>>AI CONF 2026

Stop Waiting for Data: How Generative Models Are Reshaping Insurance Analytics

Claudio Giorgio Giancaterino

Actuary & AI Scientist

Banca Intesa SanPaolo

improove

Kudos++

Executive

beSharp. we make IT run.

LEXROOM

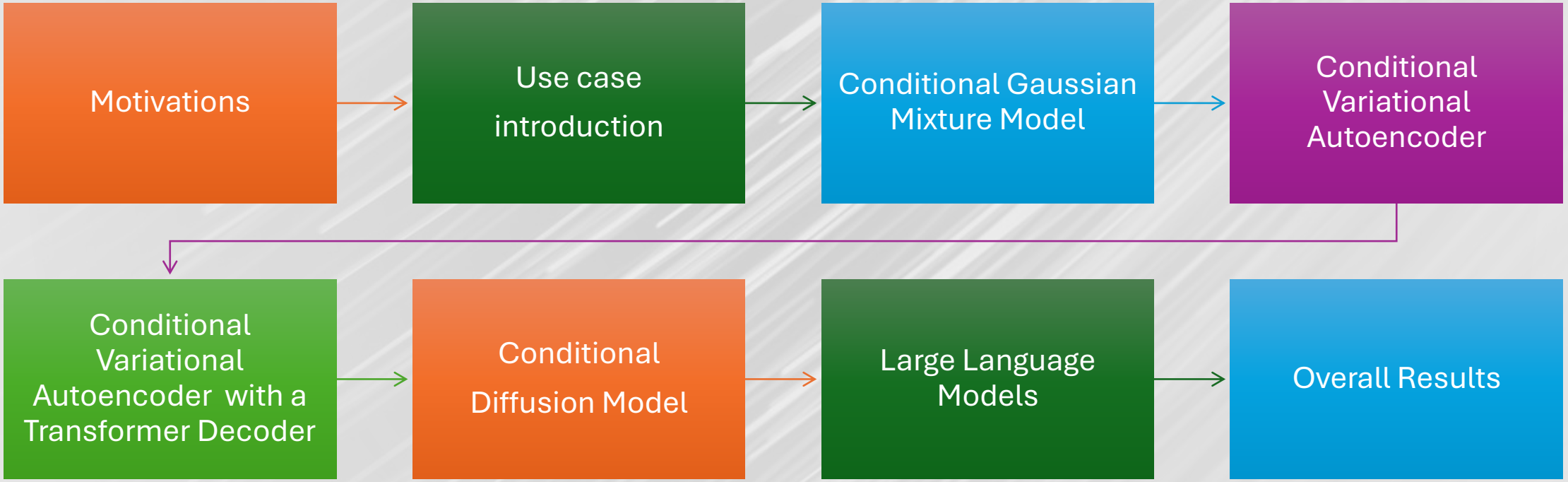
REPLY STORM

Gold

seeweb

FASTWEB + vodafone

Agenda


Motivations
Use case introduction
Conditional Gaussian Mixture Model
Conditional Variational Autoencoder
Conditional Variational Autoencoder with a Transformer Decoder
Conditional Diffusion Model
Large Language Models
Overall Results

Motivations

- Insurance Use Case
- Employ a wide range of Generative Models

Google Nano Banana

Data Scarcity in Insurance Research

The Problem: Access to realistic datasets is a major barrier to advancing actuarial research and developing insurance analytics tools.

- Most insurance data is confidential and proprietary
- Data masking and bureaucratic processes prevent disclosure
- Limited open resources lack diversity for robust modelling

The Solution: Develop simulated datasets using generative models

The Anatomy of Insurance Non-Life Risk Data: Statistical Components in the Pricing dataset

Risk Premium Components:

Frequency = Number of Claims / Number of Exposures

Severity = Losses / Number of Claims

Risk Premium = Expected Claim Frequency × Expected Cost per Claim

Datasets employed

Datasets are retrieved from the "CASdatasets" R Package

Dataset 1: ausprivauto0405 - Automobile claim datasets in Australia
It is a data frame of 9 columns and 67,856 rows:
Exposure, VehValue, VehAge, VehBody, Gender, DrivAge, ClaimOcc, ClaimNb, ClaimAmount

Dataset 2: swmotorcycle - Swedish Motorcycle Insurance dataset
It is a data frame of 9 columns and 64,548 rows:
OwnerAge, Gender, Area, RiskClass, VehAge, BonusClass, Exposure, ClaimNb, ClaimAmount

Unlocking Data Quality: Leveraging Claim Occurrence for Stable Generative Modelling

What is ClaimOcc ?

It is a binary indicator which indicates occurrence of a claim

If ClaimOcc = 0 → ClaimNb = 0 and ClaimAmount = 0

If ClaimOcc = 1 → ClaimNb ≥ 1 and ClaimAmount > 0

Why use ClaimOcc?

Structural relationship: drives logical dependency between ClaimNb and ClaimAmount

Class imbalance: separates rare claim events from common no-claim cases

Model stability: improves synthetic data quality by conditioning

Role in Generative Models

CGMM: Fits separate Gaussian mixture models for each ClaimOcc value

CVAE & CTVAE: Concatenates ClaimOcc to the latent space for conditional generation

CDM: Uses ClaimOcc as a condition signal to drive the denoising process

Synthetic Data Generation Trials

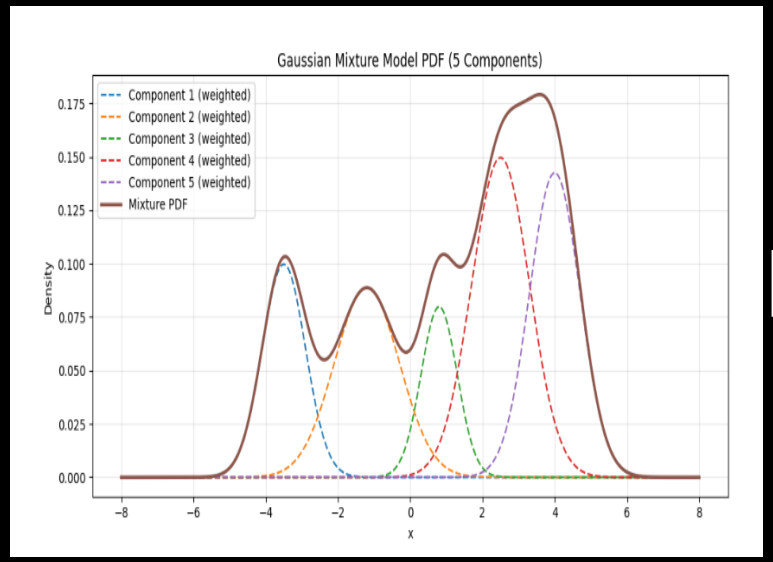
Trial 1: 80% Training → 80% Generation
Baseline Quality Assessment
What is the baseline quality of synthetic insurance data when a generative model is trained on 80% of the original data? How well does it preserve statistical distributions and enable accurate GLM predictions?

Trial 2: 60% Training → 80% Generation
Data Augmentation with Limited Training
Can a generative model trained on only 60% of the data successfully generate synthetic samples to reach 80% dataset size while maintaining statistical fidelity and predictive performance?

