HEINZ NIXDORF INSTITUT
PADERBORN UNIVERSITY

DICE

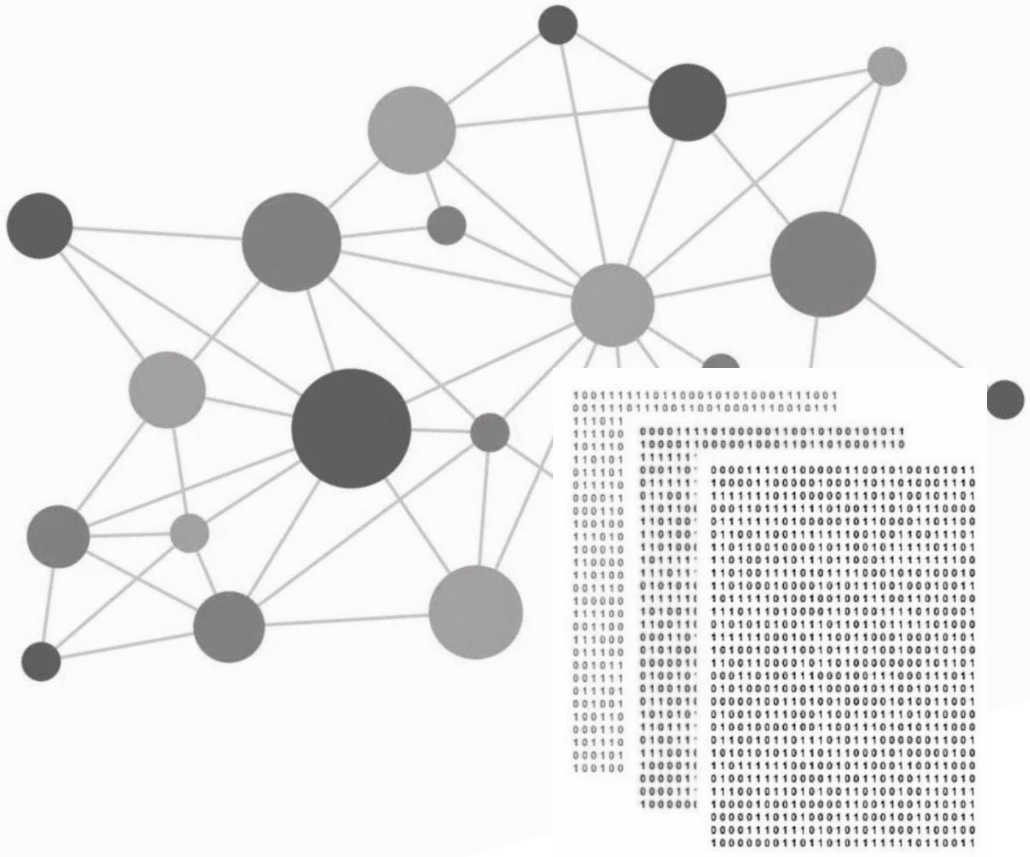
Link Prediction Under Non-targeted Attacks: Do Soft Labels Always Help?

Adel Memariani, Michael Röder, Arnab Sharma, Caglar Demir, Axel-Cyrille Ngonga Ngomo

Data Science Group (DICE), Heinz Nixdorf Institute, Paderborn University

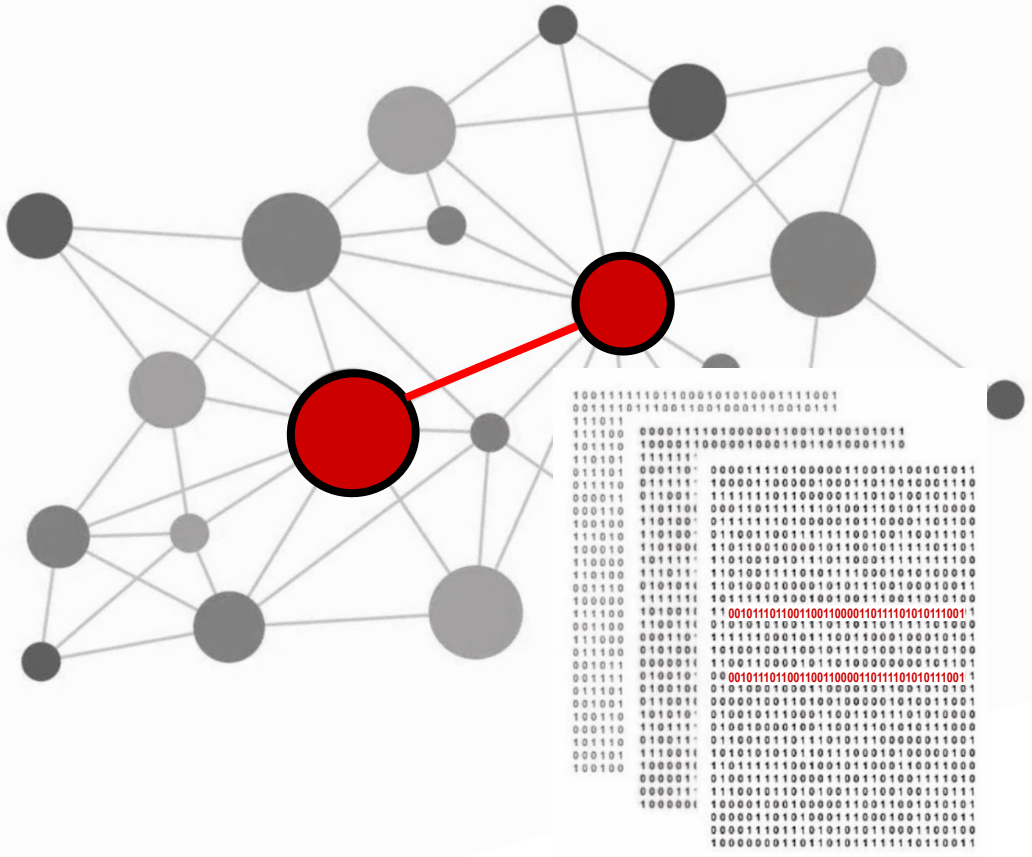
03.11.2025

Motivation



Knowledge graphs and their embeddings

Motivation



Mitigating the effect of wrong information in Knowledge Graphs

Motivation

- **Real-world Knowledge Graphs (KGs) may contain wrong information due to:**
  - Human curation mistakes
  - Automated construction
  - Data poisoning attacks

Motivation

- **Real-world Knowledge Graphs (KGs) may contain wrong information due to:**
  - Human curation mistakes
  - Automated construction
  - Data poisoning attacks

- **Fact checking is costly because:**
  - Rely on trusted reference knowledge
  - Evidence retrieval in large KGs is computationally expensive

Motivation

- **Real-world Knowledge Graphs (KGs) may contain wrong information due to:**
  - Human curation mistakes
  - Automated construction
  - Data poisoning attacks

- **Fact checking is costly because:**
  - Rely on trusted reference knowledge
  - Evidence retrieval in large KGs is computationally expensive

- **Unreliable facts can distort knowledge graph embeddings**
  - Compromise the trustworthiness of downstream!

How do data-poisoning attacks work?

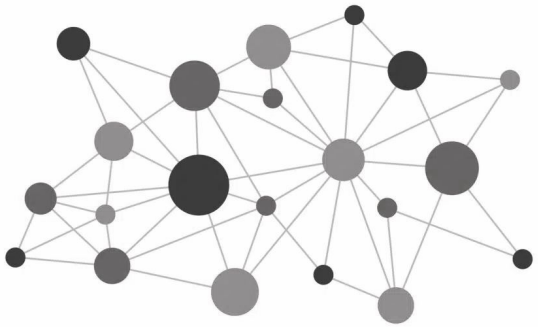
- **Targeted**: Add or remove triples to change **specific** inferences
  - Facts and the AI model are assumed to be known in advance!
  - Less practical

How do data-poisoning attacks work?

- **Targeted**: Add or remove triples to change **specific** inferences
  - Facts and the AI model are assumed to be known in advance!
  - Less practical

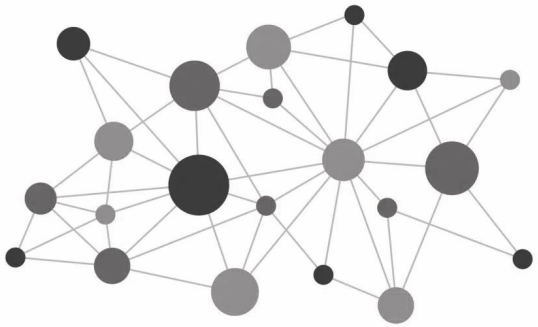
- **Non-targeted**: Add noise/remove facts to lower **overall** performance
  - Our focus in this research

How do data-poisoning attacks work?

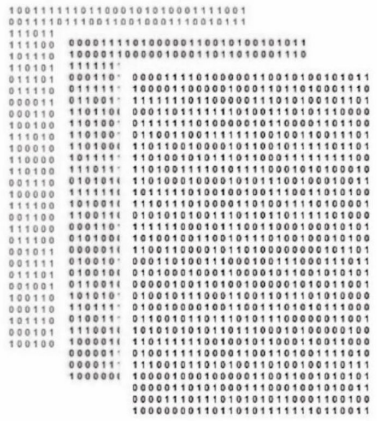


Knowledge Graph

How do data-poisoning attacks work?

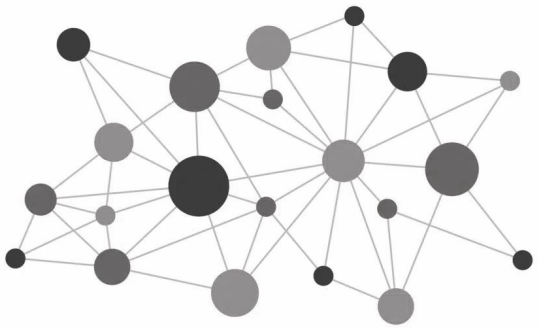


Knowledge Graph

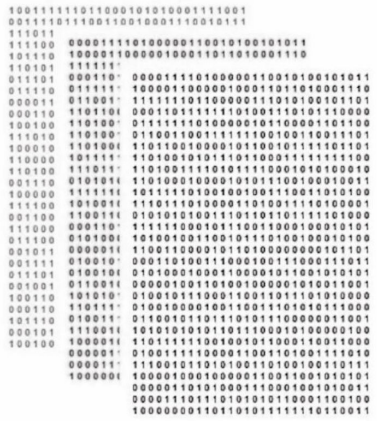


Embedding Vectors

How do data-poisoning attacks work?



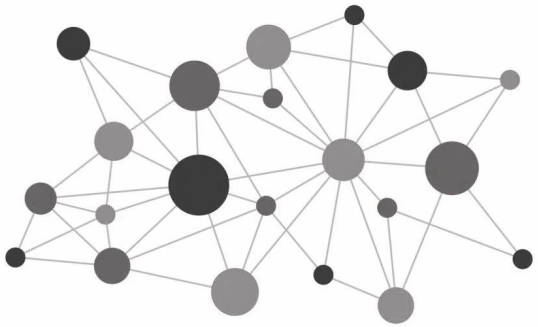
Knowledge Graph



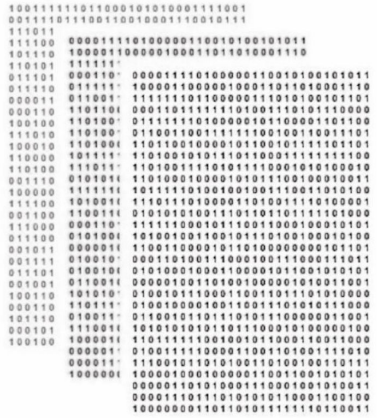
Embedding Vectors

Link Prediction

How do data-poisoning attacks work?



Knowledge Graph

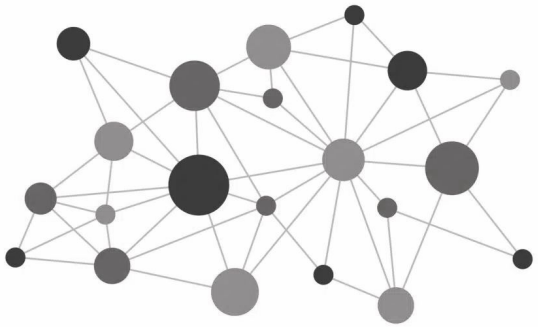


Embedding Vectors

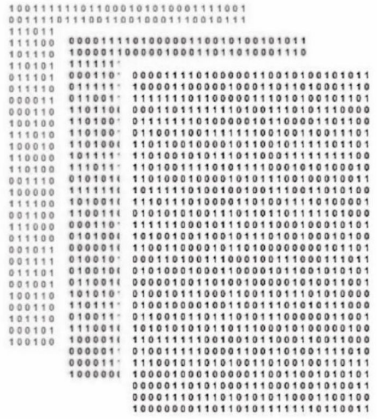
Link Prediction

*(Jupiter→surfaceType→**?**)*

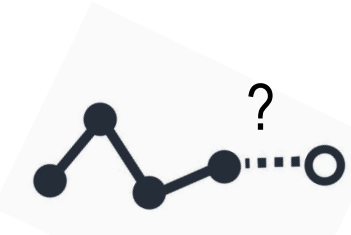
How do data-poisoning attacks work?



