CTGAN Model 1(base model without tuning)

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=128,

generator\_dim=(256, 256),

discriminator\_dim=(256, 256),

generator\_lr=2e-4,

generator\_decay=1e-6,

discriminator\_lr=2e-4,

discriminator\_decay=1e-6,

batch\_size=500,

discriminator\_steps=1,

log\_frequency=True,

verbose=False,

epochs=300,

pac=10,

cuda=True,

):

assert batch\_size % 2 == 0

CTGAN Model 2

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=256,

generator\_dim=(512, 512, 256),

discriminator\_dim=(512, 512, 256),

# Lowered the generator LR slightly

generator\_lr=5e-5,

generator\_decay=1e-6,

# Keep discriminator LR as is or tune as needed

discriminator\_lr=1e-4,

discriminator\_decay=1e-6,

# Reduced batch size to 128 for potentially better discriminator feedback

batch\_size=128,

discriminator\_steps=1,

log\_frequency=True,

# Turn on verbose to track training more closely

verbose=True,

# Increased epochs to allow more training time

epochs=2000,

# Reduced PAC from 10 to 5 to help capture sharper correlations

pac=5,

cuda=True,

):

assert batch\_size % 2 == 0

CTGAN Model 3

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=256,

generator\_dim=(512, 512, 256),

discriminator\_dim=(512, 512, 256),

# Slightly increased the generator LR to balance with the discriminator

generator\_lr=1e-4,

generator\_decay=1e-6,

# Lowered the discriminator LR to keep the generator from underfitting correlations

discriminator\_lr=5e-5,

discriminator\_decay=1e-6,

# Keep batch size at 128 for stable discriminator feedback

batch\_size=128,

# Increase discriminator steps to 2, helping refine correlations

discriminator\_steps=2,

log\_frequency=True,

verbose=True,

# Extend epochs for more training time

epochs=2500,

# Lower pac to 3 to capture sharper distinctions in multi‐categorical data

pac=3,

cuda=True,

):

assert batch\_size % 2 == 0

CTGAN Model 4

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=256,

generator\_dim=(512, 512, 256),

discriminator\_dim=(512, 512, 256),

# Keep generator LR at 1e-4 to maintain sufficient capacity to learn correlations

generator\_lr=1e-4,

generator\_decay=1e-6,

# Lower discriminator LR a bit more so the generator isn’t overpowered

discriminator\_lr=3e-5,

discriminator\_decay=1e-6,

# Keep batch size at 128 for balanced training

batch\_size=128,

# Keep 2 discriminator steps to refine correlations

discriminator\_steps=2,

log\_frequency=True,

verbose=True,

# Increase total epochs to allow more time for improved correlation matching

epochs=3000,

# Lower PAC to 2 to potentially capture more nuanced interactions

pac=2,

cuda=True,

):

assert batch\_size % 2 == 0

CTGAN Model 5

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=256,

generator\_dim=(512, 512, 256),

discriminator\_dim=(512, 512, 256),

# Keep generator LR at 1e-4 to maintain sufficient capacity to learn correlations

generator\_lr=1e-4,

generator\_decay=1e-6,

# Lower discriminator LR a bit more so the generator isn’t overpowered

discriminator\_lr=3e-5,

discriminator\_decay=1e-6,

# Keep batch size at 128 for balanced training

batch\_size=128,

# Keep 2 discriminator steps to refine correlations

discriminator\_steps=2,

log\_frequency=True,

verbose=True,

# Increase total epochs to allow more time for improved correlation matching

epochs=3000,

# Lower PAC to 2 to potentially capture more nuanced interactions

pac=2,

cuda=True,

):

assert batch\_size % 2 == 0

CTGAN Model 6

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=256,

generator\_dim=(512, 512, 256),

discriminator\_dim=(512, 512, 256),

# Keep a moderately higher generator LR to maintain correlation gains

generator\_lr=2e-4,

generator\_decay=1e-6,

# Keep the lower discriminator LR so the generator isn't overpowered

discriminator\_lr=2e-5,

discriminator\_decay=1e-6,

# Use batch size 128 for balanced updates

batch\_size=128,

# Consider raising discriminator\_steps to 3 for better pairwise fidelity

discriminator\_steps=3,

log\_frequency=True,

verbose=True,

# Use 3000 epochs unless you add early stopping

epochs=3000,

# Revert pac to 2 for more stable categorical learning

pac=2,

cuda=True,

):

