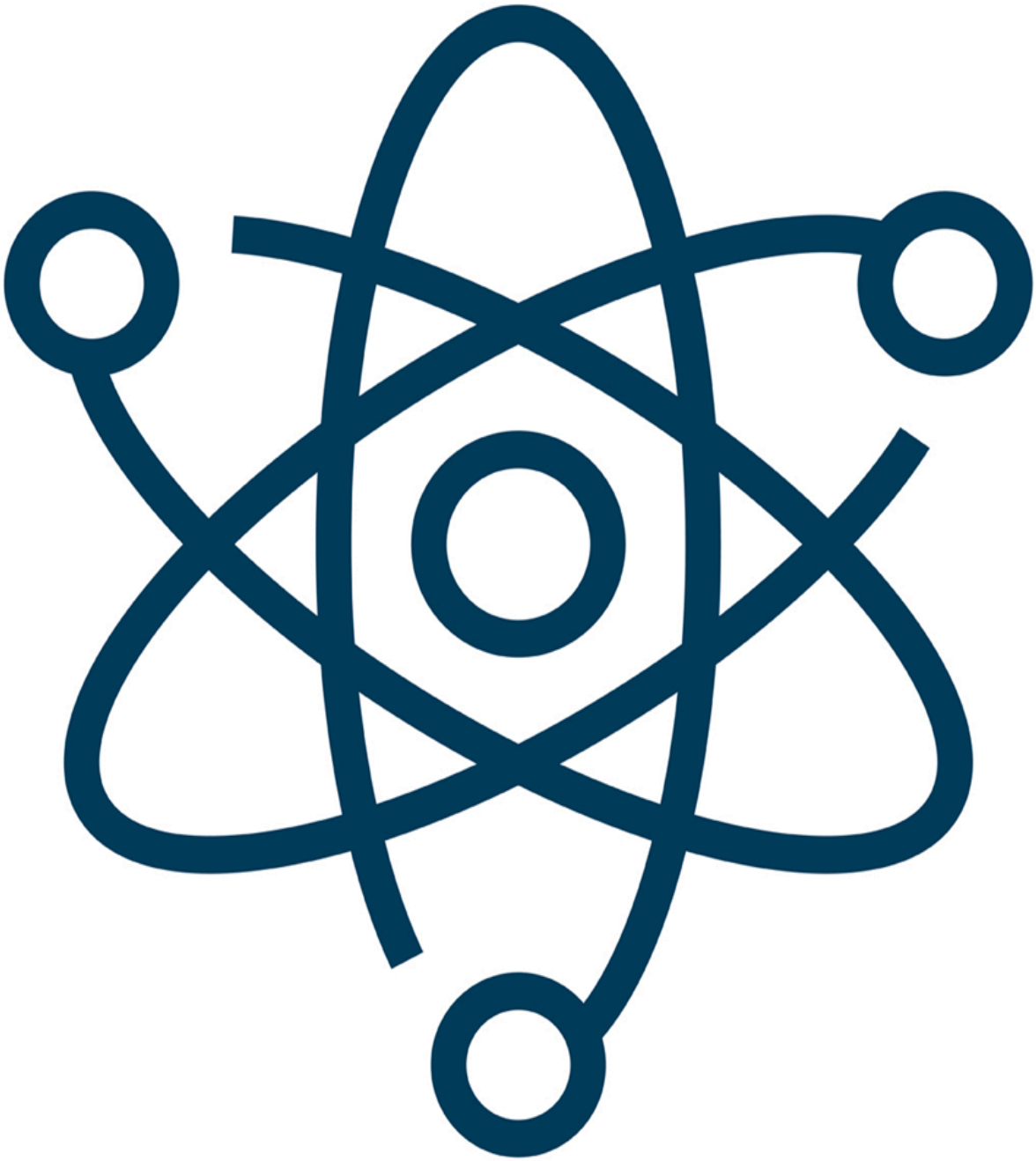
Progress Report (PHY:496 - Summer 2024)

VENKATA SESH TEJ MATTA : (10 Weeks – 06/17/2024 – 08/24/2024).



Timeline

S. No.	Week	Date	Tasks
1	1-2	06/17/2024	Ran the existing notebooks, setting up environments and debugging the code for clarity.
2		06/24/2024	Understanding the codes and getting it running in google collab and roar collab.
3		27/06/2024	Tried various ways to cope with overfitting by simplifying the underlying neural nets of the Generator and Discriminator.
4	3-4	04/7/2024	Tried using lower image resolution and using gray scale for the images. Reason : The color channels will be needed when we use many quarks, not just the up quark
5		07/11/2024	Generated plot for various cases and scenarios. (All plots attached below).
6	5-6	16/07/2024	Enhancements on above generated plots for great visualization.
7		23/07/2024	Generated plots for 3D projections shadows.(side 2D projections)
8	7-8	29/07/2024	Generated contour plots and made enhancements with inclusion of derivatives in the plots.

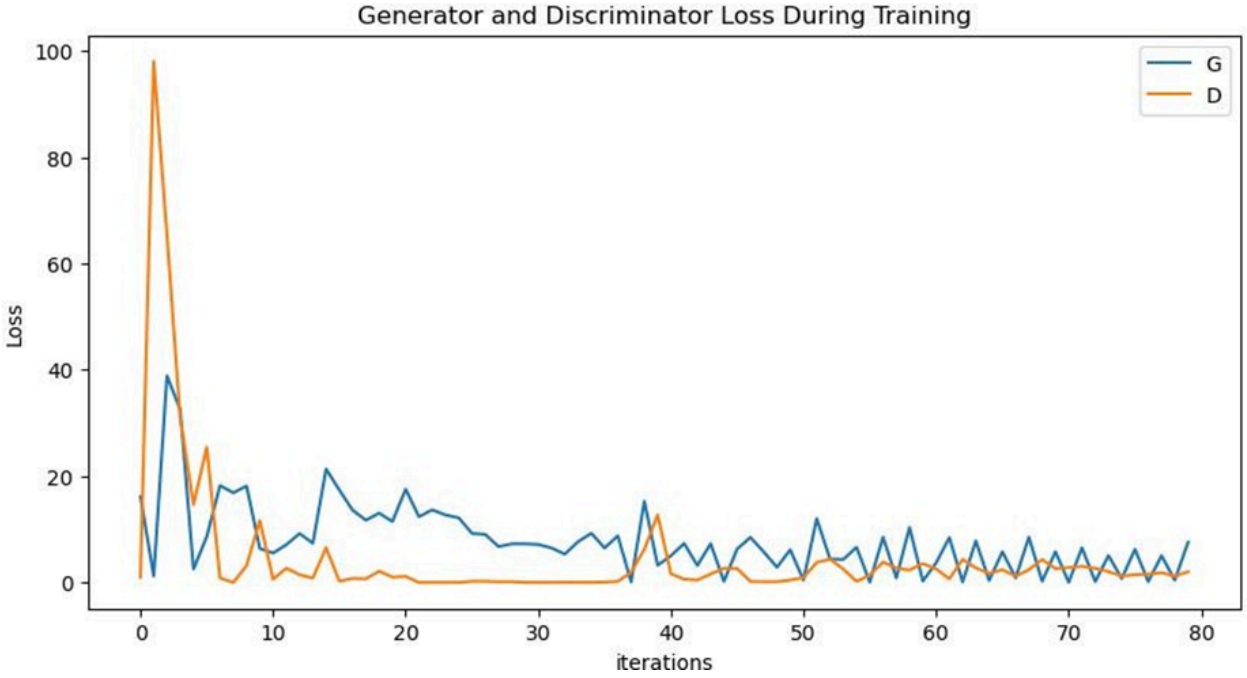
9		**08/10/2024**	**Created a basic version of plot at different slices of X- value.**
10	**9-10**	**08/15/2024**	**Made different versions for the plot above to make it similar to the original picture.**
11		**08/20/2024**	**Made further enhancements as required for the above plot.**
12		**08/24/2024**	**Updated the code for the final version of the plot with various changes.**

NOTE: All the codes for above are attached as a separate attachment.

Below I am going to attach the picture and explanations as required.

