



Improvements That Add Up

An opinionated rant on reproducibility and progress in IR



About me

I'm a 3rd year PhD student with Maarten de Rijke and Onno Zoeter at the IRLab (UvA) and the Mercury ML Lab ([booking.com](https://www.booking.com)).

Previously

- Recommender systems at Blinkist, Berlin
- M.Sc. Hasso-Plattner Institute, Potsdam
- B.Sc. University of Applied Sciences, Düsseldorf

Research interests

Unbiased learning-to-rank, user simulation, and offline evaluation

Acknowledgments

Towards reproducible machine learning research in natural language processing
SIGIR 2022 tutorial by Ana Lucic, Maurits Bleeker, Maarten de Rijke, Koustuv Sinha,
Sami Jullien, Robert Stojnic

(A great resource if you think of running a reproducibility university course)

On progress in IR

Crisis? What crisis?



Ad-Hoc Retrieval (2009)

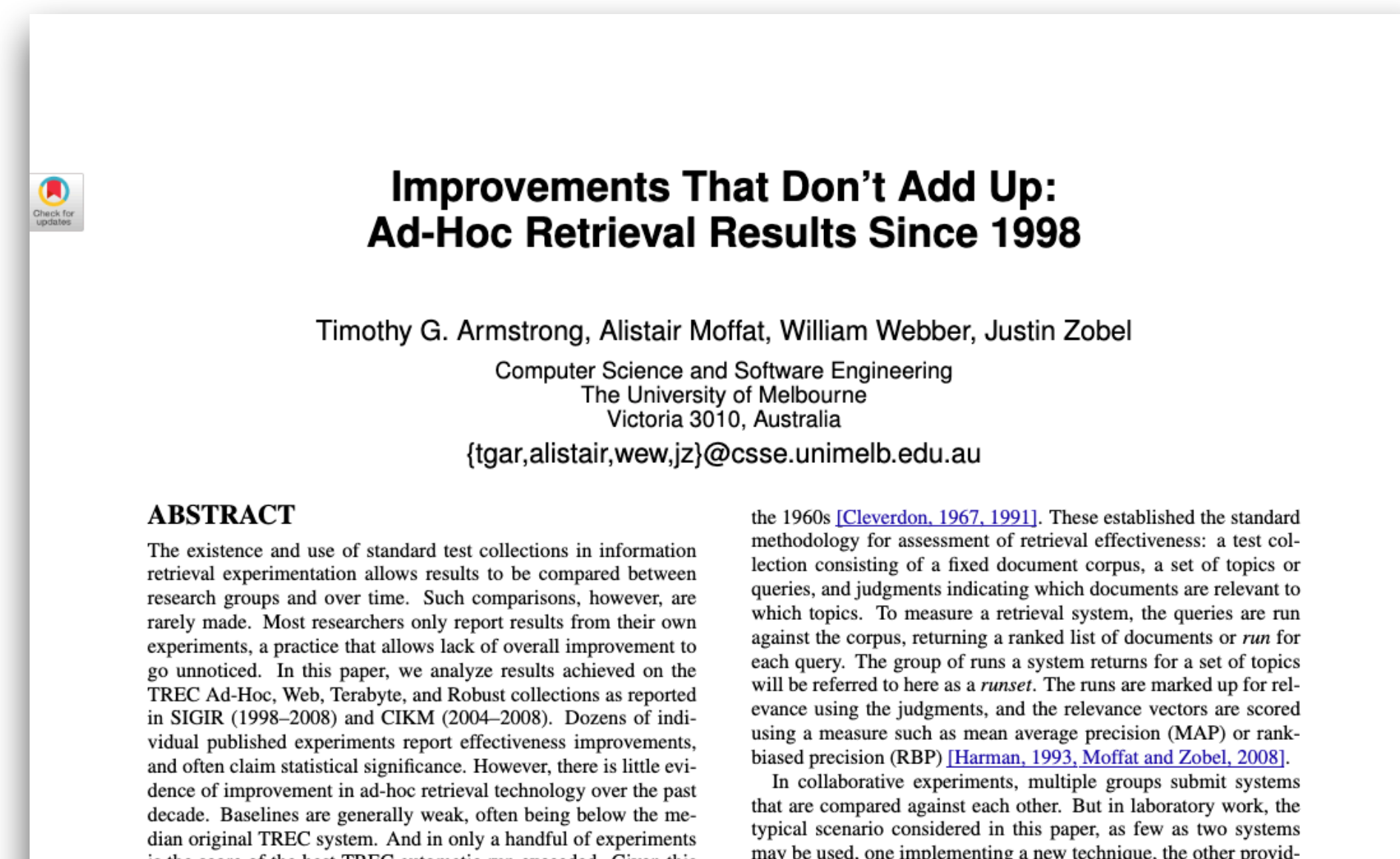
A survey of 85 SIGIR and 21 CIKM papers showed **no upward trend** in TREC ranking performance **over a decade** (1998 - 2008).