Knowledge Graph



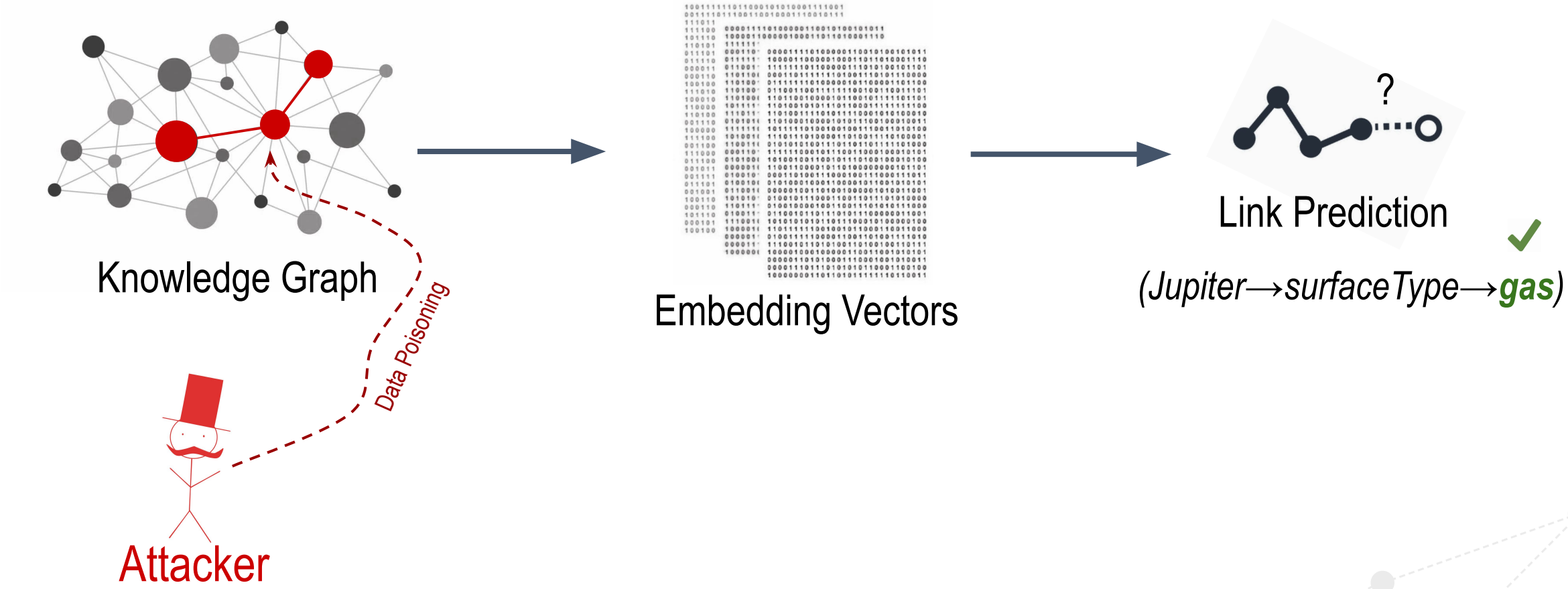
Embedding Vectors



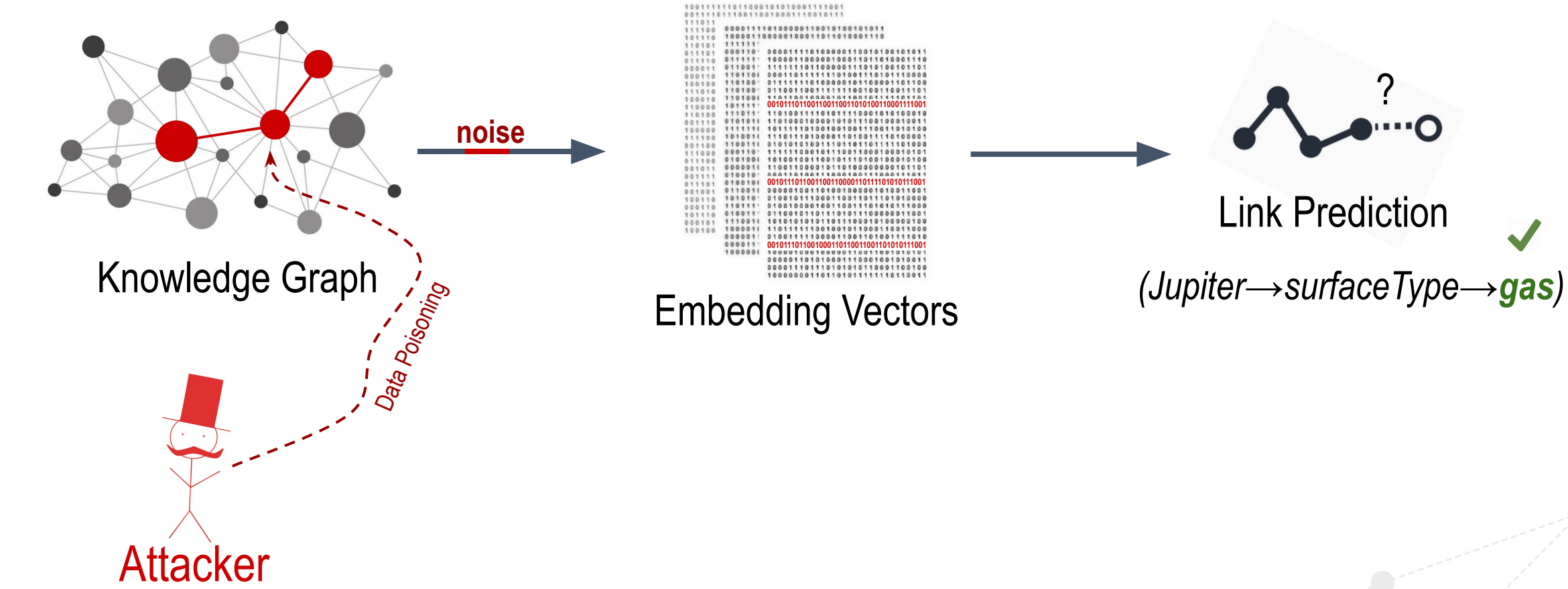
Link Prediction

*(Jupiter→surfaceType→**gas**)*

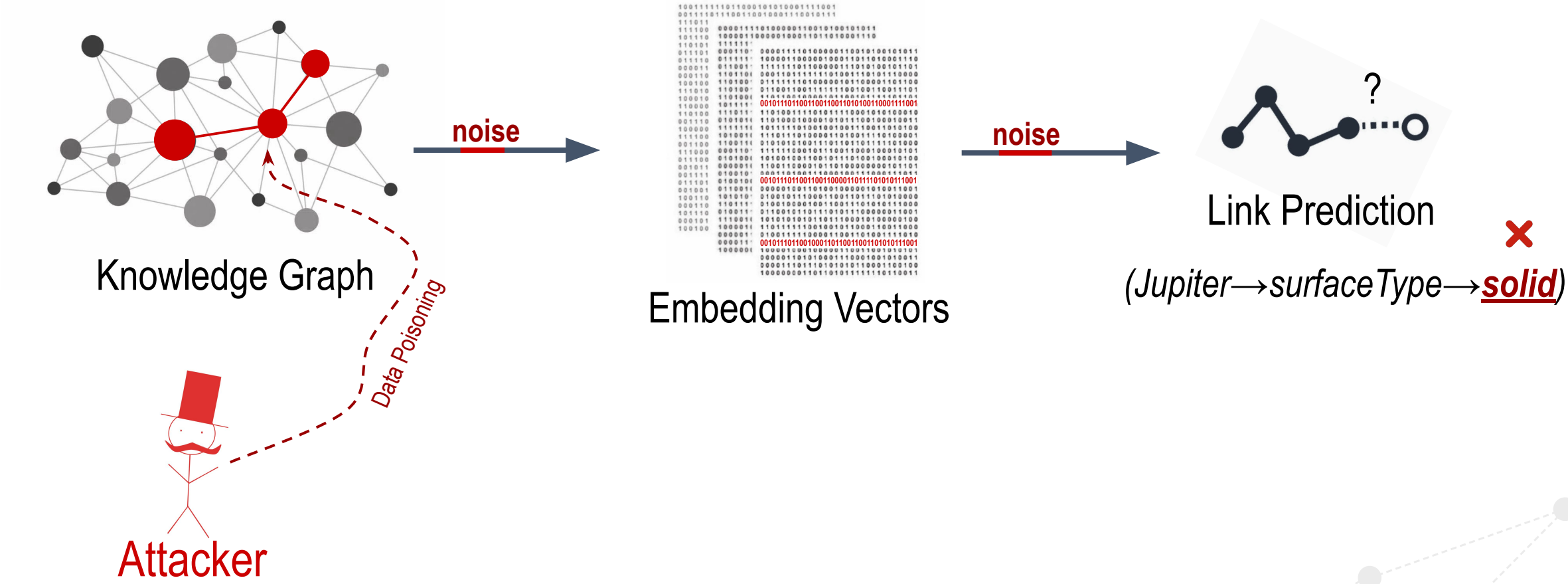
How do data-poisoning attacks work?

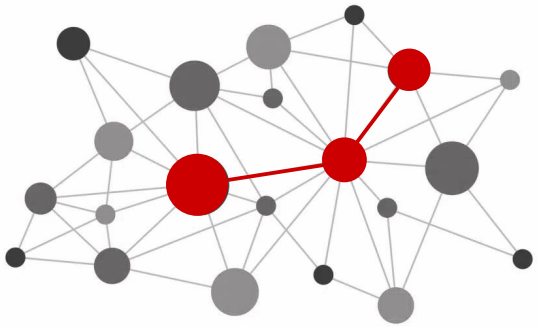


How do data-poisoning attacks work?



How do data-poisoning attacks work?





Knowledge Graph

noise

Embedding Vectors

noise

Link Prediction

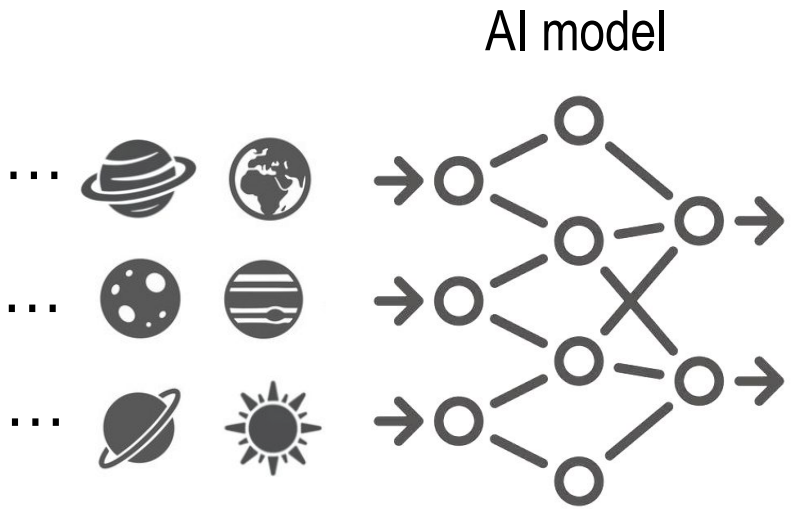
(*Jupiter*→*surfaceType*→***solid***)

