MANCHESTER 1824
The University of Manchester

AMORTIZED BAYESIAN INFERENCE USING TRANSFORMERS FOR DATA RECONSTRUCTION

A DISSERTATION SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF MASTER OF SCIENCE
IN THE FACULTY OF SCIENCE AND ENGINEERING

Year of submission

2025

Student ID: 14110008

Department of Computer Science

Contents

Word Count: 7250

List of Figures

Acknowledgements

I would like to express my sincere gratitude to the University of Manchester School of Engineering for providing me with the opportunity to undertake this MSc project.

My heartfelt thanks go to Dr. Ainur Begalinova, our project coordinator, for facilitating this opportunity and for her continuous support throughout the project.

Most importantly, I am deeply grateful to my supervisor, Dr. Jason Hartford, whose expert guidance, insightful feedback, and encouragement have been invaluable in the successful completion of this work. Without his mentorship, this project would not have been possible.

This journey has been both challenging and rewarding, and I am thankful to everyone who has contributed to making it a meaningful learning experience.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Abstract

This project investigates whether transformer architectures can perform amortised probabilistic inference in a controlled, unsupervised setting. We study *in-context density estimation (ICDE)*, where a model receives only a small set of observed context samples from an episode-specific Gaussian Mixture Model (GMM) and must infer the distribution of fully masked query points. Motivated by recent work on transformers as implicit Bayesian inference engines and meta-learners for continuous distributions, we develop a permutation-invariant Transformer–Mixture Density Network (Transformer–MDN) that maps context sets to predictive mixture parameters for each query.

Experiments on an episodic 2-D GMM benchmark evaluate the model using both qualitative diagnostics and distribution-level metrics, including Wasserstein distance, Maximum Mean Discrepancy, and Monte-Carlo KL divergence. The proposed model substantially outperforms a unimodal Gaussian baseline and approaches the performance of per-episode GMM oracles, demonstrating that a single transformer can amortise inference across diverse mixture distributions. The analysis also identifies systematic failure modes, such as overlapping components and the diagonal-covariance restriction of the MDN, which limit performance in challenging regimes.

Overall, the results provide controlled empirical evidence that transformers can approximate Bayesian-style updates from context and offer a flexible, data-driven mechanism for amortised density estimation. The study clarifies the connection between in-context learning and probabilistic reasoning and establishes ICDE as a compact benchmark for exploring these ideas.

Chapter 1

Introduction

1.1 Context and Problem Definition

The ability to reconstruct missing or incomplete data is a cornerstone of modern artificial intelligence (AI) and machine learning (ML). Real-world datasets are frequently incomplete: measurements are missing, corrupted, or withheld, and downstream systems must reason under uncertainty.

Classical strategies for handling such cases (e.g., deletion, mean/median imputation, or k-nearest neighbour imputation) are computationally cheap and often effective in practice, but they neglect distributional structure and provide little uncertainty quantification [1]. More principled statistical methods, such as EM-based maximum likelihood [2], multiple imputation [3], or Bayesian hierarchical models, capture richer uncertainty but typically require bespoke inference for each new dataset, often relying on MCMC or variational approximations that are computationally expensive at scale.

Deep generative models, such as Variational Autoencoders (VAEs) [4] and related latent-variable frameworks, attempt to amortize this cost: they introduce an *inference network* that learns a fast mapping from observations to posterior approximations across many instances,

$$q_\phi(z \mid x) \approx p(z \mid x).$$

This provides reusable inference but depends on explicitly modelling latent variables and likelihoods.

This motivates a complementary question: can transformers perform amortised probabilistic inference without explicit latent variables, by directly conditioning on context samples? To examine this, we study in-context density estimation (ICDE), a setting where a model receives a small set of context examples from an unknown distribution and must infer the distribution governing fully masked query points. Concretely, each episode provides a small context set of fully observed examples and a query set that is *fully masked at input*. The model conditions on the context and outputs a distribution over the queries.

This differs from latent-variable amortization (e.g., VAEs approximating $p(z \mid x)$) by instead directly learning $q(x \mid C)$: a distribution over new samples conditioned on

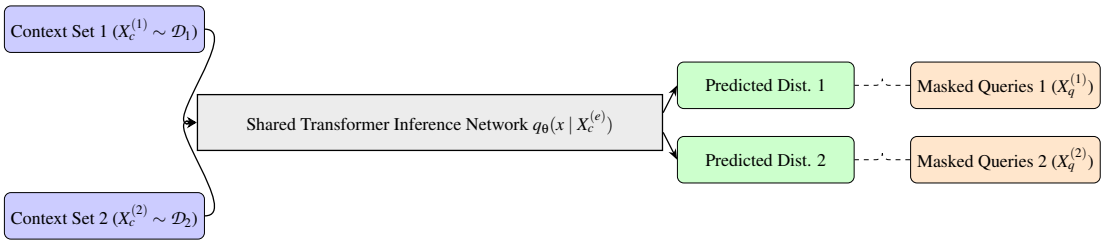

Figure 1.1: Illustration of in-context density estimation (ICDE).

a context. It aligns with recent views of transformers as inference engines that can internalize Bayesian-style updates from context [5], and is further motivated by recent evidence that large language models can approximate density estimation from in-context samples [6]. Building on this perspective, we introduced the term *in-context density estimation* (ICDE) — a framework where a single model amortizes across a family of distributions via conditioning on observed examples. Figure 1.1 illustrates this idea: each episode e provides a context set $X_c^{(e)}$ and a (possibly masked) query set $X_q^{(e)}$, drawn from a distinct distribution $\mathcal{D}_e$. A shared Transformer inference network amortizes inference across episodes; predicted distributions (green) are evaluated at the episode's query points.

Problem setup. Let an episode $\mathcal{E}$ be a sample from a meta-distribution over tasks. It contains a context set $C = \{x_i\}_{i=1}^{N_c}$ (fully visible) and a query set $Q = \{x_j^\star\}_{j=1}^{N_q}$ whose values are hidden to the model at input time. The learning objective is to train a parametric predictor f_ϕ that, given C and a mask for Q, produces a conditional density $q_\phi(\cdot \mid C)$ from which we can evaluate or sample the queries:

$$\mathcal{L}(\phi) = -\mathbb{E}_{\mathcal{E}}\left[\frac{1}{N_q}\sum_{j=1}^{N_q} \log q_\phi\big(x_j^\star \,\big|\, C\big)\right].$$

Importantly, we do *not* claim to approximate a specific Bayesian posterior $p(\cdot|C)$ under an explicit latent-variable model; rather, we test whether a Transformer can *behave as an amortized inference mechanism* that maps context to well-calibrated predictive distributions over queries.

Why a Transformer? Transformers offer two properties that make them well suited for in-context density estimation. First, self-attention provides a natural mechanism for conditioning on sets of variable size without imposing ordering assumptions, which is essential when the model must aggregate information from context samples. Second, transformers have been shown to adapt their predictions based on in-context examples in supervised settings, suggesting an ability to approximate inference procedures when trained over a distribution of related tasks[7]. Prior work also demonstrates that transformers can approximate posterior predictive behaviour in controlled function-learning

setups[6], though typically in supervised or partially observed regimes. In this project we extend this perspective to a fully masked-query setting, where the model receives no information about query values at input time. Any successful prediction must therefore arise from learning how to infer a distribution solely from the context, making the transformer's flexible conditional modelling capacity a promising choice rather than relying on fixed parametric estimators.

Role of Gaussian Mixture Models (GMMs). We use mixtures of Gaussians purely as a *controlled, multi-modal testbed* to probe whether the model captures modality, spread, and mixture proportions from few-shot context. We do *not* position our approach as a replacement for classical GMM fitting (e.g., EM), which is already efficient and well understood; instead, GMMs offer precise control over difficulty (separation, anisotropy, mixing weights) while allowing rigorous, distribution-level evaluation of in-context predictions.