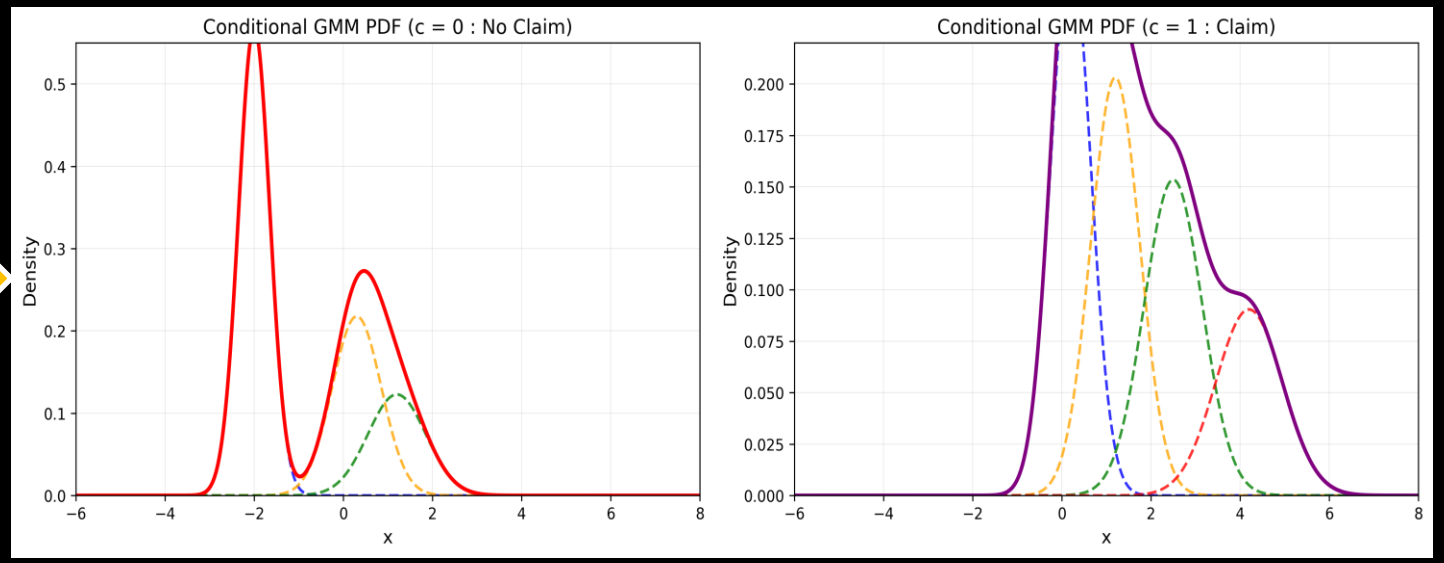
Trial 3: 80% Gender-Masked → 80% Generation
Fairness-Aware Generation
Can a gender-masked generative model generate unbiased synthetic insurance data while maintaining predictive utility, and what is the fairness-accuracy trade-off?

Conditional Gaussian Mixture Model (CGMM)

What is it?
A Gaussian Mixture Model assumes that the data are generated by a mixture of K Gaussian distributions, each with unknown parameters and corresponding to a cluster. Every Gaussian has its own mean μ_k, covariance Σ_k, and mixing weight π_k.



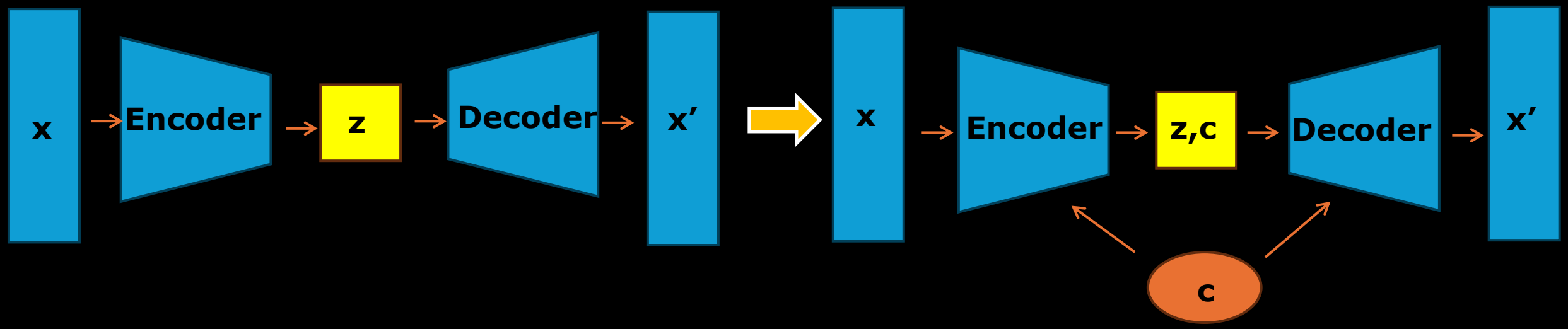
Source
ChatGPT



Synthetic record = CGMM(Gaussian component k, condition ClaimOcc)

Conditional Variational Autoencoder (CVAE)

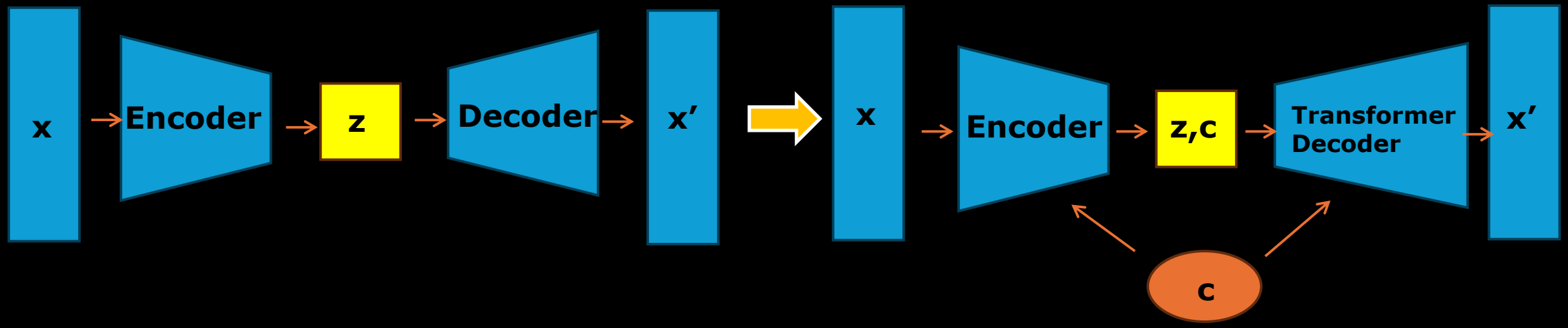
What is it?
A Variational Autoencoder (VAE) is a type of neural network that learns to represent input data as a probability distribution in a lower-dimensional space and then decode it to generate similar input data.



Synthetic record = Decoder(latent risk profile z, condition ClaimOcc)

Conditional Variational Autoencoder with a Transformer-based Decoder (CTVAE)

What is it?
It's a hybrid architecture that leverages:
-VAEs for learning a smooth, continuous latent space.
-Transformers for decoding sequences with long-range dependencies and attention mechanisms.