Goal

Robust Knowledge Graph Embedding
Reliable even with noisy data

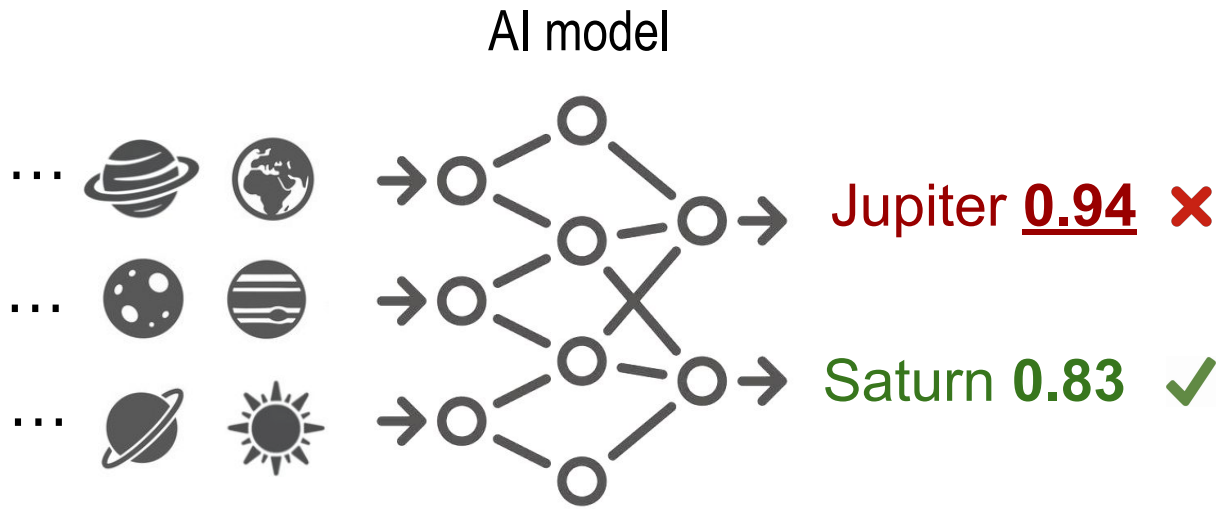
Model confidence

Model confidence



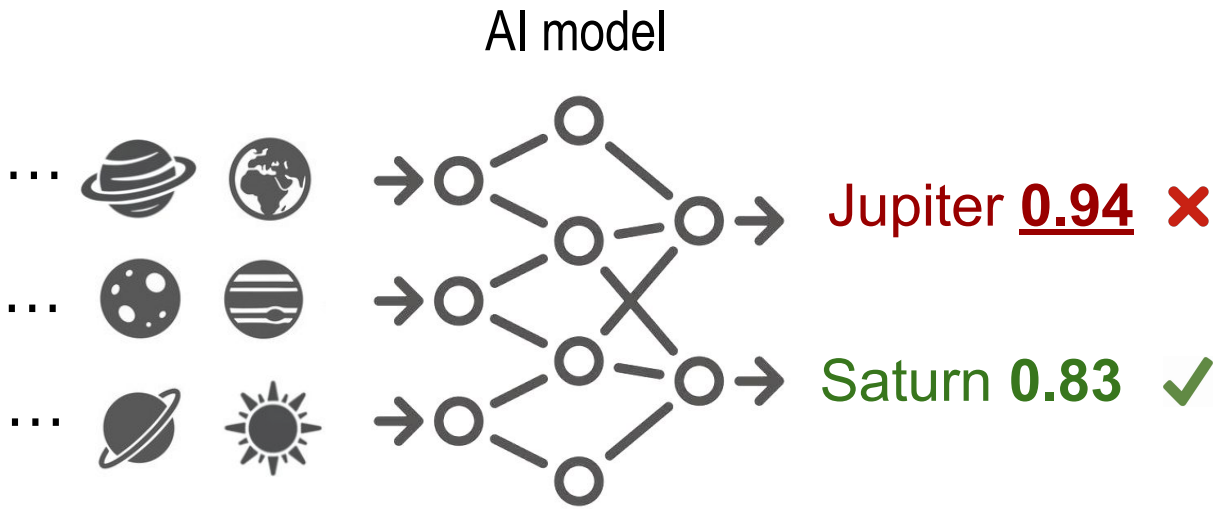
(Titan, orbits, ?)

Model confidence



(Titan, orbits, Jupiter)

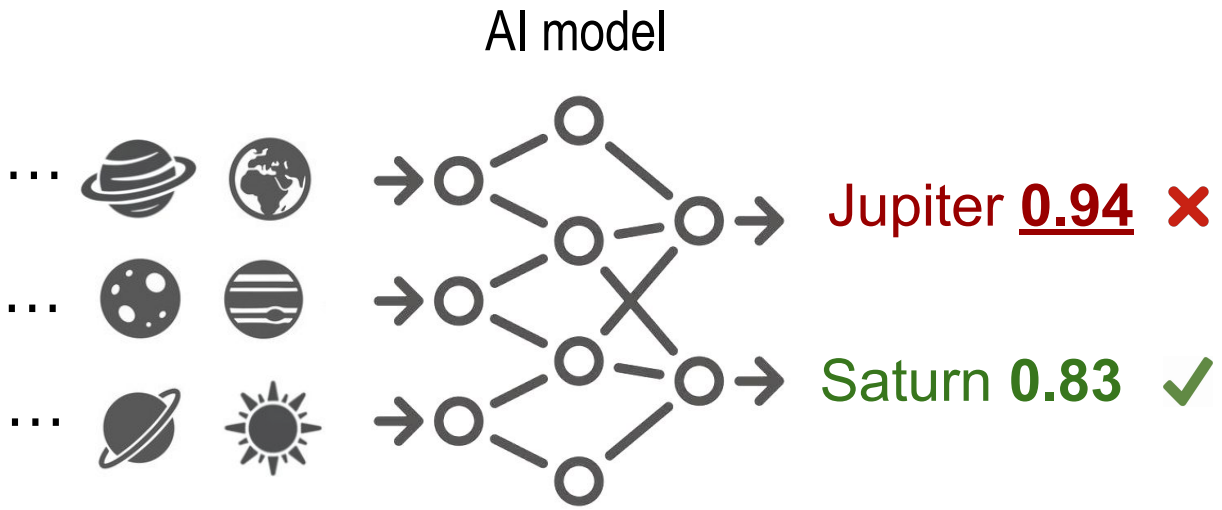
Model confidence



(Titan, orbits, Jupiter)

A highly *confident* wrong prediction 🤔

Model confidence



(Titan, orbits, Jupiter)

A highly *confident* wrong prediction

➡ Add a bit of uncertainty during training

The Challenge: Noisy training triples

The Challenge: **Noisy** training triples

Our goal: Avoid over-confidence

The Challenge: Noisy training triples

Our goal: Avoid over-confidence

Our approach: Use soft labels during training

Soft labels

Binary Class Entropy (BCE) Loss

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \left[y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right) \right]$$

Binary Class Entropy (BCE) Loss

Predicted

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \left[y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right) \right]$$

Binary Class Entropy (BCE) Loss

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \left[y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right) \right]$$

Predicted

Label

Binary: 1.0, 0.0, 1.0, 0.0 …

Label Smoothing (LS)

➔ **Is a training-time regularizer:**

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \left[y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right) \right]$$

Predicted

Label

Smoothed: 0.95, 0.05, 0.95, 0.05 …

Label Smoothing (LS)

➔ **Is a training-time regularizer:**

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \left[y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right) \right]$$

Predicted

Label

Smoothed: 0.95, 0.05, 0.95, 0.05 …

Tail prediction: $p_s(t \mid h, r) = \begin{cases} 1 - \alpha & \text{if } t' = t \\ \frac{\alpha}{(|\mathcal{E}| - 1)} & \text{otherwise} \end{cases}$

Label Smoothing (LS)

➔ **Is a training-time regularizer:**

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{|\mathcal{E}|} \sum_{i=1}^{|\mathcal{E}|} \left[y_i \log\left(\hat{y}_i\right) + (1 - y_i) \log\left(1 - \hat{y}_i\right) \right]$$

