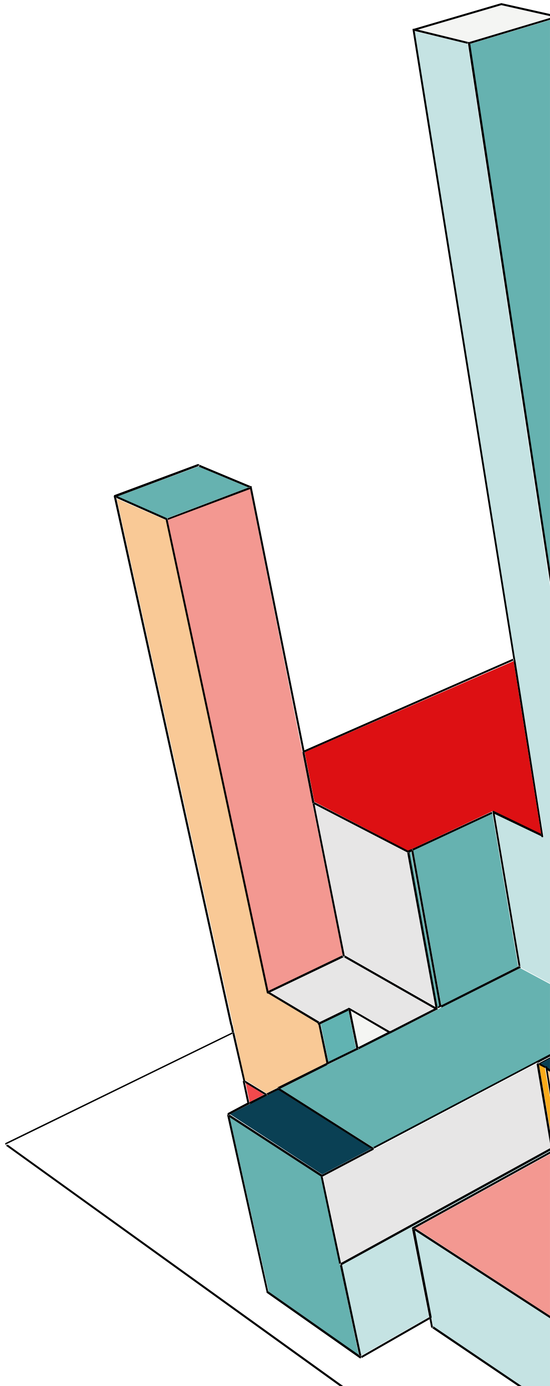
UNIFIED LOSS OPTIMIZATION FOR MULTI-RELATIONAL RECOMMENDATION

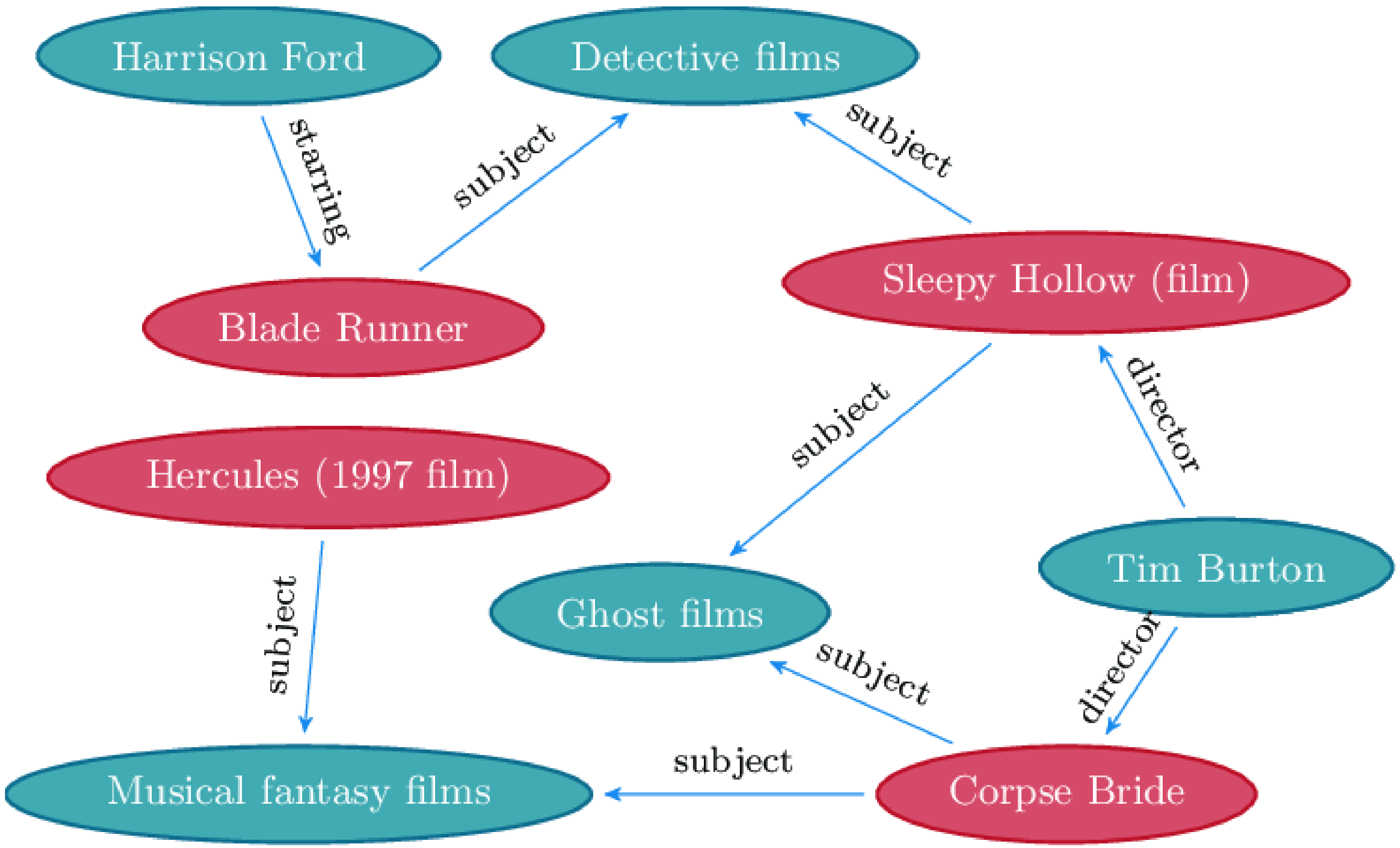
Alberto Ricchiuti
Department of Computer Science,
University Aldo Moro – Bari, Italy
Course: Semantics in Intelligent Information Access
Email:
a.ricchiuti20@studenti.uniba.it

INTRODUCTION

- **Recommender systems** are intelligent computational frameworks designed to filter and rank items according to user