Information Visualization Recommendation System
VizML vs KG4Vis

L.EIC - Capstone Project 2024/25 - PE3

João Lamas, Pedro Fernandes, Tiago Pinheiro, Tiago Rocha

Supervised by: Alexandre Valle de Carvalho

U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

U.PORTO FC FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO

Introduction

- Data visualization efficiently summarizes complex datasets for quick understanding
- Manually selecting the right visualization is time-consuming and needs expert knowledge
- This project automates the process using VizML and KG4Vis to recommend suitable visualizations for any dataset

Challenge: Due to implementation limitations, we could not fully execute VizML, so we focused on KG4Vis for evaluation.

Objectives

- Compare VizML and KG4Vis in terms of accuracy, explainability, scalability and efficiency
- Analyze how each works and when one is more suitable than the other
- Evaluate both models on a dataset using key performance metrics

Methodology

- Conducted a literature review of both systems' papers
- Studied solutions for the prototypes and executed them on the given dataset
- Optimized configurations (batch sizes, hidden dimensions) for performance

VizML: Visualization Machine Learning

VizML is a **deep learning-based** system that recommends visualization types based on extracted dataset features.

- Implements a fully connected feedforward **neural network**
- Trained on 841 numerical and structural dataset features
- Uses PyTorch with batch training and adaptive learning rates

Advantages:

- Fast inference after training (according to authors, unable to verify)
- Learns complex patterns from large datasets

Limitations:

- Limited explainability (outputs only a chart type with no rationale)
- Requires retraining to incorporate new features or chart types

KG4Vis: Knowledge Graphs for Visualization Recommendation

KG4Vis uses a **knowledge graph** to model relationships between data features and visual encodings.

- **Nodes** represent dataset attributes, chart types, and encoding options
- **Edges** encode semantic relationships (e.g., "Categorical data → Bar Chart")
- Utilizes **TransE embeddings** to learn vector representations
- Enables inference-based recommendations with explanatory chains

Advantages:

- Provides transparent, explainable recommendations
- Flexible and extensible without retraining

Limitations:

- Higher computational cost
- Longer training time, especially without GPU acceleration

Result Discussion (KG4Vis)

1. Optimal Configuration

- Best performance achieved with hidden dimension=600 and batch size=1024
- Delivered 74.5% Hits@2 accuracy with 1.98 Mean Rank (lower is better)