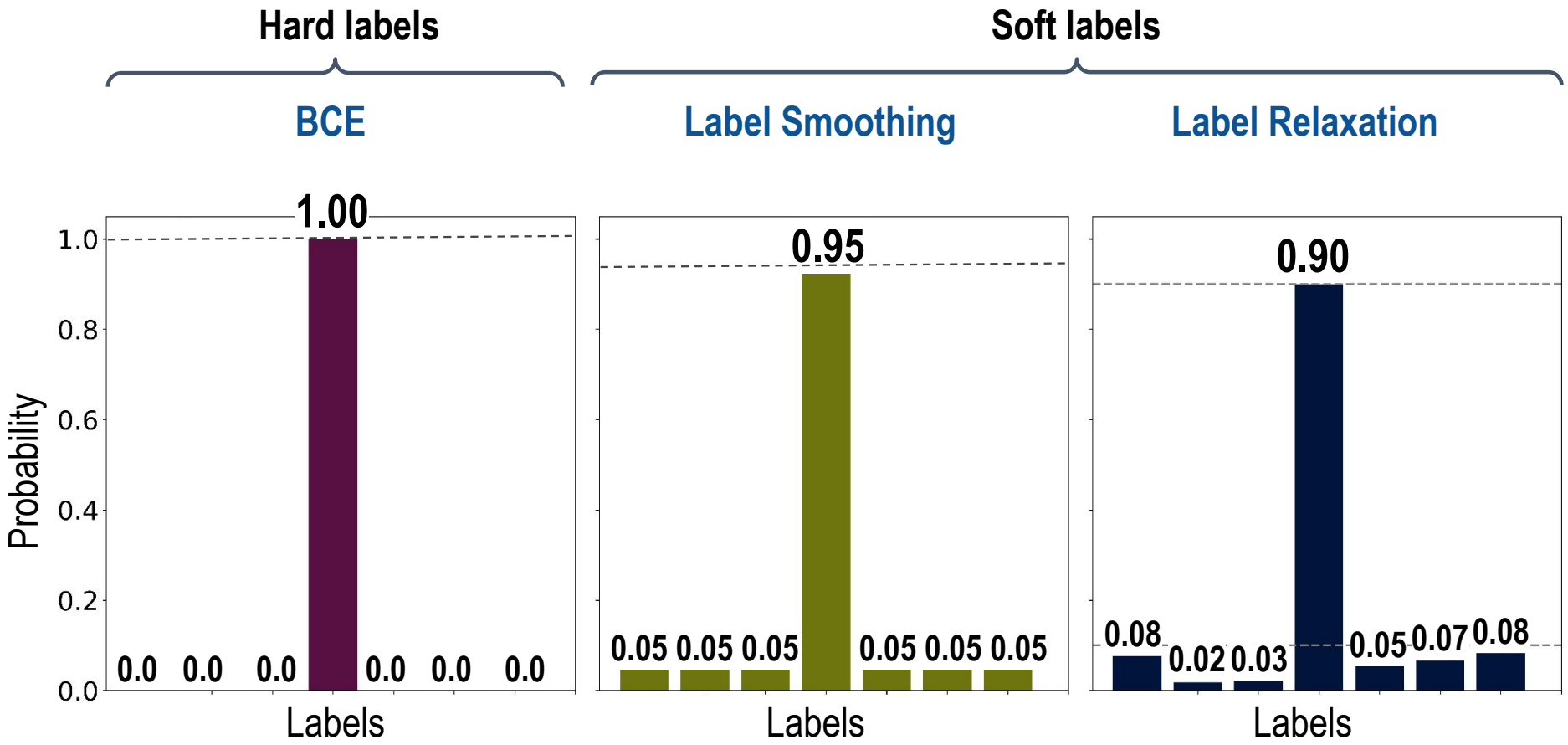
Predicted

Label

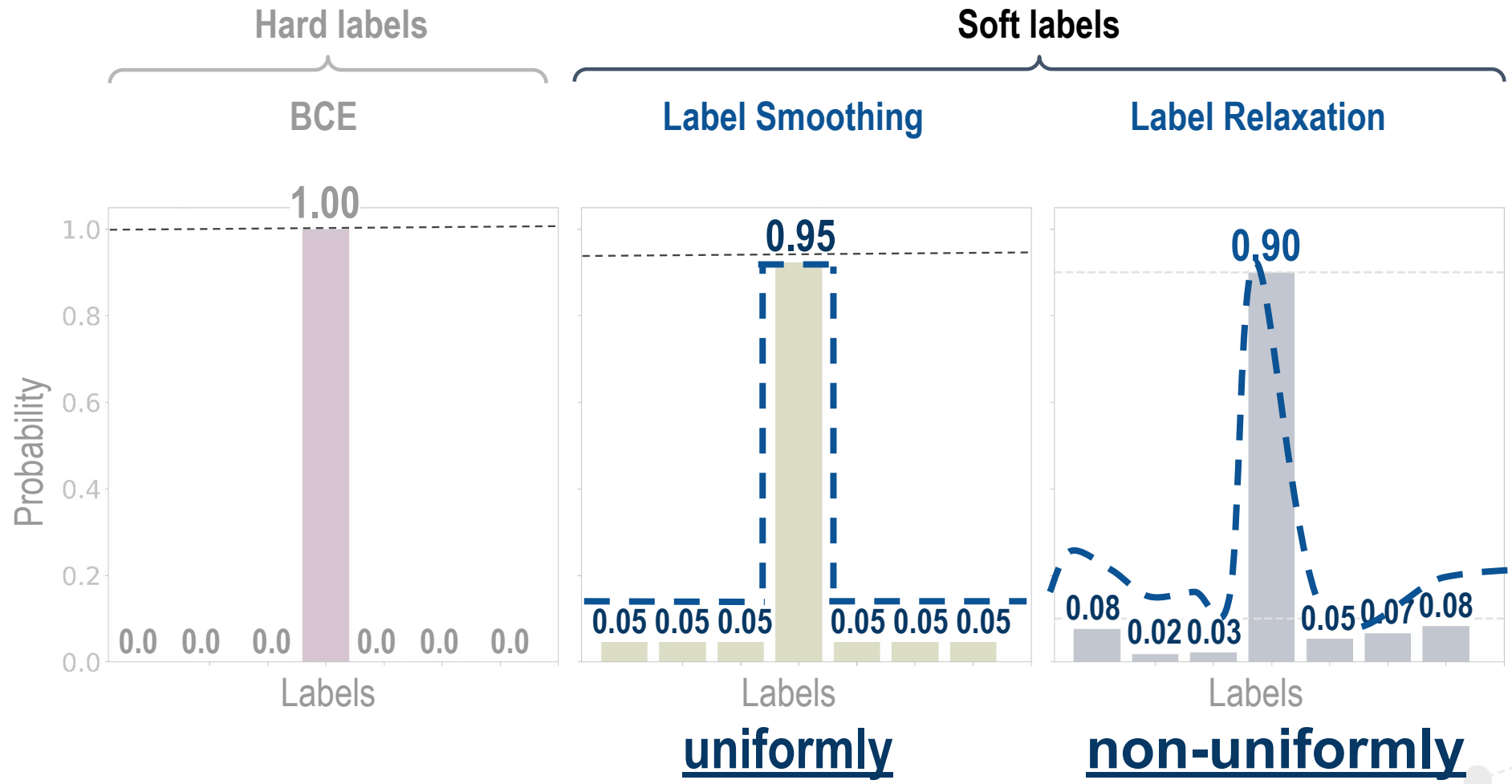
Smoothed: 0.95, 0.05, 0.95, 0.05 …

Tail prediction: $p_s(t \mid h, r) = \begin{cases} 1 - \alpha & \text{if } t' = t \\ \frac{\alpha}{(|\mathcal{E}| - 1)} & \text{otherwise} \end{cases}$

BCE and soft labels



BCE and soft labels



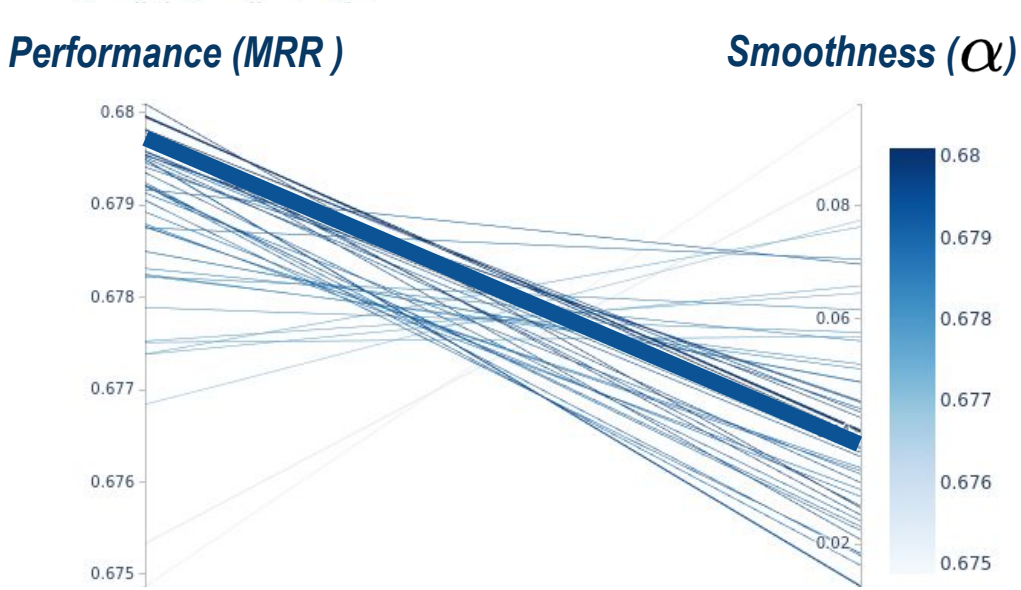
So far,
α as a smoothing factor

How to choose the optimal α value?

$$\alpha \in (0, 1]$$

LS with Bayesian Optimization

- Smoothing ratio may depend on model architecture and training dataset
- We performed **50 runs** per model–dataset combination
  - Hyper-parameter optimization!



Adaptive Label Smoothing/Relaxation

- Smoothing ratio adjusts **dynamically** during the training, instead of relying on a fixed value.

α

Adaptive Label Smoothing/Relaxation

- Smoothing ratio adjusts **dynamically** during the training, instead of relying on a fixed ratio.
- Tracks the change in Kullback-Leibler (KL) divergence between predicted and target label **distributions**.

α

Adaptive Label Smoothing/Relaxation

- Smoothing ratio adjusts **dynamically** during the training, instead of relying on a fixed ratio.
- Tracks the change in Kullback-Leibler (KL) divergence between predicted and target label **distributions**.

$$\alpha_{t+1} = \begin{cases} \min(\alpha_t + \Delta\alpha, \alpha_{\max}), & \text{if } KL_{t+1} > KL_t \\ \max(\alpha_t - \Delta\alpha, \alpha_{\min}), & \text{otherwise} \end{cases}$$

KL: Kullback–Leibler divergence
$\Delta\alpha$: the step size (hyper-parameter)
$\alpha_{min}, \alpha_{max}$: hyper-parameters
t: training step at time t

Adaptive Label Smoothing/Relaxation

$$KL_{t+1} - KL_t \uparrow$$

Model is presumed to be becoming overconfident

- Increase α
  - Add more uncertainty into the training

Adaptive Label Smoothing/Relaxation

$$KL_{t+1} - KL_t$$

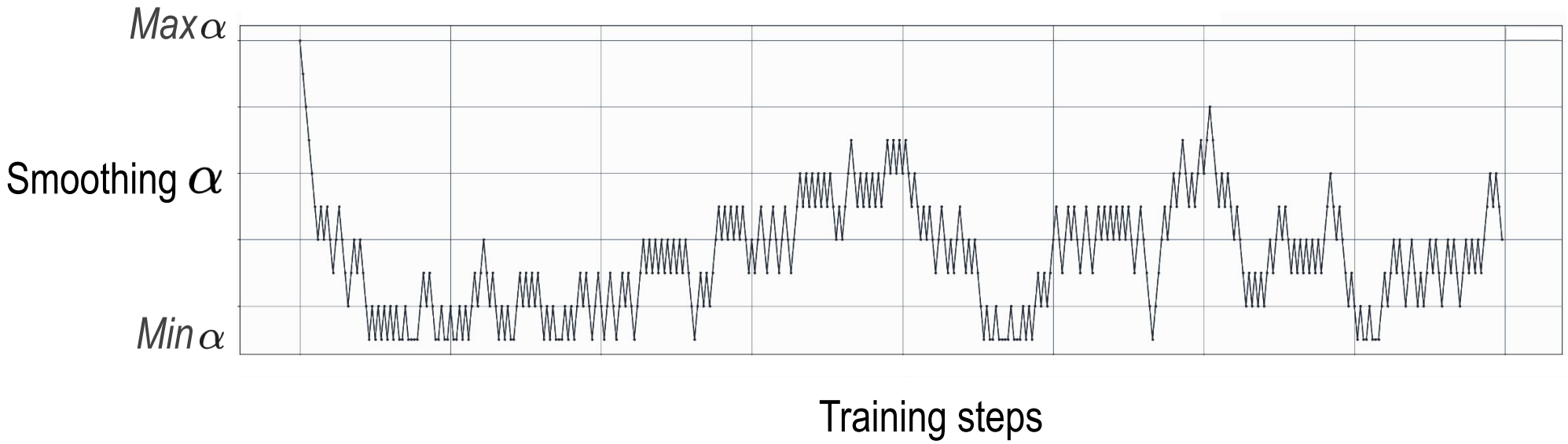
↑ Model is presumed to be becoming overconfident

- Increase α
  - Add more uncertainty into the training

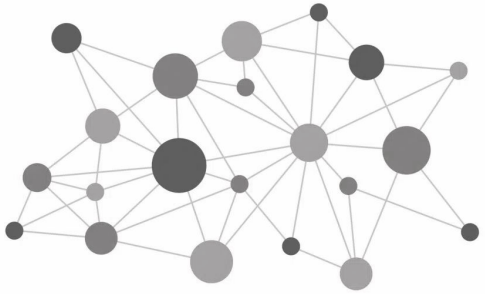
↓ Model is likely generalizing better

- Reduce α
  - Avoid over-regularization
  - Preserve current model's confidence

Adaptive Label Smoothing/Relaxation

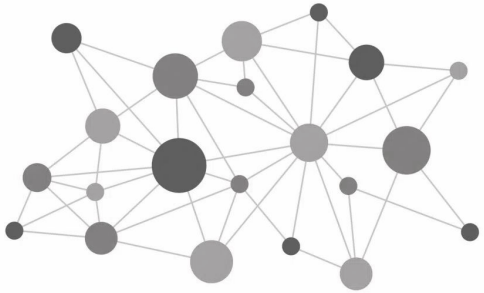


Experimental setup

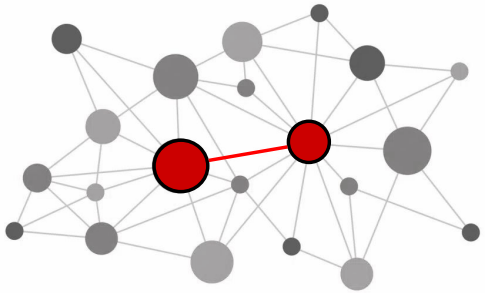


Noise-free

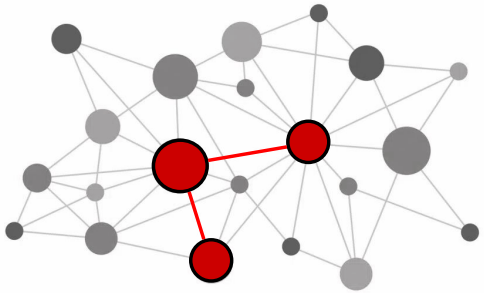
Experimental setup



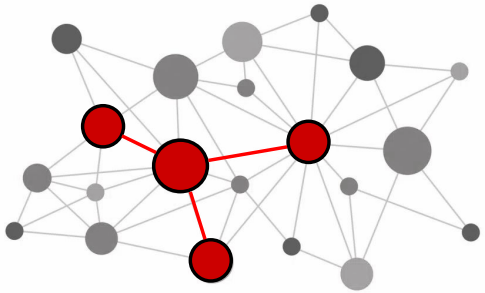
Noise-free



Low noise

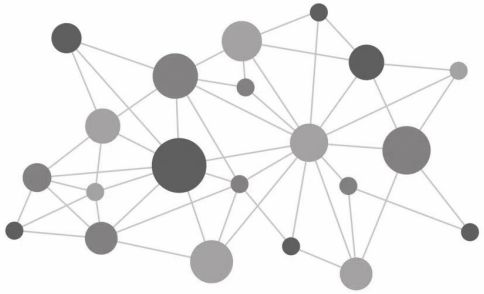


Moderate noise

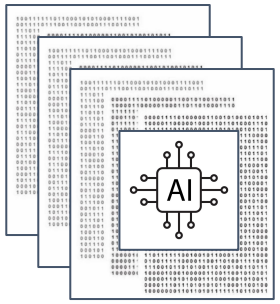


High noise

Experimental setup

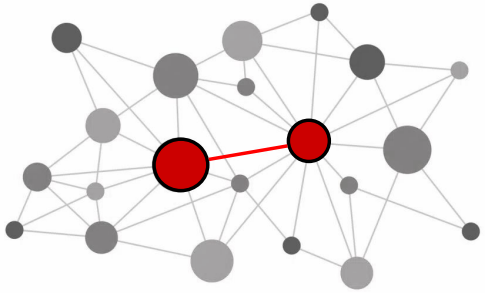


Noise-free

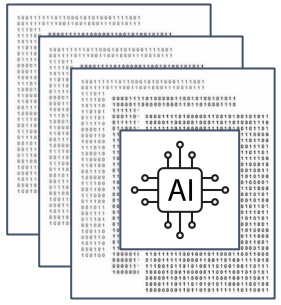


KGE Models

Hard vs. Soft labels?