preferences, thereby mitigating the problem of information overload in digital environments.
- **Knowledge-aware recommender systems** extend these paradigms by explicitly integrating structured domain knowledge, often encoded as a **knowledge graph (KG)**. In such systems, entities and their semantic relations are represented as graph-structured data, enabling **relational reasoning** and **context-aware inference**.



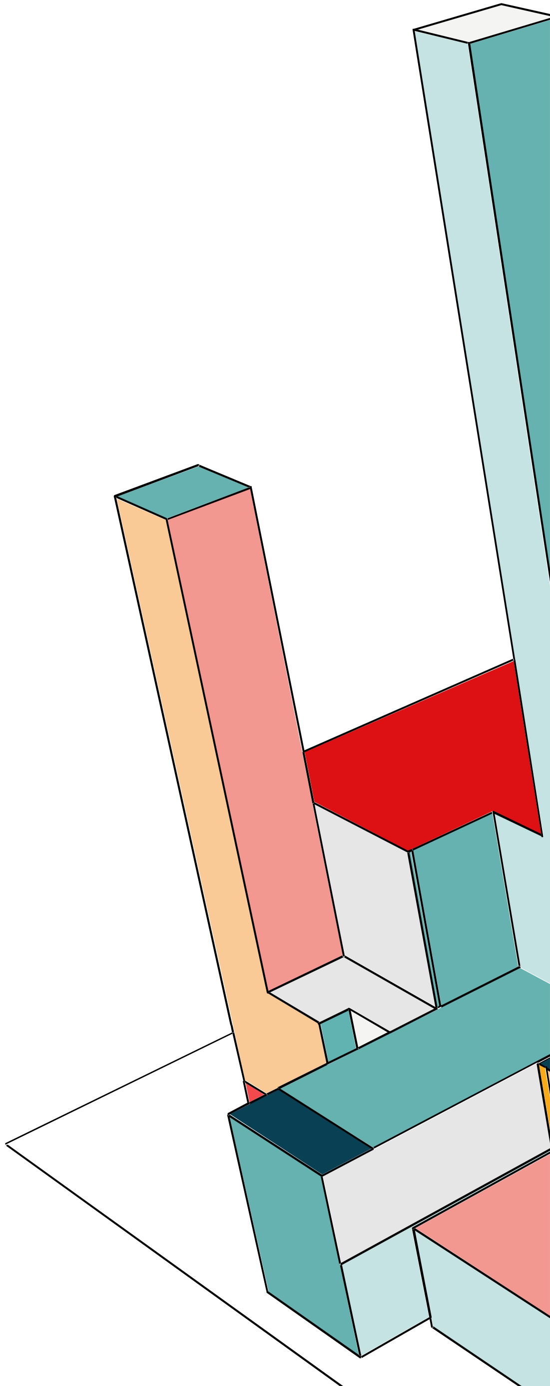
Example:

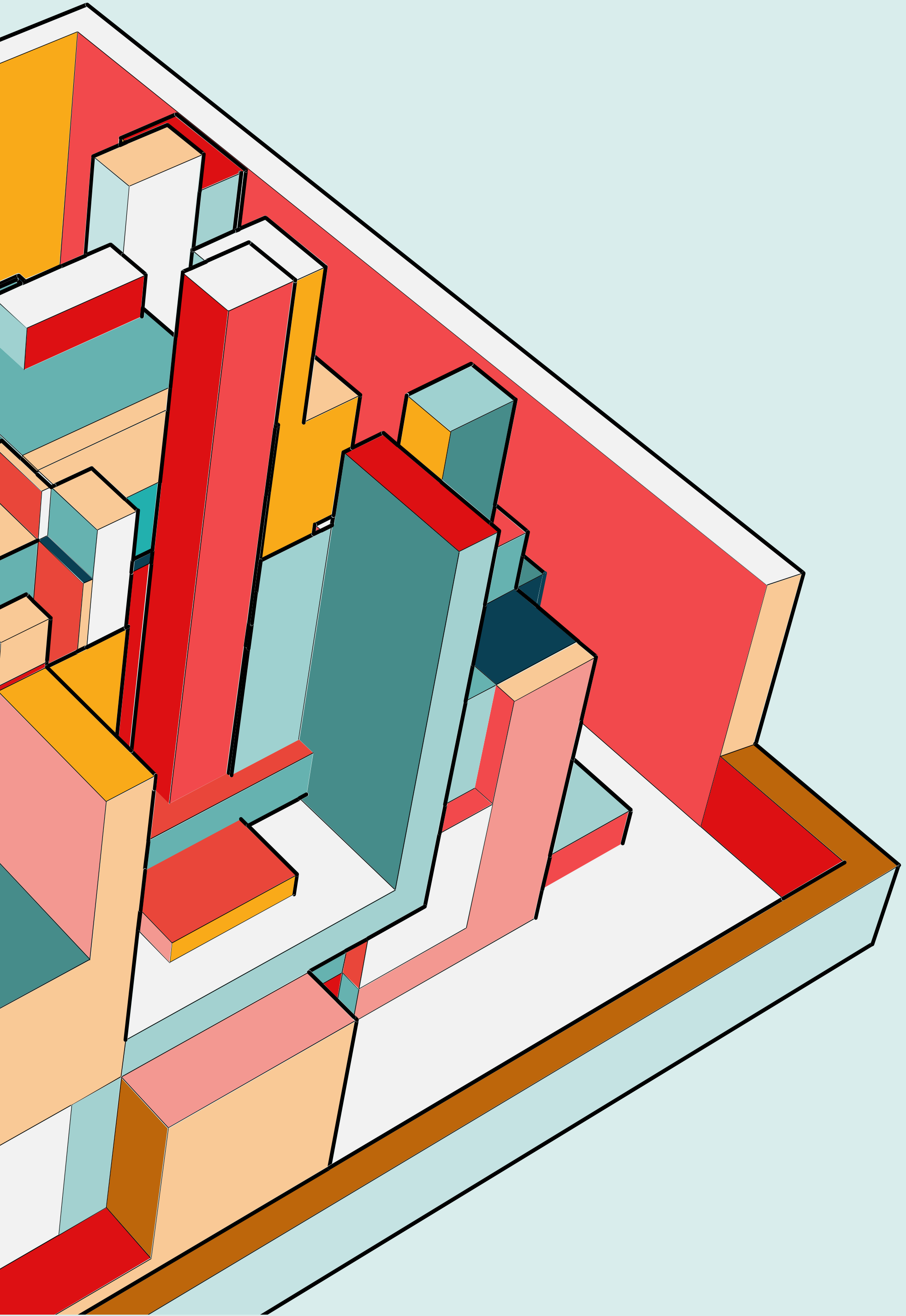


Harrison Ford
Detective films
starring
subject
subject
Blade Runner
Sleepy Hollow (film)
subject
director
Hercules (1997 film)
Tim Burton
subject
Ghost films
subject
director
subject
Musical fantasy films
Corpse Bride