assert batch\_size % 2 == 0

CTGAN Model 7

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=256,

generator\_dim=(512, 512, 256),

discriminator\_dim=(512, 512, 256),

# Slightly raise generator LR to give more capacity for capturing correlations

generator\_lr=3e-4,

generator\_decay=1e-6,

# Keep or lower discriminator LR if the correlation is still lagging

discriminator\_lr=1e-5,

discriminator\_decay=1e-6,

# Consider a smaller batch for more frequent updates, but watch stability

batch\_size=64,

# Keep 3 discriminator steps if stable or revert to 2 if training is too slow

discriminator\_steps=3,

log\_frequency=True,

verbose=True,

epochs=3000,

# Keep pac=2 for stable multi‐categorical learning

pac=2,

cuda=True,

):

assert batch\_size % 2 == 0

TVAE Model 1:(base model without tuning)

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=128,

compress\_dims=(128, 128),

decompress\_dims=(128, 128),

l2scale=1e-5,

batch\_size=500,

epochs=300,

loss\_factor=2,

cuda=True,

verbose=False,

):

TVAE Model 2:

Hyperparameters:

def \_\_init\_\_(

self,

# Increase the embedding dimension to capture more complexity

embedding\_dim=256,

# Expand hidden layers to better model high-cardinality columns

compress\_dims=(256, 256, 128),

decompress\_dims=(128, 256, 256),

# Slightly lower L2 regularization for more flexible fitting

l2scale=1e-6,

# Reduce batch size for more frequent updates and better correlation capture

batch\_size=128,

# Extend training epochs for more thorough convergence

epochs=1000,

# Adjust loss factor if numeric columns need more/less emphasis

loss\_factor=1, # or keep at 2; experiment to see which is better

cuda=True,

# Turn verbose=True to monitor training progress if you like

verbose=True,

):

Pass

TVAE Model 3:

Hyperparameters:

def \_\_init\_\_(

self,

# Further increase embedding\_dim to better represent high-cardinality data

embedding\_dim=512,

# Add more layers / capacity in compress & decompress to learn complex distributions

compress\_dims=(512, 512, 256, 128),

decompress\_dims=(128, 256, 512, 512),

# Lower L2 regularization so the model can fit the data more flexibly

l2scale=1e-7,

# Keep batch size moderate; 128 is often a good balance

batch\_size=128,

# Increase training epochs to allow more thorough convergence

epochs=1500,

# If numeric columns need more emphasis, consider higher loss\_factor (e.g., 2)

loss\_factor=2,

cuda=True,

verbose=True,

):

Pass

TVAE Model 4:

Hyperparameters:

def \_\_init\_\_(

self,

# Keep a large embedding\_dim for complex categorical data

embedding\_dim=512,

# Maintain a high-capacity architecture but slightly reduce the layer depth

compress\_dims=(512, 256, 256),

decompress\_dims=(256, 256, 512),

# Reduce L2 regularization further for more flexibility

l2scale=1e-8,

# Use a smaller batch size for more frequent updates

batch\_size=64,

# Increase epochs to allow thorough convergence

epochs=2000,

# Keep or adjust loss\_factor (e.g., 2) if numeric columns need emphasis

loss\_factor=2,

cuda=True,

verbose=True,

):

Pass

TVAE Model 5

Hyperparameters:

def \_\_init\_\_(

self,

# Keep a large embedding\_dim for complex categorical data

embedding\_dim=512,

# Maintain a high-capacity architecture but slightly reduce the layer depth

compress\_dims=(512, 256, 256),

decompress\_dims=(256, 256, 512),

# Reduce L2 regularization further for more flexibility

l2scale=1e-8,

# Use a smaller batch size for more frequent updates

batch\_size=64,

# Increase epochs to allow thorough convergence

epochs=2000,

# Keep or adjust loss\_factor (e.g., 2) if numeric columns need emphasis

loss\_factor=2,

cuda=True,

verbose=True,

):

Pass

TVAE Model 6:

Hyperparameters:

def \_\_init\_\_(

self,

# Keep the large embedding dimension for complex categorical data

embedding\_dim=512,

# If you want to push capacity further, add or expand layers slightly

compress\_dims=(512, 512, 256, 128),

decompress\_dims=(128, 256, 512, 512),

# Slightly reduce L2 regularization for more flexibility

l2scale=1e-9,

# Lower batch size further for more frequent updates (monitor stability)

batch\_size=32,

# Retain or extend epochs; consider early stopping if it stabilizes early

epochs=3000,

# Keeping loss\_factor=2 can help preserve numeric correlation

loss\_factor=2,

cuda=True,

verbose=True,

):

