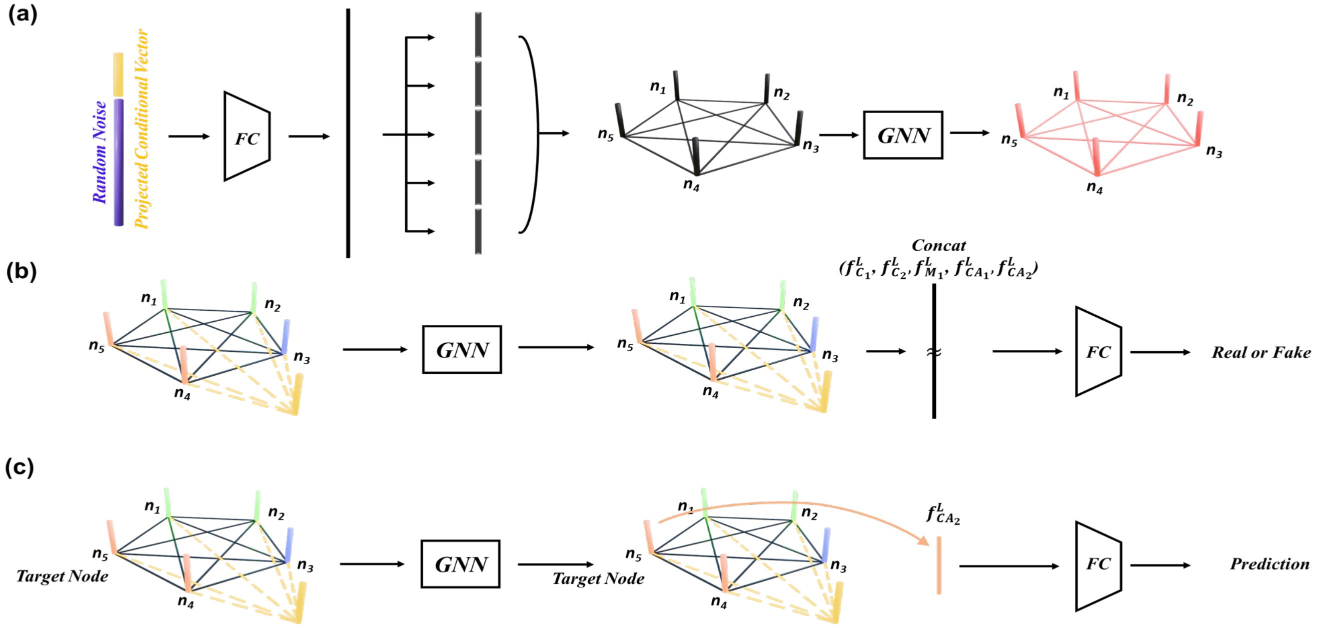


Figure 1: Generator and Discriminator. (a) shows how the generator generates a fake graph from random noise and conditional vector input. (b) shows the discriminator's graph-level task process, which discriminates whether the input graph is real or fake. After passing through $L$ GNN layers, all the node features of the final graph output are concatenated as a single vector. And we put it into the final fully connected layer. (c) shows the process of the discriminator's node-level task of predicting the target node value. After passing through GNN, the target node feature of the graph output is put into a fully connected layer.