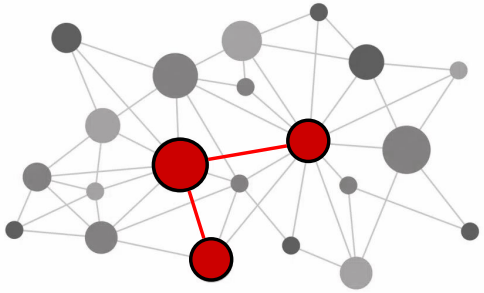


Low noise

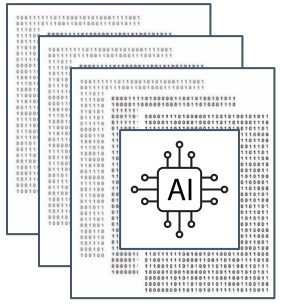


KGE Models

Hard vs. Soft labels?

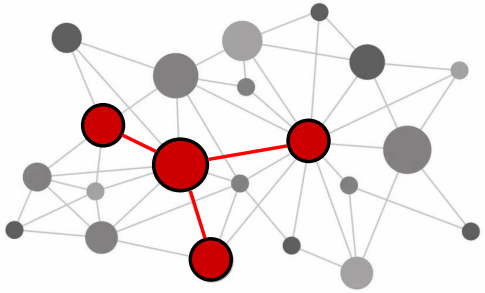


Moderate noise

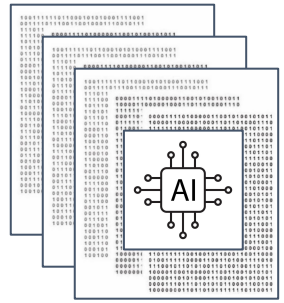


KGE Models

Hard vs. Soft labels?



High noise



KGE Models

Hard vs. Soft labels?

Experimental setup

Models: *DistMult, ComplEx, QMult, RotatE, MuRE, Keci*

Experimental setup

Models: *DistMult, ComplEx, QMult, RotatE, MuRE, Keci*

Datasets: *UMLS, KINSHIP, WN18RR, NELL-995-h100, FB15K-237*

Experimental setup

Models: *DistMult, ComplEx, QMult, RotatE, MuRE, Keci*

Datasets: *UMLS, KINSHIP, WN18RR, NELL-995-h100, FB15K-237*

Random Noise ratios: *0.0 .. 0.64*

Experimental setup

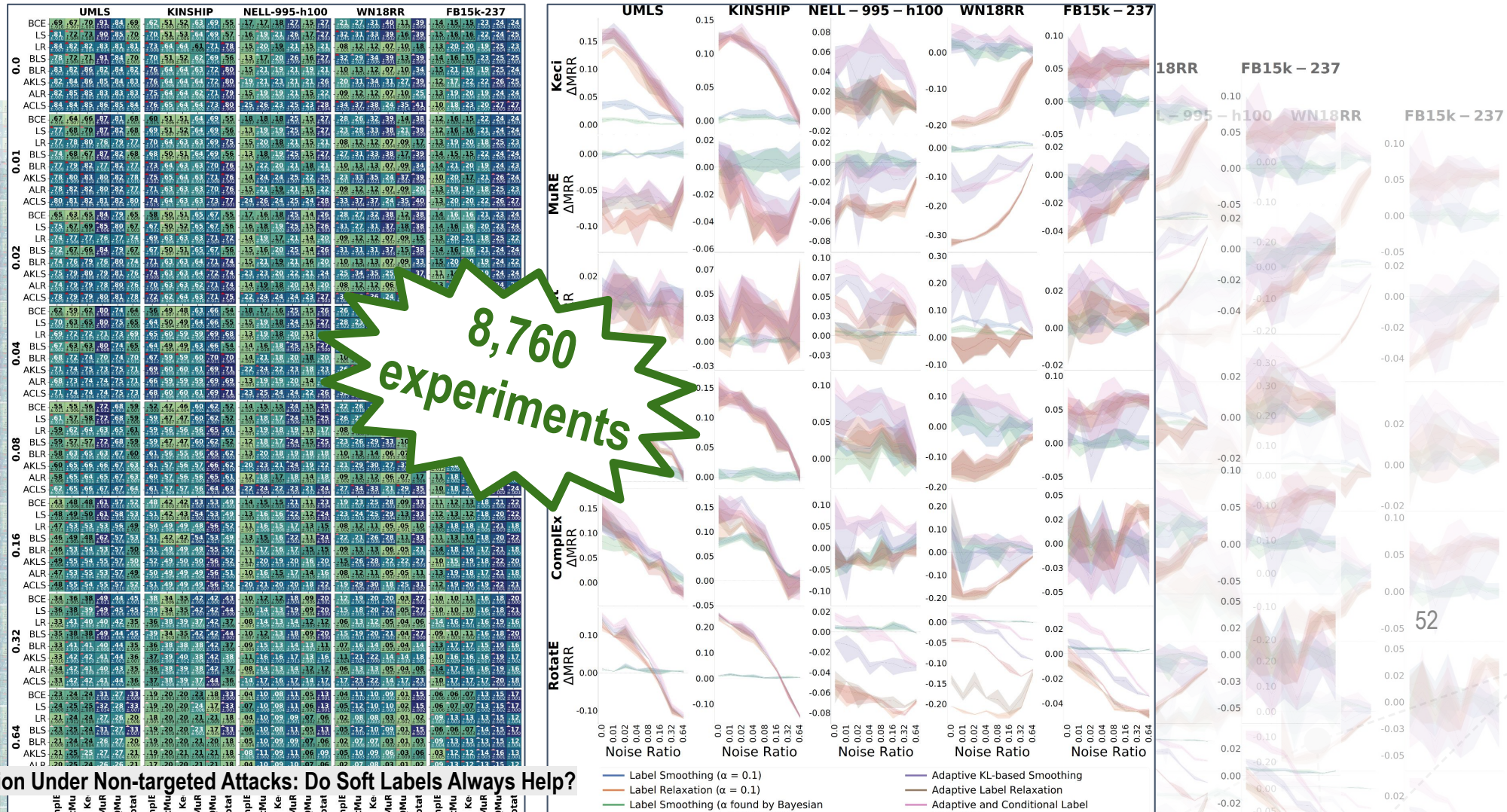
Models: *DistMult, ComplEx, QMult, RotatE, MuRE, Keci*

Datasets: *UMLS, KINSHIP, WN18RR, NELL-995-h100, FB15K-237*

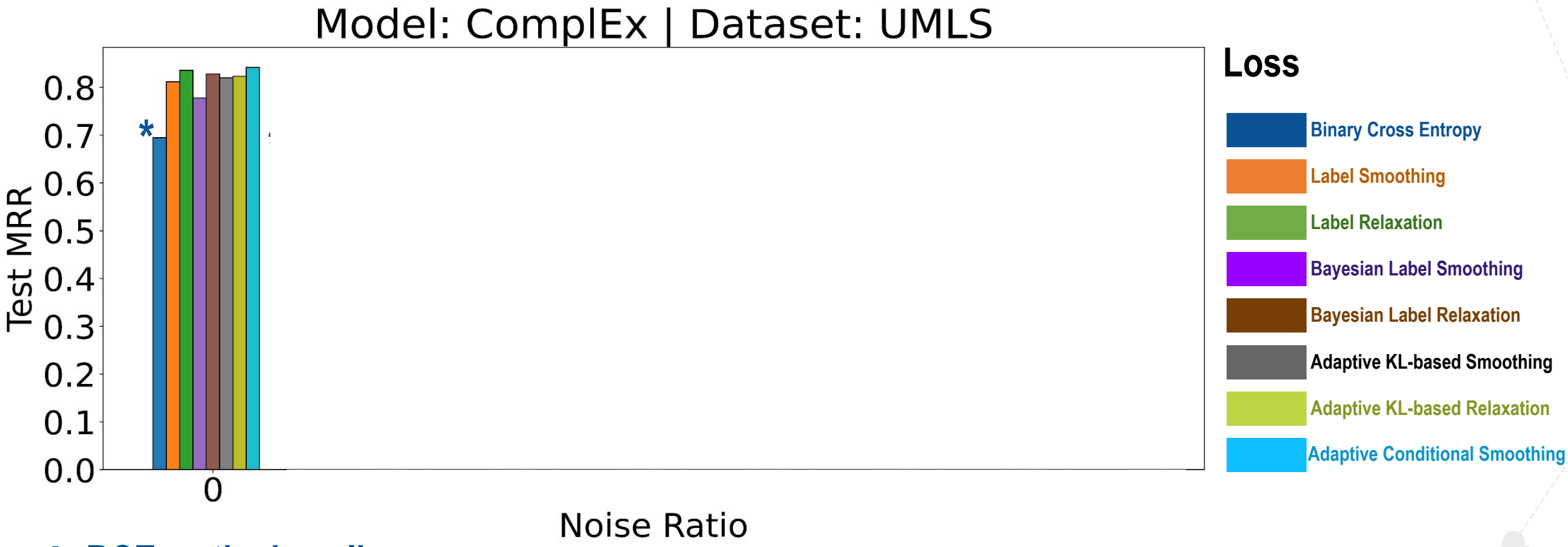
Random Noise ratios: *0.0 .. 0.64*

Loss functions:
- **Hard:** *BCE*
- **Soft:** *Smoothing, Relaxation, Bayesian/Adaptive versions*

Results

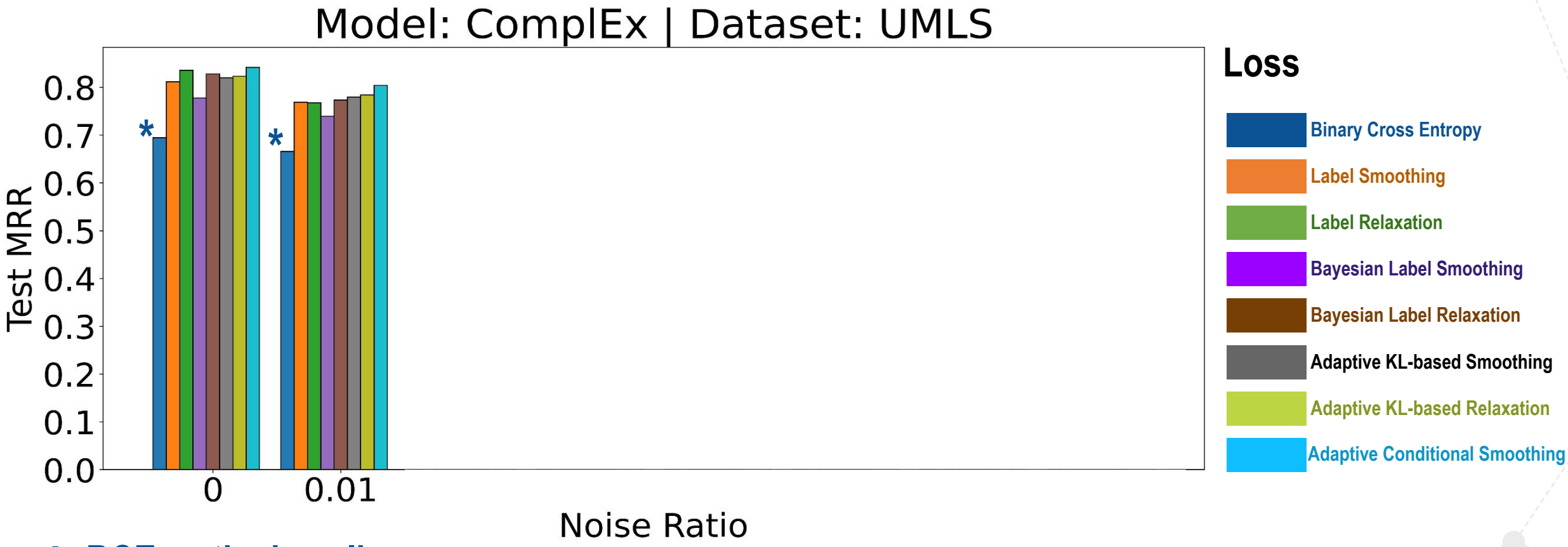

UMLS
KINSHIP
NELL-995-h100
WN18RR
FB15k-237
BCE
LS
LR
BLS
BLR
AKLS
ALR
ACLS
0.0
0.01
0.02
0.04
0.08
0.16
0.32
0.64
8,760
experiments
UMLS
KINSHIP
NELL – 995 – h100
WN18RR
FB15k – 237
Keci ΔMRR
MuRE ΔMRR
ComplEx ΔMRR
RotatE ΔMRR
Noise Ratio
Label Smoothing (α = 0.1)
Label Relaxation (α = 0.1)
Label Smoothing (α found by Bayesian
Adaptive KL-based Smoothing
Adaptive Label Relaxation
Adaptive and Conditional Label