MOTIVATION

- Modern recommenders operate over multi-relational graphs with rich semantics.
- Many pipelines optimize separate objectives for knowledge-graph completion and user–item ranking.
- Fragmented training can dilute specialization and complicate model comparison and attribution.





CONTRIBUTIONS

Q: What happens when KG embedding quality, recommendation accuracy, and regularization are trained under a single objective?

Built a unified framework in RecBole coupling CompGCN and TransE under one loss.

Provided a rigorous theoretical walk-through and reproducible PyTorch blueprint.

Released a harmonized evaluation protocol and ablations on composition operators.

BACKGROUND: TRANSE

Translational scoring: minimize $\|E(h) + R(r) - E(t)\|_2$ for true triples; maximize for negatives.

Simple, efficient baseline with competitive performance.

Used here as a KG component inside a unified recommendation pipeline.

BACKGROUND: COMPGCN

Relation-aware message passing with composition of entity and relation embeddings.

Composition operators: subtraction, element-wise product, addition, and circular correlation (ccorr).

Directional transforms (out/in/loop) and degree-normalized aggregation.

BACKGROUND: COMPGCN LAYER

- Compose ϕ(e_s, r) via chosen operator (e.g., ccorr with FFT/IFFT).
- Apply direction-specific linear maps W_out, W_in, W_loop.
- Aggregate normalized messages; apply BN, dropout, and nonlinearity.
- Relation update via W_rel with basis decomposition to reduce parameters.

$$h_v = f\left(\sum_{(u,r)\in\mathcal{N}(v)} W_{\lambda(r)} \phi(x_u, z_r) \right)$$

METHODS

- RecBole is an **open-source unified recommendation framework** developed to support research and experimentation in recommender systems. It is implemented in Python and built on **PyTorch**, providing a highly modular and extensible architecture.

- Implementation of CompGCN and TransE under a unified loss

UNIFIED OBJECTIVE

- **Margin-based ranking loss**: for each positive triple (h, r, t) ,a corrupted negative triple $(h, r, t^{-})$is generated. The scoring function $f(h, r, t)$evaluates the plausibility of the triple, and the hinge loss enforces that positive triples are scored higher than negatives by at least a margin γ.
- **Bayesian Personalized Ranking (BPR) loss:** for recommendation. It ensures that for a given user u, the score of a positively interacted item $i^{+}$is larger than that of a negatively sampled item i^{-} ,using the logistic sigmoid $\sigma(\cdot)$.
- **L2-norm regularization:** term that penalizes large parameter values, helping to avoid overfitting.

UNIFIED OBJECTIVE

- *Single scalar loss*: margin-based KG term + BPR ranking term + ℓ_2 regularization.
- Single backprop step.
- Fixed mixing coefficient **α**; margin **γ** for KG hinge; dot-product user–item score.

$$\mathcal{L} = \frac{1}{|\mathcal{B}_{\mathrm{KG}}|} \sum_{(h,r,t)\in\mathcal{B}_{\mathrm{KG}}} \max\big(0,\ f(h,r,t) - f(h,r,t^{-}) + \gamma\big)$$
$$+\ \alpha \frac{1}{|\mathcal{B}_{\mathrm{Rec}}|} \sum_{(u,i^{+},i^{-})\in\mathcal{B}_{\mathrm{Rec}}} -\ln\sigma\big(s_{ui^{+}} - s_{ui^{-}}\big)\ +\ \lambda\|\Theta\|_2^2$$

DATA & GRAPH CONSTRUCTION

- *MovieLens-100K* with users, items, and enriched side information
- Users: 943
- Items (movies): 1,682
- Ratings: 100,000 explicit ratings
- Rating scale: integers from 1 to 5
- Sparsity: about 93.7%
- Temporal coverage: ratings collected between September 1997 and April 1998
- Genres: 19

EXPERIMENTAL SETUP

Table 1: Core training hyperparameters by model (capacity & optimization).