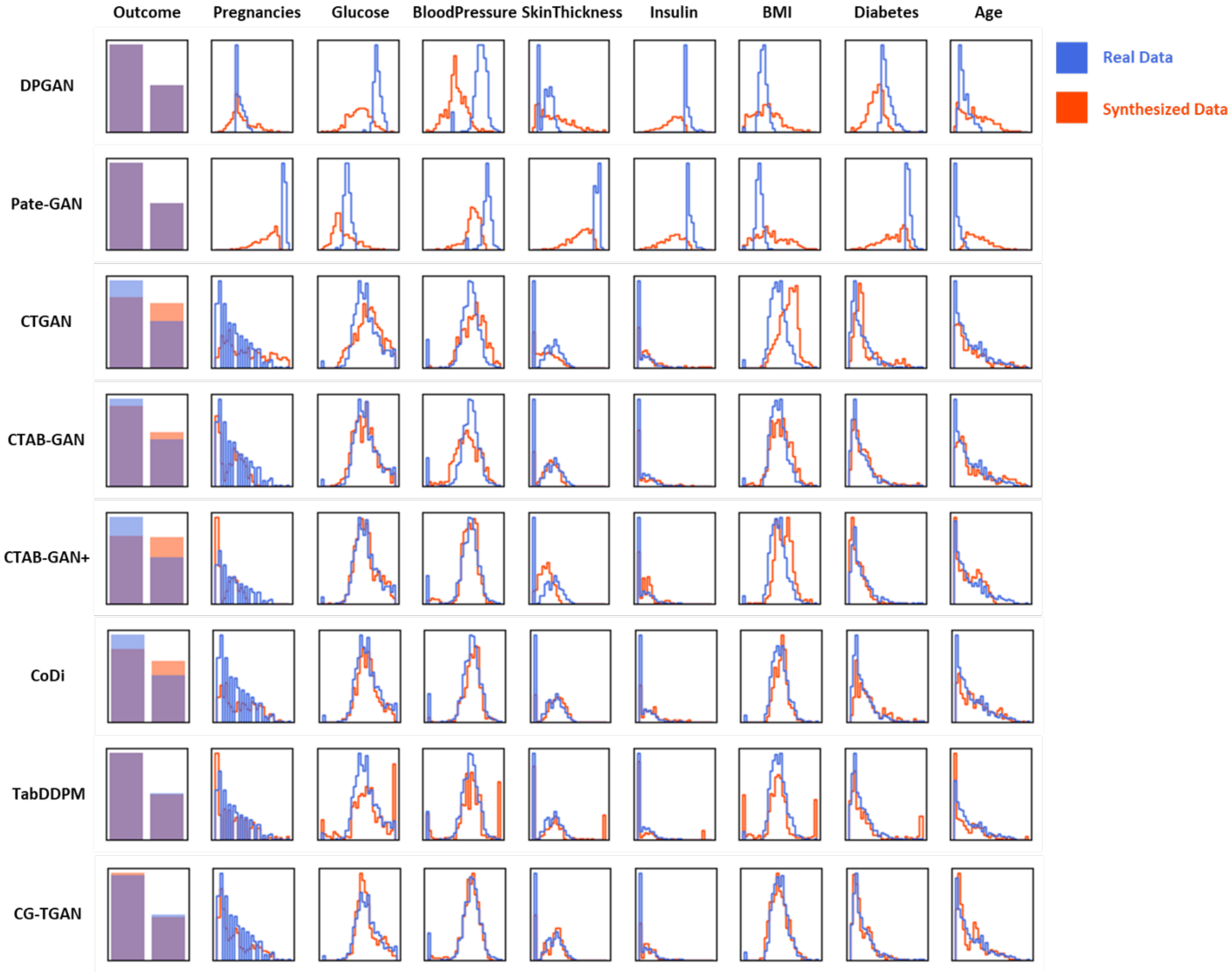