Example Result



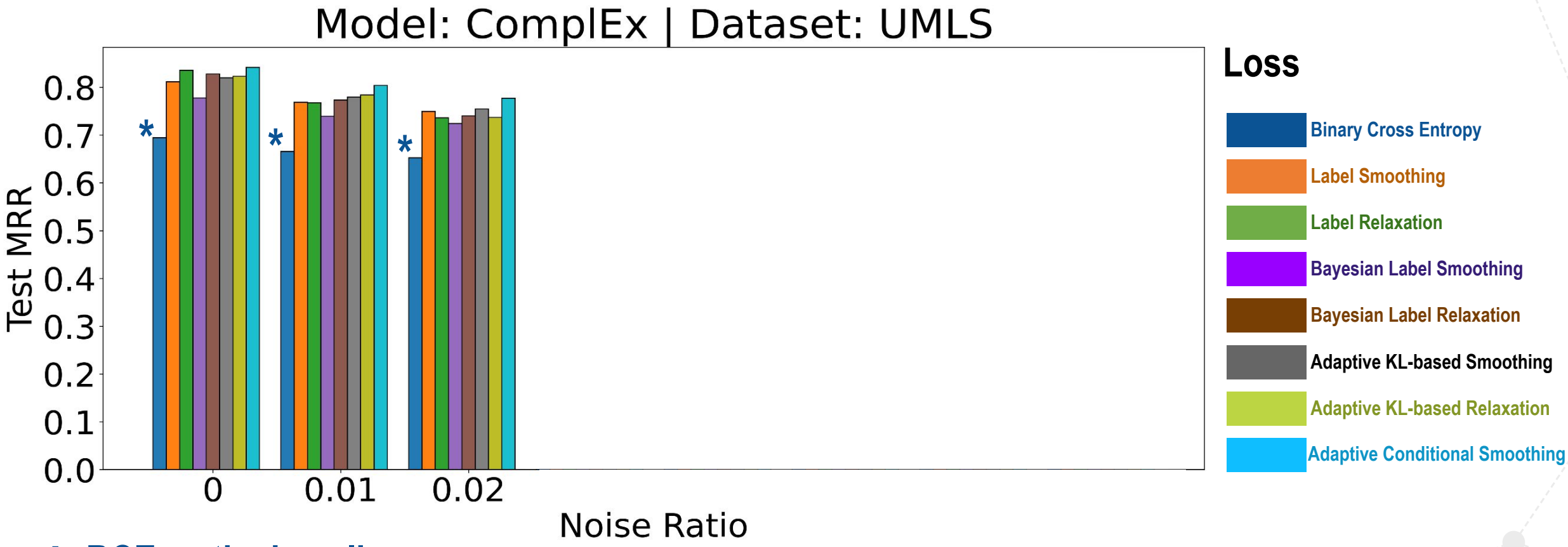
* BCE as the baseline

Example Result



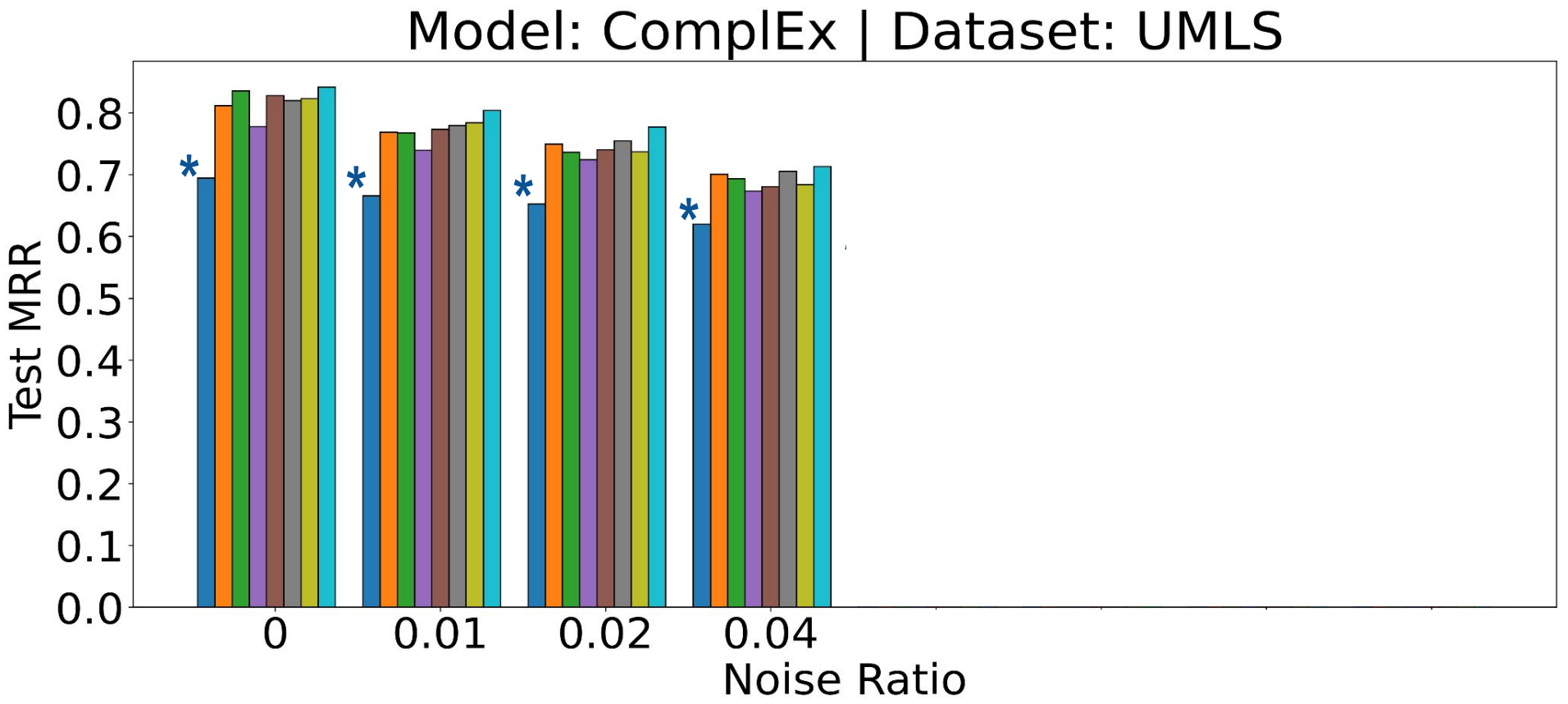
∗ BCE as the baseline

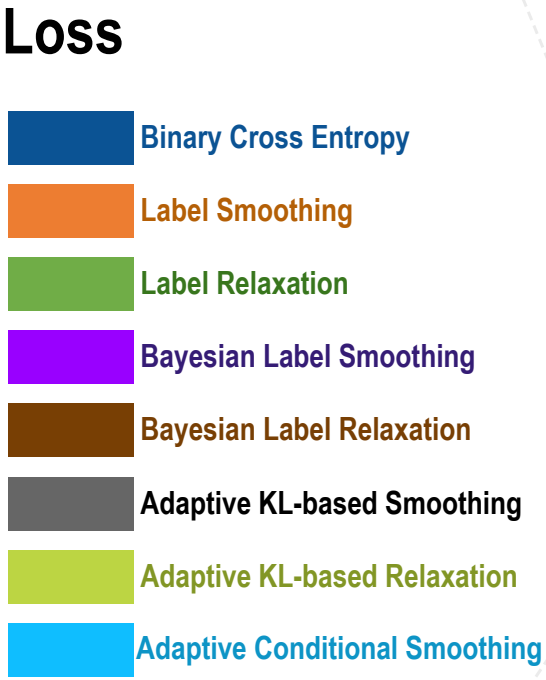
Example Result



∗ BCE as the baseline

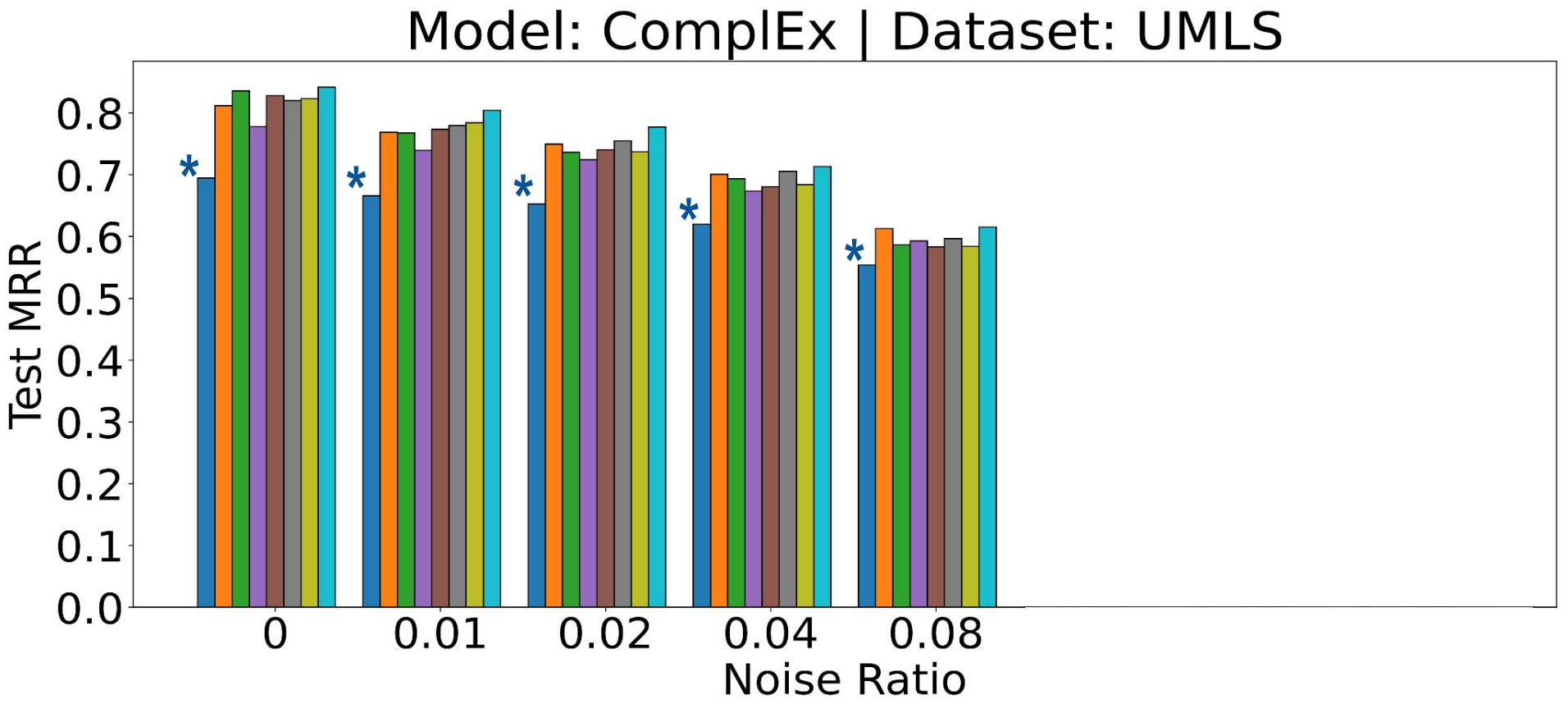
Example Result

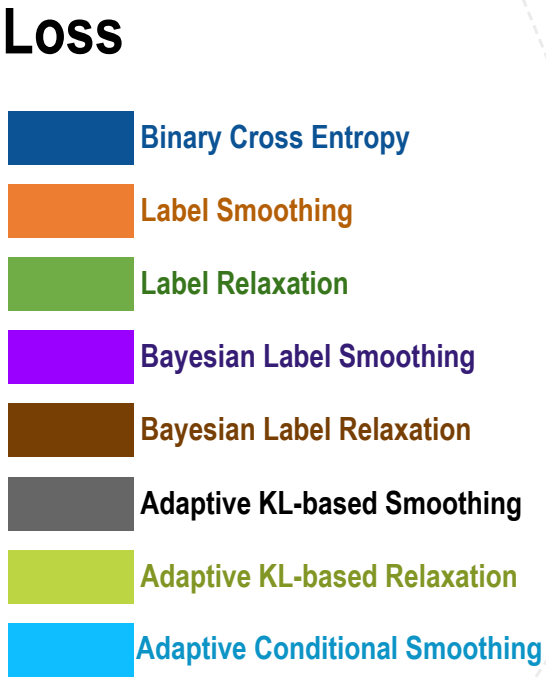




* BCE as the baseline

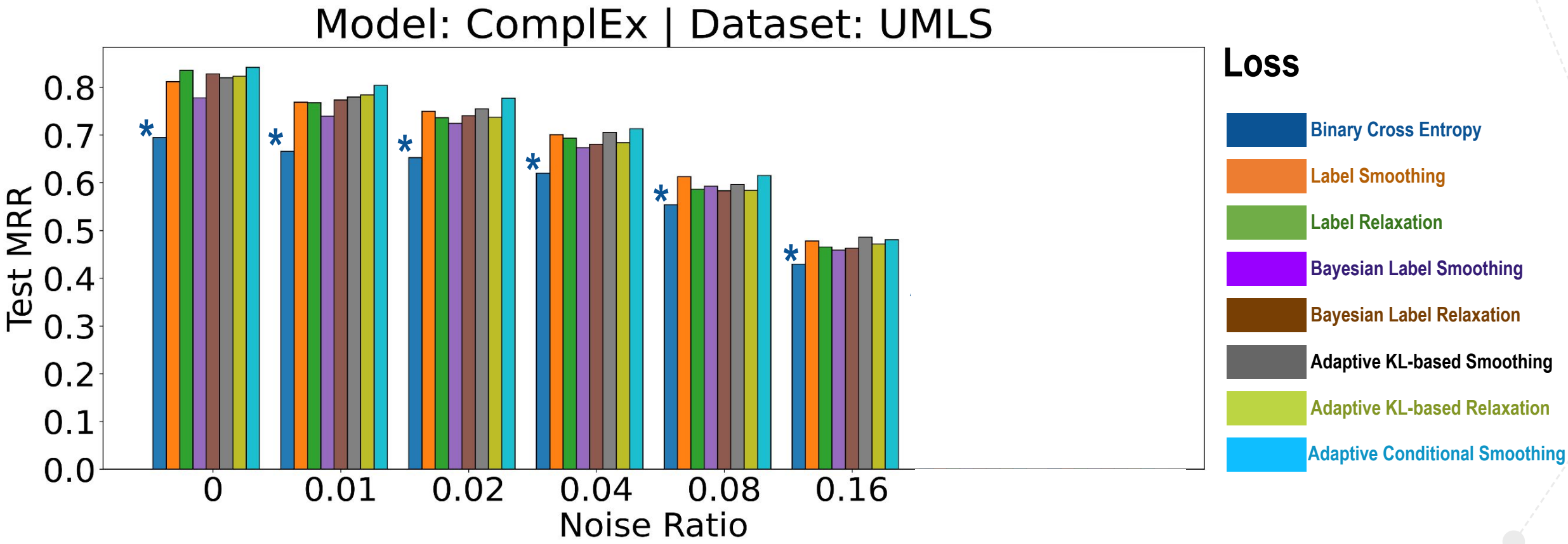
Example Result





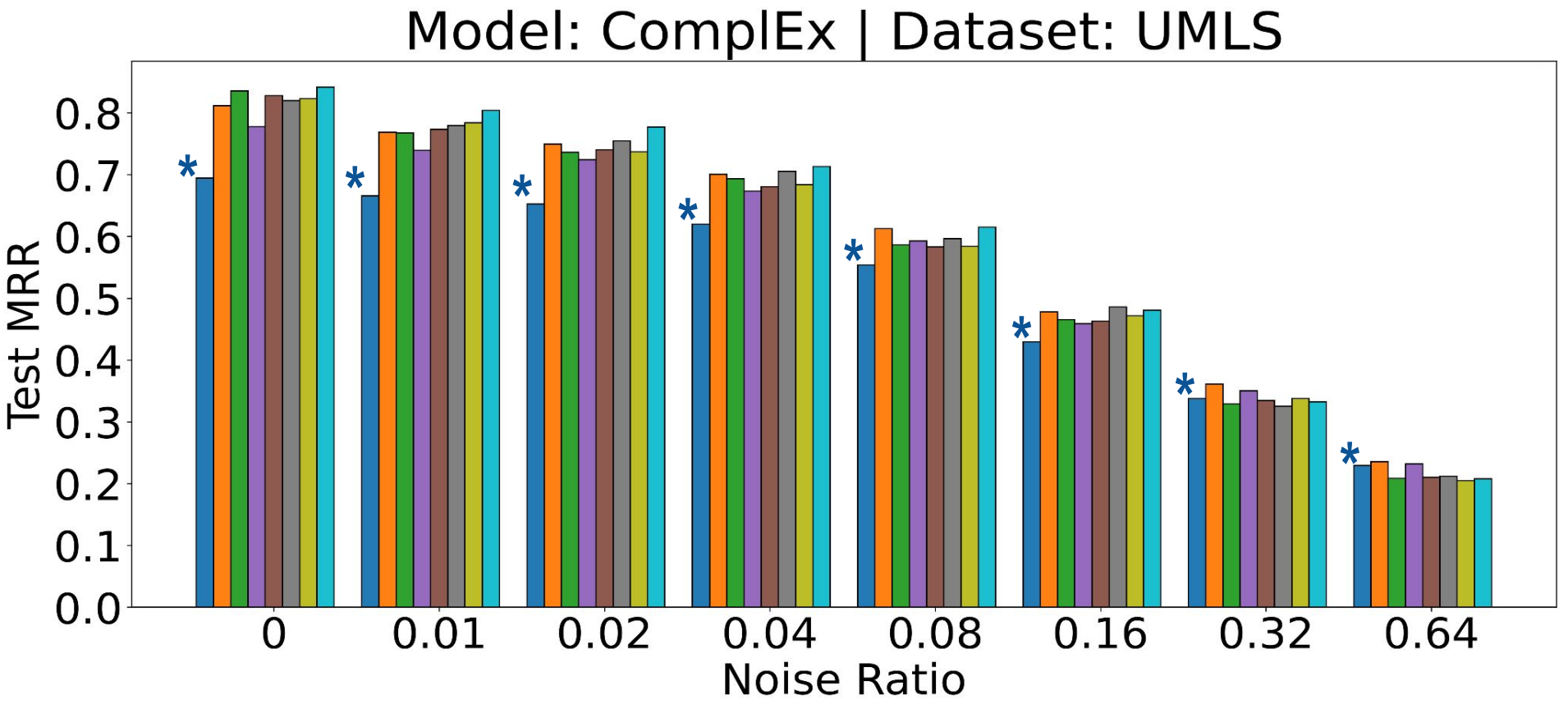
∗ BCE as the baseline

Example Result



* BCE as the baseline

Example Result

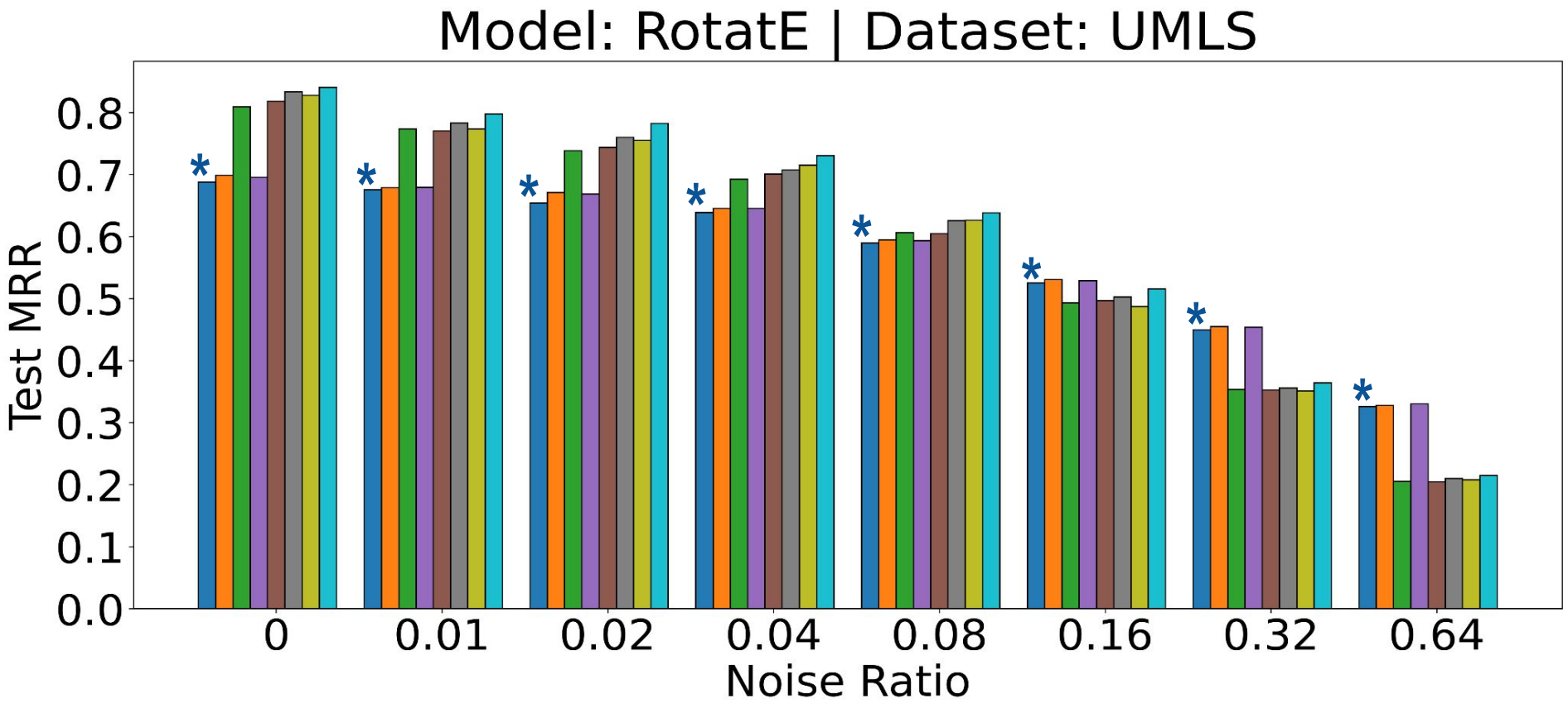


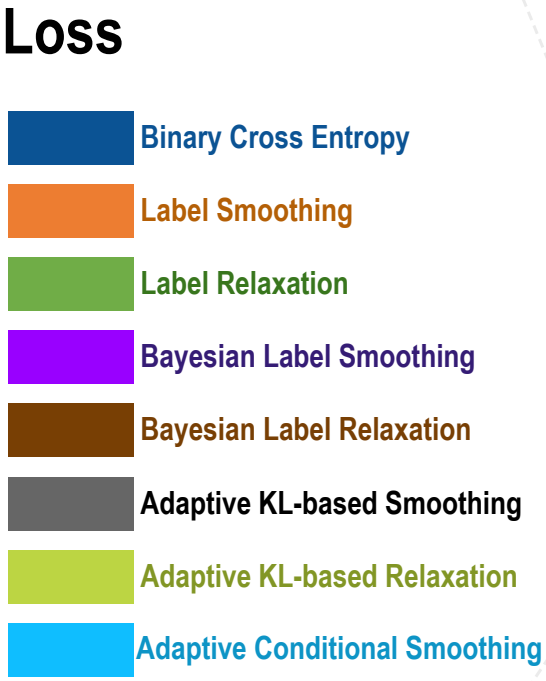
Loss

- Binary Cross Entropy
- Label Smoothing
- Label Relaxation
- Bayesian Label Smoothing
- Bayesian Label Relaxation
- Adaptive KL-based Smoothing
- Adaptive KL-based Relaxation