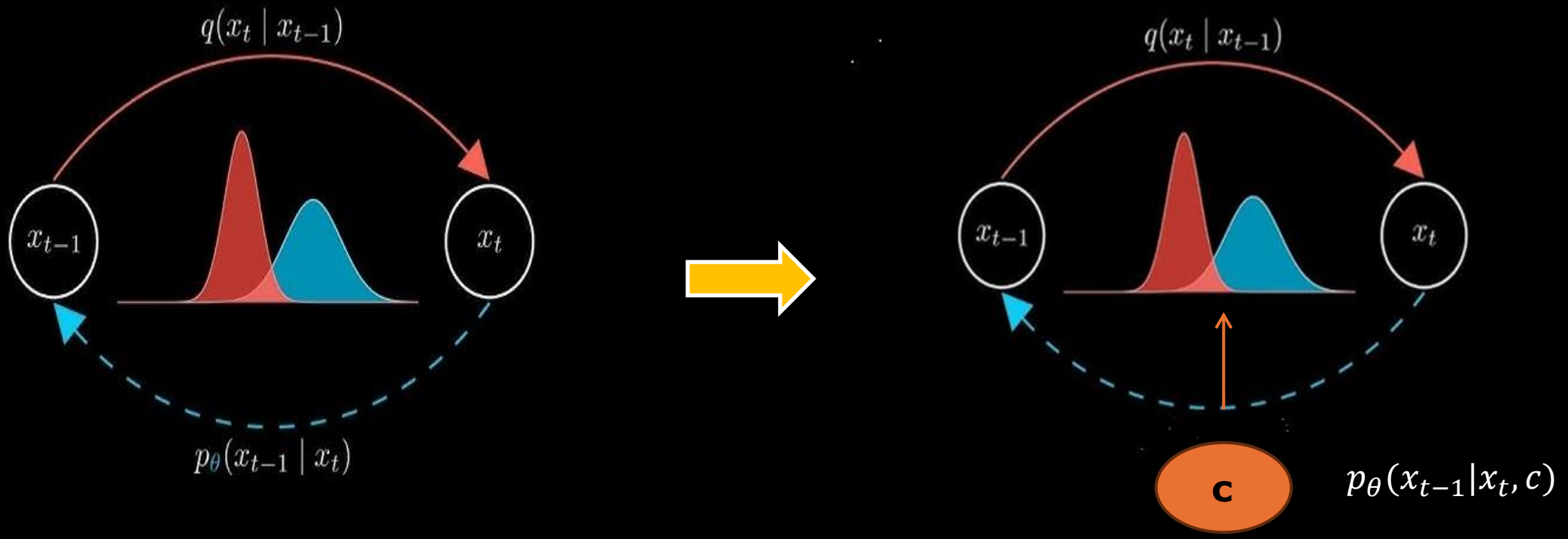


Synthetic record = Transformer Decoder(latent risk profile z, condition ClaimOcc)

Conditional Diffusion Model (CDM)

What is it?
A Diffusion Model learns to generate data by simulating a gradual noising-and-denosing process.

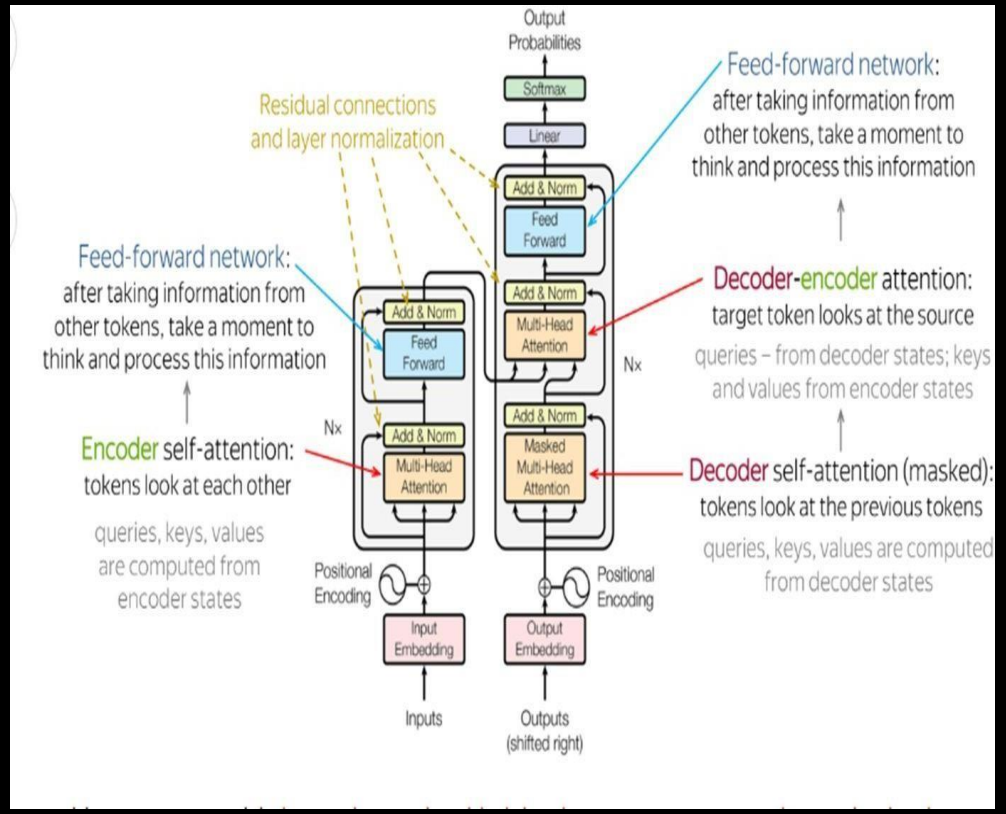
source



Synthetic record = iterative reverse denoising conditioned on ClaimOcc

Large Language Model (LLM)

What is it?
Language Models can be defined as a probability distribution over sequences of words.
Given a vocabulary of words, a Language Model assigns a probability to each sequence of words, and the purpose is to predict the next word in the sequence.
Large Language Models like BERT and GPT are trained on a vast amount of text data, to learn the structure, meaning, and usage of language.



Synthetic record = LLM(next-token generation , condition ClaimOcc)

The prompt provides the LLM with the project context, the dataset structure, variable distributions, and sample records. It defines the target distribution for Claim Count and Claim Amount, the business rules that connect claim occurrence, frequency, and severity, and the main actuarial relationships between variables, such as vehicle value and vehicle age.

source

Validation by Consistency records

```python
# Find inconsistencies
inconsistent_records = new_samples_df[
    ~(((new_samples_df["ClaimNb"] == 0) & (new_samples_df["ClaimOcc"] == 0) & (new_samples_df["ClaimAmount"] == 0)) |
      ((new_samples_df["ClaimNb"] > 0) & (new_samples_df["ClaimOcc"] > 0) & (new_samples_df["ClaimAmount"] > 0)))
]

print(f"Number of inconsistent records: {len(inconsistent_records)}")
print(inconsistent_records.head())  # Show a few inconsistent rows
```

```
Number of inconsistent records: 1
       Exposure  VehValue  VehAge  VehBody  Gender  DrivAge  ClaimNb  \
49759  0.736911  5.434924       3        6       1        6      1.0

       ClaimAmount  ClaimOcc
49759          0.0         1
```

The consistency test validates the **logical relationships** between three insurance claim variables in the synthetic data:

-ClaimNb (Number of claims)
-ClaimOcc (Claim occurrences)
-ClaimAmount (Claim amount)

No Claims: All three = 0 (no claims occurred)
Claims Exist: All three > 0 (claims occurred with positive values)

Validation by Kolmogorov-Smirnov Test

```
]:  # Kolmogorov-Smirnov test
    for column in X_train.columns:
        original = X_train[column].values
        generated = new_samples_df[column].values
        statistic, p_value = ks_2samp(original, generated)
        print(f"KS Test for {column}: Statistic={statistic}, P-value={p_value}")
```