1.2 Background and Literature Review

Transformers, first introduced for sequence modelling in natural language processing [7], have rapidly become one of the most versatile architectures in machine learning. Beyond language, they now power state-of-the-art models in computer vision, audio, reinforcement learning, and scientific domains. A growing body of research has also explored their potential for *probabilistic inference*. In parallel, the concept of *in-context learning* has emerged, where a transformer is prompted with examples of input–output pairs and adapts its predictions to new queries without any parameter updates [8]. This highlights the ability of transformers to implicitly perform inference by conditioning on context. Building on this idea, Müller, Hollmann, and colleagues demonstrated that transformers can approximate Bayesian inference when trained on Gaussian Process (GP) samples [5]. In this setting, the transformer acts as an *amortized inference engine*, learning to condition on observed data and predict posterior function values without explicit likelihood evaluations. This perspective motivates treating transformers not only as sequence predictors, but as general-purpose inference networks.

However, most of these works focus on supervised tasks, i.e. learning conditional distributions of the form $p(y \mid x)$. In contrast, the question of whether transformers can perform *unsupervised* inference i.e. directly modelling distributions $p(x)$ in context remains underexplored.Recent evidence shows that large language models can indeed approximate density estimation from in-context samples, exhibiting geometric behaviours akin to kernel density estimators [6]. Our work extends this idea by applying ICDE to Gaussian mixture meta-datasets, thereby testing transformers as implicit amortized inference networks in multi-modal density estimation settings.

To situate this approach, it is useful to recall what is meant by unsupervised learning. Unlike supervised tasks, which predict labelled outputs, unsupervised learning focuses on estimating latent structure or full distributions over the data itself. Classical

approaches include Expectation–Maximization (EM) for mixture models [2], multiple imputation techniques [3], and Bayesian hierarchical models. While these methods provide principled uncertainty estimates, they can be computationally expensive as inference must be repeated for each new dataset. Deep generative models such as Variational Autoencoders (VAEs) [4] and Generative Adversarial Networks (GANs) [9] have sought to amortize this cost by learning reusable inference networks, but their stability and flexibility depend heavily on architectural choices.

Finally, Chen et al. [10] examined a complementary question: can transformers emulate *classical unsupervised algorithms* such as EM or spectral methods for Gaussian mixture models? Their focus was on algorithmic emulation and training transformers to reproduce clustering and parameter estimation steps, rather than direct probabilistic inference. This contrasts with our approach, which instead evaluates whether transformers can serve as inference engines that approximate posterior distributions under structured missingness.

1.3 Gap and Challenges

Despite progress in generative modelling and probabilistic inference, several challenges remain unresolved.

First, while in-context learning with transformers has been widely studied for supervised tasks (learning $p(y \mid x)$), much less work has investigated their ability to perform *unsupervised in-context density estimation* i.e. approximating the underlying data distribution $p(x)$ given only a small set of context examples. In this setting, queries are naturally *fully masked* at input, since the task is to infer their distribution solely from the context. However, systematic evaluation of transformers in such density-estimation tasks remains limited. Existing masked modeling approaches such as MLMs or MAEs [11, 12] rely on *partial masking*, making the fully masked query setting considered here a significantly more stringent and underexplored benchmark.

Second, although amortized inference is well established in the context of VAEs and related latent-variable frameworks, relatively little work has considered transformers explicitly as inference networks for density estimation. Prior studies have largely focused on their predictive capacity in supervised learning, rather than their ability to approximate posterior-like distributions in unsupervised settings.

Third, existing transformer-based approaches differ in emphasis. Prior work has shown that transformers can approximate Bayesian inference in function-space settings [5] and generalise distributions via in-context conditioning [8]. Other studies, such as Chen et al. [10], explored whether transformers can emulate classical algorithms like EM for Gaussian Mixture Models (GMMs). However, these works do not directly address the role of transformers as *amortized inference networks for mixture distributions under fully masked queries*, which remains largely unexplored.

By contrast, the specific space of transformers as *direct amortized inference networks for mixture distributions with structured masking*, where queries are fully hidden

in an unsupervised setting, and must be reconstructed distributionally from context remains underexplored.

Finally, the evaluation of such models still lacks consensus. Many generative modelling studies report reconstruction error or likelihood scores, but distributional fidelity has often been neglected. For in-context density estimation, principled measures such as Kullback–Leibler divergence, Wasserstein distance, and Maximum Mean Discrepancy (MMD) are essential to assess whether predicted samples genuinely approximate the ground-truth distribution, rather than collapsing to point estimates.

1.4 Objectives

The overarching aim of this project is to investigate whether transformers can be effectively used as *amortized inference networks for in-context density estimation*. Specifically, we aim to:

1. **Benchmark Construction:** Construct a controlled benchmark based on episodic Gaussian mixture models (GMMs), ensuring variation in modality, cluster separation, and component covariances.

2. **Transformer-based Inference Model:** Design and implement a permutation-invariant Transformer–Mixture Density Network (Transformer–MDN) capable of predicting mixture-model parameters for masked queries.

3. **Evaluation Framework:** Develop a principled evaluation framework using distributional similarity metrics (Wasserstein, symmetrised KL, MMD).

4. **Insights and Contribution:** Analyse the strengths and limitations of transformer-based amortised inference in multi-modal synthetic settings, situating the findings relative to existing literature on Bayesian inference with transformers[6], algorithmic emulation[8], and in-context density modelling[13]

1.5 Summary

In summary this project proposes in-context density estimation (ICDE) as a compact benchmark for studying amortised probabilistic inference with transformers. By focusing on fully masked queries and episodic mixture distributions, the framework isolates the model's ability to infer densities from context alone. The Transformer–MDN architecture is trained and evaluated on synthetic episodes to assess its ability to recover multimodal structure, mixture proportions, and uncertainty. This provides an empirical foundation for understanding the role of transformers as inference mechanisms beyond supervised settings.

Chapter 2

Methodology

2.1 Dataset Design

2.1.1 Motivation

Rather than training on a single fixed dataset, this project employs an *episodic Gaussian Mixture Model (GMM) meta-dataset*. Each episode defines a separate probabilistic task: a new mixture with randomly sampled number of components $K \in \{2, \ldots, K_{\max}\}$, mixture weights, component means, and diagonal covariances. From each sampled mixture, we draw two disjoint sets: a context set of fully observed examples and a query set used solely as supervision.

This episodic design mirrors the principles of meta-learning, where a model is trained across diverse tasks and must generalise across episodes by inferring distributional structure directly from context. By varying K and component parameters across episodes[5, 13], the model is prevented from memorizing a fixed parametric form and is instead forced to amortize inference over a family of mixtures, effectively learning to reconstruct the query distribution from limited context.

Crucially, unlike many missing-data imputation studies that use *partial masking*, here the *query set is fully masked*. At training and inference time, the transformer never observes query values as input; instead, it must predict a distribution over them conditioned only on the context. This setting constitutes a more stringent test of whether a transformer can operate as an amortized inference network capable of reconstructing unseen distributions without direct supervision on query inputs.

2.1.2 GMM Parameter Sampling

For each episode e, the generative process is:

$$K \sim \text{Uniform}\{2, \ldots, K_{\max}\}, \tag{2.1}$$
$$\pi \sim \text{Dirichlet}(\mathbf{1}_K), \tag{2.2}$$
$$\mu_k \sim \mathcal{N}(\mathbf{0}, \sigma_\mu^2 I_D), \tag{2.3}$$
$$\sigma_k \sim \text{Uniform}[0.3, 2.0]^D, \tag{2.4}$$

where K is the number of mixture components, π_k are the mixture weights, μ_k are the component means, and σ_k are diagonal covariance terms.

2.1.3 Episode Construction

As shown in figure 2.1 below, each episode returns:

- **Context set:** $X_c = \{x_i\}_{i=1}^{N_c}$, fully observed samples used as conditioning data.
- **Query set:** $X_q = \{x_j\}_{j=1}^{N_q}$, fully masked inputs that the model must reconstruct.

