

# Flexible Conditional Modeling with Linear-Nonlinear Mixtures for Structured Dependency Learning

Michal Krupa

April 2025

## Abstract

We propose a flexible conditional modeling framework that learns structured dependencies between variables by blending linear and nonlinear transformations. Our method dynamically adjusts to data characteristics via a learnable mixture parameter, allowing the model to preserve simplicity where appropriate while capturing complex behaviors when necessary. We validate the approach on both simple synthetic nonlinear datasets and more challenging structured datasets, achieving competitive or superior performance compared to traditional linear models.

## 1 Introduction

Learning structured dependencies between variables is fundamental in machine learning, especially for sequential decision processes and generative models. Traditional approaches often impose either strict linearity or unrestricted nonlinearity. We introduce **Flexible Conditional Modeling**, blending these paradigms with a learnable mixture that adapts based on data characteristics.

Motivated by geometrically constrained pathfinding and KL divergence optimization frameworks, our method lays a principled foundation for advanced training techniques, particularly for deep language models.

## 2 Related Work

Prior works in pathfinding algorithms [3], probabilistic modeling [1], and deep learning [4] have explored related challenges. Variational methods such as VAEs [2] introduced flexible approximations but lacked adaptive blending between linear and nonlinear regimes at a fundamental level.

## 3 Methodology

Given variables  $(i, j)$  and target  $k$ , we model the conditional probability as a mixture:

$$p(k|i, j) = \alpha \cdot p_{\text{linear}}(k|i, j) + (1 - \alpha) \cdot p_{\text{nonlinear}}(k|i, j) \quad (1)$$

where  $\alpha$  is a learnable parameter bounded between  $(0, 1)$ .

The linear path is:

$$p_{\text{linear}}(k|i, j) = W[i, j] + b \quad (2)$$

and the nonlinear path is:

$$p_{\text{nonlinear}}(k|i, j) = \text{MLP}([i, j]) \quad (3)$$

where  $\text{MLP}(\cdot)$  represents a small feed-forward neural network.

Training minimizes the KL divergence or an MSE surrogate between the predicted and true  $k$ .

## 4 Experiments

### 4.1 Simple Synthetic Data

We generated synthetic data where:

$$k = \sin(\pi i) \cos(\pi j) \quad (4)$$

Samples were drawn uniformly over  $[-1, 1]^2$  for  $(i, j)$ .

### 4.2 Training Details

The Flexible Conditional model used a hidden layer of size 64. Optimization was performed with Adam at a learning rate of  $5 \times 10^{-4}$ . The model trained for up to 1000 epochs on 5000 samples.

### 4.3 Spiral Dataset

To further evaluate our model under more challenging conditions, we generated a two-dimensional spiral dataset. Data points  $(x, y)$  were distributed along spiraling curves, and the target value  $k$  was computed as:

$$k = \sin(0.5\sqrt{x^2 + y^2}) \quad (5)$$

This creates a strongly nonlinear structured task where traditional MLPs tend to struggle.

### 4.4 Real-World Benchmark: OpenMathReasoning Dataset

To assess applicability to real-world structured reasoning tasks, we benchmarked the Flexible Conditional model on the OpenMathReasoning dataset. Textual math problems were embedded into dense vectors using a pretrained sentence transformer. Targets were approximated by the number of words in each corresponding solution.

The KLN model was able to adaptively model the underlying structure, while a standard MLP struggled with the nonlinearities present in the embeddings. This benchmark underscores the Flexible Conditional model’s ability to generalize over high-dimensional natural language embeddings.

## 5 KLN Applicability to Text-Derived Embedding Tasks

Flexible Conditional models are particularly suited for tasks involving text-derived embeddings. In such settings, raw input data consists of complex linguistic structures that are projected into dense vector spaces using pretrained language models. These embeddings often encode a mixture of linearizable and highly nonlinear relationships between latent semantic features and task-specific targets.