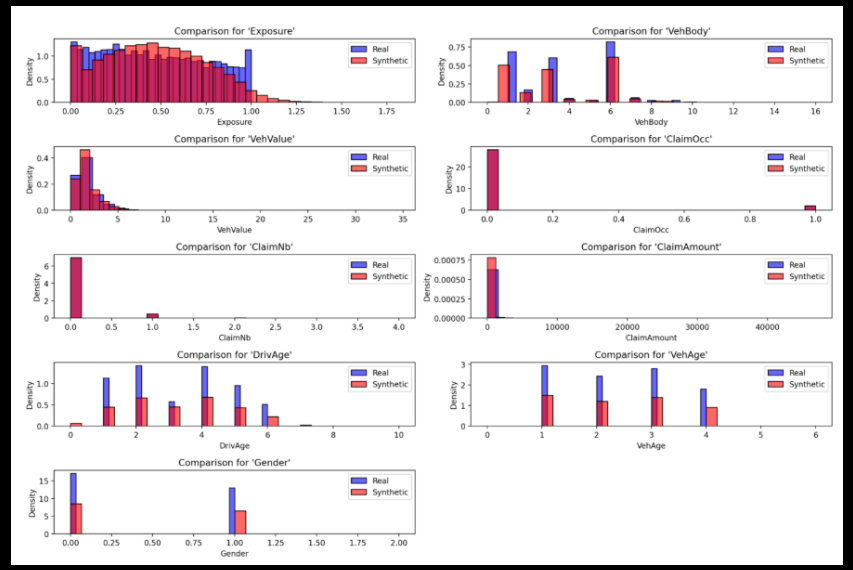
```
KS Test for Exposure: Statistic=0.13863465866466618, P-value=0.0
KS Test for VehValue: Statistic=0.2830457614403601, P-value=0.0
KS Test for VehAge: Statistic=0.07691297824456114, P-value=1.3001617749392413e-137
KS Test for VehBody: Statistic=0.20864591147786948, P-value=0.0
KS Test for Gender: Statistic=0.001444111027756878, P-value=0.9999999970492804
KS Test for DrivAge: Statistic=0.12738184546136533, P-value=0.0
KS Test for ClaimOcc: Statistic=0.0, P-value=1.0
KS Test for ClaimNb: Statistic=0.0017254313578394243, P-value=0.9999982405003918
KS Test for ClaimAmount: Statistic=0.016185296324081055, P-value=1.6976741222647111e-06
```

The Kolmogorov-Smirnov (KS) test compares the distributions of original and synthetic data for each feature. A **high p-value (>0.05)** indicates that the two distributions are statistically similar, while a **low p-value (<0.05)** suggests significant differences.

Validation by Data Visualization

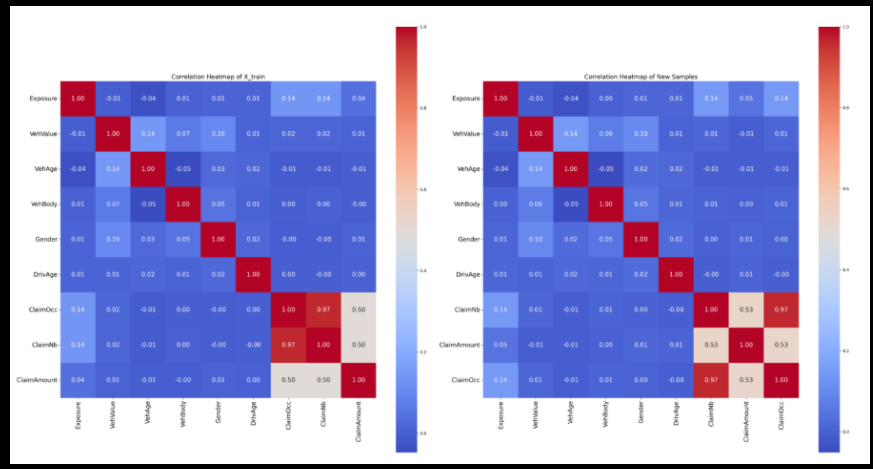
Univariate Analysis

Displays the alignment of the univariate statistical properties for each feature across the real and synthetic data



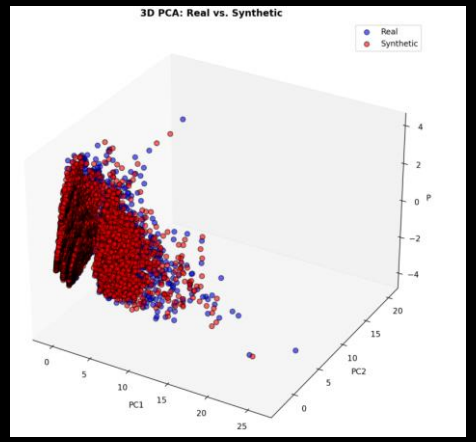
Correlation Matrix

Assesses the structural fidelity of the synthetic data by comparing the correlation matrix to the original data



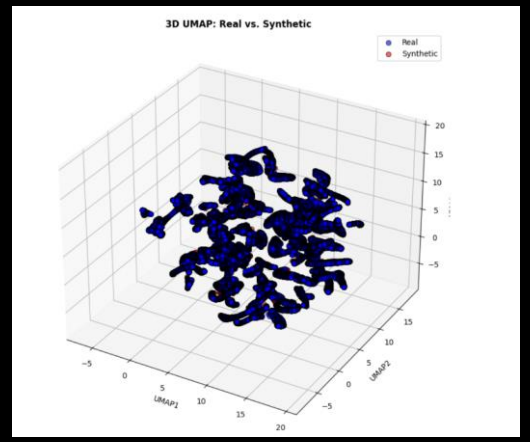
3D PCA

Compares the explained variance captured by the principal components of the synthetic data against the real data



3D UMAP

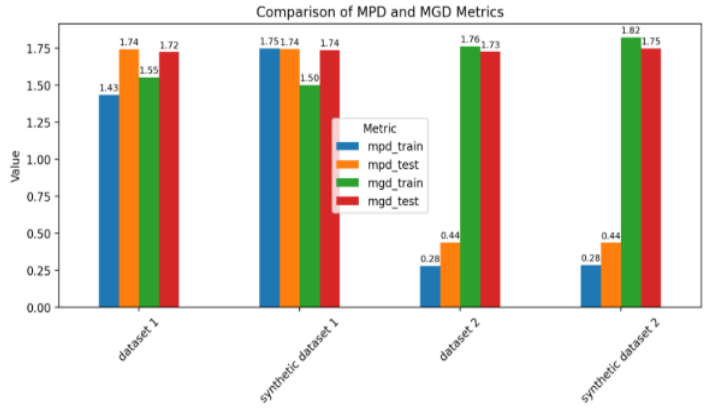
Evaluates the degree to which the synthetic data preserves the local and global topological structure of the original data



Validation by Predictive Modelling

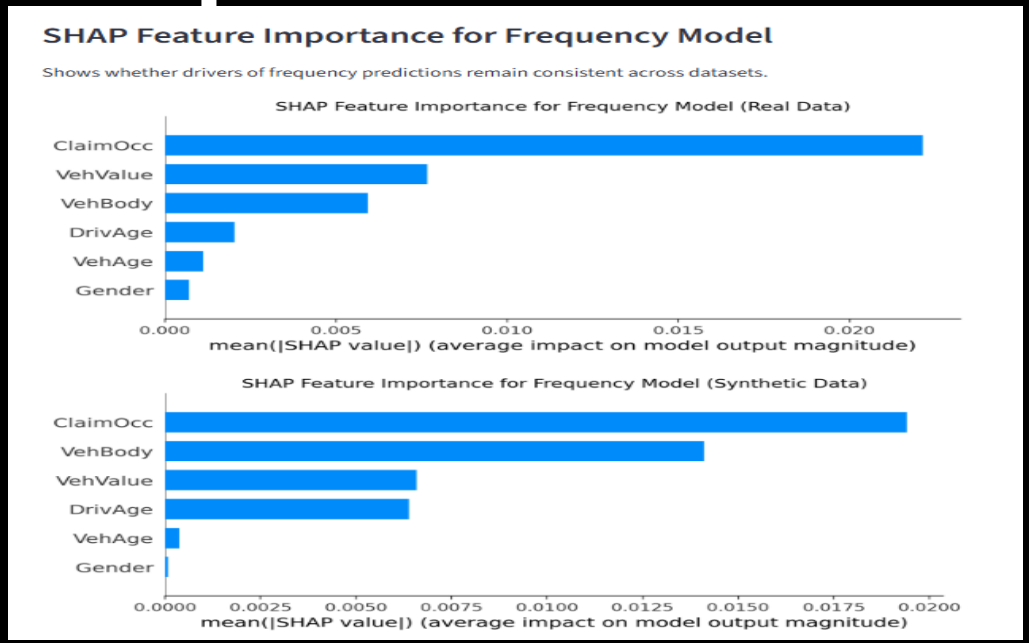
Frequency and Severity prediction
GLM on both synthetic and real data to evaluate performance using deviance metrics (lower is better).