Model	Emb Dim	LR	Batch	Epochs	Optim.	Layers
BPR	64	0.001	2048	300	Adam	—
CFKG	64	0.001	2048	300	Adam	—
CKE	64	0.001	2048	300	Adam	—
CompGCN	128	0.001	512	30	Adam	2
ItemKNN	—	0.001	2048	300	Adam	—
KGCN	64	0.001	2048	300	Adam	—
TransE	128	0.001	512	30	Adam	—

Notation

- U: set of users; u ∈ U is a user.
- R_u: set of relevant items for user u.
- R̂_u@K: top-K recommended items for user u (ordered).
- rel_u(i): relevance grade for item i (0,1,2,...) for user u.
- rank_u(i): position of item i in the ranked list for user u.

Precision@K. Proportion of recommended items in the top-K that are relevant:

$$\text{Precision@}K(u) = \frac{\left|\mathcal{R}_u \cap \hat{\mathcal{R}}_u^{@K}\right|}{K}, \qquad \text{Precision@}K = \frac{1}{|U|}\sum_{u \in U} \text{Precision@}K(u). \tag{1}$$

Recall@K. Proportion of relevant items that appear in the top-K recommendations:

$$\text{Recall@}K(u) = \frac{\left|\mathcal{R}_u \cap \hat{\mathcal{R}}_u^{@K}\right|}{|\mathcal{R}_u|}, \qquad \text{Recall@}K = \frac{1}{|U|}\sum_{u \in U} \text{Recall@}K(u). \tag{2}$$

nDCG@K (Normalized Discounted Cumulative Gain). Rewards placing (highly) relevant items earlier in the list. Define the DCG at cutoff K:

$$\mathrm{DCG@}K(u) = \sum_{i \in \hat{\mathcal{R}}_u^{@K}} \frac{2^{rel_u(i)} - 1}{\log_2(\mathrm{rank}_u(i) + 1)}, \tag{3}$$

where $\mathrm{rank}_u(i) \in \{1, \ldots, K\}$ is the position of item i in the ranked list for user u. Let $\mathrm{IDCG@}K(u)$ be the maximum possible $\mathrm{DCG@}K(u)$ obtained by sorting items in descending $rel_u(i)$. Then

$$\mathrm{nDCG@}K(u) = \frac{\mathrm{DCG@}K(u)}{\mathrm{IDCG@}K(u)}, \qquad \mathrm{nDCG@}K = \frac{1}{|U|} \sum_{u \in U} \mathrm{nDCG@}K(u). \tag{4}$$