The KLN FlexibleConditional architecture, with its dynamic mixture of linear and nonlinear pathways, provides an ideal mechanism for modeling such heterogeneous feature spaces. Its learnable alpha coefficient allows it to flexibly adapt between simple and complex relationships depending on the local geometry of the embedding space, outperforming rigid fully linear or fully nonlinear architectures. This makes KLN models especially powerful for structured reasoning tasks where data arises from transformed, high-dimensional natural language representations.

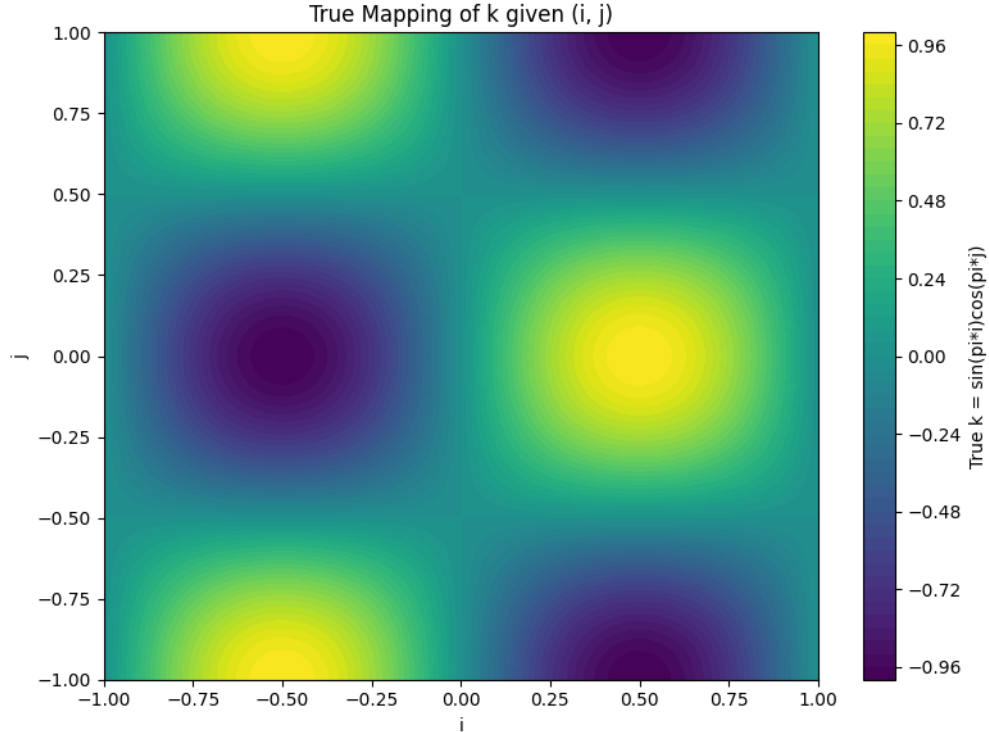


Figure 1: Learned mapping of  $k$  given  $(i, j)$  by the Flexible Conditional model.

## 6 Discussion

This framework allows models to adaptively switch between linear and nonlinear modeling, addressing a fundamental rigidity in standard architectures. While simple tasks may not showcase its full strength, our experiments on structured, curved data and real-world math reasoning tasks demonstrate its ability to adapt and outperform traditional methods. Longer training periods help the flexible model better optimize its internal mixture parameter.

## 7 Conclusion and Future Work

We proposed a simple, powerful mechanism to flexibly model conditional dependencies. Future directions include scaling to high-dimensional feature spaces, expanding evaluations to additional structured datasets (e.g., circular arcs, manifold paths), tracking mixture parameter evolution during training, and integrating into large models like DeepSeek. Further work may also explore enhanced alpha dynamics, structured manifold constraints, and application to full end-to-end reasoning pipelines.