2. Total Training Time

- Configuration **1:** ~2h
- Configuration **2**: ~23h

3. Unexpected Insight

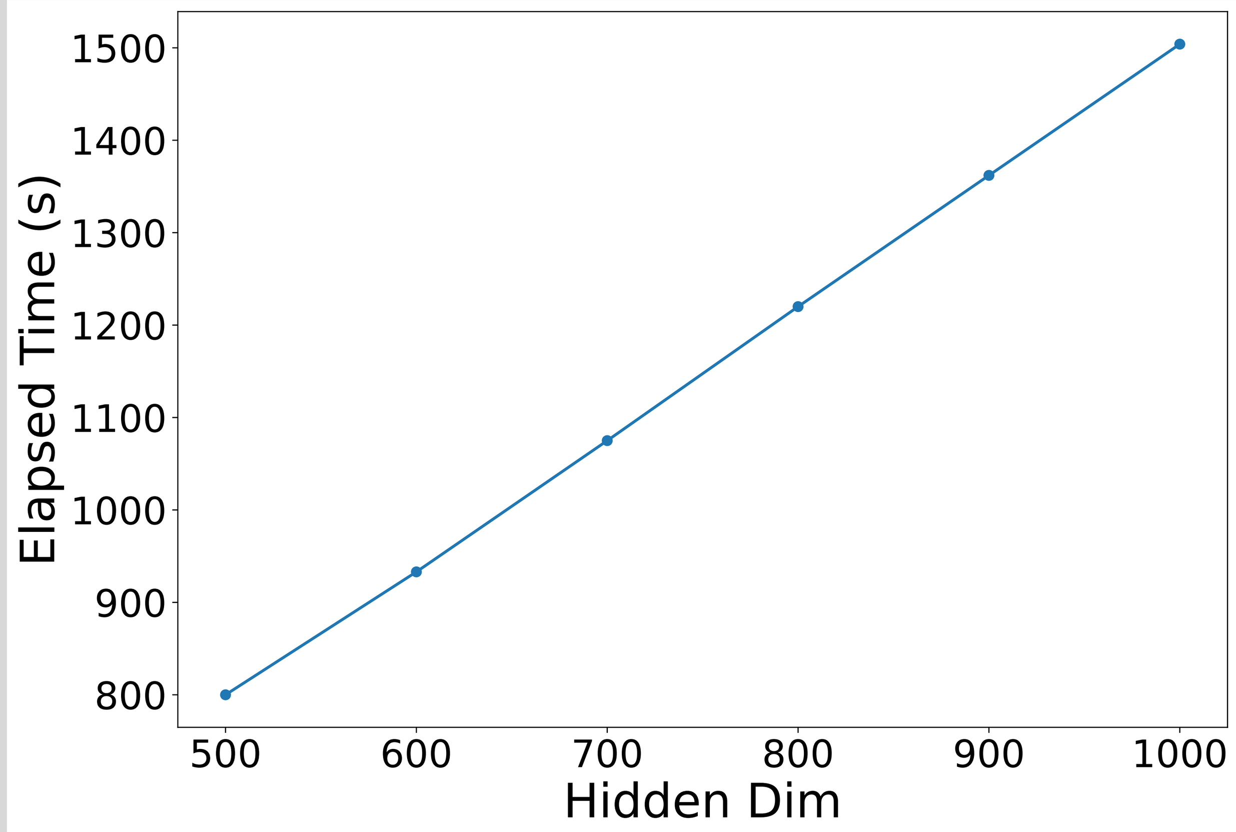
Smaller hidden dimensions (500-600) outperformed larger ones (800-1000), suggesting:

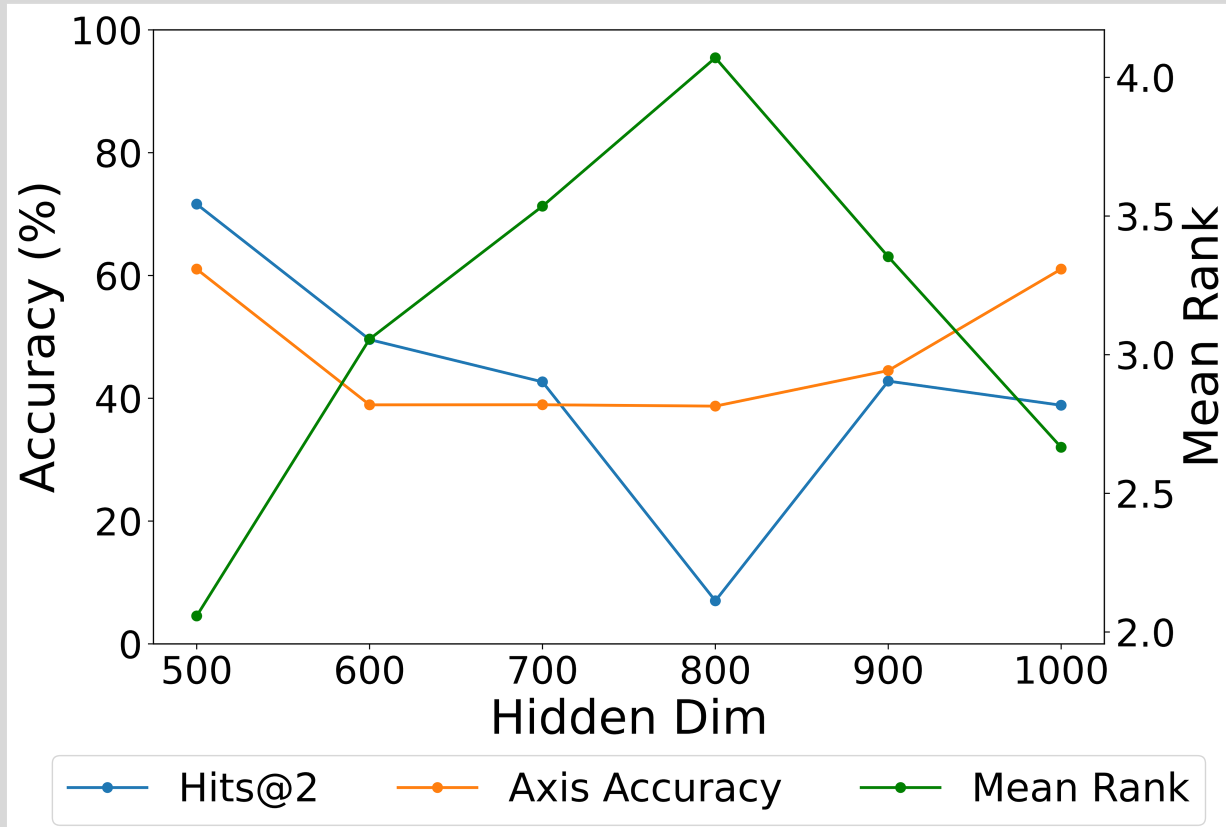
- Better generalization due to lower model complexity
- Reduced risk of overfitting