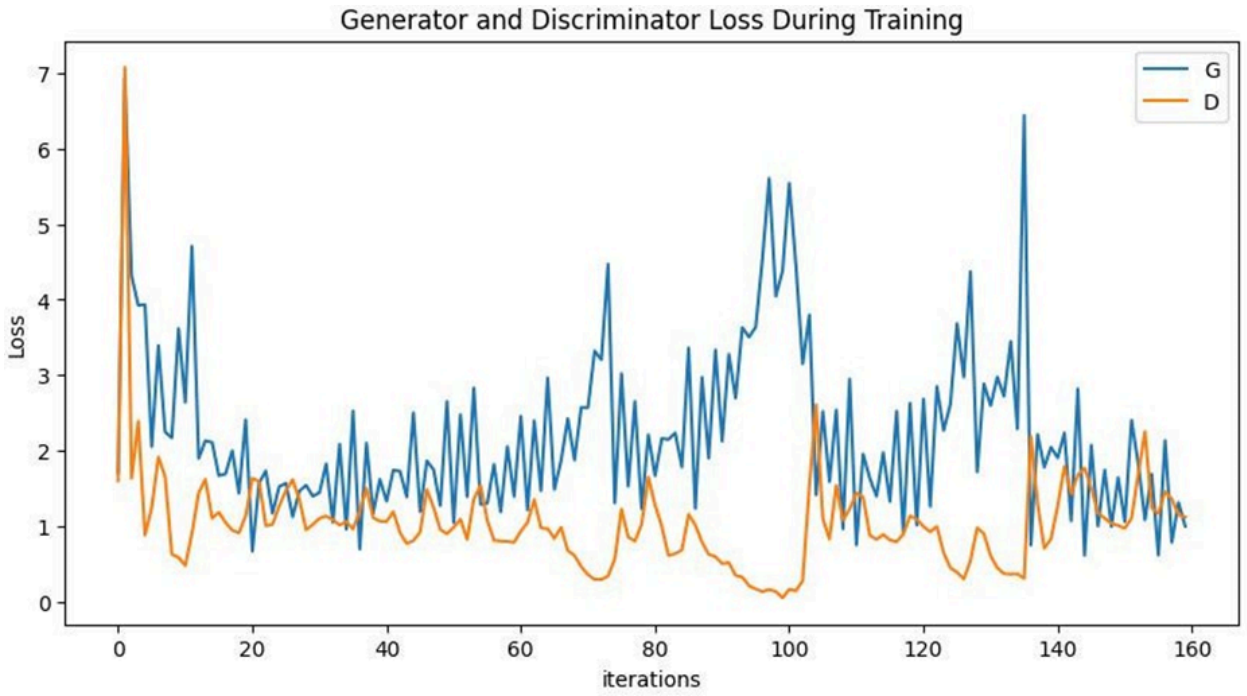
GOAL: Study the GAN notebook, try to cope with overfitting by simplifying the underlying neural nets of the Generator and Discriminator. Try using lower image resolution, maybe 10x10, 20x20 etc. Use gray scale for the images. The color channels will be needed when we use many quarks, not just the up quark.

STEPS: Originally the loss looked something like this

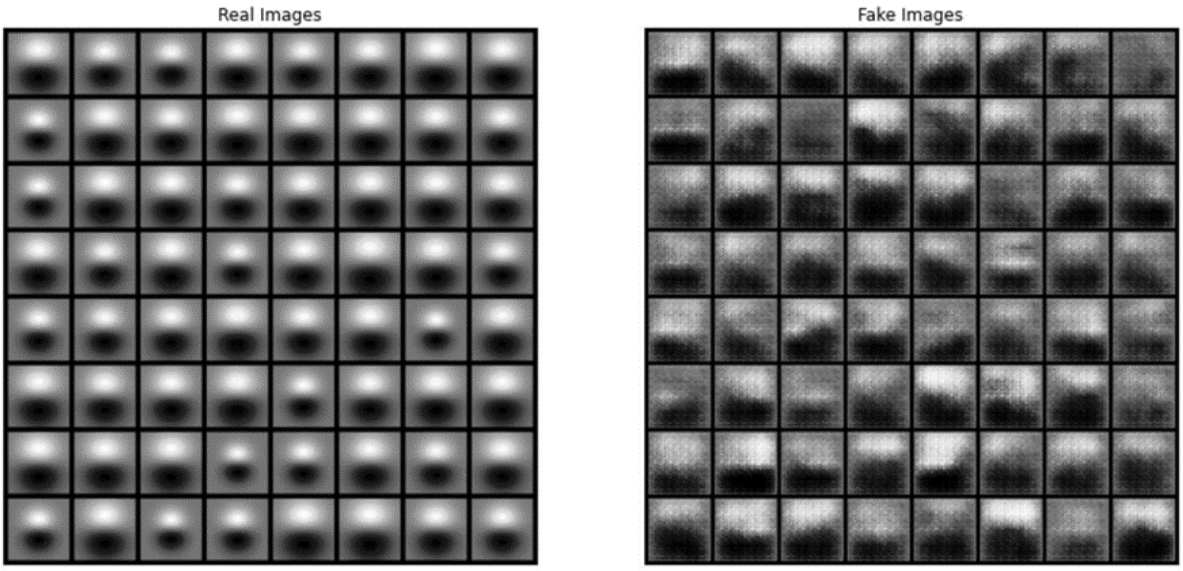


After using the gray scale and made the layers simple , the loss may look like this[y-axis]

```
Starting Training Loop...
[0/20][0/8]     Loss_D: 1.6025  Loss_G: 1.7116  D(x): 0.3819    D(G(z)): 0.3667 / 0.2954
[1/20][0/8]     Loss_D: 0.6307  Loss_G: 2.1719  D(x): 0.7817    D(G(z)): 0.2696 / 0.1395
[2/20][0/8]     Loss_D: 1.0529  Loss_G: 1.6947  D(x): 0.6408    D(G(z)): 0.3644 / 0.2579
[3/20][0/8]     Loss_D: 1.2682  Loss_G: 1.5271  D(x): 0.6992    D(G(z)): 0.5161 / 0.3317
[4/20][0/8]     Loss_D: 1.0842  Loss_G: 1.0532  D(x): 0.5566    D(G(z)): 0.3009 / 0.4209
[5/20][0/8]     Loss_D: 1.0636  Loss_G: 1.3319  D(x): 0.5492    D(G(z)): 0.2722 / 0.3870
[6/20][0/8]     Loss_D: 0.9576  Loss_G: 1.2747  D(x): 0.6822    D(G(z)): 0.3318 / 0.3653
[7/20][0/8]     Loss_D: 0.8151  Loss_G: 1.8201  D(x): 0.7229    D(G(z)): 0.3566 / 0.2109
[8/20][0/8]     Loss_D: 0.9678  Loss_G: 2.9658  D(x): 0.8887    D(G(z)): 0.5219 / 0.0872
[9/20][0/8]     Loss_D: 0.2964  Loss_G: 3.2059  D(x): 0.8768    D(G(z)): 0.1396 / 0.0668
[10/20][0/8]    Loss_D: 1.2827  Loss_G: 1.6695  D(x): 0.4990    D(G(z)): 0.2305 / 0.3386
[11/20][0/8]    Loss_D: 0.6288  Loss_G: 1.9024  D(x): 0.6608    D(G(z)): 0.1591 / 0.2535
[12/20][0/8]    Loss_D: 0.1359  Loss_G: 4.5381  D(x): 0.9531    D(G(z)): 0.0783 / 0.0223
[13/20][0/8]    Loss_D: 2.6094  Loss_G: 1.4179  D(x): 0.4454    D(G(z)): 0.2862 / 0.4665
[14/20][0/8]    Loss_D: 0.8828  Loss_G: 1.6373  D(x): 0.5681    D(G(z)): 0.2417 / 0.2592
[15/20][0/8]    Loss_D: 0.9960  Loss_G: 2.6853  D(x): 0.9144    D(G(z)): 0.5305 / 0.1073
[16/20][0/8]    Loss_D: 0.9796  Loss_G: 1.7205  D(x): 0.4847    D(G(z)): 0.0346 / 0.3854
[17/20][0/8]    Loss_D: 2.1974  Loss_G: 0.7519  D(x): 0.3103    D(G(z)): 0.0101 / 0.6331
[18/20][0/8]    Loss_D: 1.7713  Loss_G: 0.6163  D(x): 0.2775    D(G(z)): 0.1005 / 0.6273
[19/20][0/8]    Loss_D: 1.6976  Loss_G: 1.8296  D(x): 0.4884    D(G(z)): 0.2979 / 0.4283
```



These are the real and fake images after the training process is completed in grayscale

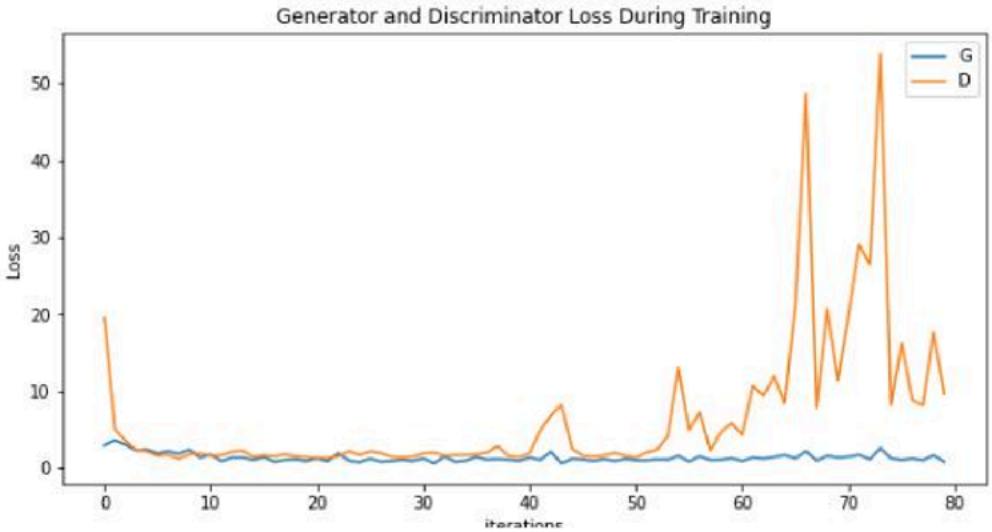


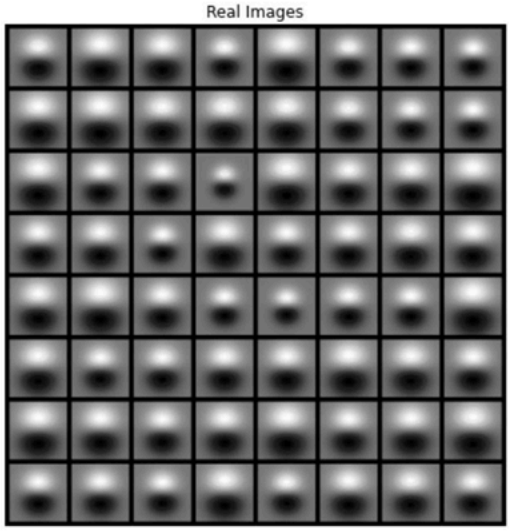
I have tried to still reduce the loss taking https://arxiv.org/pdf/1701.07875 as reference and below are the results for the same