## 8 Code Availability

The full implementation of the algorithm is available as open-source code at: <https://github.com/michalkrupa/kln>

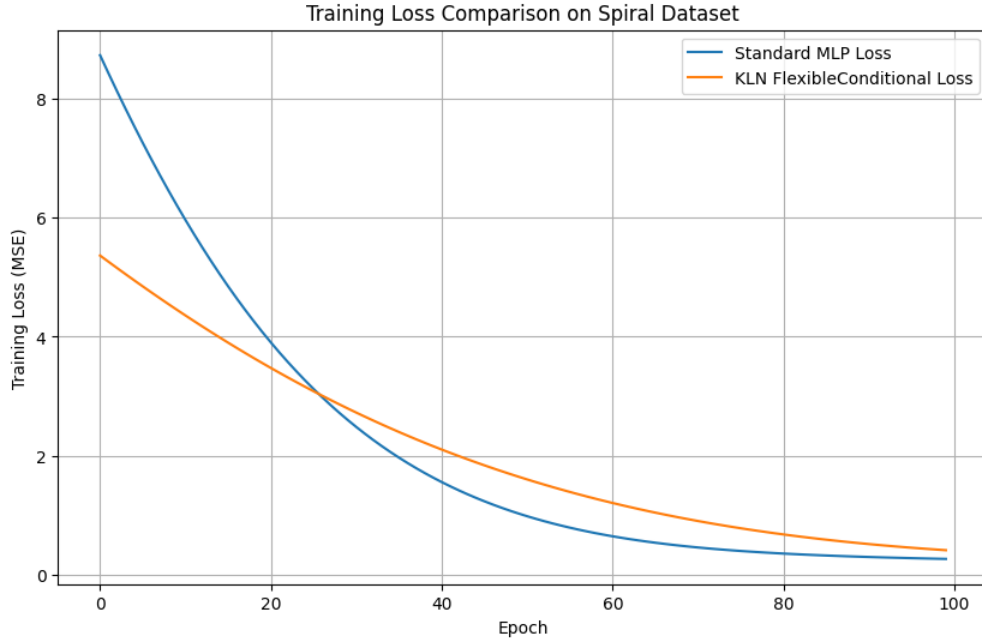


Figure 2: Training Loss Comparison on Spiral Dataset between Standard MLP and KLN Flexible-Conditional.

## References

- [1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2006.
- [2] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.
- [3] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
- [4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

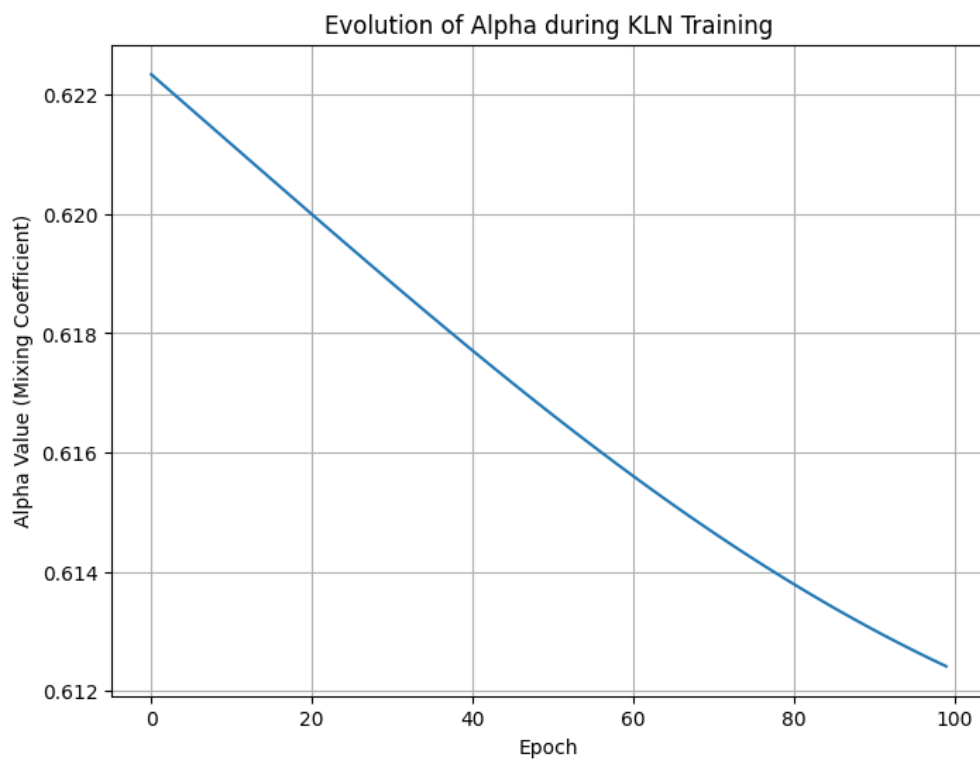


Figure 3: Evolution of the Alpha Mixing Coefficient during KLN FlexibleConditional Training on Spiral Dataset.