	mpd_train	mpd_test	mgd_train	mgd_test
dataset 1	1.4345	1.7412	1.5509	1.7221
synthetic dataset 1	1.7457	1.7427	1.4991	1.7361
dataset 2	0.2788	0.4368	1.7615	1.7251
synthetic dataset 2	0.2848	0.4357	1.8222	1.7478



Validation by Feature Importance

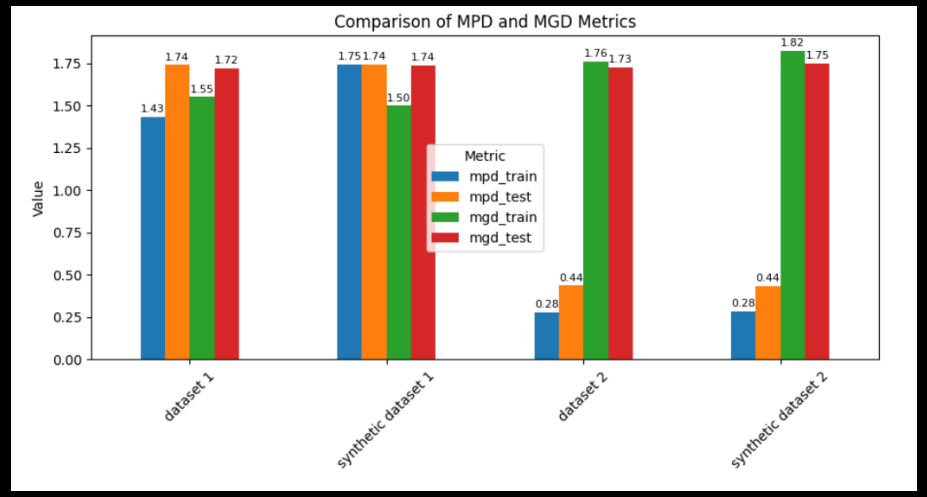
SHAP Feature Importance
Shows whether drivers of frequency and severity predictions remain consistent across datasets.



CGMM Predictive Modeling Results

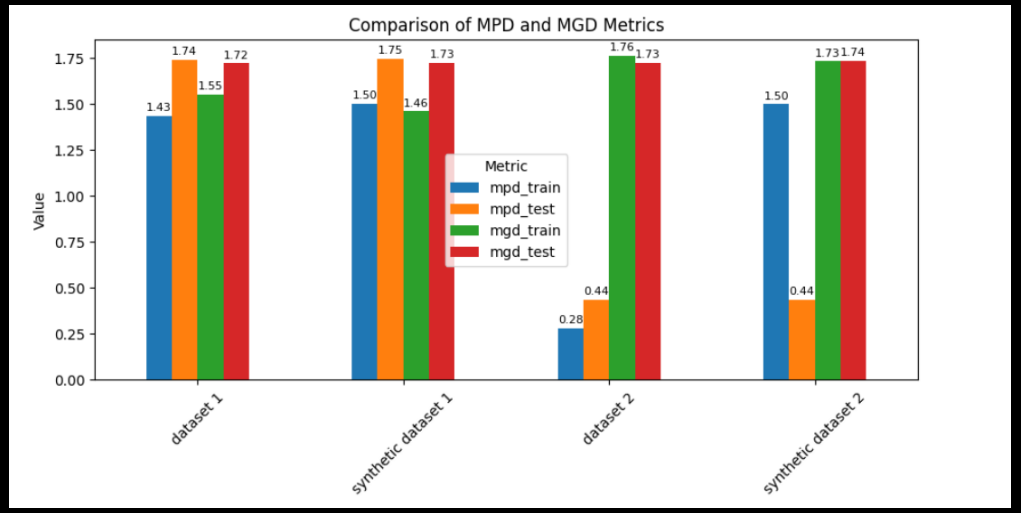
Takeaway: CGMM preserves test predictive performance, but Dataset 2 exhibits repeated instability in the frequency model in Trials 2 and 3.

Trial 1



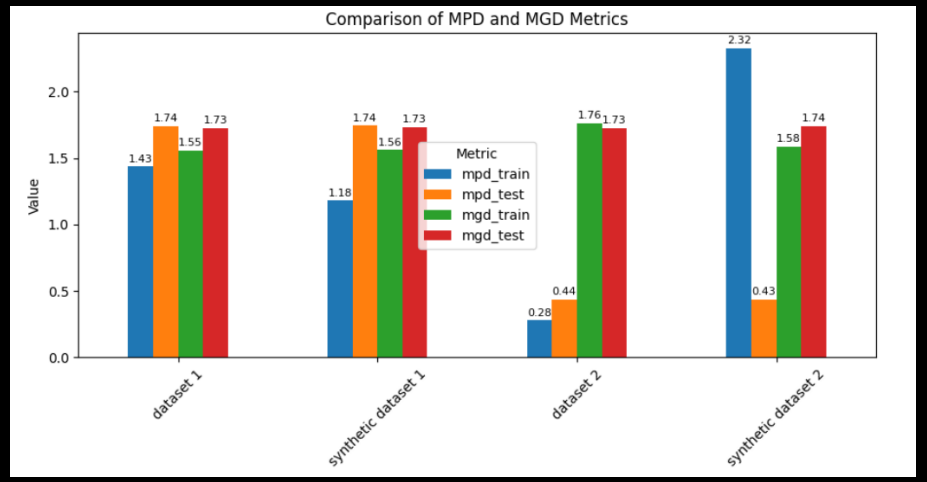
Trial 1: Stable across frequency and severity