The authors point to the “selection of weak baselines that can create an illusion of incremental improvement” and “insufficient comparison with previous results” [1].

And while many papers reported **improvements**, they **did not add up to overall progress**.

[1] Armstrong, Timothy G., et al.

Improvements that don't add up: ad-hoc retrieval results since 1998. In CIKM 2009.



**Improvements That Don't Add Up:
Ad-Hoc Retrieval Results Since 1998**

Timothy G. Armstrong, Alistair Moffat, William Webber, Justin Zobel
Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia
{tgar,alistair,web,jz}@csse.unimelb.edu.au

ABSTRACT

The existence and use of standard test collections in information retrieval experimentation allows results to be compared between research groups and over time. Such comparisons, however, are rarely made. Most researchers only report results from their own experiments, a practice that allows lack of overall improvement to go unnoticed. In this paper, we analyze results achieved on the TREC Ad-Hoc, Web, Terabyte, and Robust collections as reported in SIGIR (1998–2008) and CIKM (2004–2008). Dozens of individual published experiments report effectiveness improvements, and often claim statistical significance. However, there is little evidence of improvement in ad-hoc retrieval technology over the past decade. Baselines are generally weak, often being below the median original TREC system. And in only a handful of experiments is the performance of the best TREC system significantly better than the 1960s [Cleverdon, 1967, 1991]. These established the standard methodology for assessment of retrieval effectiveness: a test collection consisting of a fixed document corpus, a set of topics or queries, and judgments indicating which documents are relevant to which topics. To measure a retrieval system, the queries are run against the corpus, returning a ranked list of documents or *run* for each query. The group of runs a system returns for a set of topics will be referred to here as a *runset*. The runs are marked up for relevance using the judgments, and the relevance vectors are scored using a measure such as mean average precision (MAP) or rank-biased precision (RBP) [Harman, 1993, Moffat and Zobel, 2008].