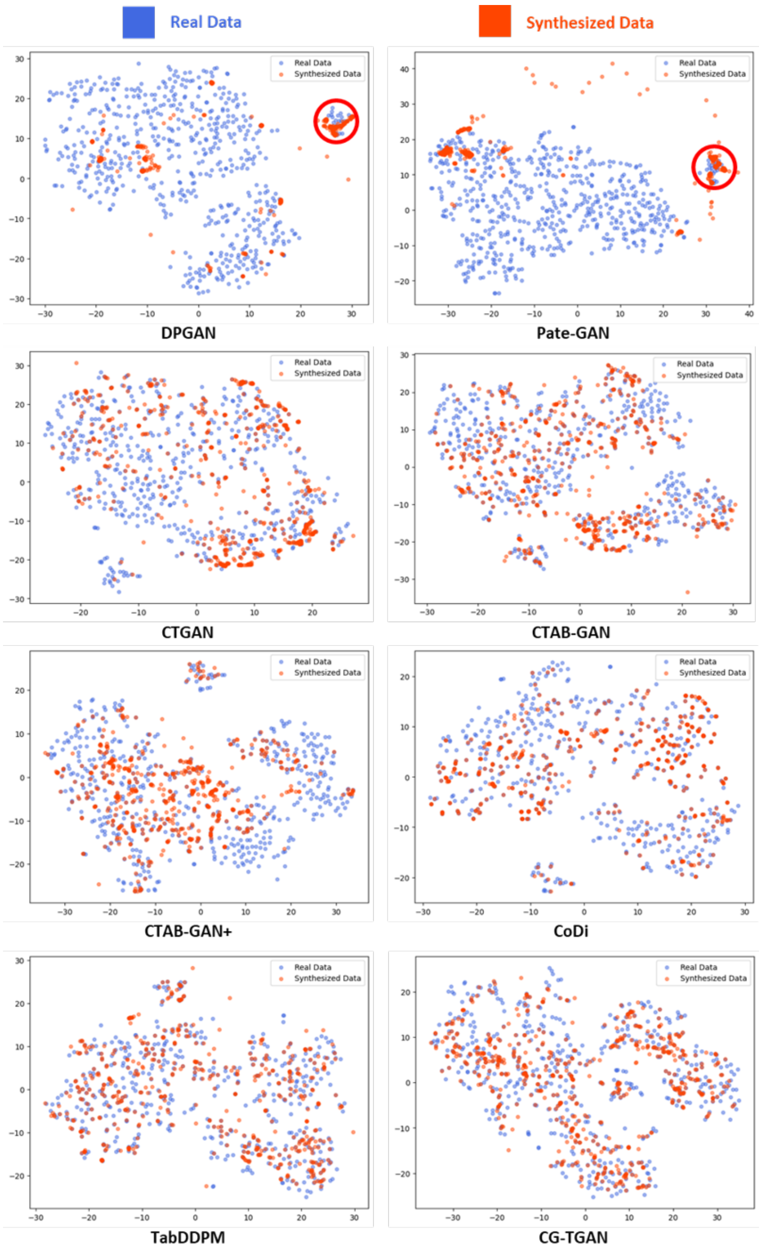