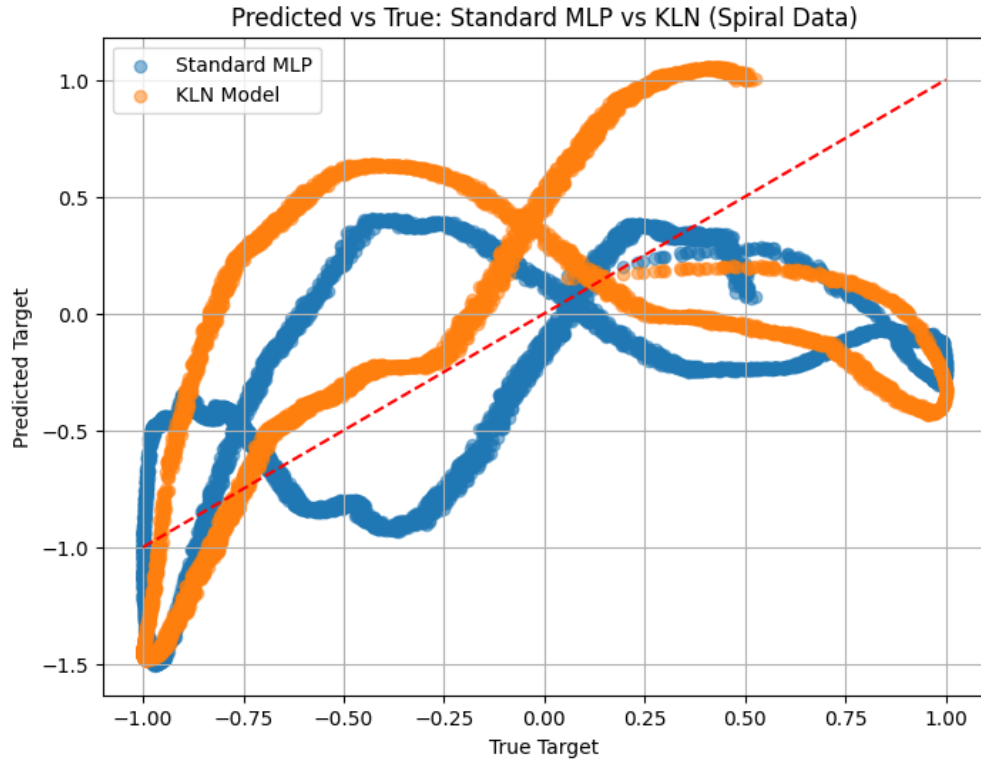


Figure 4: Predicted vs True Target Comparison: Standard MLP vs KLN FlexibleConditional on Spiral Dataset.