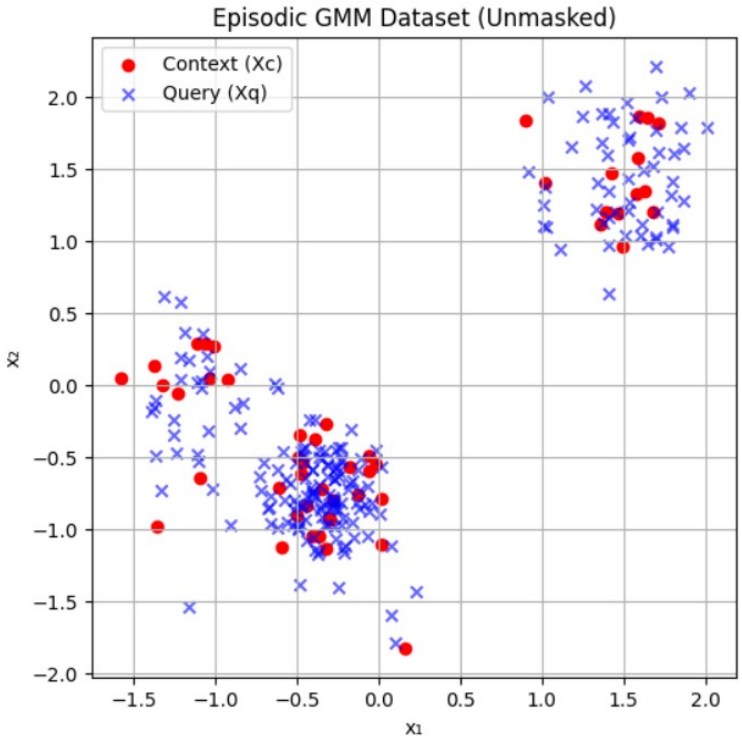

Figure 2.1: An example episode sampled from the Gaussian Mixture Model (GMM) dataset. The red circles represent context samples and the blue crosses denote query samples.

2.1.4 Masking Policy (Fully Masked Queries)

The masking strategy plays a central role in determining what information is available to the model. In our setup, the *context set* is fully visible while the *query set* is *fully masked*. This ensures that the model cannot trivially rely on partial query information, but instead must infer a distribution over queries conditioned solely on the context.Such a design makes the problem closer to *amortized inference*, as the transformer must generalise from context statistics to unseen points. This approach is conceptually aligned with earlier work in distribution estimation that employed masked prediction as a mechanism for modelling joint distributions. For instance, MADE (*Masked Autoencoder for Distribution Estimation*) reconstructs variables under a masking scheme so that each variable is predicted without access to future inputs, effectively enforcing valid conditional density estimation [14]. More recently, Masked Conditional Density Estimation (MaCoDE) has further demonstrated that fully-masked feature prediction can be treated as a conditional density estimation problem, particularly in structured data settings [15]. Our masking policy follows this paradigm, extending it to the in-context setting with GMM-based episodic tasks.

Formally, each query vector $x_j \in \mathbb{R}^D$ is replaced by a mask token:

$$\tilde{x}_j = m \odot x_j, \quad m = \mathbf{0}_D,$$

where m is a binary mask. Since $m = \mathbf{0}_D$, this corresponds to full masking, i.e. no feature of x_j is directly revealed to the model. Thus the model must learn the conditional density

$$p_\theta(x_j \mid X_c), \quad \forall x_j \in X_q,$$

relying entirely on the statistical structure of the context.

2.1.5 Per-Episode Normalisation

To stabilise training, all samples in an episode are normalised using context statistics:

$$\tilde{X} = \frac{X - \mu_c}{\sigma_c + \varepsilon}, \quad \mu_c = \frac{1}{N_c}\sum_{i=1}^{N_c} x_i, \quad \sigma_c = \sqrt{\frac{1}{N_c}\sum_{i=1}^{N_c}(x_i - \mu_c)^2}.$$

This ensures scale invariance across episodes, while μ_c and σ_c are stored for denormalisation at evaluation time. Normalization techniques are widely known to improve training stability and convergence in deep models. In particular, Layer Normalization (even when applied per token) helps prevent internal covariate shift, stabilizing the scale of activations across layers and enabling the model to generalize better [16, 17]. By conditioning with context-derived statistics, the transformer is supplied with per-episode normalization cues, analogous to Context Normalization layers that adjust input activations based on auxiliary context [18]. This design choice is thus theoretically motivated by established normalization principles.

2.1.6 Illustrative Example

Figure 2.2 depicts what masked queries look like in an episode(data available to the transformer)

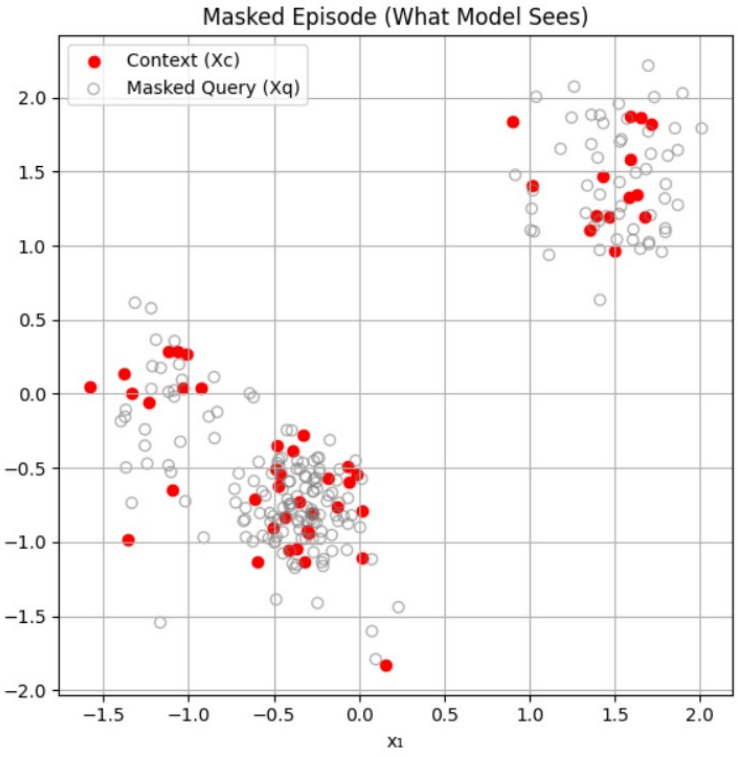

Figure 2.2: A plot depicting what is visible to the model in an episode.

2.2 Model Architecture

The central aim of this project is to test whether a Transformer can function as an *amortised inference network* in a setting where queries are *fully masked*. Rather than performing posterior inference from scratch for every new query set, the model learns a shared mapping from observed contexts to predictive distributions, thereby amortising the inference cost across episodes. We adopt a **permutation-invariant Transformer** architecture, designed to operate on sets rather than sequences. This choice reflects the fact that both context and query points are exchangeable, and any valid inference procedure should be invariant to their ordering. To capture predictive uncertainty, the Transformer outputs the parameters of a Mixture Density Network (MDN), providing a flexible family of distributions capable of modelling multi-modal targets.

The architecture proceeds in four main stages: (i) tokenisation and masking, (ii) Transformer-based set encoding, (iii) mixture density output heads, and (iv) training via negative log-likelihood (NLL).

Problem setup. Each training *episode* provides a set of fully observed *context* points $\mathcal{C} = \{x_i^{(c)}\}_{i=1}^{N_c} \subset \mathbb{R}^D$ and a set of *query* points $\mathcal{Q} = \{x_j^{(q)}\}_{j=1}^{N_q} \subset \mathbb{R}^D$. At inference time the model sees $\mathcal{C}$ and must predict the density of $x^{(q)}$.

Tokenisation and masking. We construct one token per element of $\mathcal{C} \cup \mathcal{Q}$. For each token we build a feature vector by concatenating the masked value, the mask, and per-episode statistics:

$$z = \big[x \odot m,\ m,\ \mu_{\text{ctx}},\ \sigma_{\text{ctx}}\big] \in \mathbb{R}^{2D+2D}, \tag{2.5}$$

where $m \in \{0,1\}^D$ is a binary mask; for context tokens $m = \mathbf{1}$ so $x \odot m = x$, and for query tokens $m = \mathbf{0}$ so $x \odot m = \mathbf{0}$. The statistics $(\mu_{\text{ctx}}, \sigma_{\text{ctx}})$ are computed from C and appended to all tokens to provide episode-specific normalization.

A learned linear projection maps z into the Transformer dimension d_{model}, augmented with a *segment embedding* to distinguish between context and query tokens.

Transformer encoder. Tokens are concatenated in the order $[\mathcal{C} \,|\, \mathcal{Q}]$ (no positional encoding) and passed through a L-layer Transformer encoder with H attention heads. Because no positional encodings are used, the encoder is permutation-equivariant, ensuring that the final predictions are invariant to the ordering of context/query points.