- Adaptive Conditional Smoothing

∗ BCE as the baseline

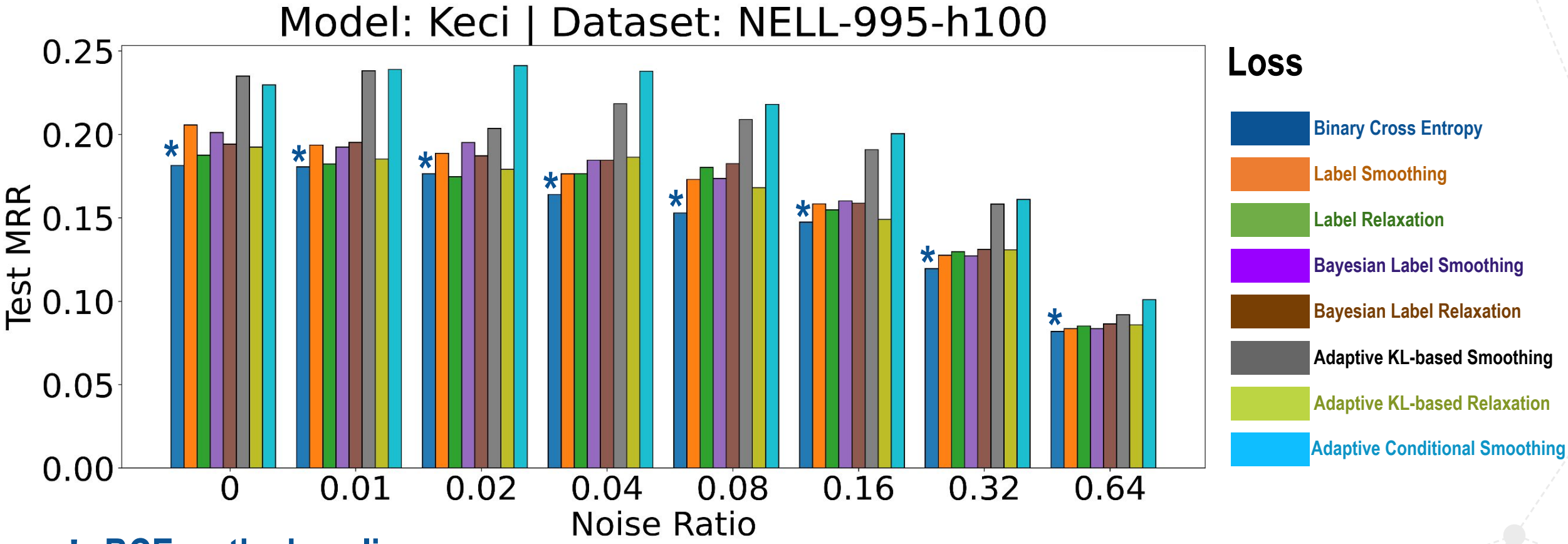
Example Result





∗ BCE as the baseline

Example Result



∗ BCE as the baseline

Statistical Testing

- Beyond improvements, we assessed their **significance**
  - Soft-label loss function that are **better** than BCE:

Statistical Testing

- Beyond improvements, we assessed their **significance**
  - Soft-label loss function that are **better** than BCE

- 8 noise ratios across 5 datasets, resulting in 48 groups
    - Significant improvements across **41 out of 48** groups

Conclusions

- **Soft labels** enable more robust link prediction under noisy data.
- Suitability of **KL divergence** for label smoothing.
- Bayesian optimization helps, but is expensive;
  - **Adaptive** smoothing is better with less **computational cost**.