Critical Observations

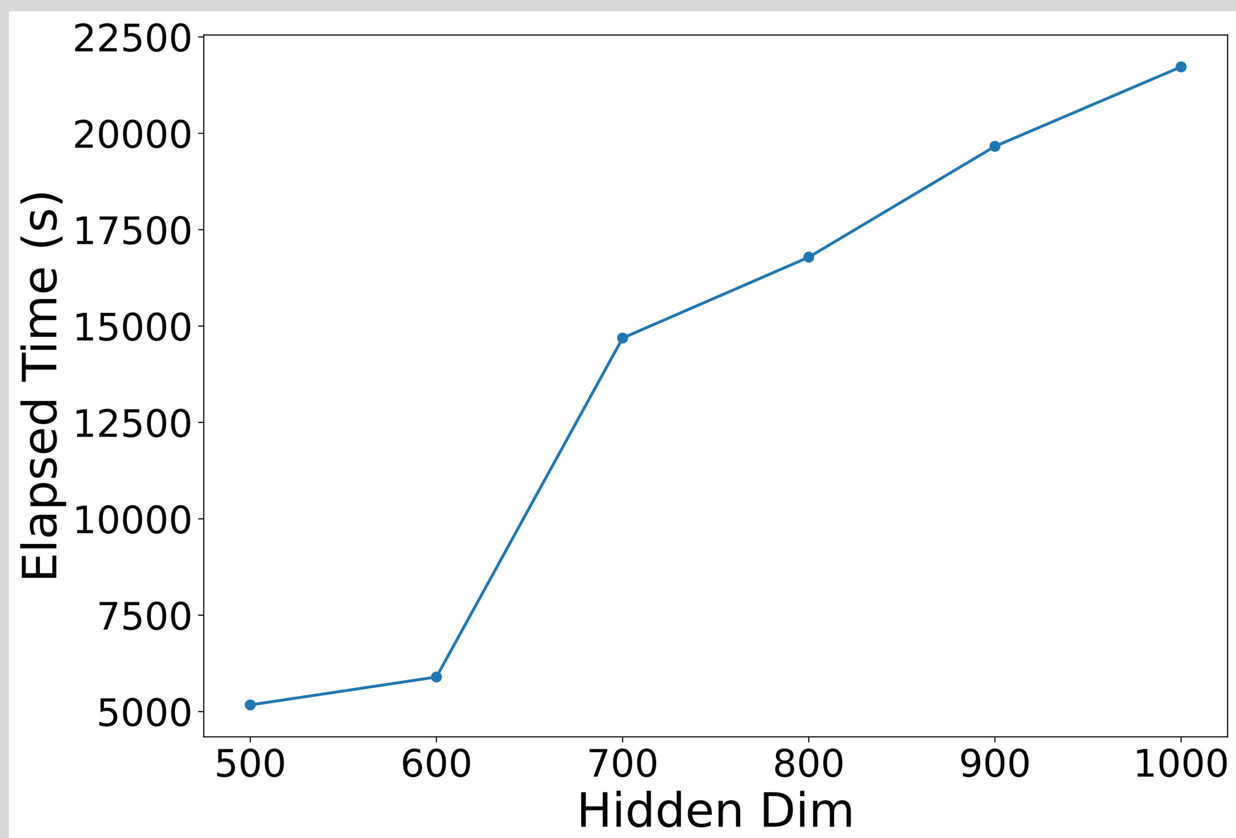
- **Hardware Impact:** GPU acceleration drastically reduced training time, although batch size also played a key role
- **Precision Trade-off:** Larger batch sizes (1024) led to higher accuracy and lower Mean Rank, despite longer training times