Mixture density head. From the query slice of the encoded sequence, we predict parameters of a K-component diagonal Gaussian mixture:

$$\pi = \text{softmax}\big(W_\pi \tilde{H}_{\text{qry}}\big), \tag{2.6}$$

$$\mu = \text{reshape}\big(W_\mu \tilde{H}_{\text{qry}}\big), \tag{2.7}$$

$$\sigma = \exp\big(\text{reshape}(W_{\log\sigma} \tilde{H}_{\text{qry}})\big). \tag{2.8}$$

This yields the predictive density

$$p(x^{(q)} \mid \mathcal{C}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x^{(q)}; \mu_k, \text{diag}(\sigma_k^2)). \tag{2.9}$$

Training objective. With queries fully masked at the input level, supervision comes only from their true values under the predictive distribution. We minimise the mean negative log-likelihood:

$$\mathcal{L}_{\text{MDN}} = -\frac{1}{N_q} \sum_{j=1}^{N_q} \log \sum_{k=1}^{K} \pi_{jk} \, \mathcal{N}\big(x_j^{(q)}; \mu_{jk}, \text{diag}(\sigma_{jk}^2)\big). \tag{2.10}$$

Justification. Transformers are chosen over MLPs or RNNs because (i) they naturally encode set structure via self-attention, (ii) they provide flexible context aggregation, and (iii) they scale well to varying N_c and N_q. The MDN head ensures that predictive uncertainty and multi-modality are explicitly captured, aligning the model with the goals of Bayesian inference through amortisation.

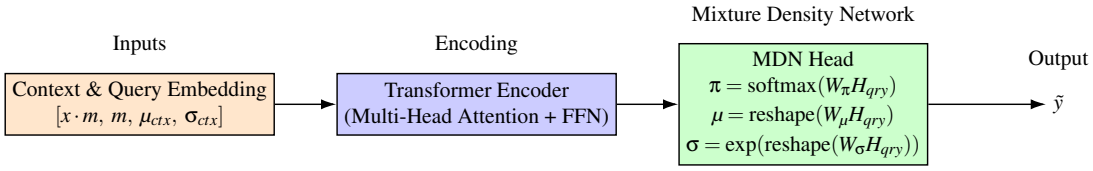

Figure 2.3: Architecture of the Transformer with Mixture Density Network (MDN) head. Inputs are embedded as $[x \cdot m, m, \mu_{ctx}, \sigma_{ctx}]$, passed through a Transformer encoder, and decoded into mixture parameters (π, μ, σ) for output prediction $\tilde{y}$.

2.3 Training Strategy

The training of the proposed Transformer–MDN follows a supervised episodic scheme, where each episode provides a context set X_c and a fully masked query set X_q. As shown in Figure 2.4, the model first encodes the inputs using a permutation-invariant transformer encoder. The query tokens, which contain no observed values, are processed jointly with the context tokens to produce hidden representations. These are passed through mixture density network (MDN) heads that output mixture weights π, means μ, and standard deviations σ for each query point.

The parameters are optimised by minimising the negative log-likelihood (NLL) of the true query values under the predicted mixture distribution, with an additional entropy bonus applied to prevent variance collapse. The resulting objective is given in Equation (1). Optimisation is carried out using AdamW, with early stopping based on validation loss to prevent overfitting. This setup ensures that the model learns to generalise across masking patterns and stabilises training under the fully masked query setting.

2.3.1 Objective Function

Since the model outputs mixture parameters π, μ, σ, training is performed using the negative log-likelihood (NLL) of the query points under the predicted mixture distribution:

$$\mathcal{L}_{\text{NLL}} = -\frac{1}{BN_q} \sum_{b=1}^{B} \sum_{i=1}^{N_q} \log \left(\sum_{k=1}^{K} \pi_{b,i,k}\, \mathcal{N}(x_{b,i} \mid \mu_{b,i,k}, \sigma^2_{b,i,k}) \right).$$

Here, B is the batch size, N_q the number of queries, and K the number of mixture components. This loss encourages the network to place probability mass around the true query values rather than point predictions.

2.3.2 Variance Regularization

A small entropy bonus term is added to prevent the mixture components from collapsing to zero variance:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} - \lambda \mathbb{E}\left[\tfrac{1}{2}\log \sigma^2\right],$$

where λ is a hyperparameter controlling the strength of this regularisation. In practice, $\lambda = 10^{-3}$ stabilises training without inflating predictive uncertainty.

2.3.3 Optimisation

Training is performed using the AdamW optimizer with learning rate 10^{-3} and weight decay 10^{-4}. The choice of AdamW is motivated by its improved handling of weight decay compared to standard Adam, which helps avoid overfitting in transformer-based architectures. Training proceeds for 40 epochs with early stopping based on validation NLL.

2.3.4 Supervision via Maximum Likelihood

Although query tokens are fully masked during the forward pass, their true values are used only for supervision in the loss function. Specifically, the predicted mixture distribution over each query is trained to maximize the likelihood of the corresponding ground-truth query point. This setup ensures that the model learns to allocate probability mass around true query values, without any direct leakage of those values into the input.

2.3.5 Validation and Model Selection

A separate validation set is generated from independent GMMs to ensure generalisation. The best model checkpoint is selected based on the lowest validation objective, reflecting the model's ability to reconstruct queries from unseen distributions.

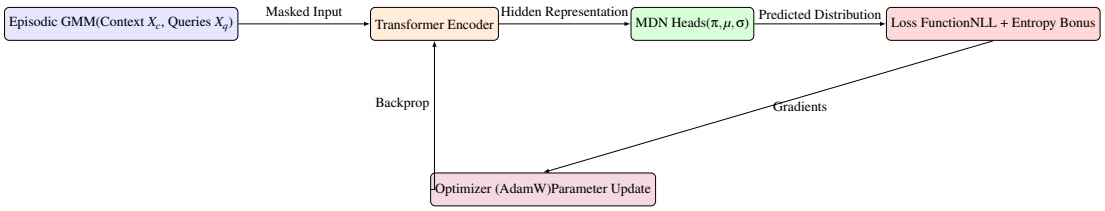

Figure 2.4: Training pipeline for the Transformer MDN. Each episode provides context samples X_c and fully masked queries X_q. The transformer encoder processes the inputs, the MDN heads predict mixture parameters (π, μ, σ), and training is driven by negative log-likelihood (NLL) with entropy regularisation. The AdamW optimizer updates model parameters.

Chapter 3

Evaluation and Reflection

3.1 Evaluation Setup

We evaluate the trained **Transformer–MDN** on held-out episodic GMM data using the same episode structure as training. Each test episode provides a *context set* $\mathcal{C} = \{x_i\}_{i=1}^{N_c}$ (fully visible) and a *query set* $\mathcal{Q} = \{x_j^\star\}_{j=1}^{N_q}$ whose values are used only for supervision and never revealed to the model at inference time. As in training, per-episode normalization is applied using context statistics:

$$\mu_{\text{ctx}} = \frac{1}{N_c}\sum_{i=1}^{N_c} x_i, \qquad \sigma_{\text{ctx}} = \sqrt{\frac{1}{N_c}\sum_{i=1}^{N_c}(x_i-\mu_{\text{ctx}})^2}\ \text{(elementwise)},$$

and all inputs to the model are standardized accordingly. After prediction, model samples are de-normalized via

$$\tilde{x} \mapsto x = \tilde{x} \odot \sigma_{\text{ctx}} + \mu_{\text{ctx}}. \tag{3.1}$$

Fully masked queries. The transformer receives tokens for both context and queries. Context tokens encode observed values; query tokens carry only a segment indicator and the mask ($m = 0$) with zeroed features. Thus, no information from $x_j^\star$ leaks into the model; the model must amortize $p(x \mid \mathcal{C})$ and *generate* query values.

Predictive distribution and sampling. For each query token the MDN head outputs mixture parameters $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ with diagonal covariances $\Sigma_k = \text{diag}(\sigma_k^2)$. The predictive density is

$$p_\theta(x \mid \mathcal{C}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}\big(x; \mu_k, \Sigma_k\big).$$

We draw Monte Carlo samples $x^{(s)} \sim p_\theta(\cdot \mid \mathcal{C})$ (typically one sample per query for reporting) and de-normalize via (3.1).