Trial 2



Trial 2: Dataset 1 stable; Dataset 2 flagged on frequency train

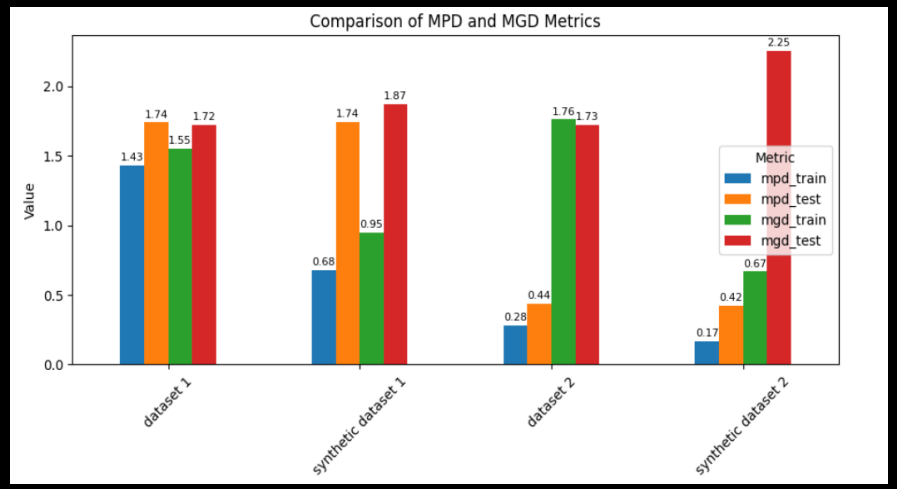
Trial 3



Trial 3: Dataset 1 stable; Dataset 2 confirms frequency train instability

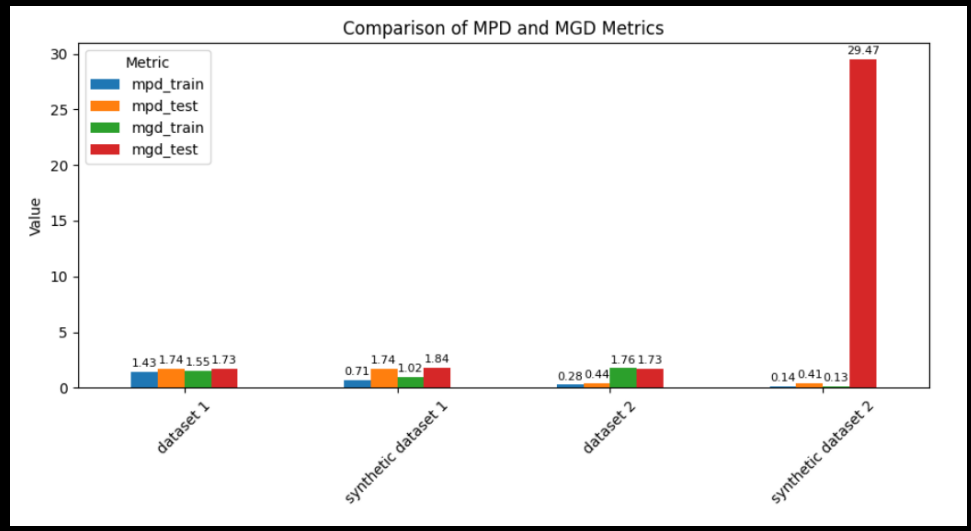
CVAE Predictive Modeling Results

Trial 1

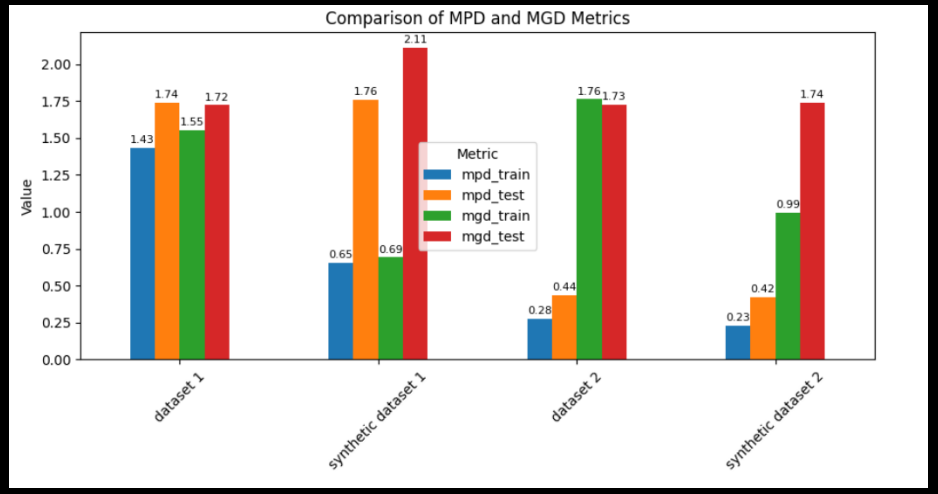


Takeaway: CVAE preserves frequency test performance, but Dataset 1 shows repeated frequency overfitting, while Dataset 2 fails terribly on severity in Trial 3.

Trial 3



Trial 2



Trial 1: Dataset 1 frequency/severity overfitting; Dataset 2 shows severity overfitting

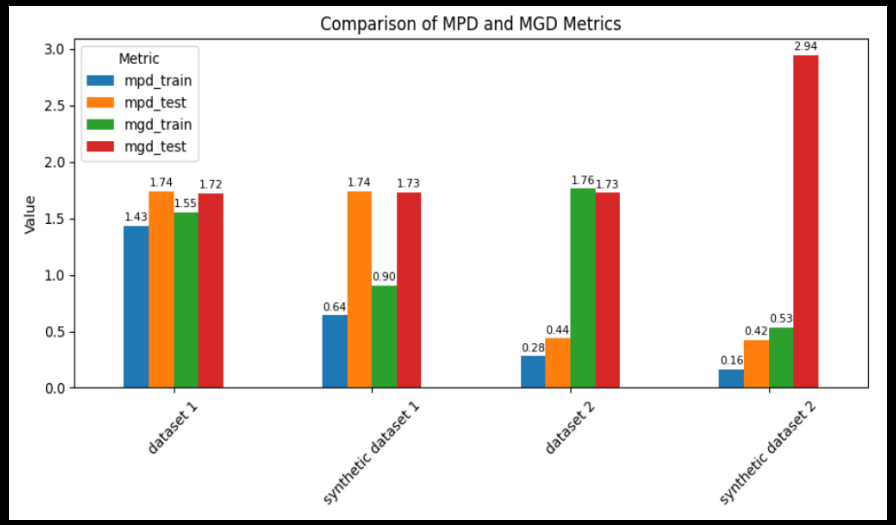
Trial 2: Dataset 1 confirms frequency/severity overfitting; Dataset 2 preserve severity test performance

Trial 3: Frequency stable; Dataset 2 fails on severity test

CTVAE Predictive Modeling Results

Takeaway: CTVAE preserves frequency test performance across all trials, but it shows repeated frequency overfitting and recurrent severity instability for Synthetic Dataset 2.