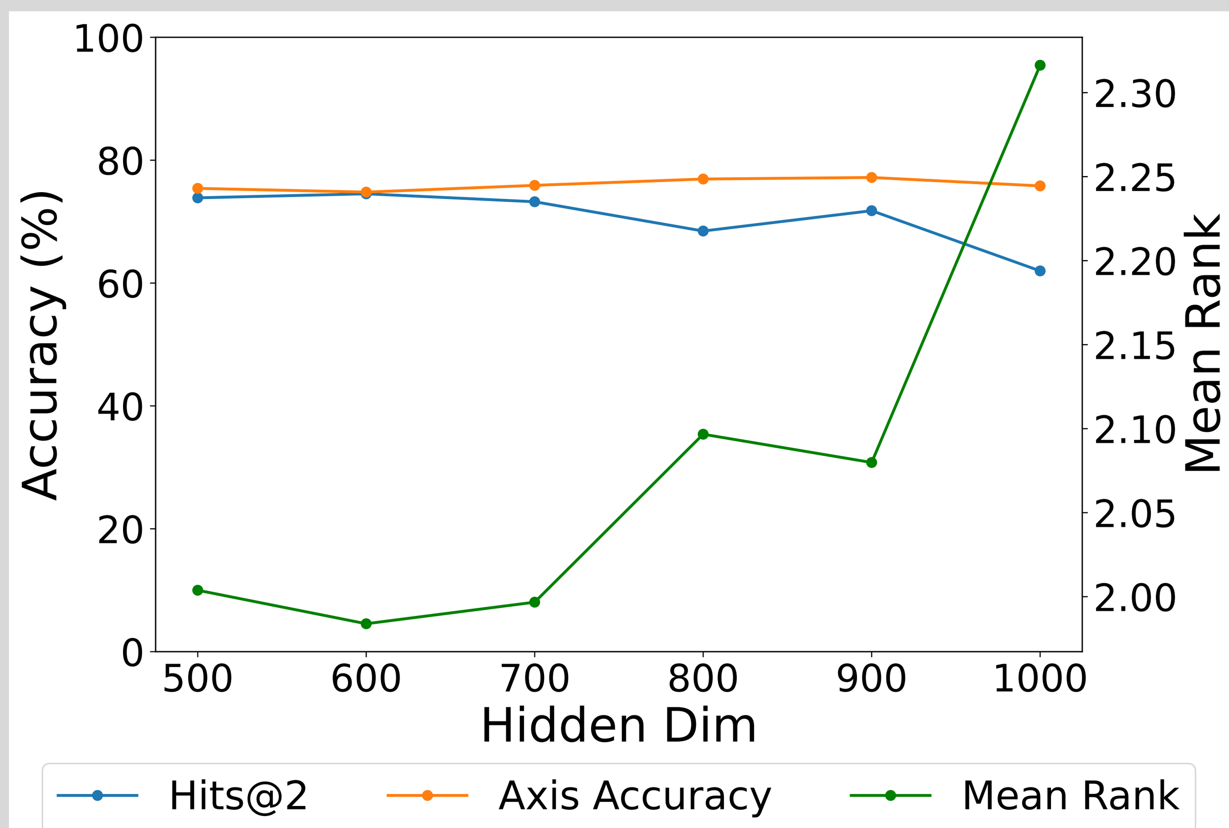
Comparative Analysis





Configuration 1 (CUDA-enabled): Batch size = 32





Configuration 2 (CPU only): Batch size = 1024

Conclusion

This project explored KG4Vis for visualization recommendation. Despite VizML's challenges, the work offered insights into embedding-based learning and explainable AI.

Future work may focus on: a) Successfully implementing and benchmarking VizML; b) Exploring hybrid methods combining ML speed with knowledge graph interpretability.