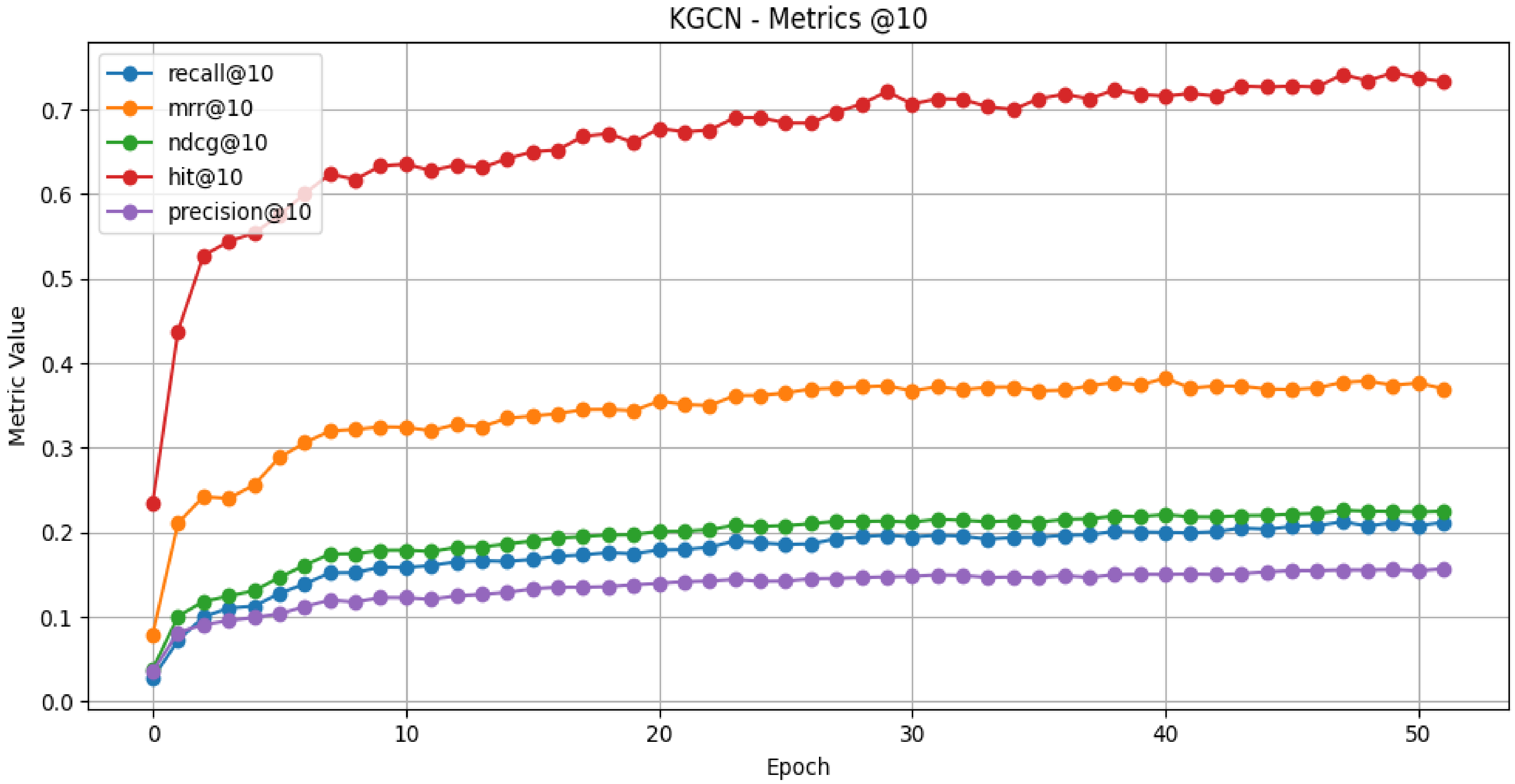
MRR (Mean Reciprocal Rank). Emphasizes the position of the first relevant item:

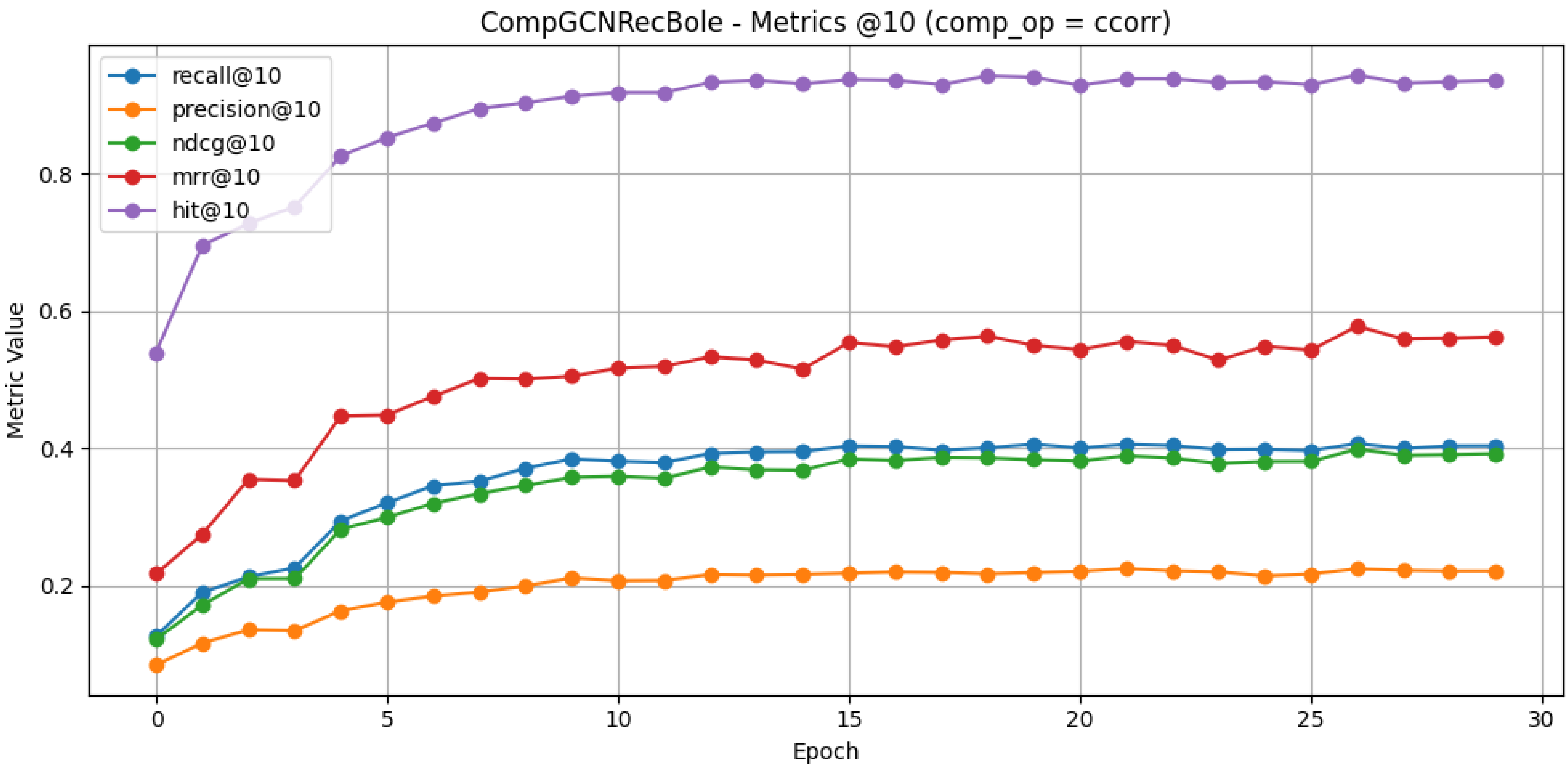
$$\mathrm{RR}(u) = \begin{cases} \frac{1}{\min\{\mathrm{rank}_u(i) : i \in \mathcal{R}_u\}} & \text{if } \mathcal{R}_u \cap \hat{\mathcal{R}}_u^{@K} \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases} \qquad \mathrm{MRR} = \frac{1}{|U|} \sum_{u \in U} \mathrm{RR}(u). \tag{5}$$