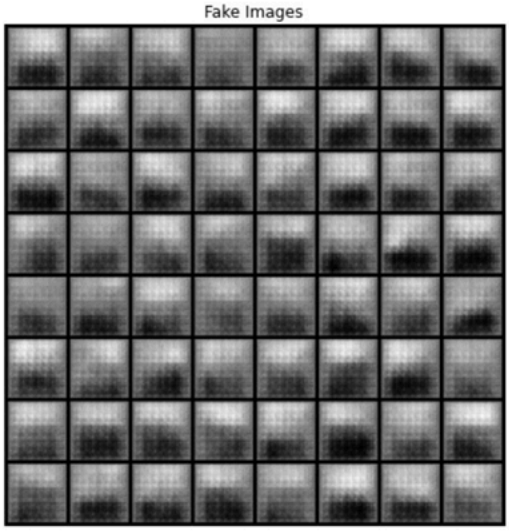
gradient Penalty Implementation :Gradient penalty enforces the gradient norm of the discriminator's output with respect to its input to be close to 1, which

improves the training stability.

```
Starting Training Loop...
[0/20][0/4]     Loss_D: 19.4908 Loss_G: 2.9714  D(x): 0.6639    D(G(z)): 0.5696 / 0.0890
[1/20][0/4]     Loss_D: 2.2034  Loss_G: 2.4409  D(x): 0.7129    D(G(z)): 0.5607 / 0.1083
[2/20][0/4]     Loss_D: 1.8247  Loss_G: 2.4280  D(x): 0.7559    D(G(z)): 0.5088 / 0.1128
[3/20][0/4]     Loss_D: 2.1197  Loss_G: 1.3579  D(x): 0.6609    D(G(z)): 0.5426 / 0.3117
[4/20][0/4]     Loss_D: 1.5328  Loss_G: 0.8078  D(x): 0.4385    D(G(z)): 0.3442 / 0.5097
[5/20][0/4]     Loss_D: 1.3105  Loss_G: 1.2490  D(x): 0.6218    D(G(z)): 0.4806 / 0.3022
[6/20][0/4]     Loss_D: 1.7117  Loss_G: 0.7501  D(x): 0.5757    D(G(z)): 0.5090 / 0.5487
[7/20][0/4]     Loss_D: 1.4048  Loss_G: 1.0796  D(x): 0.5732    D(G(z)): 0.4386 / 0.3577
[8/20][0/4]     Loss_D: 1.6061  Loss_G: 1.4942  D(x): 0.7218    D(G(z)): 0.5910 / 0.2548
[9/20][0/4]     Loss_D: 2.0251  Loss_G: 1.0883  D(x): 0.4803    D(G(z)): 0.3279 / 0.4343
[10/20][0/4]    Loss_D: 1.9574  Loss_G: 1.3659  D(x): 0.6446    D(G(z)): 0.4492 / 0.2735
[11/20][0/4]    Loss_D: 2.5484  Loss_G: 1.2109  D(x): 0.7792    D(G(z)): 0.6446 / 0.3325
[12/20][0/4]    Loss_D: 1.9992  Loss_G: 0.9138  D(x): 0.4980    D(G(z)): 0.3899 / 0.4594
[13/20][0/4]    Loss_D: 2.4139  Loss_G: 1.1145  D(x): 0.5908    D(G(z)): 0.4128 / 0.3484
[14/20][0/4]    Loss_D: 7.2742  Loss_G: 1.5325  D(x): 0.6811    D(G(z)): 0.5527 / 0.3129
[15/20][0/4]    Loss_D: 4.3198  Loss_G: 0.8985  D(x): 0.4923    D(G(z)): 0.3167 / 0.4462
[16/20][0/4]    Loss_D: 8.4449  Loss_G: 1.7003  D(x): 0.6799    D(G(z)): 0.3815 / 0.2322
[17/20][0/4]    Loss_D: 20.7096 Loss_G: 1.6002  D(x): 0.7447    D(G(z)): 0.5722 / 0.3049
[18/20][0/4]    Loss_D: 26.4406 Loss_G: 1.1309  D(x): 0.6141    D(G(z)): 0.3346 / 0.4028
[19/20][0/4]    Loss_D: 8.8437  Loss_G: 1.2502  D(x): 0.6497    D(G(z)): 0.4394 / 0.3096
```

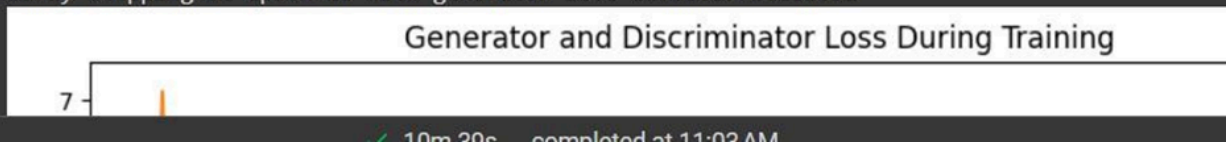




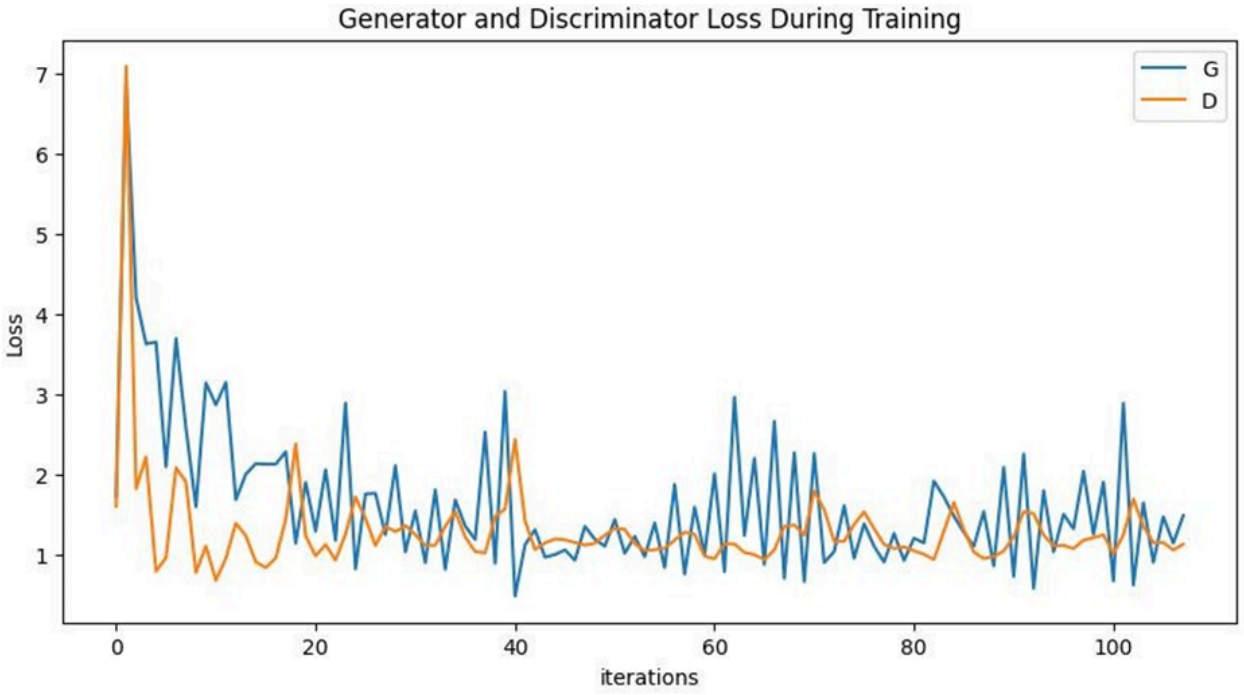


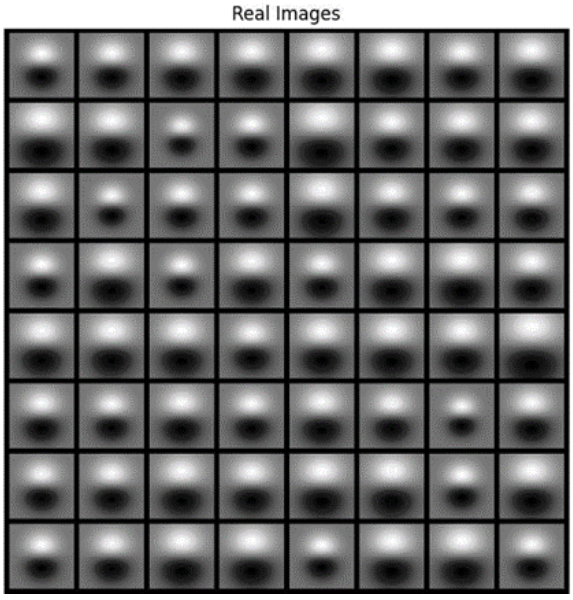
These are the results when I use early stopping with patience 10 and 40 iterations