Pass

TVAE Model 7:

Hyperparameters:

def \_\_init\_\_(

self,

embedding\_dim=512,

# Keep strong capacity but reduce depth slightly to avoid overfitting

compress\_dims=(512, 512, 256),

decompress\_dims=(256, 512, 512),

# Increase L2 regularization back to reduce overfitting of numeric correlation

l2scale=5e-9, # or try 1e-8 if you see too much overfitting

# Raise batch size for more stable category frequencies

batch\_size=64,

# Keep or reduce epochs if you suspect overfitting; try early stopping

epochs=3000,

# Keep loss\_factor=2 for numeric emphasis

loss\_factor=2,

cuda=True,

verbose=True,

):

Pass

CTABGAN Model 1:(base model without tuning)

Hyperparameters:

def \_\_init\_\_(self,

class\_dim=(256, 256, 256, 256),

random\_dim=100,

num\_channels=64,

l2scale=1e-5,

batch\_size=500,

epochs=150):

CTABGAN Model 2:

Hyperparameters:

class CTABGANSynthesizer:

def \_\_init\_\_(self,

# Increase the network capacity for better nuanced distributions

class\_dim=(512, 512, 256, 256),

# Expand latent space to capture more complex patterns

random\_dim=128,

# Keep num\_channels the same unless you see underfitting

num\_channels=64,

# Slightly reduce L2 regularization to help match correlation better

l2scale=1e-6,

# Lower batch size for more frequent updates (monitor stability)

batch\_size=256,

# Increase epochs for a more thorough training

epochs=300)

CTABGAN Model 3:

Hyperparameters:

class CTABGANSynthesizer:

def \_\_init\_\_(self,

# Slightly shallower capacity to reduce correlation overshoot

class\_dim=(512, 256, 256),

# Keep random\_dim as 128 if diversity is good

random\_dim=128,

# Keep num\_channels if not underfitting

num\_channels=64,

# Increase L2 regularization to curb overemphasis on correlation

l2scale=5e-6,

# Lower batch size for more frequent updates (monitor stability)

batch\_size=128,

# Try adding more epochs or keep 300; if overshoot persists, reduce

epochs=300):

...

CTABGAN Model 4:

Hyperparameters:

class CTABGANSynthesizer:

def \_\_init\_\_(

self,

# Narrow the model layers further to reduce overshoot

class\_dim=(256, 256, 256),

# Keep random\_dim if diversity is good; or lower to 64 if you see overfitting

random\_dim=128,

num\_channels=64,

# Increase L2 further to constrain correlation overshoot

l2scale=1e-5,

# Use a moderate batch size; if stable, you can go even smaller

batch\_size=128,

# Keep or reduce epochs to limit overfitting; consider early stopping

epochs=200

):

...

CTABGAN 5 Model:

Hyperparameters:

class CTABGANSynthesizer:

def \_\_init\_\_(

self,

# Further reduce capacity to avoid excessive correlation overshoot

class\_dim=(256, 128, 128),

# Lower random\_dim if you suspect high diversity is fueling correlation overshoot

random\_dim=64,

num\_channels=64,

# Slightly raise L2 regularization again to constrain correlation

l2scale=2e-5,

# Consider a smaller batch if stable, or keep 128

batch\_size=128,

# Possibly reduce epochs to 150 or 100, or add early stopping

epochs=150

):

...

CTABGAN Model 6:

Hyperparameters:

class CTABGANSynthesizer:

def \_\_init\_\_(

self,

# Even narrower layers to reduce numeric correlation overshoot

class\_dim=(128, 128, 64),

# Keep or reduce random\_dim if correlation remains overemphasized

random\_dim=64,

num\_channels=64,

# Increase L2 further to curb correlation overshoot

l2scale=5e-5,

# Possibly keep batch\_size at 128; smaller can help if stable

batch\_size=128,

# Reduce epochs to 100 or incorporate early stopping

epochs=100

):

...

CTABGAN Model 7:

Hyperparameters:

class CTABGANSynthesizer:

def \_\_init\_\_(

self,

# Even more narrow to avoid strong correlation overshoot

class\_dim=(128, 64, 64),

# Potentially reduce random\_dim further if correlation remains overshoot

random\_dim=32,

num\_channels=64,

# Raise L2 again to strongly curb correlation overshoot

l2scale=1e-4,

# Try smaller batch for more frequent updates (watch for stability)

batch\_size=64,

# Fewer epochs or add early stopping to keep correlation from drifting

epochs=80

):

...