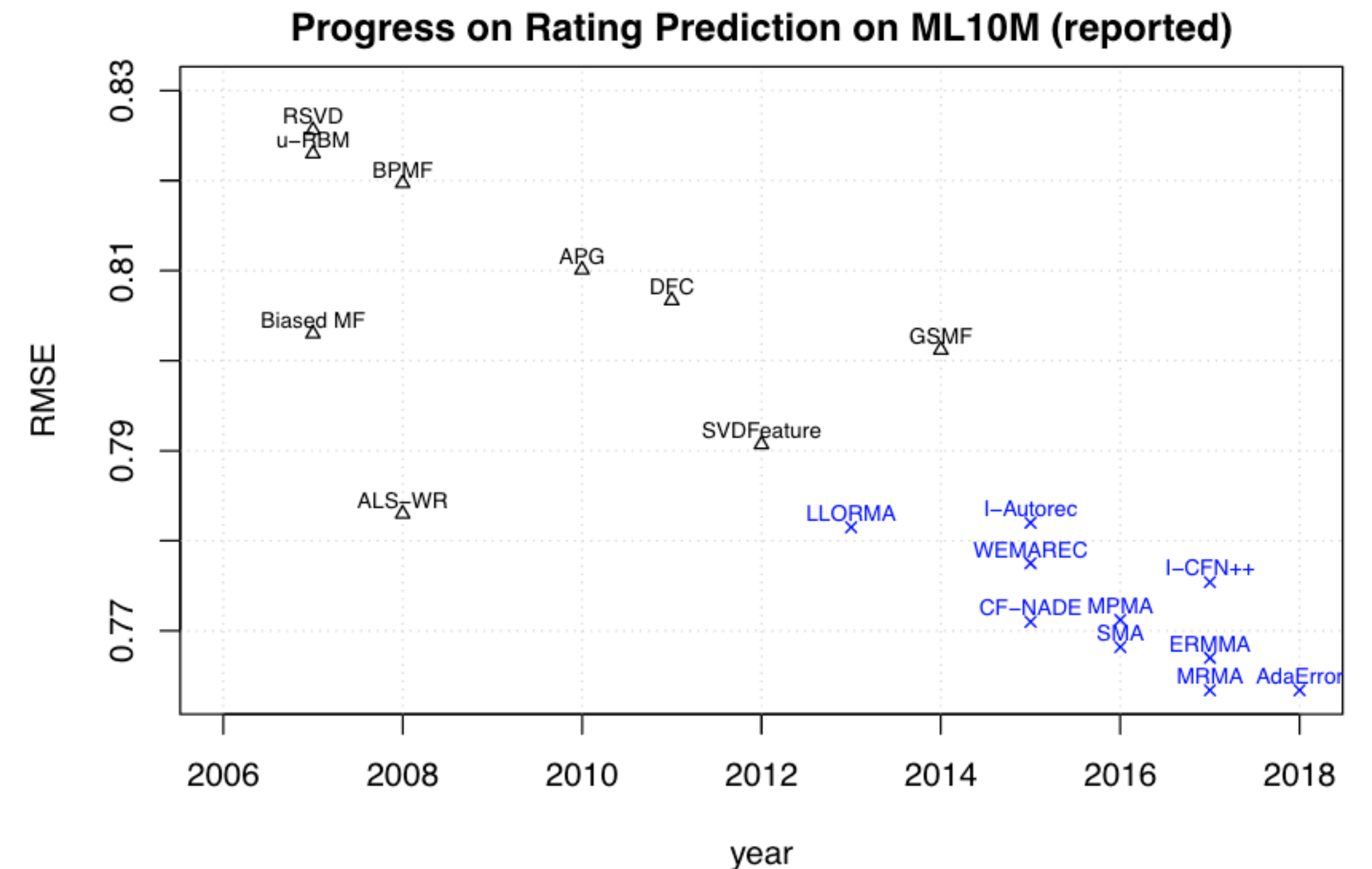
In collaborative experiments, multiple groups submit systems that are compared against each other. But in laboratory work, the typical scenario considered in this paper, as few as two systems may be used, one implementing a new technique, the other provid-

Rating Prediction (2019)

Ten recommender systems* were surveyed that reported **rating prediction improvements on ML-10M.**

Rendle et al. [1] report **improved baseline performance** with proper tuning.

Properly **tuned matrix factorization outperformed all ten methods.**



*Considered conferences: ICML, NeurIPS, WWW, SIGIR, and WWW.