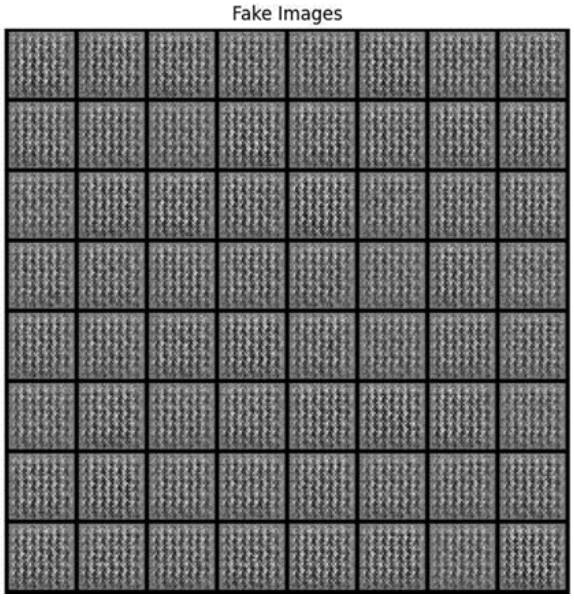
```
Starting Training Loop...
[0/40][0/4]     Loss_D: 1.6048  Loss_G: 1.7283  D(x): 0.3820    D(G(z)): 0.3670 / 0.2938
[1/40][0/4]     Loss_D: 0.7952  Loss_G: 3.6564  D(x): 0.7998    D(G(z)): 0.2797 / 0.0453
[2/40][0/4]     Loss_D: 0.7786  Loss_G: 1.6001  D(x): 0.6666    D(G(z)): 0.2303 / 0.2474
[3/40][0/4]     Loss_D: 1.3931  Loss_G: 1.6874  D(x): 0.5639    D(G(z)): 0.3768 / 0.2605
[4/40][0/4]     Loss_D: 0.9599  Loss_G: 2.1312  D(x): 0.6463    D(G(z)): 0.3471 / 0.2480
[5/40][0/4]     Loss_D: 0.9859  Loss_G: 1.2946  D(x): 0.5468    D(G(z)): 0.2615 / 0.3606
[6/40][0/4]     Loss_D: 1.7219  Loss_G: 0.8234  D(x): 0.2856    D(G(z)): 0.1465 / 0.5393
[7/40][0/4]     Loss_D: 1.2893  Loss_G: 2.1127  D(x): 0.6264    D(G(z)): 0.4697 / 0.2040
[8/40][0/4]     Loss_D: 1.1205  Loss_G: 1.8113  D(x): 0.7247    D(G(z)): 0.5205 / 0.2095
[9/40][0/4]     Loss_D: 1.0409  Loss_G: 1.1916  D(x): 0.6332    D(G(z)): 0.3877 / 0.3625
[10/40][0/4]    Loss_D: 2.4379  Loss_G: 0.4863  D(x): 0.2069    D(G(z)): 0.1770 / 0.6624
[11/40][0/4]    Loss_D: 1.1966  Loss_G: 1.0027  D(x): 0.6086    D(G(z)): 0.4806 / 0.3963
[12/40][0/4]    Loss_D: 1.1437  Loss_G: 1.1948  D(x): 0.5418    D(G(z)): 0.3622 / 0.3860
[13/40][0/4]    Loss_D: 1.1387  Loss_G: 1.2299  D(x): 0.6476    D(G(z)): 0.4583 / 0.3228
[14/40][0/4]    Loss_D: 1.1823  Loss_G: 1.8776  D(x): 0.7399    D(G(z)): 0.5321 / 0.1979
[15/40][0/4]    Loss_D: 0.9495  Loss_G: 2.0120  D(x): 0.7867    D(G(z)): 0.4836 / 0.1633
[16/40][0/4]    Loss_D: 1.0008  Loss_G: 2.2052  D(x): 0.8395    D(G(z)): 0.5014 / 0.1331
[17/40][0/4]    Loss_D: 1.3733  Loss_G: 2.2740  D(x): 0.8394    D(G(z)): 0.6282 / 0.1289
[18/40][0/4]    Loss_D: 1.1668  Loss_G: 1.0407  D(x): 0.6997    D(G(z)): 0.5018 / 0.4106
[19/40][0/4]    Loss_D: 1.3272  Loss_G: 1.1054  D(x): 0.5263    D(G(z)): 0.3758 / 0.3769
[20/40][0/4]    Loss_D: 1.0494  Loss_G: 1.2067  D(x): 0.6625    D(G(z)): 0.4501 / 0.3408
[21/40][0/4]    Loss_D: 1.6526  Loss_G: 1.4818  D(x): 0.5986    D(G(z)): 0.4697 / 0.3790
[22/40][0/4]    Loss_D: 0.9770  Loss_G: 0.8614  D(x): 0.5240    D(G(z)): 0.2692 / 0.4649
[23/40][0/4]    Loss_D: 1.5144  Loss_G: 0.5805  D(x): 0.2793    D(G(z)): 0.1398 / 0.5973
[24/40][0/4]    Loss_D: 1.0776  Loss_G: 1.3255  D(x): 0.5398    D(G(z)): 0.3087 / 0.3772
[25/40][0/4]    Loss_D: 1.0154  Loss_G: 0.6737  D(x): 0.5050    D(G(z)): 0.2349 / 0.5579
[26/40][0/4]    Loss_D: 1.1521  Loss_G: 0.9066  D(x): 0.4266    D(G(z)): 0.2296 / 0.4316
Early stopping at epoch 26 with generator loss 0.706827700138092
```