Metrics. We report distributional similarity between model samples and ground-truth query points per episode, then aggregate across episodes as mean ± std. In particular, we compute:

1. **Wasserstein distance.** We compute the multi-dimensional Wasserstein distance $W(\hat{p}, p)$ between predicted and true query samples using `scipy`'s `wasserstein_distance_nd`. This gives a geometry-aware measure of the optimal transport cost required to align the two empirical distributions. Compared to MMD, WD is more sensitive to outliers: even a small fraction of predicted points lying far from the true support can substantially inflate the score. This lack of robustness has been analyzed in the optimal transport literature, where small contaminations or heavy-tailed noise can significantly inflate Wasserstein-based metrics [19, 20].. This sensitivity partly explains the higher variance observed across episodes in our evaluation.

2. **MMD with Gaussian kernel (2D joint).** We compare the *predicted* sample set $\hat{\mathcal{Z}} = \{\hat{\mathbf{z}}_i\}_{i=1}^n \subset \mathbb{R}^2$ to the *ground-truth* query set $\mathcal{Z} = \{\mathbf{z}_j\}_{j=1}^m \subset \mathbb{R}^2$, where $\mathbf{z} = (x^{(1)}, x^{(2)})$ are 2D coordinates. Using an RBF kernel $k_\sigma(\mathbf{u}, \mathbf{v}) = \exp\big(-\|\mathbf{u} - \mathbf{v}\|^2/(2\sigma^2)\big)$, we compute the (biased) squared MMD estimator:

$$\widehat{\mathrm{MMD}}^2 = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{i'=1}^{n} k_\sigma(\hat{\mathbf{z}}_i, \hat{\mathbf{z}}_{i'}) + \frac{1}{m^2}\sum_{j=1}^{m}\sum_{j'=1}^{m} k_\sigma(\mathbf{z}_j, \mathbf{z}_{j'}) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} k_\sigma(\hat{\mathbf{z}}_i, \mathbf{z}_j).$$

   This estimator is termed *biased* because it includes the diagonal self-similarity terms (e.g. $k(x_i, x_i)$), which yield a positive bias in expectation. Despite this, the biased form is widely used due to its lower variance and simpler implementation compared to the unbiased variant [21].

3. **Kullback–Leibler Divergence (KL):** estimated using a Monte Carlo approximation for Gaussian mixtures, following the sampling-based approach proposed by Hershey and Olsen [22]. The difficulty of computing exact KL divergence between mixture distributions, as discussed in prior work such as Durrieu and Thiran [23], motivates this estimation strategy.

Protocol. We evaluate on synthetic GMM episodes with known ground truth, enabling controlled comparisons free from noise or dataset artifacts. This isolates the core task of reconstructing multimodal structure from limited context.

For each episode we: (i) input standardized context and masked queries to the model, (ii) generate predictions from the Transformer–MDN, (iii) de-normalize using episode statistics, (iv) compute metrics (Wasserstein Distance, MMD, KL Divergence), (v) report aggregated results.

We also provide scatterplots, marginal histograms, and analytic contour plots based on both GMM and MDN parameters to visualize alignment across sample, marginal, and full-density levels.

3.2 Qualitative Evaluation

To complement quantitative metrics, we present qualitative comparisons between model predictions and ground-truth distributions. These visualisations serve three complementary purposes: (i) *scatterplots* illustrate the alignment of predicted samples with true queries in joint 2D space; (ii) *contour plots* compare the analytic densities of the ground-truth GMM against the predicted MDN, providing a direct comparison of probability mass in two dimensions; (iii) *histograms* show calibration of one-dimensional marginals, highlighting how well the model captures per-dimension variability.

Together, these plots provide a multi-level perspective: from raw samples, to marginal statistics, to analytic densities. By plotting contours derived directly from ground-truth and predicted mixture parameters, we avoid artifacts from kernel smoothing and obtain a clearer picture of how well the MDN captures multimodal structure.

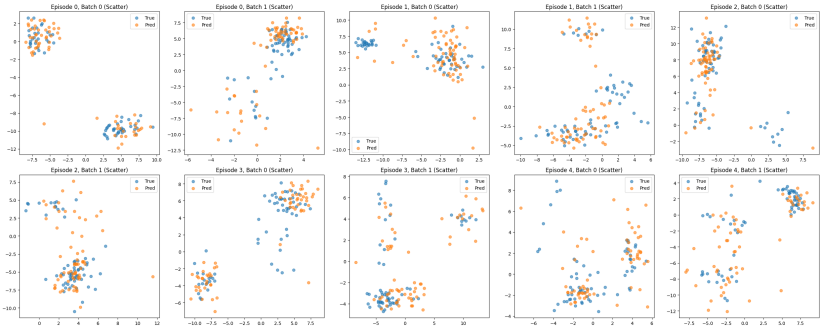

Figure 3.1: Scatterplots of predicted samples (orange) vs. ground-truth queries (blue) across five test episodes. Each subplot corresponds to a batch, showing the model's ability to reconstruct multimodal structure under fully masked queries.

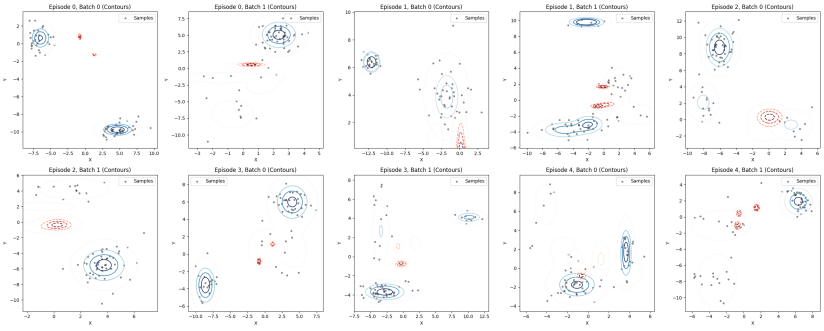

Figure 3.2: Analytic contour plots comparing ground-truth GMM densities (blue) and predicted MDN densities (red, dashed). Grey points denote samples from the predicted MDN. This provides a faithful visualization of the model's density estimation ability, highlighting alignment and mismatches between predicted and true probability mass.

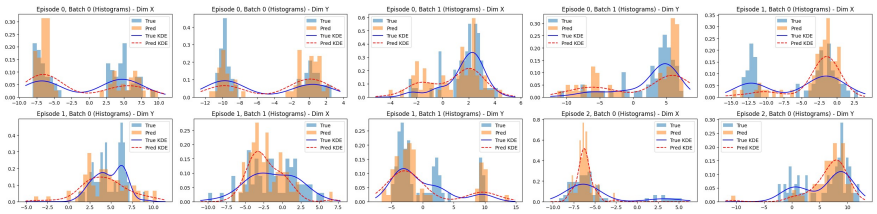

Figure 3.3: Histograms of marginal distributions along Dimension X and Dimension Y for representative episodes(Only a few were possible to be given. Rest in the appendix) Predicted samples (orange) are compared against ground-truth queries (blue), with KDE overlays (solid: true, dashed: predicted) to illustrate calibration of one-dimensional marginals.

Limitations. While the Transformer–MDN offers a flexible framework for amortized inference on synthetic GMMs, several limitations arise from the mixture density network formulation. First, covariances are restricted to be diagonal, preventing the model from explicitly capturing correlations between dimensions. Contour plots therefore reflect only axis-aligned variability, with no off-diagonal structure. Second, mixture density networks are known to overestimate variance in regions where modes overlap, sometimes producing overly diffuse predictions. Finally, the Gaussian mixture family itself constrains the class of representable distributions, limiting expressivity relative to more general nonparametric approaches. These limitations should be kept in mind when interpreting both quantitative metrics and qualitative plots.

3.3 Quantitative Evaluation

To complement the qualitative analysis in Section 3.2, we report quantitative metrics that assess the fidelity of reconstructed samples relative to ground-truth queries. Specifically, we use two classes of distributional similarity measures:

- **Wasserstein Distance (WD):** measures the minimum "cost of transport" between predicted and true distributions. Sensitive to cluster displacement and spread.
- **Maximum Mean Discrepancy (MMD):** with an RBF kernel, assesses similarity between predicted and true *joint distributions* in two dimensions.