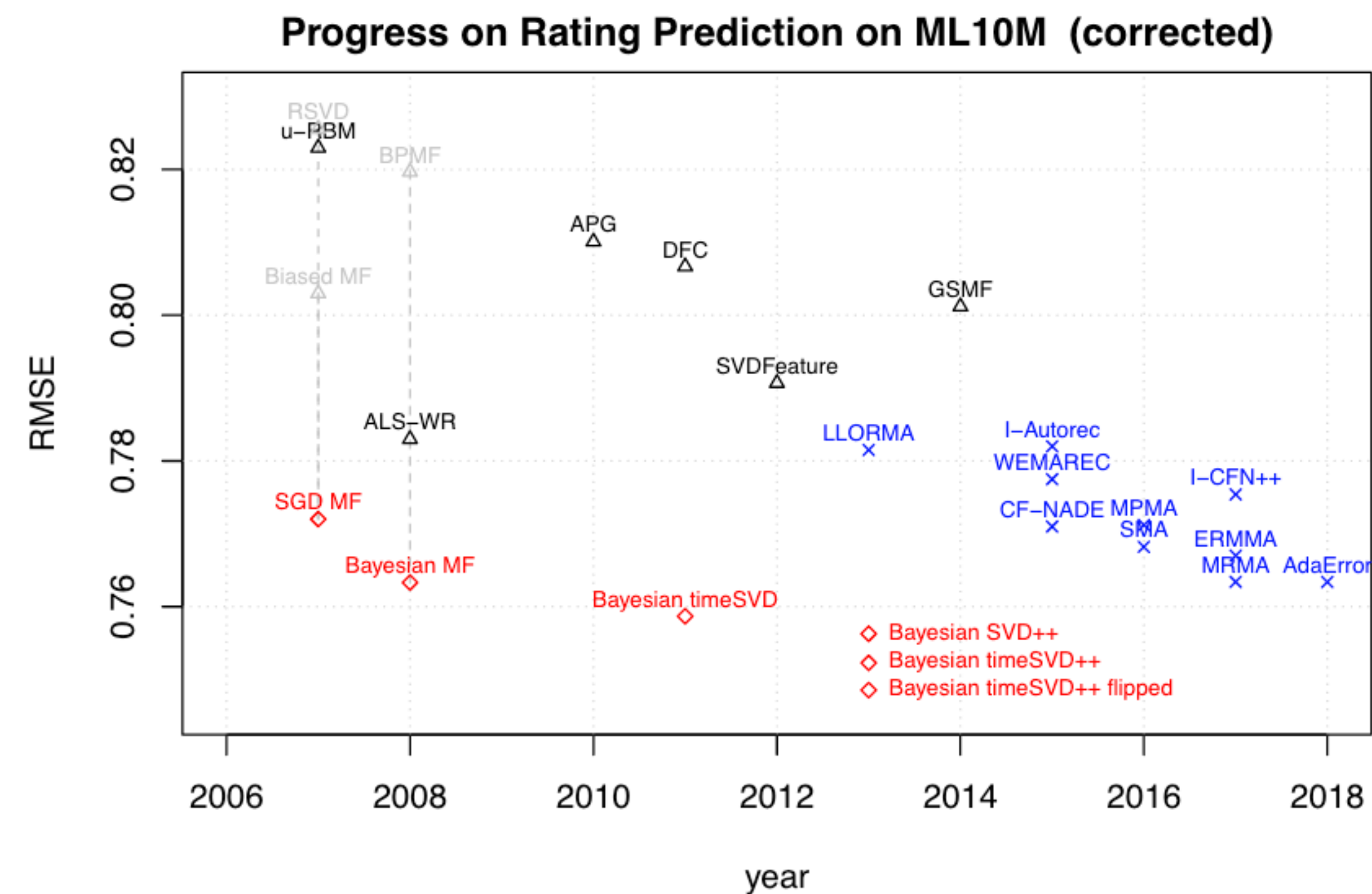
[1] Rendle, Steffen, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. arXiv preprint arXiv:1905.01395 (2019).

Rating Prediction (2019)

Ten recommender systems* were surveyed that reported **rating prediction improvements on ML-10M**.

Rendle et al. [1] report **improved baseline performance** with proper tuning.

Properly **tuned matrix factorization outperformed all ten methods**.



*Considered conferences: ICML, NeurIPS, WWW, SIGIR, and WWW.

[1] Rendle, Steffen, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. arXiv preprint arXiv:1905.01395 (2019).

Neural Recommender Systems (2019)

Survey of 18 neural top-n recommender systems published at RecSys, KDD, SIGIR, TheWebConf between 2015 and 2018.

Only 7/18 papers could be reproduced.

6/7 papers were outperformed by simple **heuristics** (KNN and graph-based methods).

One paper outperformed heuristics but not consistently a strong linear method (SLIM).

[1] Ferrari Dacrema, Maurizio, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In RecSys 2019.



Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches

Maurizio Ferrari Dacrema
Politecnico di Milano, Italy
maurizio.ferrari@polimi.it

Paolo Cremonesi
Politecnico di Milano, Italy
paolo.cremonesi@polimi.it

Dietmar Jannach
University of Klagenfurt, Austria
dietmar.jannach@aau.at

ABSTRACT

Deep learning techniques have become the method of choice for researchers working on algorithmic aspects of recommender systems. With the strongly increased interest in machine learning in general, it has, as a result, become difficult to keep track of what represents the state-of-the-art at the moment, e.g., for top-n recommendation tasks. At the same time, several recent publications point out problems in today's research practice in applied machine learning, e.g., in terms of the reproducibility of the results or the choice of the baselines when proposing new models.