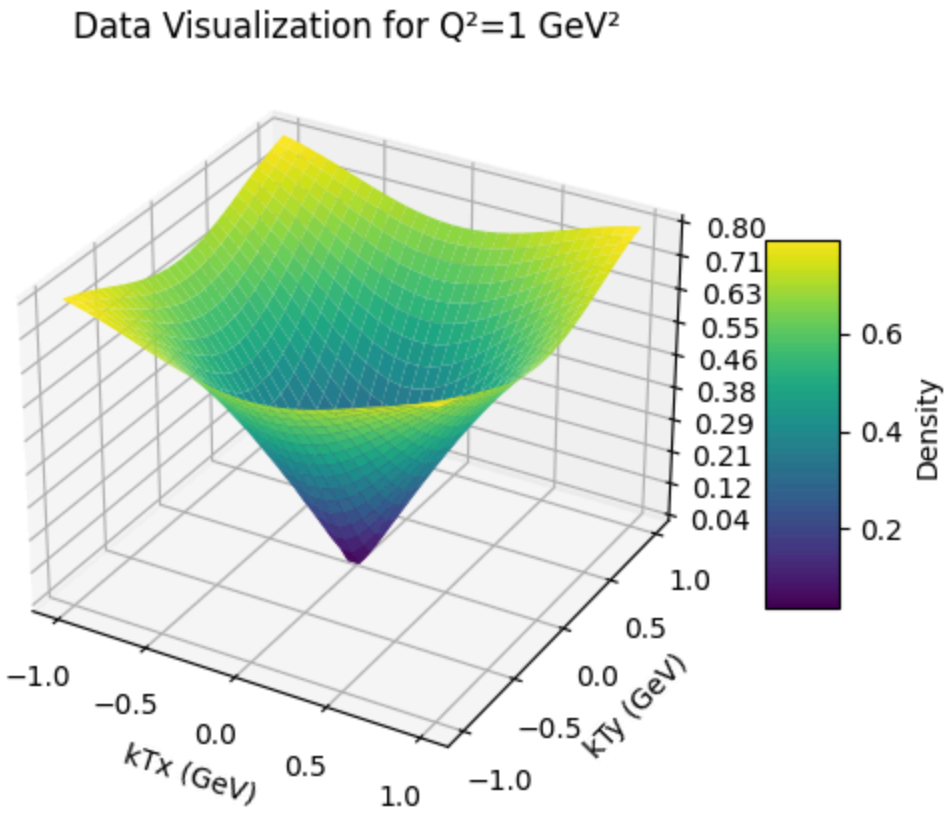
10m 20s completed at 11:02 AM

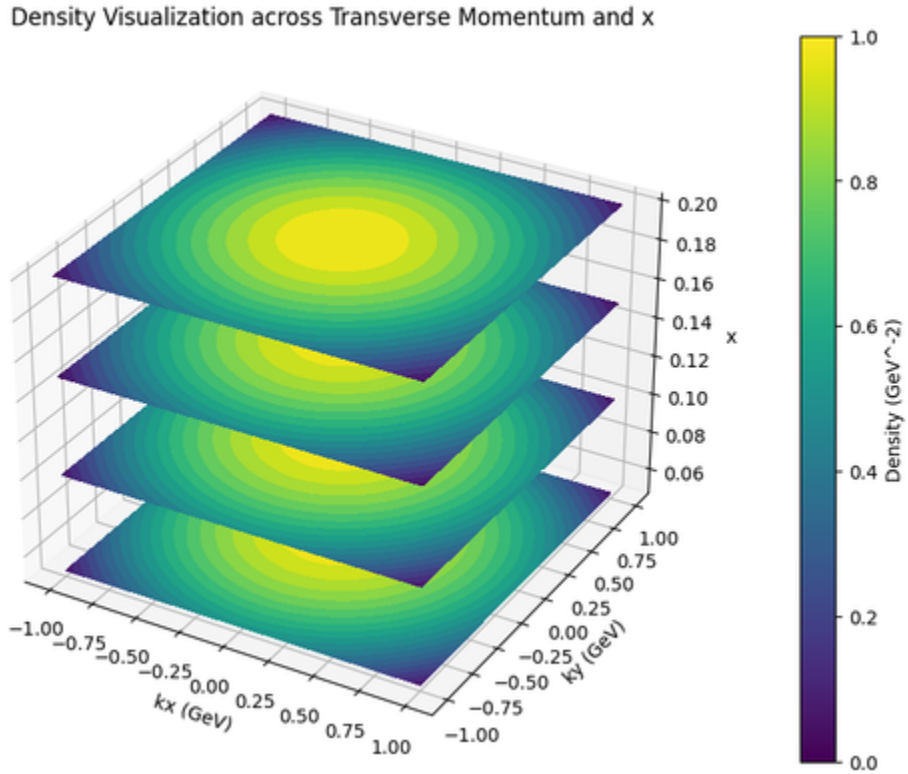


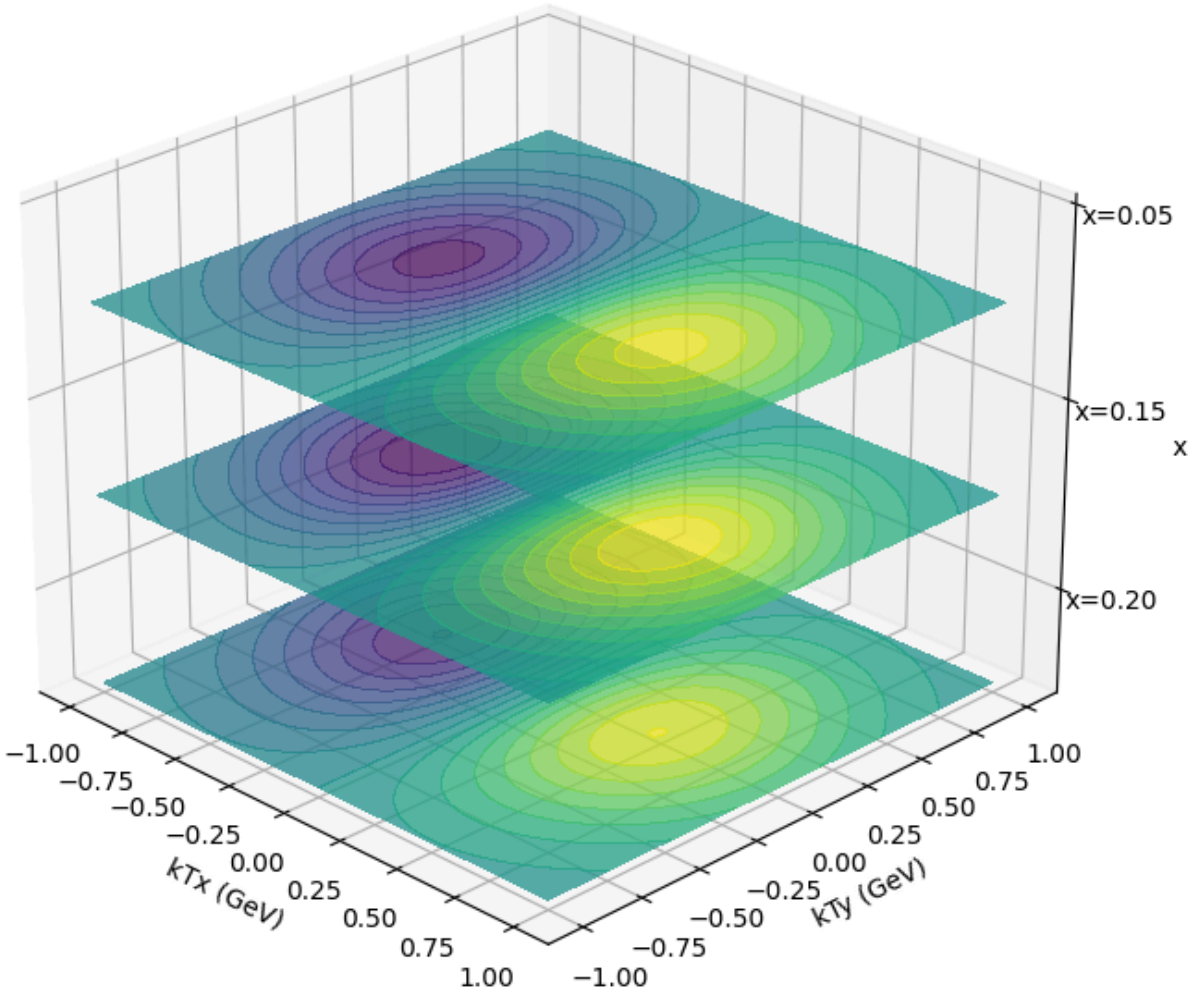


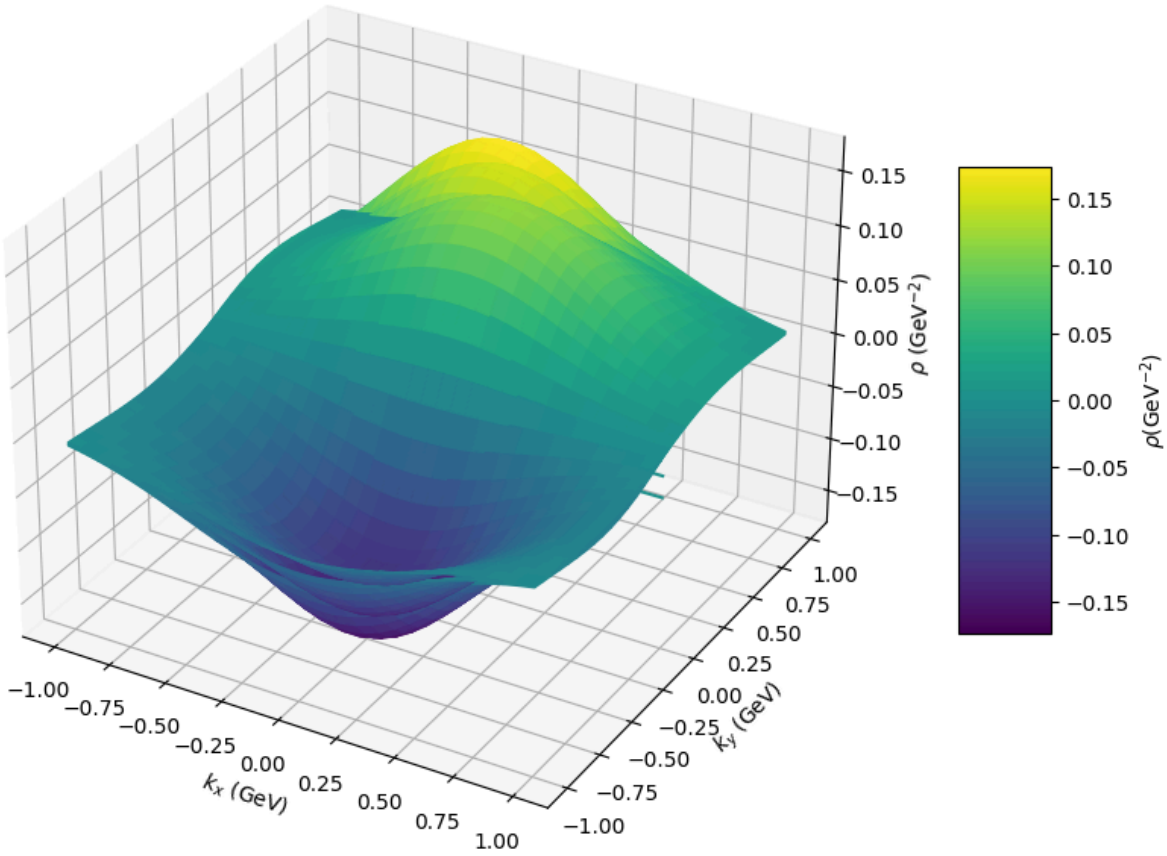


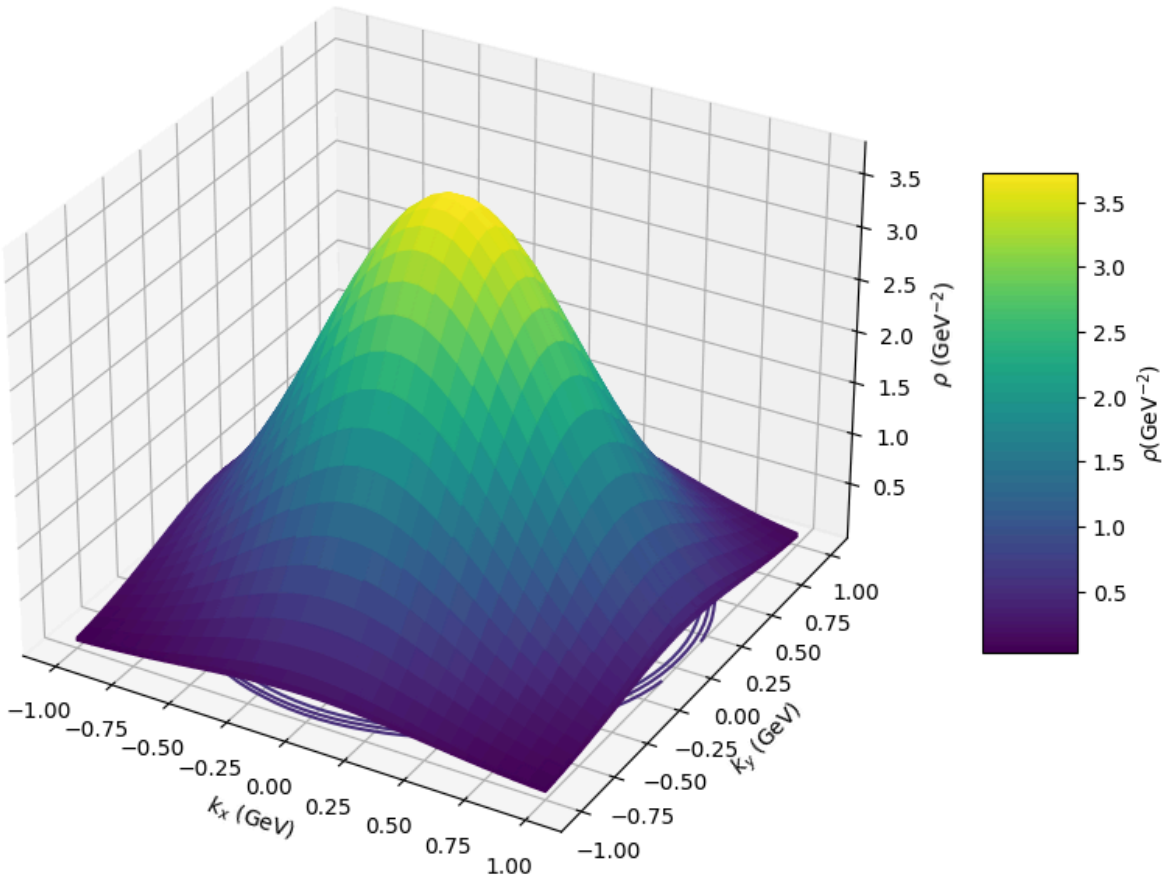
Below I have attached some best pictures (plots) but every plot along with all the trails of the plots are included in the code.

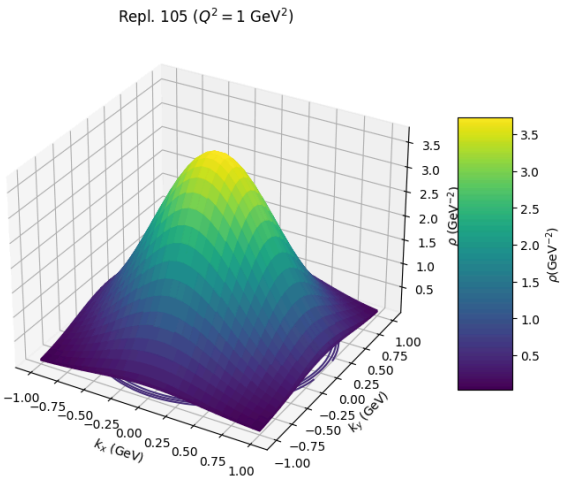

Data Visualization for Q²=1 GeV²
0.80
0.71
0.63
0.55
0.46
0.38
0.29
0.21
0.12
0.04
0.6
0.4
0.2
Density
1.0
0.5
0.0
−0.5
−1.0
kTy (GeV)