**CTAB-GAN+ Hyperparameter Tuning and Results Interpretation**

**1. Baseline Model (Model 1)**

We began with a CTAB-GAN+ configuration that closely followed the recommendations from the original implementation, using epochs=150, batch\_size=500, and default values for other parameters. This model served as our baseline.

* **Outcome**: The synthetic data achieved 100% validity and structure, with an overall SDV Quality Score of 0.93. The univariate fidelity (Column Shapes ~94.69%) and pairwise trends (~91.15%) were high. However, the Pearson correlation between *Year of diagnosis* and *Survival months* in the synthetic data (-0.463) overshot the real value (-0.420).
* **Conclusion**: The model produced excellent fidelity but slightly exaggerated the negative correlation. Our goal in subsequent iterations was to preserve fidelity while better aligning the synthetic correlation with the real data.

**2. Model 2: Increased Capacity and Training Time**

We expanded class\_dim and random\_dim, reduced batch\_size, and increased epochs to 300 to allow the model more expressive power and convergence time.

* **Outcome**: Quality scores improved (0.94), but the correlation overshoot worsened to -0.523. Univariate distributions (~95.26%) and column pair trends (~92.1%) showed gains.
* **Conclusion**: While this configuration captured distributions more precisely, it led to stronger overfitting of numeric relationships.

**3. Model 3: Regularization and Reduced Batch Size**

We slightly reduced the model depth and increased L2 regularization (l2scale=5e-6). Batch size was lowered to 128 to allow more frequent updates.

* **Outcome**: The correlation improved slightly to -0.518. Distributional fidelity remained consistent (~93.53%).
* **Conclusion**: Regularization helped limit the correlation overshoot while maintaining high fidelity.

**4. Model 4: Further Constraint of Capacity and Epochs**

We narrowed the model’s structure (class\_dim=(256, 256, 256)), increased L2 to 1e-5, and reduced training epochs to 200.

* **Outcome**: This run yielded our best overall performance: Quality Score of 0.94, column shapes ~95.96%, and pair trends ~92.56%. The correlation overshoot was reduced to -0.460.
* **Conclusion**: This configuration best balanced realism and fidelity. It came closest to matching the true correlation without sacrificing structural accuracy.

**5. Model 5: Latent Space Reduction**

To further refine correlation, we reduced random\_dim to 64 and increased l2scale to 2e-5, while maintaining a compact model and 150 epochs.

* **Outcome**: Quality remained high (0.94), with a correlation of -0.469. Some minor dips in fidelity scores were noted.
* **Conclusion**: Shrinking the latent space and increasing regularization modestly improved the correlation alignment.

**6. Model 6: Aggressive Regularization and Epoch Reduction**

We further reduced capacity (class\_dim=(128, 128, 64)), increased regularization (l2scale=5e-5), and shortened training to 100 epochs.

* **Outcome**: Quality dipped slightly (0.93), and correlation overshot to -0.477. While still strong, both distributional and correlation metrics showed marginal decline.
* **Conclusion**: These hyperparameters may have led to underfitting. The correlation was not effectively corrected, indicating a potential lower limit of model compression.

**7. Model 7: Final Refinement**

We implemented our most constrained configuration: class\_dim=(128, 64, 64), random\_dim=32, l2scale=1e-4, batch\_size=64, and epochs=80.

* **Outcome**: We observed modest quality recovery (~93.27%) and a correlation of -0.473. Univariate and bivariate fidelity remained high (Column Shapes ~95.36%, Pair Trends ~91.18%).
* **Conclusion**: While the correlation continued to overshoot, it was within an acceptable range and the model retained strong data fidelity. This final configuration demonstrated the limits of further compression without compromising synthetic quality.