Figure 2: Comparison of synthetic data histograms. These are the data distributions synthesized by DPGAN, Pate-GAN, CTGAN, CTAB-GAN, CTAB-GAN+, CoDi, TabDDPM and CG-TGAN for the Diabetes dataset. The royal-blue color plot represents the real data distribution, and the orange-red color plot represents the synthesized data distribution. We can see that the data distribution synthesized by CG-TGAN best matches with the real data distribution.



DPGAN

Pate-GAN

CTGAN

CTAB-GAN

CTAB-GAN+

CoDi

TabDDPM

CG-TGAN

Figure 3: t-SNE projection on Diabetes dataset. The royal-blue color plot represents the real data distribution, and the orange-red color plot represents the synthesized data distribution. We can see that the data distribution synthesized by CG-TGAN best matches with the real data distribution.

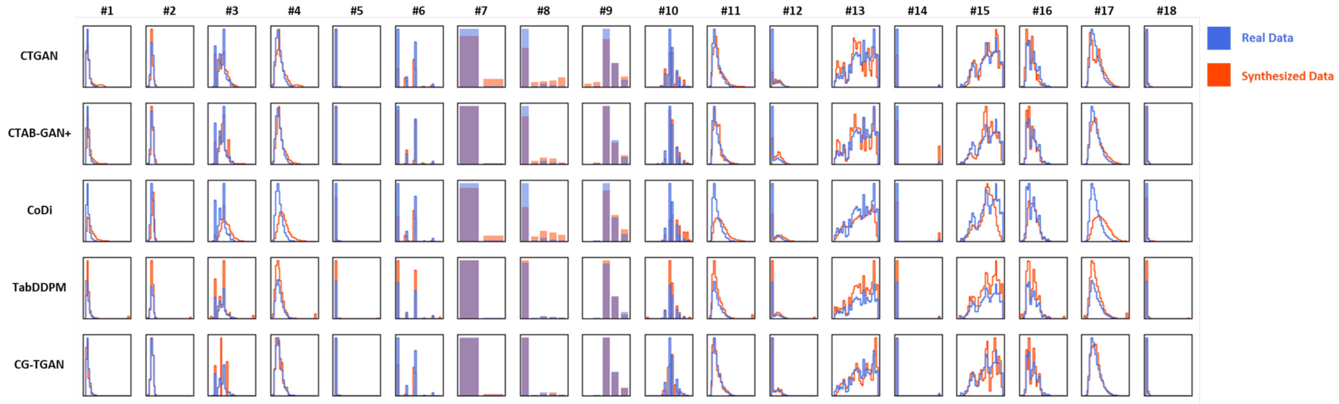


Figure 4: Comparison of synthetic data histograms. These are the data distributions synthesized by CTGAN, CTAB-GAN+, CoDi, TabDDPM and CG-TGAN for the Diabetes dataset. #1, Price. #2, Bedrooms. #3, Bathrooms. #4, SQFT Living. #5, SQFT Lot Mixed. #6, Floors. #7, Waterfront. #8, View. #9, Condition. #10, Grade. #11, SQFT Above. #12, SQFT Basement. #13, YR Built. #14, YR Renovated Mixed. #15, Lat. #16, Long. #17, SQFT Living15. #18, SQFT Lot15. The royal-blue color plot represents the real data distribution, and the orange-red color plot represents the synthesized data distribution. We can see that the data distribution synthesized by CG-TGAN best matches with the real data distribution.

Table 1: The datasets we used in our experiments. The N notation means the number of columns that were not used because they are meaningless.

| Dataset | #Train | #Test | #C | #M | #CA | #N | Task Type |
|---|---|---|---|---|---|---|---|
| Abalone | 3341 | 836 | 8 | 0 | 1 | 0 | RG |
| Insurance | 1070 | 268 | 4 | 0 | 3 | 0 | RG |
| King | 17290 | 4323 | 13 | 2 | 3 | 3 | RG |
| Adult | 32561 | 16281 | 4 | 2 | 9 | 0 | BC |
| Diabetes | 614 | 154 | 8 | 0 | 1 | 0 | BC |
| Gesture | 7898 | 1975 | 32 | 0 | 1 | 0 | MC |
| Wilt | 3871 | 968 | 5 | 0 | 1 | 0 | BC |

Table 2: CG-TGAN hyperparameters used in the experiments for each dataset. $n_{dg}$, the number of iterations of the discriminator's graph-level task per update. $n_{update}$, the number of updates for CG-TGAN learning. $b$, the batch size.

|  | Number of GC | Embedding Dimensions | Learning Rate | $n_{dg}$ | $n_{update}$ | $b$ |
|---|---|---|---|---|---|---|
| Abalone | 3 | {64, 128, 64} | 1e-4 | 5 | 5000 | 256 |
| Insurance | 3 | {64, 128, 64} | 1e-4 | 5 | 5000 | 256 |
| King | 3 | {64, 128, 64} | 1e-4 | 5 | 5000 | 256 |
| Adult | 4 | {64, 128, 128, 64} | 1e-4 | 5 | 5000 | 256 |
| Diabetes | 3 | {64, 128, 64} | 1e-4 | 5 | 5000 | 256 |
| Gesture | 3 | {64, 128, 64} | 1e-4 | 5 | 5000 | 256 |
| Wilt | 4 | {64, 128, 128, 64} | 1e-4 | 5 | 5000 | 256 |