−1.0
−0.5
0.0
0.5
1.0
kTx (GeV)

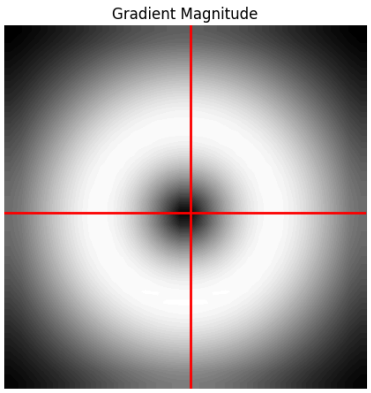

Density Visualization across Transverse Momentum and x
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
x
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
ky (GeV)
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
kx (GeV)
1.0
0.8
0.6
0.4
0.2
0.0
Density (GeV^-2)

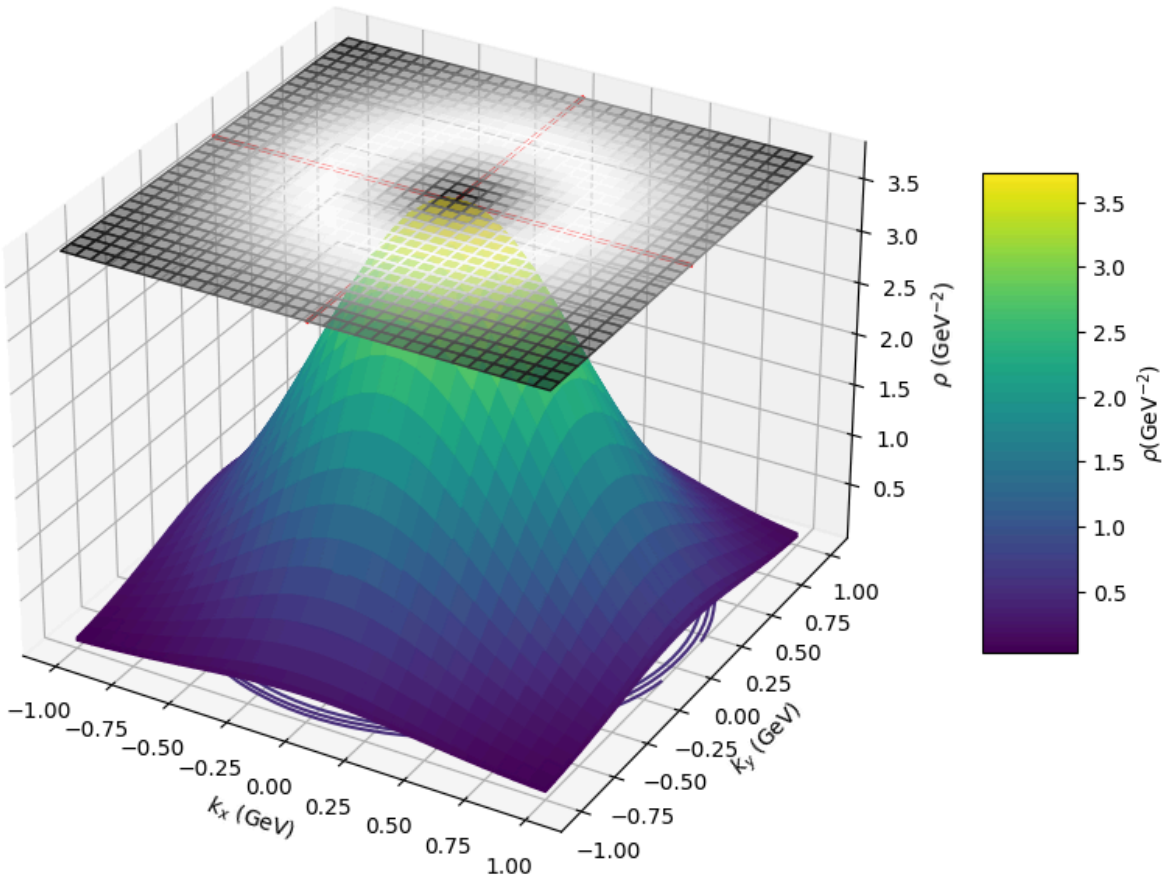

x=0.05
x=0.15
x=0.20
x
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
kTx (GeV)
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
kTy (GeV)