Baselines. To contextualise performance, we include two reference baselines: (i) a *lower bound*, corresponding to a global Gaussian fitted across all query points (a poor amortised approximation); and (ii) an *upper bound*, given by fitting a per-episode Gaussian mixture with optimised number of components ($K \in \{1,\dots,6\}$). The upper bound acts as an oracle that adapts directly to each test episode, and therefore represents the best achievable alignment without amortisation.

Table 3.1: Distributional similarity metrics across test episodes (mean $\pm$ standard deviation). The proposed model is compared against a Gaussian lower bound and a per-episode GMM upper bound.

Method	**WD**	**MMD**
Lower (Gaussian)	6.70 ± 2.20	0.171 ± 0.088
Transformer–MDN (ours)	2.09 ± 1.05	0.060 ± 0.030
Upper (per-episode GMM)	0.79 ± 0.29	0.017 ± 0.007

The results in Table 3.1 show that our amortised model lies much closer to the upper bound than to the lower bound, indicating successful in-context inference of the target distributions. While a gap remains relative to the oracle GMM, the substantial improvement over the Gaussian baseline demonstrates that the transformer learns useful amortisation rather than regressing to context means.

Per-episode variation. To better understand variability across episodes, we also plot per-episode scores for WD and MMD (Figure 3.4). These reveal that the majority of episodes achieve close alignment, while a minority of challenging episodes (typically with overlapping or imbalanced clusters) account for most of the variance.

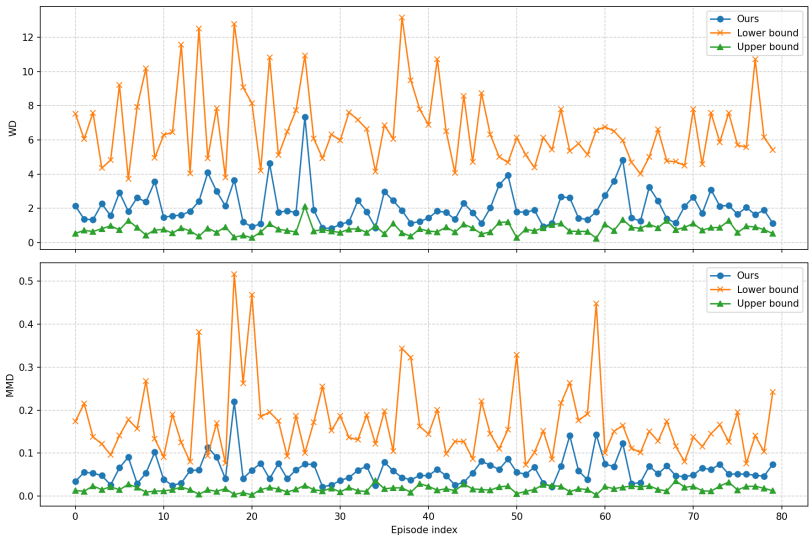

Figure 3.4: Per-episode WD and MMD scores across 100 test episodes. The proposed model (orange) consistently outperforms the Gaussian lower bound (blue), and approaches the per-episode GMM upper bound (green). A small subset of difficult episodes dominates the variance.

The WD scores indicate that the model preserves global cluster structure but suffers when mixture components overlap, leading to increased transport costs. Meanwhile, the low MMD scores demonstrate that despite per-dimension variability, the model successfully recovers the overall *joint distributional structure* of the queries.

3.3.1 KL Divergence.

Unlike Wasserstein Distance and MMD, the Kullback–Leibler (KL) divergence between Gaussian mixtures has no closed-form expression, due to the intractability of integrating over log-mixtures. Prior work, such as Durrieu and Thiran [23], has explored tractable bounding strategies for this problem, highlighting the challenges involved in exact computation.

In this project, we instead adopt a Monte Carlo estimation approach, drawing samples from the predicted MDN distribution $p(x)$ and computing the average log-likelihood ratio with respect to a chosen reference distribution $q(x)$, as described by Hershey and Olsen [22]:

$$\mathrm{KL}(p \,\|\, q) = \mathbb{E}_{x\sim p(x)}[\log p(x) - \log q(x)]$$

This allows us to directly compare distributions despite the absence of a closed-form solution, while accounting for the inherent sampling variance of the estimator.

To contextualise KL values, we report three variants with different choices of $q(x)$:

- **KL(Pred‖True):** the principal measure, where $q(x)$ is the ground-truth episodic GMM with known mixture parameters. This yields the most faithful comparison, since the true generative distribution is exactly available in our synthetic benchmark.

- **KL(Pred‖Ep-GMM):** an *upper baseline*, where $q(x)$ is a per-episode GMM fitted directly to query samples using BIC model selection over $K \in \{1,\ldots,6\}$. This reference adapts flexibly to each episode and can capture multimodality, but risks overfitting on small or imbalanced queries, which in turn inflates KL estimates.

- **KL(Pred‖Glob-Gauss):** a *lower baseline*, where $q(x)$ is a single Gaussian fitted across all query points in the test set. This amortised baseline is deliberately weak—it ignores per-episode structure—but provides a stable floor for KL values and highlights how much benefit is gained from modelling multimodality.

Together, these three variants provide a spectrum of comparisons: KL(Pred‖True) quantifies alignment to the exact target distribution, KL(Pred‖Ep-GMM) assesses closeness to a flexible but potentially overfit oracle, and KL(Pred‖Glob-Gauss) benchmarks against a simple unimodal approximation. This framing makes it possible to interpret both the absolute KL magnitudes and their variability across episodes.

Across 80 held-out episodes, the KL divergence to the true GMM averages 1.47 ± 1.41, while the global Gaussian baseline yields 1.11 ± 0.60, and the per-episode fitted GMM baseline 2.28 ± 2.21. The relatively low mean values indicate that the Transformer–MDN is generally able to recover densities that align closely with the underlying generative distributions. Interestingly, in some episodes the global Gaussian baseline produces lower divergence values than the ground-truth GMM reference. This

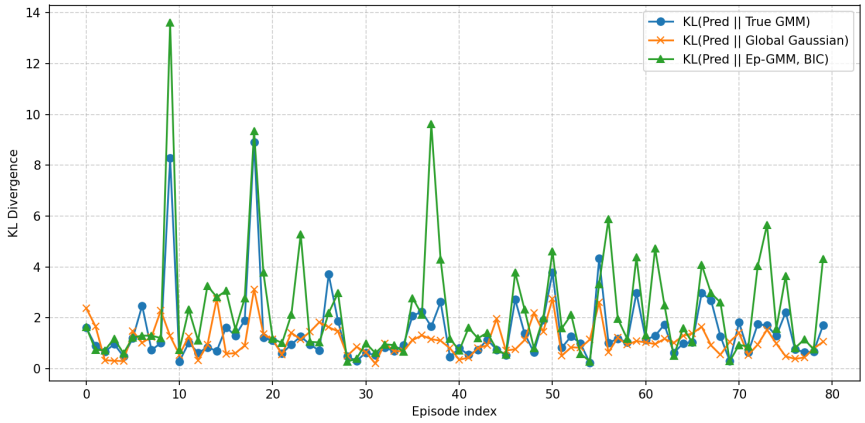

Figure 3.5: Per-episode KL divergences between MDN predictions and reference distributions. Blue: KL(Pred‖True GMM) — principal comparison; Orange: KL(Pred‖Global Gaussian) — lower baseline; Green: KL(Pred‖Ep-GMM, BIC) — upper baseline. Occasional spikes reflect episodes where the fitted reference either oversimplifies (global Gaussian) or overfits (Ep-GMM) the underlying distribution.

is because the global Gaussian, despite being less expressive, provides a smoother coverage across the distribution's support, whereas the oracle GMM can overfit to small or imbalanced query sets, penalising samples outside tightly concentrated modes. Overall, the global Gaussian remains stable but underestimates multimodality, while the fitted GMM baseline adapts more closely to local structure but introduces greater variance under challenging scenarios.

3.4 Interpretation of Results and Reflection

The evaluation results, both qualitative and quantitative, provide complementary insights into the behaviour of the proposed Transformer–MDN framework for amortized inference.