In this work, we report the results of a systematic analysis of algorithmic proposals for top-n recommendation tasks. Specifically, we considered 18 algorithms that were presented at top-level research conferences in the last years. Only 7 of them could be reproduced with reasonable effort. For these methods, it however turned out that 6 of them can often be outperformed with comparably simple heuristic methods, e.g., based on nearest-neighbor or graph-based techniques. The remaining one clearly outperformed the baselines but did not consistently outperform a well-tuned non-neural linear ranking method. Overall, our work sheds light on a number of potential problems in today's machine learning scholarship and calls for improved scientific practices in this area.

systems. Novel methods were proposed for a variety of settings and algorithmic tasks, including top-n recommendation based on long-term preference profiles or for session-based recommendation scenarios [36]. Given the increased interest in machine learning in general, the corresponding number of recent research publications, and the success of deep learning techniques in other fields like vision or language processing, one could expect that substantial progress resulted from these works also in the field of recommender systems. However, indications exist in other application areas of machine learning that the achieved progress—measured in terms of accuracy improvements over existing models—is not always as strong as expected.

Lin [25], for example, discusses two recent neural approaches in the field of information retrieval that were published at top-level conferences. His analysis reveals that the new methods do not significantly outperform existing baseline methods when these are carefully tuned. In the context of recommender systems, an in-depth analysis presented in [29] shows that even a very recent neural method for session-based recommendation can, in most cases, be outperformed by very simple methods based, e.g., on nearest-neighbor techniques. Generally, questions regarding the true progress that is achieved in such applied machine learning settings are not new, nor tied to research based on deep learning.

And many more examples...

- **Session-based recommendation**
Simple KNN-based methods beat complex methods [1].
- **Next (shopping) basket recommendation**
Frequency-based heuristics beat most deep methods [2].
- **Neural IR**
On the use of weak baselines in neural IR [3].
- **Sequential recommendation**
On implementation differences and tuning of BERT4Rec as a baseline [4].
- **Learning to rank**
Most neural LTR methods are outperformed by gradient boosting [5].
- **Unbiased learning to rank**
Improvements from simulations do not translate to real-world data [6].

[1] Ludewig, Malte, and Dietmar Jannach. Evaluation of session-based recommendation algorithms. In UMUAI 2018.

[2] Li, Ming, et al. A next basket recommendation reality check. In TOIS 2023.

[3] Lin, Jimmy. The neural hype and comparisons against weak baselines. In SIGIR Forum 2019.

[4] Petrov, Aleksandr, and Craig Macdonald. A systematic review and replicability study of bert4rec for sequential recommendation. In RecSys 2022.

[5] Qin, Zhen, et al. Are neural rankers still outperformed by gradient boosted decision trees?. In ICLR 2021.

[6] Hager, Philipp, et al. Unbiased Learning to Rank Meets Reality: Lessons from Baidu's Large-Scale Search Dataset. In SIGIR 2024.

A larger problem in machine learning

Reproducibility has been a problem in, e.g.:

- Deep reinforcement learning [1]
- Generative adversarial networks [2]
- Metric learning [3]
- Deep Bandits [4]
- Computer vision [5]
- Forecasting [6]
- Natural language processing [7]

[1] Henderson, Peter, et al. Deep reinforcement learning that matters. In AAAI 2018.

[2] Lucic, Mario, et al. Are gans created equal? A large-scale study. In NeurIPS 2018.

[3] Musgrave, Kevin, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In ECCV 2020.

[4] Riquelme, Carlos, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. In ICLR 2018.

[5] Bouthillier, Xavier, César Laurent, and Pascal Vincent. Unreproducible research is reproducible. In ICML 2019.

[6] Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and Machine Learning forecasting methods: Concerns and ways forward. In PloS 2018.

[7] Belz, Anya, et al. A systematic review of reproducibility research in natural language processing. In EACL 2021.

Why care about reproducibility?

“It is a truism within the community that at least one clear win is needed for acceptance at a top venue.

*Yet, a moment of reflection recalls that **the goal of science is not wins, but knowledge [1].**”*

[1] Sculley, David, et al. Winner's curse? On pace, progress, and empirical rigor. In ICLR workshop 2018.



What is reproducibility anyway?

ACM Definitions

Repeatability

Same team, same experimental setup

Reproducibility

Different team, same experimental setup

Replicability

Different team, different experimental setup

Other conferences, other definitions ...

NeurIPS definitions

Reproducible

Same experimental setup, same data

Replicable

Same experimental setup, different data

Robust

Different experimental setup, same data

Generalizable

Different experimental setup, different data

		Data	
		Same	Different
Code & Analysis	Same	Reproducible	Replicable
	Different	Robust	Generalisable

Defining reproducibility at NeurIPS [1]

[1] Pineau, Joelle, et al. Improving reproducibility in machine learning research. In JMLR 2021.