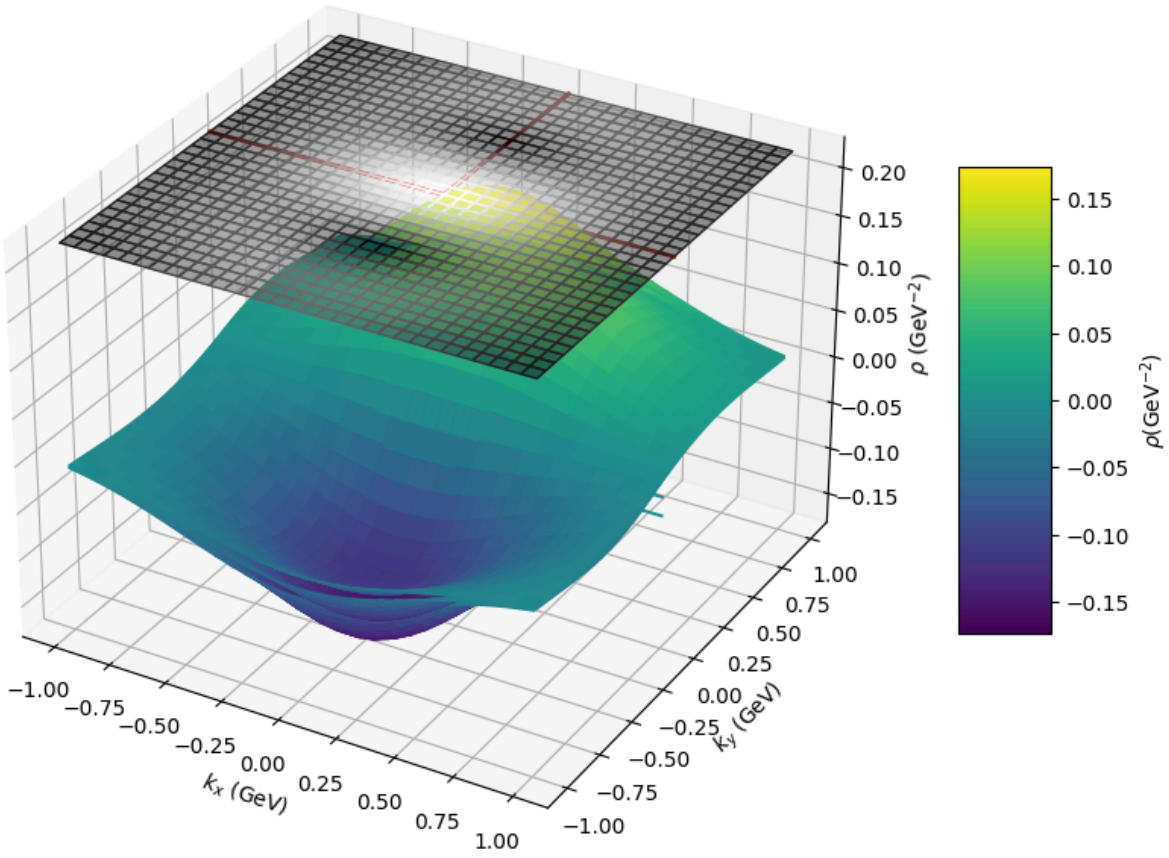

Repl. 105 ($Q^2 = 1$ GeV^2)
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
ρ (GeV^{-2})
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
k_x (GeV)
k_y (GeV)
ρ(GeV^{-2})

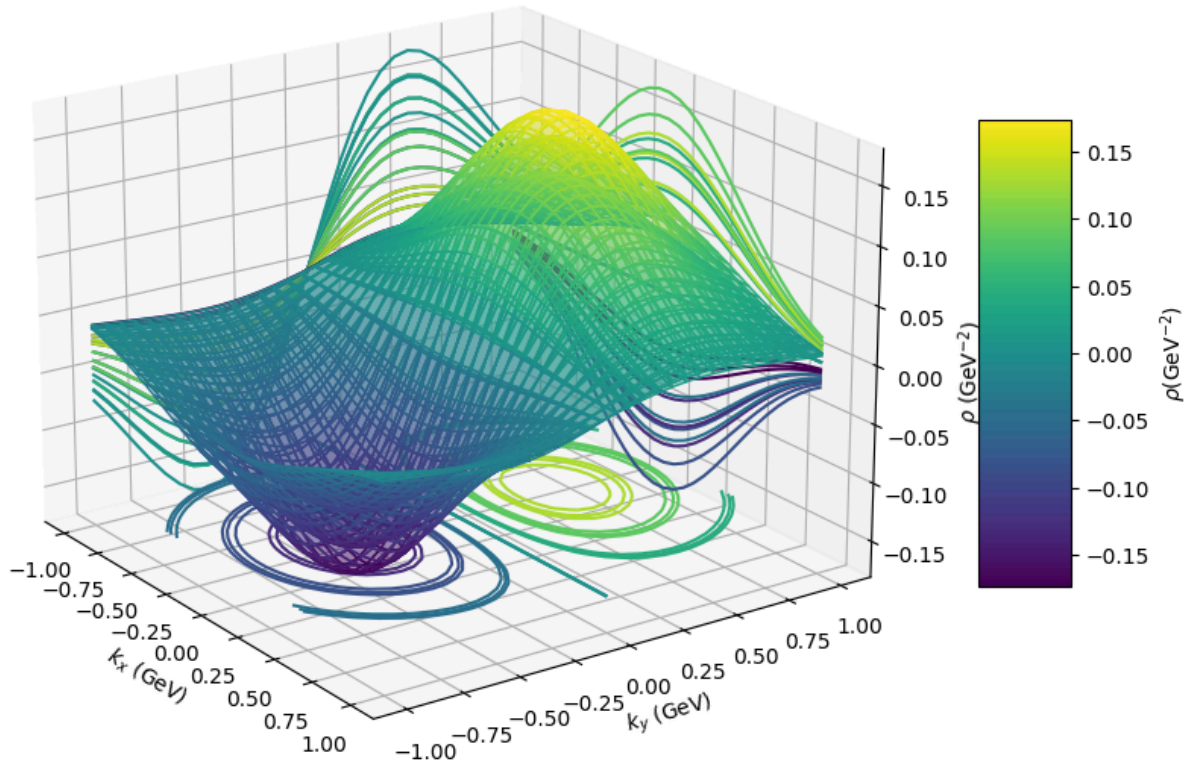

Repl. 105 ($Q^2 = 1$ GeV^2)
k_x (GeV)
k_y (GeV)
ρ (GeV^{-2})
ρ(GeV^{-2})

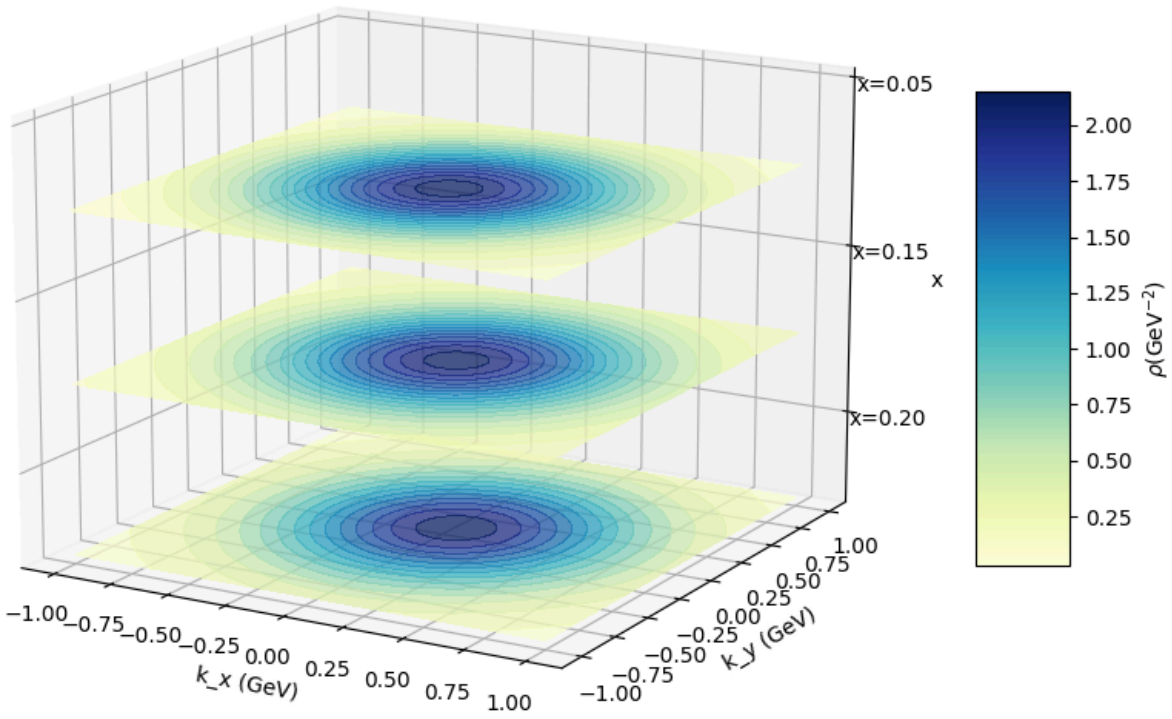

Repl. 105 ($Q^2 = 1$ GeV^2)
k_x (GeV)
k_y (GeV)
ρ (GeV^{-2})
ρ(GeV^{-2})