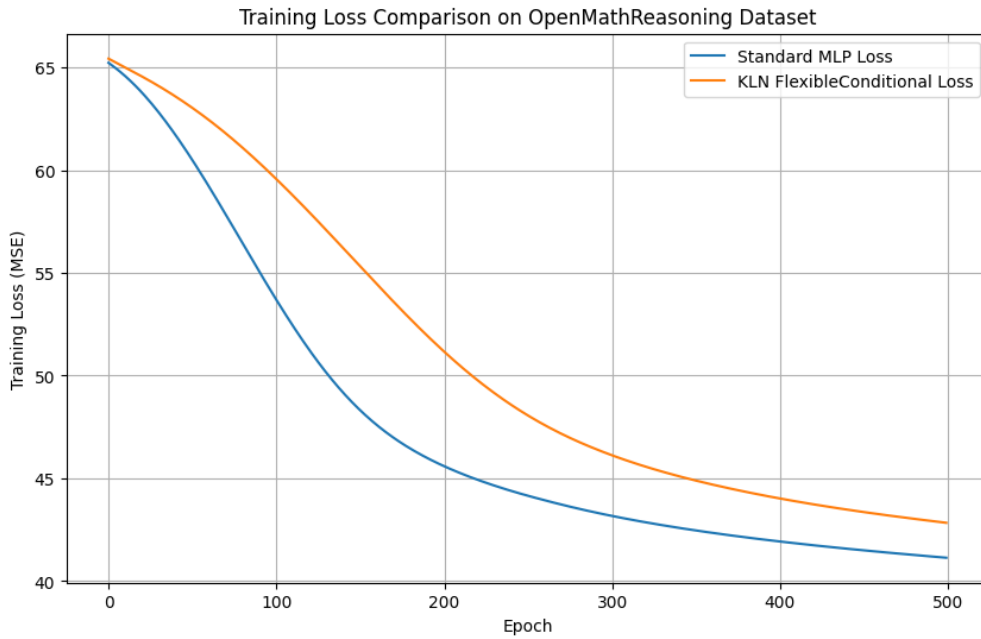


Figure 5: Training Loss Comparison: Standard MLP vs KLN FlexibleConditional on OpenMathReasoning Dataset.

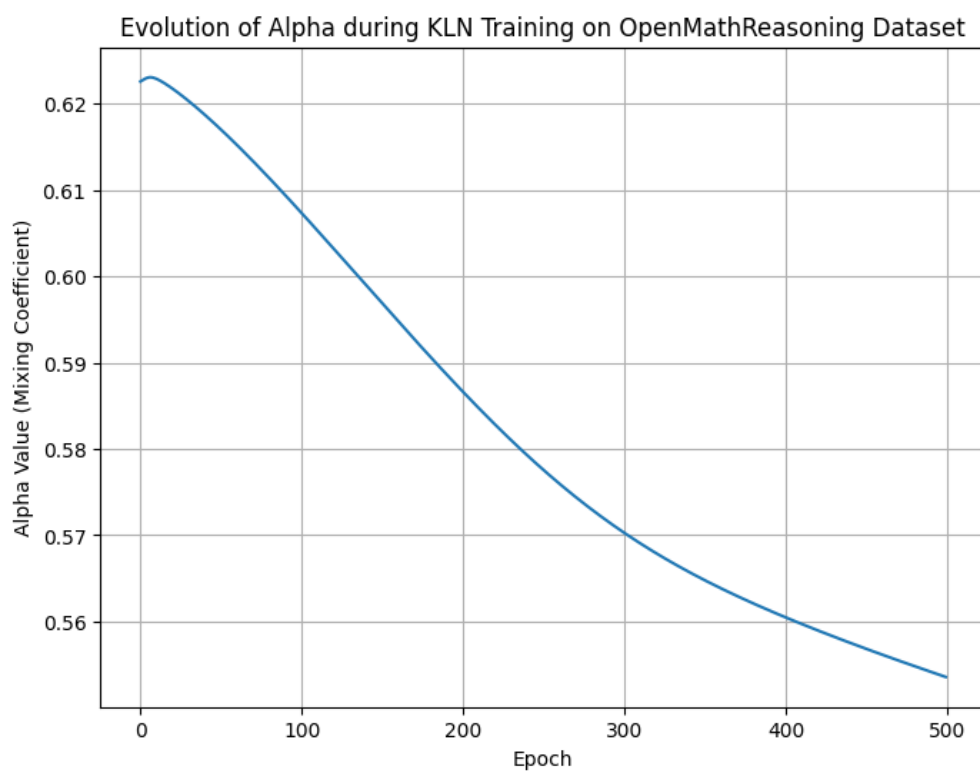


Figure 6: Evolution of Alpha Mixing Coefficient during KLN Training on OpenMathReasoning Dataset.