**8. Summary of Tuning Strategy and Rationale**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Reason for Tuning | Observed Effect |
| class\_dim | Controlled model capacity to reduce overfitting | Helped mitigate correlation overshoot |
| random\_dim | Adjusted latent diversity | Smaller values reduced exaggerated correlation but risked underfitting |
| l2scale | Increased regularization | Stronger L2 penalties dampened correlation overshoot |
| batch\_size | Enabled more frequent updates | Helped refine category frequencies and detail fidelity |
| epochs | Shortened to avoid late-stage correlation drift | Balanced convergence with correlation control |

**🧠 CTGAN: Interpretation and Rationale for Hyperparameter Tuning**

**1. Baseline (Model 1)**

* **Validity & Structure:** 100%
* **Quality Score:** ~88.76%
* **Correlation (Year vs. Survival):** -0.290
* **Observations:**
  + Strong univariate fidelity (~91.75%)
  + Moderate pairwise trend capture (~85.76%)
  + Underestimated numeric correlation between “Year of diagnosis” and “Survival months”

**🔧 Action Taken for Model 2**

* **Changes:**
  + Reduced pac from 10 → 5
  + Increased epochs to 2000
  + Lowered batch\_size to 128
  + Lowered generator\_lr to 5e-5
* **Rationale:**
  + Lower pac enables sharper distribution capture
  + More epochs allow finer training
  + Smaller batch improves discriminator feedback
  + Lower generator LR avoids instability while learning nuanced patterns
* **Results:**
  + Correlation improved to -0.304
  + Quality increased to ~90.03%

**🔧 Model 3**

* **Changes:**
  + generator\_lr raised to 1e-4
  + discriminator\_lr lowered to 5e-5
  + discriminator\_steps = 2
  + pac = 3
  + epochs = 2500
* **Rationale:**
  + Raising generator learning rate helped improve correlation without mode collapse
  + Lowering discriminator LR gave the generator more flexibility
  + Additional discriminator step sharpened multivariate relationships
  + Lowering pac further improved categorical fidelity
* **Results:**
  + Best correlation yet: -0.330
  + Pairwise trend: ~87.48%
  + Overall Score: ~90.33%

**🔧 Model 4**

* **Changes:**
  + Lowered discriminator\_lr to 3e-5
  + pac reduced to 2
  + epochs = 3000
* **Rationale:**
  + Further soften discriminator to enhance generator learning of subtle correlations
  + Lower pac aimed to capture complex categorical interactions
  + Higher epochs gave the model more training cycles to optimize fidelity
* **Results:**
  + Correlation further improved to -0.339
  + Best overall quality so far: **91.07%**

**🔁 Model 5**

* **Same settings as Model 4**
* **Observation:**
  + Correlation rose slightly to -0.348
  + Slight drop in column pair trends → **tradeoff**
  + Overall score decreased slightly (~90.41%)
  + Still consistent with prior tuning intent: better numeric correlation

**🔧 Model 6**

* **Changes:**
  + generator\_lr increased to 2e-4
  + discriminator\_lr decreased to 2e-5
  + discriminator\_steps = 3
* **Rationale:**
  + Further nudged learning rates for better correlation
  + 3 discriminator steps aimed to refine multivariate structure
  + Holding pac=2 ensured categorical consistency
* **Results:**
  + Correlation: ~-0.349 (stable)
  + Improved univariate and pairwise trends (up to 90.74%)

**🔧 Model 7**

* **Changes:**
  + generator\_lr = 3e-4
  + discriminator\_lr = 1e-5
  + batch\_size = 64
* **Rationale:**
  + Aggressively boosted generator’s capacity to chase correlation
  + Small batch size increases update frequency
  + Lower discriminator LR reduces overfitting risk
* **Results:**
  + Highest pairwise trend score: **88.05%**
  + Highest overall score: **90.82%**
  + But correlation dropped to -0.303 → indicating a potential overemphasis on pairwise shapes at the cost of numerical association

**🧩 Synthesis & Strategy**

|  |  |  |
| --- | --- | --- |
| Metric | Best Model | Strategy That Helped |
| Overall Quality | Model 4 | Lower discriminator LR + pac=2 + epochs=3000 |
| Best Correlation | Model 6 | Generator 2e-4 + Discriminator 2e-5 + D-steps=3 |
| Best Pairwise Trends | Model 7 | Higher generator LR + Lower discriminator LR + batch=64 |
| Best Univariate | Model 4 | pac=2 + batch=128 + balanced LR |

We iteratively adjusted hyperparameters with the goal of improving both:

* **Numerical correlation fidelity** (especially between Year of diagnosis and Survival months)
* **General distribution alignment** (column shapes and pairwise trends)

Each tuning step focused on specific tradeoffs:

* **Improving generator freedom** (by adjusting LR)
* **Balancing convergence** (by tuning batch size, discriminator steps)
* **Stabilizing training** (via pac and epochs)

**TVAE Model Development and Hyperparameter Tuning Strategy**

**Baseline Model (TVAE Model 1)**

* **Configuration:**  
  embedding\_dim=128, compress\_dims=(128, 128), decompress\_dims=(128, 128), l2scale=1e-5, batch\_size=500, epochs=300, loss\_factor=2
* **Results:**
  + **Validity & Structure:** 100%
  + **Quality Score:** ~79.91%
  + **Pearson Correlation (Synthetic):** ~-0.401
* **Observations:**  
  The model produced valid data with reasonably good univariate fidelity but struggled to replicate complex multivariate relationships and correlations, especially in high-cardinality categorical features.