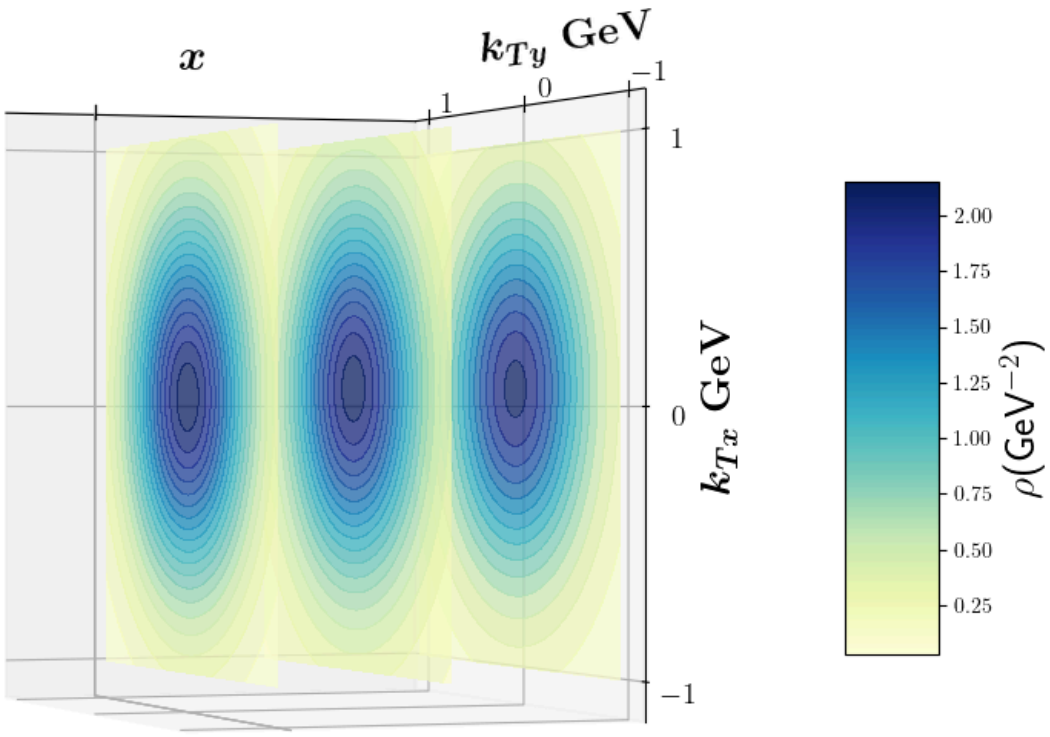

Gradient Magnitude

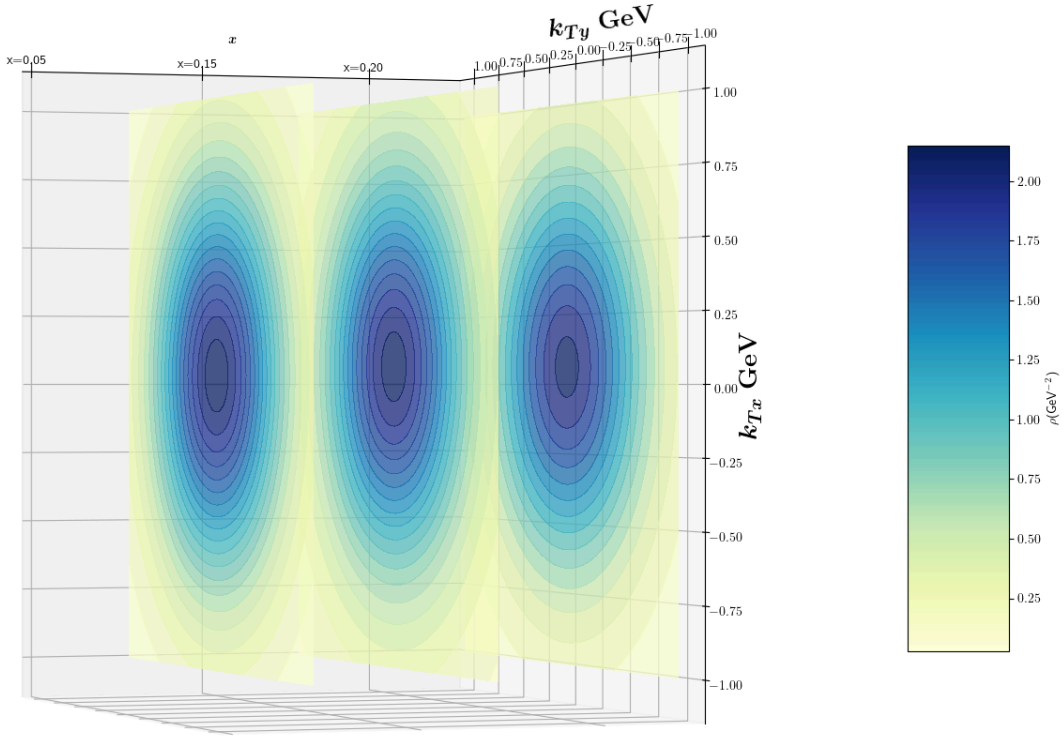

Repl. 105 ($Q^2 = 1$ GeV^2)
ρ (GeV^{-2})
k_x (GeV)
k_y (GeV)
ρ(GeV^{-2})

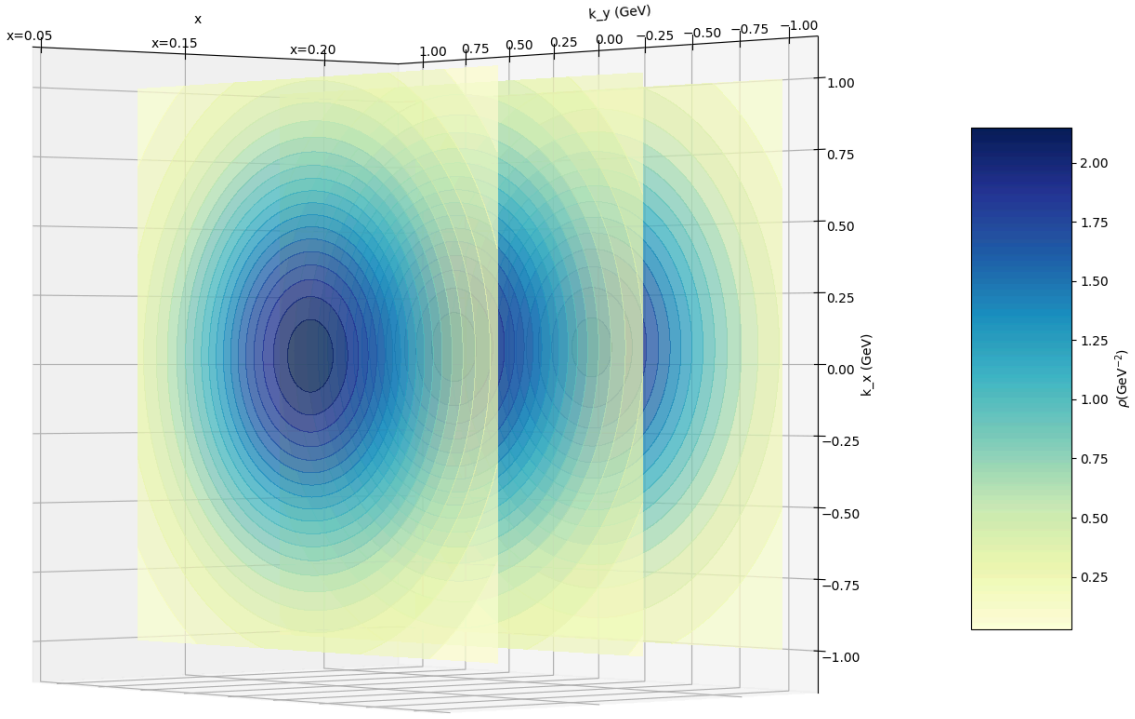

Repl. 105 ($Q^2 = 1$ GeV^2)
k_x (GeV)
k_y (GeV)
ρ (GeV^{-2})
ρ(GeV^{-2})


Repl. 105 ($Q^2 = 1$ GeV^2)
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
ρ (GeV^{-2})
ρ(GeV^{-2})
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
k_x (GeV)
k_y (GeV)


x=0.05
x=0.15
x=0.20
x
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
k_x (GeV)
k_y (GeV)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
ρ(GeV⁻²)


x
k_{Ty} GeV
1
0
−1
1
0
−1
k_{Tx} GeV
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
$\rho(\mathrm{GeV}^{-2})$


x
x=0.05
x=0.15
x=0.20
k_{Ty} GeV
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00
k_{Tx} GeV
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00
ρ(GeV^{-2})
2.00 1.75 1.50 1.25 1.00 0.75 0.50 0.25


x
x=0.05
x=0.15
x=0.20
k_y (GeV)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
k_x (GeV)
ρ(GeV⁻²)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25