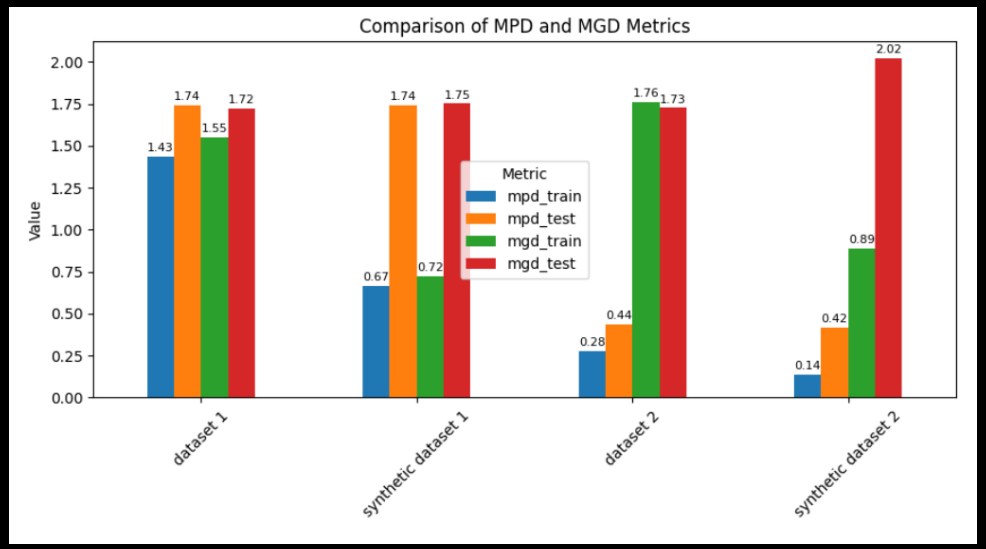
Trial 1



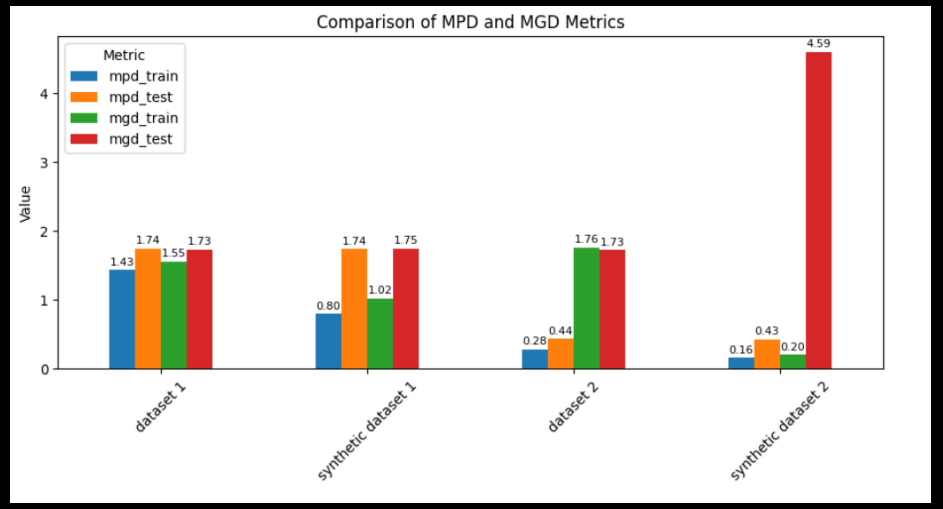
Trial 1: Dataset 1 frequency/severity overfitting; Dataset 2 fails on severity

Trial 2: Dataset 1 confirms frequency/severity overfitting; Dataset 2 shows moderate severity weakness

Trial 2



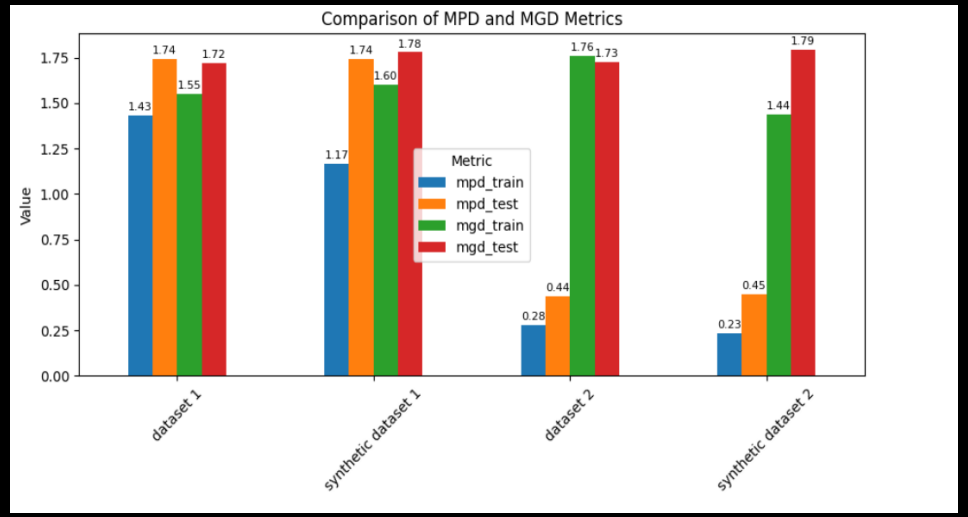
Trial 3



Trial 3: Frequency performance aligned with real data; Dataset 2 fails on severity but improve in results than CVAE

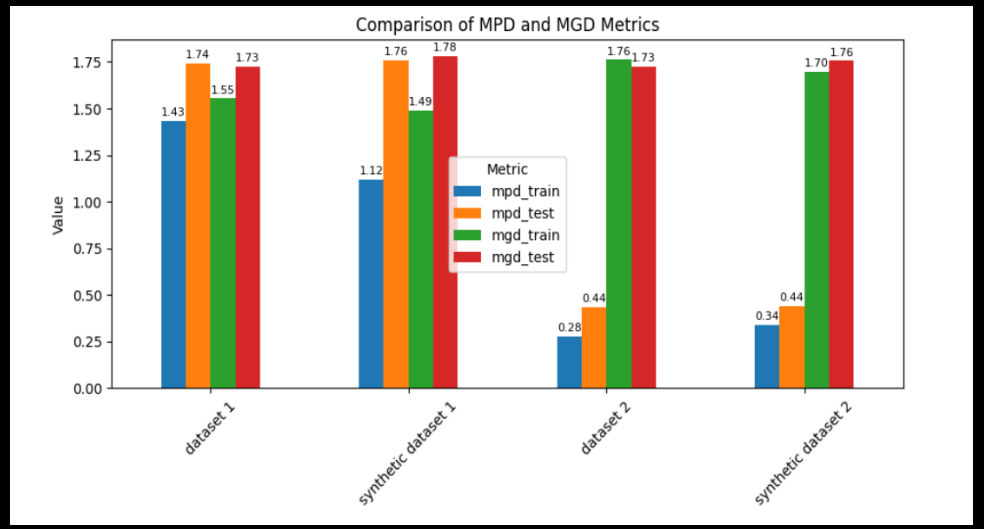
CDM Predictive Modeling Results

Trial 1



Takeaway: CDM preserves test predictive performance well across trials, with only one relevant instability in Dataset 2 during Trial 2.

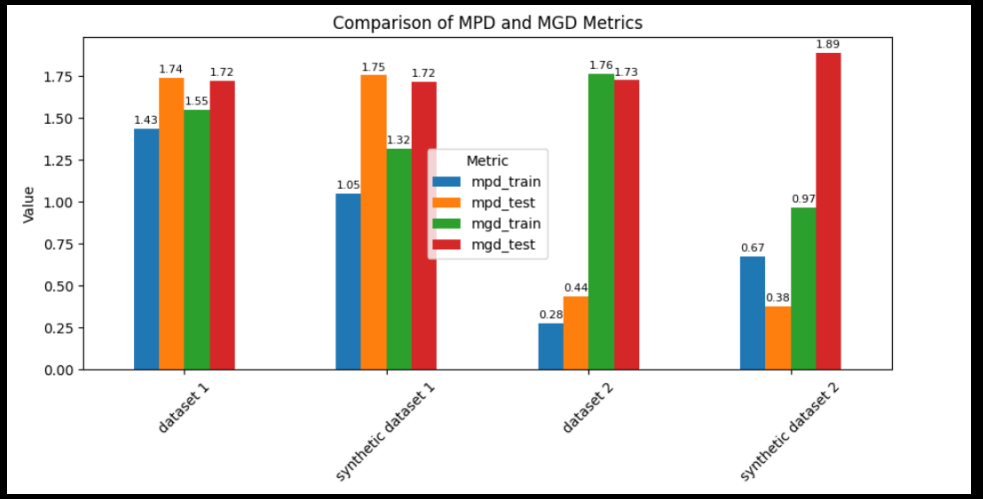
Trial 3



Trial 1: Both datasets stable across frequency and severity, close to the real-data benchmarks.

Trial 2: Dataset 1 stable; Dataset 2 flagged mainly on frequency train, with some additional severity instability.