**TVAE Model 2**

* **Rationale for Changes:**
  + **embedding\_dim** increased to 256 to capture more complex representations.
  + **compress/decompress\_dims** expanded for better feature abstraction.
  + **batch\_size** reduced to 128 for more frequent gradient updates.
  + **epochs** extended to 1000 for deeper convergence.
* **Results:**
  + **Quality Score:** ~83.25%
  + **Pearson Correlation:** -0.420 (perfectly matched real data)
* **Interpretation:**  
  Substantial gains in pairwise trends and near-perfect numeric correlation—an ideal progression from baseline.

**TVAE Model 3**

* **Further Scaling Up:**
  + **embedding\_dim=512**, more layers added in both encoder/decoder.
  + **epochs** increased to 1500.
  + **l2scale** lowered to 1e-7 for flexibility.
* **Results:**
  + **Quality Score:** ~80.04%
  + **Pearson Correlation:** -0.401
* **Trade-Off:**  
  Deeper architecture did not translate into improved multivariate relationships. Correlation slightly weakened, possibly due to overfitting or optimization plateau.

**TVAE Model 4**

* **Adjustment Focused on Stability:**
  + **compress\_dims/decompress\_dims** made shallower.
  + **l2scale** reduced to 1e-8.
  + **batch\_size=64**, **epochs=2000**
* **Results:**
  + **Quality Score:** ~81.61%
  + **Correlation:** -0.429 (strongest alignment to real -0.420)
* **Key Outcome:**  
  Achieved optimal correlation strength and improved pairwise trends compared to Model 3.

**TVAE Model 5**

* **Goal:** Balance high correlation and general fidelity.
  + Same architecture as Model 4.
* **Results:**
  + **Quality Score:** 84.23% (best overall so far)
  + **Pearson Correlation:** -0.408
* **Conclusion:**  
  Boosted column shapes and pairwise trends, slightly reduced correlation strength. Excellent compromise.

**TVAE Model 6**

* **Pushed Further for Correlation Recovery:**
  + **batch\_size** reduced to 32
  + **l2scale** lowered to 1e-9
  + **epochs=3000**
* **Results:**
  + **Quality Score:** 82.0%
  + **Correlation:** -0.416
* **Insight:**  
  Correlation was strengthened, but slight trade-off in univariate and bivariate fidelity—common with lower regularization and deeper convergence.

**TVAE Model 7**

* **Back to Balanced Setting:**
  + **l2scale=5e-9**, **batch\_size=64**
  + **Shallower architecture**
* **Results:**
  + **Quality Score:** 81.98%
  + **Correlation:** -0.399
* **Takeaway:**  
  Attempted to stabilize univariate/bivariate quality while retaining correlation. Some improvement in stability, but slight drop in correlation strength.

**Summary of Rational Hyperparameter Adjustments**

|  |  |
| --- | --- |
| Parameter | Adjustment Rationale |
| embedding\_dim | Increased progressively (128 → 512) to better capture complex, high-cardinality data |
| compress/decompress\_dims | Scaled up and down to balance model capacity and training stability |
| l2scale | Reduced (1e-5 → 1e-9) to allow more flexible learning of category frequencies |
| batch\_size | Lowered (500 → 32) to enable frequent gradient updates for capturing rare patterns |
| epochs | Extended up to 3000 to allow complete convergence |
| loss\_factor | Kept at 2 to emphasize numeric reconstruction, particularly for correlated variables |

**Strategic Observations**

* **Model performance peaked** with Models 4 and 5, each excelling in different dimensions (correlation vs. overall fidelity).
* **High-capacity, deeper models (Model 3, 6)** showed diminishing returns or stability issues.
* **Lower batch sizes** improved correlation but sometimes hurt overall alignment.
* **Future iterations** may consider early stopping, rare-category grouping, and variable transformation (e.g., log transform for skewed numerics like Survival Months).