Future works

- Focus on more structured inconsistencies:
  - Systematic errors in information extraction
  - Misalignments in ontological hierarchies
- Adaptive version of smoothing based on similarity among entities
- Hyper-parameter optimization based on other strategies

Thank you

Adel Memariani
adel.memariani@upb.de

Data Science Group (DICE)
Heinz Nixdorf Institute
Paderborn University

Link Prediction Under Non-targeted Attacks: Do Soft Labels Always Help?

- **Soft labels** enable more robust link prediction under noisy data
- Suitability of **KL divergence** for label smoothing
- Bayesian optimization helps, but is expensive
  - **Adaptive** smoothing is better with **less computational** cost

SAIL
SUSTAINABLE LIFE-CYCLE OF INTELLIGENT SOCIO-TECHNICAL SYSTEMS

Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen

enexa

Funded by the European Union

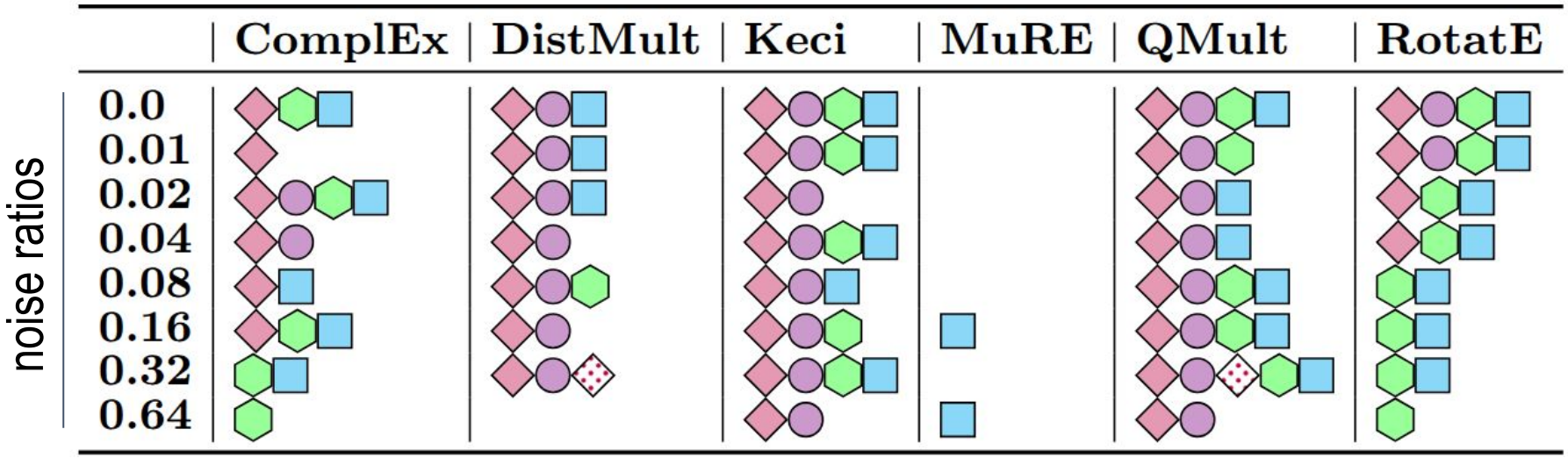
WHALE

This work has been supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL under the grant no NW21-059D, the project "WHALE" (LFN 1-04) funded under the Lamarr Fellow Network programme by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW), and the European Union's Horizon Europe research and innovation programme under grant agreement No 101070305.

Appendix

Statistical Testing

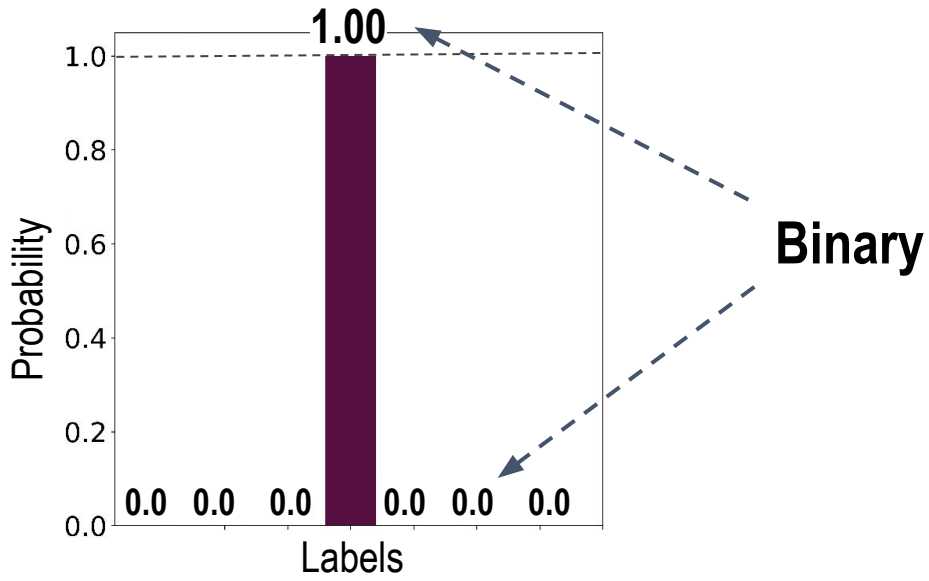
- **Significant improvements** in link prediction performance across **41 out of 48** groups



Wilcoxon signed-rank, $p < 0.05$, on MRR scores. Soft label-based loss functions that outperformed BCE and are statistically significant. Adaptive Conditional Smoothing AKLS Adaptive KL-based Smoothing LS Label Smoothing BLR Bayesian Label Relaxation BLS Bayesian Label Smoothing

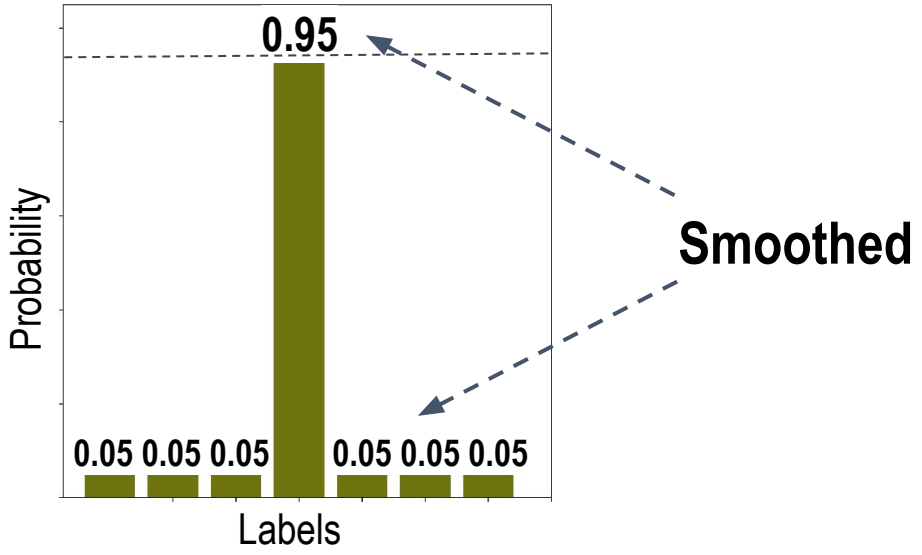
Binary Class Entropy (BCE) Loss

- Target as 100% true class, 0% everything else



Label Smoothing (LS)

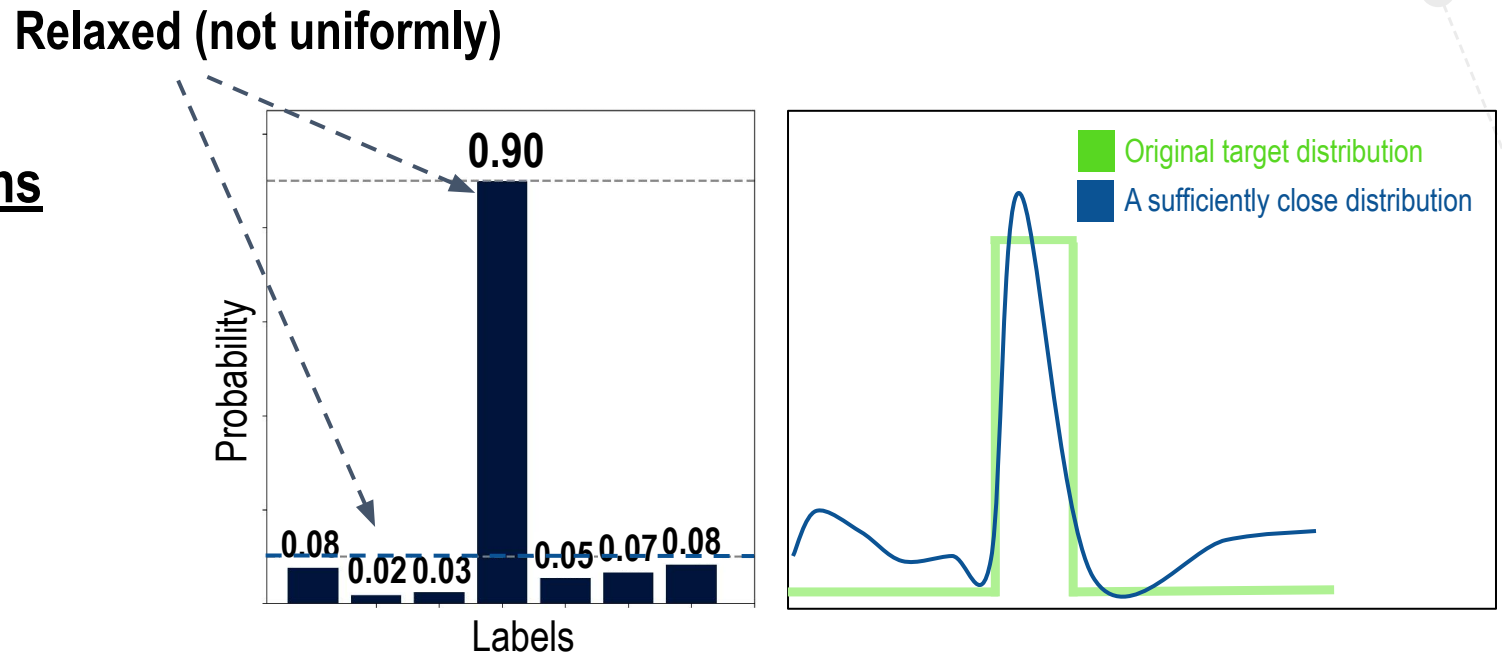
- Redistributes a small portion of the probability mass **uniformly** across all other labels.



$$p_s(t \mid h, r) = \begin{cases} 1-\alpha & \text{if } t' = t \\ \frac{\alpha}{(|\mathcal{E}|-1)} & \text{otherwise} \end{cases}$$

Label Relaxation (LR)

- Relax the target into a **set of distributions**
- A **set** of acceptable targets.
  - Instead of a precise probability
  - Labels vary within a range.



$$Q_\alpha^{(h,r,t)} = \{p' \in \mathcal{P}(Y) \mid p'(t \mid h, r) \geq 1 - \alpha\}$$

Credal set of distributions that are sufficiently close to the original target