3.4.1 Alignment with Objectives

The model meets its primary objective of reconstructing the underlying distribution of query points from fully masked inputs. Qualitative scatterplots (Figure 3.1) demonstrate that the predicted samples reliably recover the *cluster structure* of the ground-truth Gaussian mixture queries. Likewise, histograms (Figure 3.3) show that marginal means and variances are generally well aligned. These observations confirm that the model does not regress to context means, but instead learns to approximate the multimodal posterior distribution.

Quantitatively, comparison with baselines (Table 3.1) shows that the Transformer–MDN substantially outperforms the Gaussian lower bound and lies closer to the per-episode oracle. Per-episode plots (Figure 3.4) further reveal that most episodes achieve strong alignment, with only a few difficult cases driving the observed variance.

3.4.2 Observed Limitations

Despite these encouraging results, several limitations are evident. Performance is less stable when mixture components overlap or have imbalanced weights. In such cases, predicted samples become diffuse and transport costs rise, leading to higher Wasserstein distances. This indicates that the model struggles with precise allocation of probability mass when modes are close together. Nevertheless, even in these challenging scenarios the model consistently outperforms the Gaussian baseline, with the gap to the per-episode oracle explaining most of the shortfall.

Another limitation arises from the metrics themselves. While WD, KL, and MMD provide principled measures of distributional similarity, they are sensitive to sample size and hyperparameters. For example, KL divergence can spike when empirical variances are small, WD can be inflated by outliers, and MMD depends heavily on kernel bandwidth. Thus, while the scores consistently favour the proposed approach, their interpretation should be contextualised against these sensitivities.

3.4.3 Reflection on Methodology

The choice of a transformer encoder as the amortized inference mechanism appears well motivated. Its permutation-invariant tokenisation scheme supports flexible masking without positional encoding, and attention provides a natural way to model global dependencies among context points. However, given the low dimensionality of the synthetic benchmark, the architecture may be heavier than necessary; simpler models could achieve comparable results at lower cost.

Equally important is the decision to parameterise predictive densities with a mixture density network (MDN). This design entails two structural assumptions. First, restricting components to diagonal Gaussians prevents the model from capturing correlations between dimensions. This explains why analytical density contour plot sometimes suggest spurious elliptical contours. Second, MDNs often inflate variance in regions where clusters overlap, reflecting a trade-off between covering multiple modes and maintaining sharp predictions. These effects are not implementation bugs but direct consequences of the MDN formulation. Future work could explore richer parameterisations, such as full-covariance Gaussians, normalising flows, or nonparametric density estimators to relax these constraints.

3.4.4 Summary

Overall, the evaluation shows that the proposed Transformer–MDN balances accuracy with uncertainty representation. It reconstructs multimodal structure without collapsing to trivial solutions, yet remains less stable in overlap-heavy regimes where probability mass allocation is ambiguous. The combination of qualitative visualisations and quantitative metrics provides a coherent assessment of performance. Future work should extend beyond synthetic GMMs to higher-dimensional and real-world datasets, while refining both the evaluation metrics and the expressiveness of the predictive distributions.

Chapter 4

Project Achievement

The artefact produced in this project, a transformer-based mixture density network trained for amortized inference under fully masked query settings, can be evaluated along two key dimensions: its *complexity and scope*, and the *quality of execution*.

4.1 Complexity and Scope

This project addresses a non-trivial and relatively unexplored problem: using transformers as inference networks for reconstructing full distributions when no query inputs are observed. Unlike standard imputation tasks, which assume partial observability, this setting requires the model to infer the posterior distribution purely from context statistics. Several factors contribute to the artefact's complexity:

- A controlled synthetic benchmark was created using episodic Gaussian Mixture Models (GMMs), with randomised means, covariances, and mixture weights. This provided a rigorous yet interpretable testbed, while ensuring clear separation between training, validation, and test splits.

- The learning objective combined differentiable probabilistic losses, including Gaussian negative log-likelihood (NLL) for direct supervision and auxiliary regularisation (e.g. entropy penalties), to prevent variance collapse and improve stability under full masking.

- A dual evaluation pipeline was designed, spanning both *qualitative* (scatterplots, histograms, contour maps) and *quantitative* (Wasserstein distance, Maximum Mean Discrepancy, KL divergence) measures. This allowed the artefact to be assessed in terms of marginal fidelity, joint structure, and divergence from reference baselines.

- The transformer architecture was adapted to act as a Bayesian inference network: context and query tokens were embedded jointly, with attention layers providing permutation-invariant conditioning on available context.

These design choices ensure that the artefact achieves a meaningful level of complexity and scope appropriate for MSc-level research, while remaining tractable and interpretable.

4.2 Execution Quality

The implementation demonstrates reliability and technical soundness in several respects:

- Training stability was achieved through a carefully balanced objective, preventing trivial solutions such as collapse to context means or variance blow-up.
- Generalisation beyond trivial baselines was confirmed: the model reconstructs multimodal structures across unseen episodes and captures global cluster geometry under fully masked queries.
- The codebase was implemented modularly, separating dataset generation, model architecture, training pipelines, and evaluation scripts. This not only facilitated reproducibility but also enabled extensions, such as swapping evaluation metrics or testing alternative density heads.

4.3 Reliability

Reliability was established through multiple independent runs and per-episode analysis. While absolute metric values varied due to stochasticity, trends remained consistent, suggesting robustness of the overall approach. Importantly, dataset splits were performed at the episode level, avoiding leakage and ensuring fairness of evaluation.

4.4 Summary

In summary, the project achieves its stated objectives by producing a technically sound artefact of moderate-to-high complexity. Although the experimental evaluation was restricted to two-dimensional Gaussian mixtures, this was an intentional choice: the low-dimensional setting enables controlled benchmarking, interpretability of results, and clear visualisation of distributional reconstructions. Within this scope, the project goes beyond conventional imputation tasks by addressing the harder problem of fully masked inference, while the execution quality ensures both reliability and reproducibility. The transformer–MDN consistently demonstrated its ability to perform amortized inference in a fully masked setting, outperforming simple Gaussian baselines and approaching per-episode oracle models in several metrics.

Chapter 5

Conclusions

The goal of this project was to determine whether transformers might be applied as amortised inference networks for fully unsupervised data reconstruction. The study expanded this concept from Gaussian Process benchmarks to Gaussian Mixture Models (GMMs), which produced a regulated but difficult testbed. This was motivated by previous work on transformers as Bayesian inference engines. The objectives, as indicated in the introduction, included generating a benchmark dataset, establishing a transformer-based inference model, designing an acceptable evaluation framework, and analysing performance under fully masked query settings.

5.1 Summary of Findings

The initiative accomplished numerous major outcomes:

- A synthetic episodic dataset was developed, giving fine control over mixture complexity while guaranteeing rigorous separation of training, validation, and test episodes. This provides a reliable testbed for amortised inference.

- A transformer–MDN architecture was implemented, capable of anticipating mixture weights, means, and variances. Results demonstrated that the model can amortize inference across diverse episodes, recovering multimodal structure even when all queries are concealed.

- A complete assessment framework was established, integrating qualitative visualisations (scatterplots, histograms, contour maps) with quantitative metrics (Wasserstein distance, KL divergence, and MMD). Findings showed that themodel reconstructs cluster structure reliably and produces marginals with reasonable calibration. Strong performance was reported on well-separated mixtures, but small-variance or overlapping modes showed restrictions

5.2 Reflections on the Approach

Methodologically, the project highlights the possibilities of transformer-based inference: permutation-invariant context encoding, flexibility to masking techniques, and quick posterior approximation with amortisation. At the same time, it emphasises limitations:

- The MDN head, while efficient, introduces variance inflation in overlapping clusters and cannot capture correlations due to its diagonal covariance assumption.
- Careful regularisation (such as entropy bonuses) was necessary for robust training, showing sensitivity to hyperparameters.
- Evaluation was restricted to low-dimensional synthetic data. This was an intentional choice for interpretability and controlled benchmarking, but limits claims regarding scalability to high-dimensional or real-world domains.

5.3 Future Work

Several natural extensions emerge from this work:

- **Scaling:** Extending beyond two dimensions to evaluate robustness in higher-dimensional spaces and more complex mixture structures.
- **Applications:** Applying the framework to real-world datasets, such as spatio-temporal biological or physical systems, to demonstrate practical utility.
- **Architectural advances:** Replacing MDN heads with more expressive density estimators such as normalising flows, diffusion-based decoders, or full-covariance mixtures.
- **Evaluation metrics:** Incorporating additional measures (e.g. energy distance, calibration error) to better capture posterior quality and robustness.