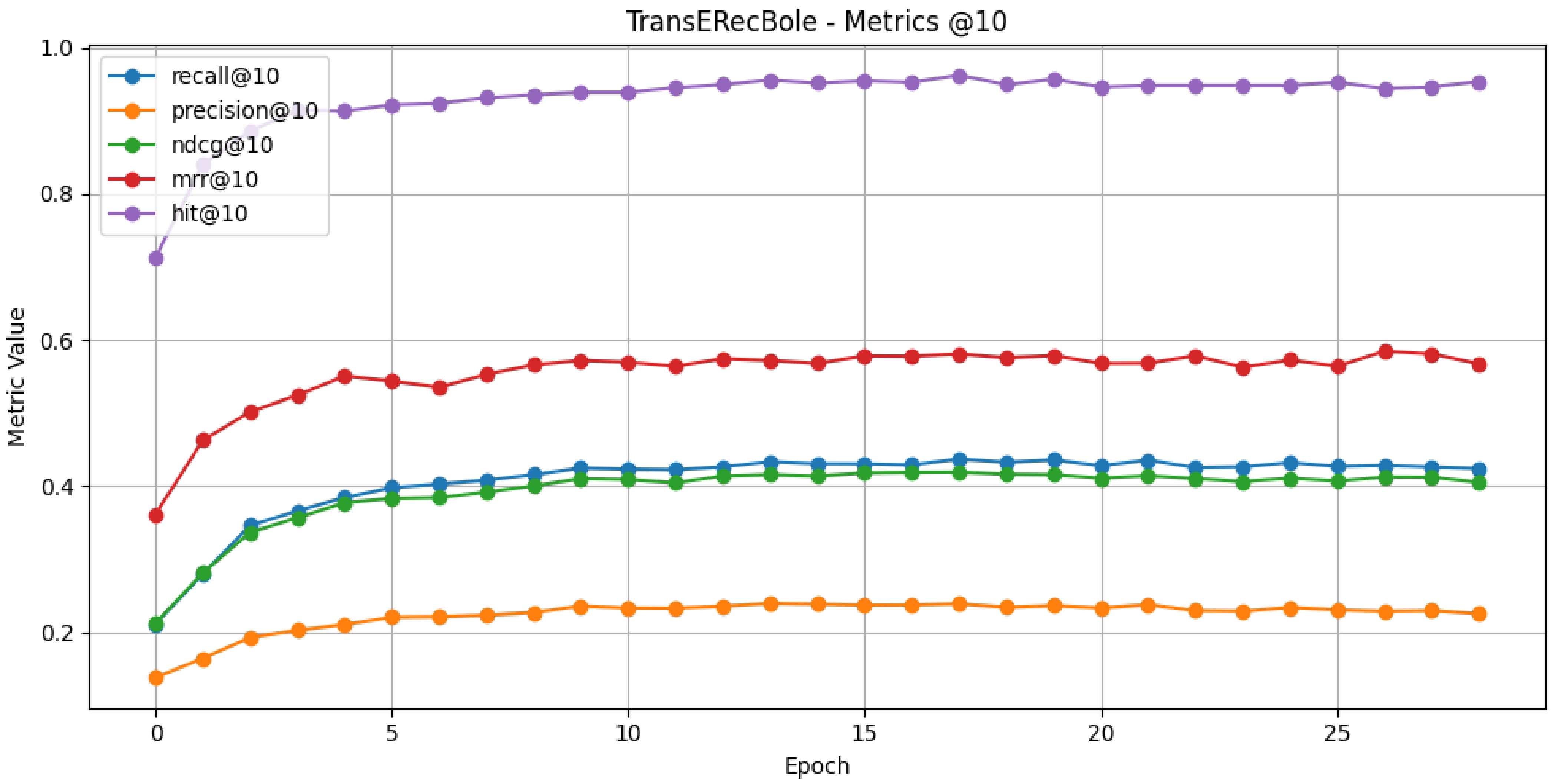
Hit Ratio@K (Hit@K). Indicates whether at least one relevant item appears in the top-K:

$$\mathrm{Hit@}K(u) = I\left[\mathcal{R}_u \cap \hat{\mathcal{R}}_u^{@K} \neq \emptyset\right], \qquad \mathrm{Hit@}K = \frac{1}{|U|} \sum_{u \in U} \mathrm{Hit@}K(u), \tag{6}$$

where $I[\cdot]$ is the indicator function.


KGCN - Metrics @10
recall@10
mrr@10
ndcg@10
hit@10
precision@10
Metric Value
Epoch
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
10
20
30
40
50


CompGCNRecBole - Metrics @10 (comp_op = ccorr)
recall@10
precision@10
ndcg@10
mrr@10
hit@10
Metric Value
Epoch
0.2
0.4
0.6
0.8
0
5
10
15
20
25
30


TransERecBole - Metrics @10
recall@10
precision@10
ndcg@10
mrr@10
hit@10
Metric Value
Epoch
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25

RESULTS

- TransE achieves the highest mean score (0.419 at best epoch), leading Recall@10 and NDCG@10.
- CompGCN with ccorr is competitive (mean 0.402), outperforming mult and sub variants.
- Collaborative baselines (BPR, ItemKNN) and classical knowledge-aware models (CKE, KGCN) trail in this regime.

RESULTS

Model	Best epoch	Avg. score	Recall	Precision	NDCG	MRR
BPR	61	0.25005	0.2168	0.1609	0.2340	0.3885
CFKG	3	0.146375	0.1169	0.0947	0.1283	0.2456
CKE	46	0.252275	0.2187	0.1603	0.2349	0.3952
CompGCNRecBole (ccorr)	26	0.402175	0.4072	0.2246	0.3988	0.5781
CompGCNRecBole (mult)	29	0.369100	0.3848	0.2058	0.3611	0.5247
CompGCNRecBole (sub)	21	0.373900	0.3991	0.2087	0.3698	0.5180
ItemKNN	0	0.240100	0.2058	0.1554	0.2213	0.3779
KGCN	47	0.242875	0.2126	0.1556	0.2260	0.3773
TransERecBole	17	0.419125	0.4373	0.2391	0.4192	0.5809

Table 3: Performance at the best epoch for each model.

FUTURE WORKS

Future work should

- Test on larger datasets
- Consider different hyperparameters

CONCLUSIONS

- A single objective coupling KG and Rec is compatible with competitive top-k ranking.
- TransE leads in this experimental regime; CompGCN (ccorr) remains close and flexible.
- Unified optimization clarifies trade-offs and simplifies cross-model comparison.

THANKS FOR YOUR ATTENTION!

Alberto Ricchiuti

Department of Computer Science

University of Bari

Email: a.ricchiuti20@studenti.uniba.it