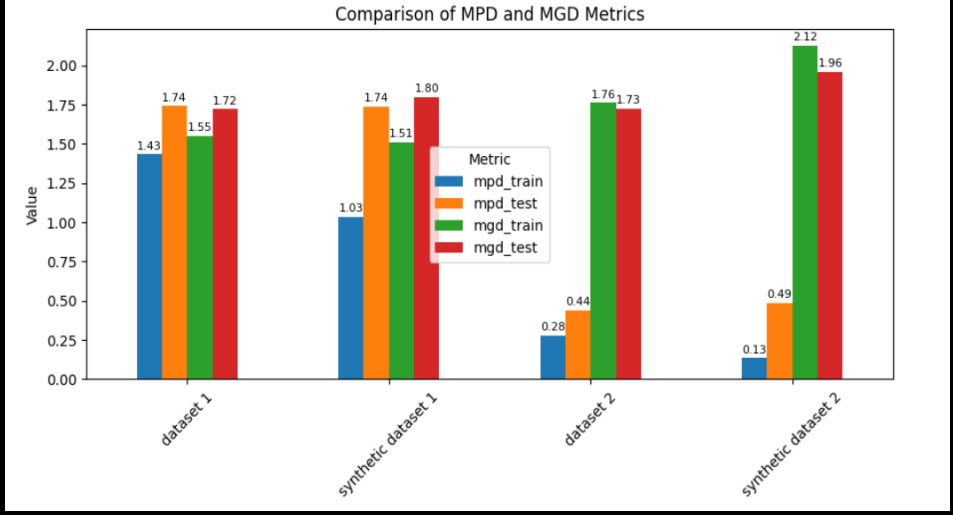
Trial 2



Trial 3: Dataset 1 stable; Dataset 2 Dataset 2 realigns with the real-data benchmark.

LLM Predictive Modeling Results

Trial 1

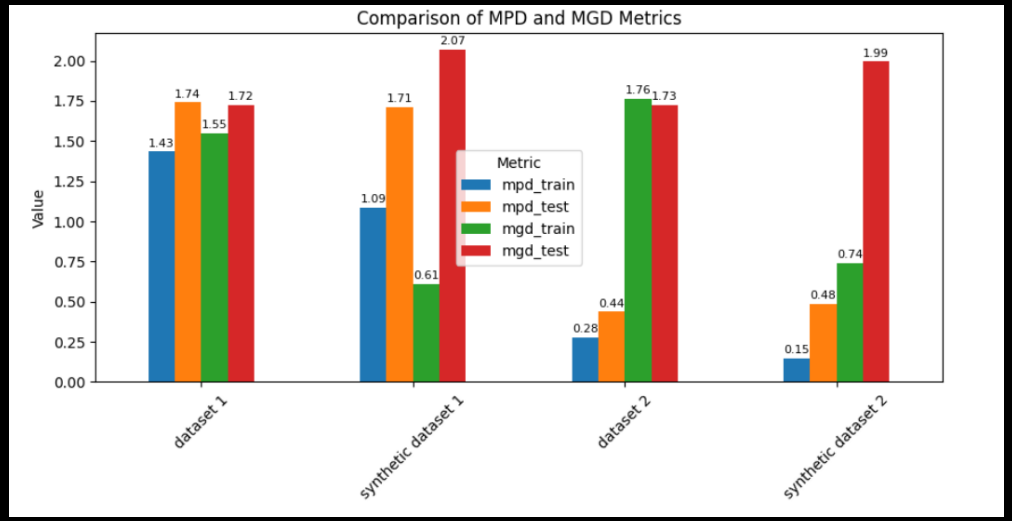


Trial 1: Frequency test performance is stable; Dataset 1 shows little training overfitting signals, while Dataset 2 is a little weak on severity.

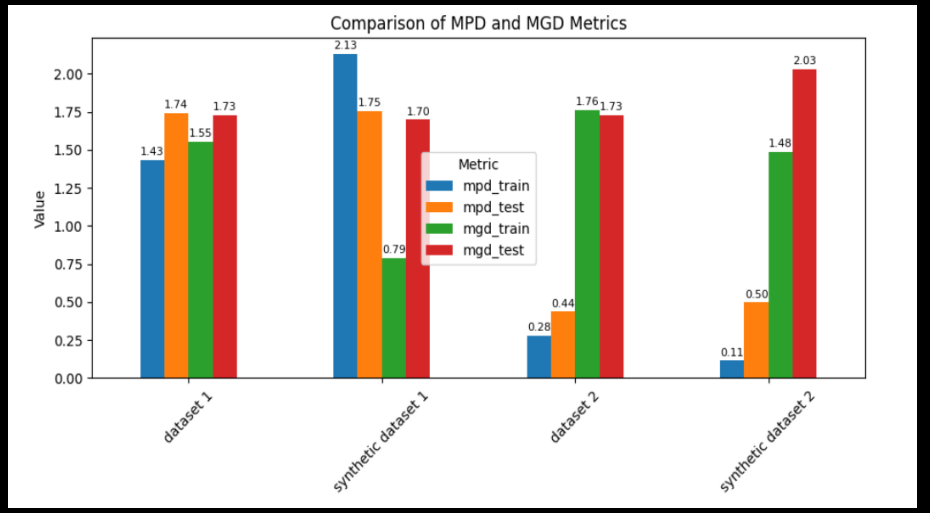
Trial 2: Frequency remains acceptable, but severity overfitting becomes clear in both datasets.

Takeaway: LLM preserves business logic very well and keeps frequency performance test broadly stable, but severity calibration remains a little fragile, especially for Dataset 2.

Trial 2



Trial 3



Trial 3: Dataset 1 frequency train calibration becomes unstable; Dataset 2 on severity is improved, but weak in results.

Overall Results

Focusing on predictive modeling performance and the preservation of statistical properties, I recommend the Conditional Gaussian Mixture Model, the Conditional Diffusion Model, and the Large Language Model. The two Conditional Variational Autoencoders show strong training performance, but suffer from severe overfitting. In subsequent trials, model performance degrades, but in some cases it remains acceptable.

Watch the results from the web demo app.

Conclusions

What I've learned

- ✅ Conditional Gaussian Mixture Models are competitive as generative models for tabular data. Good compromise between ease of use and effectiveness.
- ✅ Large Language Models are promising for synthetic data generation due to their performance and ease of use. Complex model implementation isn't required, but detailed prompting and deep data analysis are needed to write a good prompt.
- ✅ Conditional Diffusion Model is competitive in performance, but it requires computational and coding effort.
- ✅ Generative Adversarial Networks are missing in this exploration due to inconsistent results.

References

-Jan Goodfellow and Yoshua Bengio and Aaron Courville, 2016, *Deep Learning*, MIT Press.
-Mario V. Wuthrich, Ronald Richman, Benjamin Avanzi, Mathias Lindholm, Michael Mayer, Jürg Schelldorfer, Salvatore Scognamiglio, 2025, *AI Tools for Actuaries*, SSRN.
-David Foster, 2023, *Generative Deep Learning, 2nd Edition*, O'Reilly.
-Jake VanderPlas, 2016, *Python Data Science Handbook*, O'Reilly.
-Jamotton, Charlotte; Hainaut, Donatien, 2023, *Variational autoencoder for synthetic insurance data*, ISBA.
-Harshvardhan GM, Mahendra Kumar Gourisaria, Manjusha Pandey, Siddharth Swarup Rautaray, 2020, *A comprehensive survey and analysis of generative models in machine learning*, ScienceDirect.
Repository: https://github.com/claudio1975/Generative_Modelling
Web_APP:
Generative Models 4 Insurance Data
Article: https://medium.com/@c.giancaterino/stop-waiting-for-data-how-generative-models-are-reshaping-insurance-analytics-ec102a2e5177

Thank you!

👉 slides & videos: https://www.improove.tech/videos

>>AI CONF 2026

improove