5.4 Final Remarks

In conclusion, this project provides evidence that transformers can act as effective amortized inference networks for data reconstruction under fully masked conditions. By combining principled synthetic benchmarks with distribution-sensitive evaluation, it contributes both a methodological perspective and an empirical demonstration of feasibility. Although challenges remain in scaling, variance calibration, and application to real data, the artefact is technically sound, well engineered, and offers a strong foundation for further research. In particular, the work underscores the promise of transformer-based models as flexible Bayesian inference engines, bridging the gap between deep learning architectures and classical statistical inference.

Bibliography

[1] R. J. A. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. Hoboken, NJ: John Wiley & Sons, 3 ed., 2019.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[3] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. New York: John Wiley & Sons, 1987.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. International Conference on Learning Representations (ICLR)*, 2014. arXiv:1312.6114.

[5] S. Müller, N. Hollmann, S. Pineda Arango, J. Grabocka, and F. Hutter, "Transformers can do bayesian inference," *arXiv preprint arXiv:2112.10510*, 2021. Published at ICLR 2022; code: https://github.com/automl/TransformersCanDoBayesianInference.

[6] T. J. B. Liu, N. Boullé, R. Sarfati, and C. J. Earls, "Density estimation with llms: a geometric investigation of in-context learning trajectories," *arXiv preprint arXiv:2410.05218*, 2024.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS) 30*, 2017.

[8] A. Reuter, T. G. J. Rudner, V. Fortuin, and D. Rügamer, "Can transformers learn full bayesian inference in context?," in *International Conference on Machine Learning (ICML) 2025 Poster*, 2025. OpenReview poster.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[10] Z. Chen, R. Wu, and G. Fang, "Transformers as unsupervised learning algorithms: A study on gaussian mixtures," *arXiv preprint arXiv:2505.11918*, 2025.

Proposes TGMM, a transformer-based method that learns to solve GMM tasks across mixture components; shows transformers can approximate EM and spectral methods.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, Association for Computational Linguistics, 2019.

[12] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv preprint arXiv:2111.06377*, 2021.

[13] S. Mittal, "In-context parametric inference: Point or distribution ...," *arXiv preprint arXiv:2502.11617*, 2025.

[14] M. Germain, K. Gregor, I. Murray, and H. Larochelle, "Made: Masked autoencoder for distribution estimation," in *International Conference on Machine Learning (ICML)* (F. Bach and D. M. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 881–889, PMLR, 2015.

[15] S. An, G. Woo, J. Lim, C. Kim, S. Hong, and J.-J. Jeon, "Masked language modeling becomes conditional density estimation for tabular data synthesis," *arXiv preprint arXiv:2405.20602*, 2024. Accepted at the AAAI Conference 2025.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[17] Q. Wang, B. Li, T. Xiao, J. Zhu, and C. Li, "Learning deep transformer models for machine translation," in *arXiv preprint arXiv:1906.01787*, 2019.

[18] Anonymous, "Context normalization layer with applications," *arXiv preprint arXiv:2303.07651*, 2023.

[19] K. Nadjahi, R. Flamary, N. Courty, and I. Redko, "Statistical optimal transport poses sample complexity challenges," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[20] Z. Goldfeld, J. Weed, and P. Rigollet, "Convergence of optimal transport and quantization problems," *The Annals of Statistics*, vol. 47, no. 2, pp. 782–810, 2019.

[21] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[22] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, p. 317–320, IEEE, 2007.

[23] J.-L. Durrieu, J.-P. Thiran, and F. P. Kelly, "Lower and upper bounds for approximation of the kullback–leibler divergence between gaussian mixture models," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4833–4836, IEEE, 2012.

Appendix A

List of Notations and Symbols

Notation used throughout the report.

Symbol	Description
K	Number of Gaussian mixture components in an episode.
π_k	Mixture weight of the k-th Gaussian component.
μ_k	Mean vector of the k-th Gaussian component.
σ_k	Diagonal covariance (standard deviation) of the k-th Gaussian component.
$X_c = \{x_i\}_{i=1}^{N_c}$	Context set of fully observed samples.
$X_q = \{x_j\}_{j=1}^{N_q}$	Query set of fully masked samples.
μ_c	Empirical mean of the context set (used for per-episode normalisation).
σ_c	Empirical standard deviation of the context set (used for per-episode normalisation).
$x \odot m$	Elementwise product of input x with mask m.
m	Binary mask vector ($m = 1$ for context, $m = 0$ for queries).
$\mu_{\text{ctx}}, \sigma_{\text{ctx}}$	Stored context statistics appended to all tokens for normalisation.
z	Token embedding vector constructed as $[x \odot m, m, \mu_{\text{ctx}}, \sigma_{\text{ctx}}]$.
d_{model}	Transformer embedding dimension.
H_{qry}	Encoded hidden representation of query tokens from Transformer.
π, μ, σ	Mixture Density Network (MDN) outputs: weights, means, and standard deviations.
$p(x \mid C)$	Predictive density of query points conditioned on context.
L_{NLL}	Negative log-likelihood loss of predicted queries.
L	Final training objective with entropy regularisation.
λ	Regularisation hyperparameter for variance entropy term.
$W_\pi, W_\mu, W_{\log\sigma}$	Trainable linear projections producing MDN parameters.
$W(p,q)$	Wasserstein distance between predicted and ground-truth distributions.
MMD	Maximum Mean Discrepancy metric for distributional similarity.

Symbol	Description
$\mathrm{KL}(p \parallel q)$	Kullback–Leibler divergence between two distributions p and q.

A.1 Plots

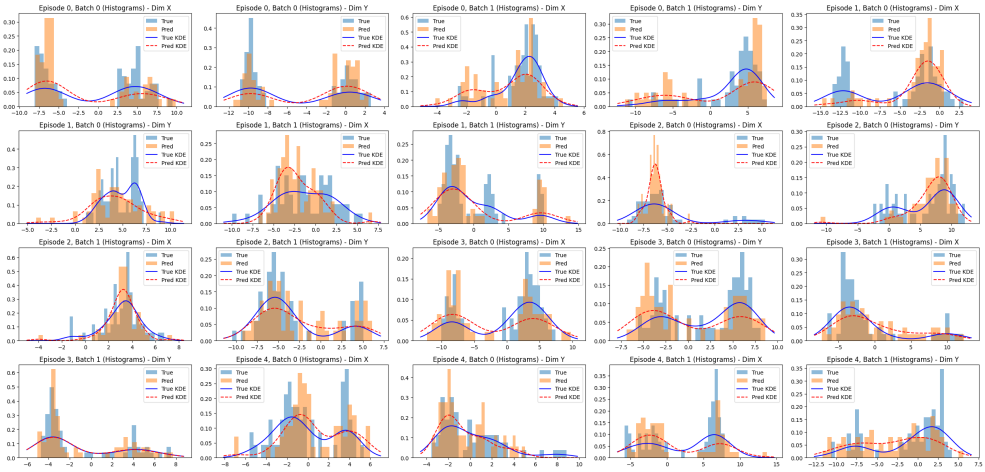

Figure A.1: Set of all the histograms generated for the Marginal probabilities throughout the episodes

A.2 Code Listing

A compact PyTorch implementation of one dataset episode is shown below:

Listing A.1: Episodic GMM dataset with fully masked queries.

```
class EpisodicGMM(Dataset):
    def __getitem__(self, idx):
        # sample GMM parameters
        K, pis, means, sigs = self._sample_gmm()
        Xc = self._sample_points(K, pis, means, sigs, self.Nc)
        Xq = self._sample_points(K, pis, means, sigs, self.Nq)

        # normalisation
        mu, std = Xc.mean(0, keepdims=True), Xc.std(0, keepdims=True)+1e-
        Xc = (Xc - mu) / std
        Xq = (Xq - mu) / std

        # full masking of queries
        Mq = torch.zeros_like(Xq)
        return torch.tensor(Xc), Mq, torch.tensor(Xq)
```

Link to code: https://colab.research.google.com/drive/1FkddkT9q2S5UD6_IfwpLGA0FtIAju8DF?usp=sharing