... but similar notions

As Gundersen [1] observes: *“reproducibility is an elusive concept”, but some ideas are similar:*

Re-run code

The published code/setup is executable and gives similar results

Re-implement idea

A method can be implemented and gives similar results

Idea/lesson generalizes (actual progress)

We can draw similar conclusions in new experimental setups

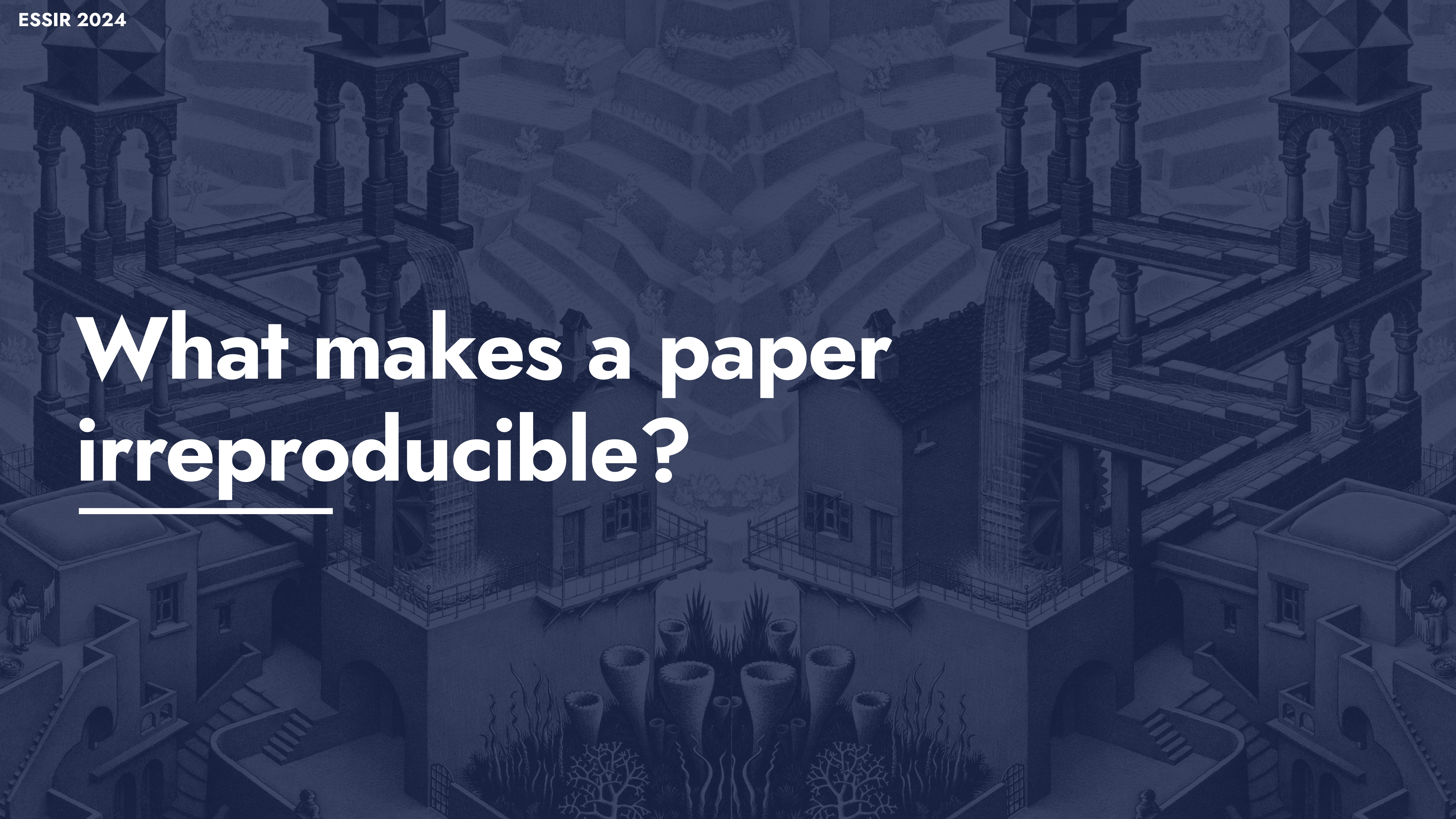
[1] Gundersen, Odd Erik. The fundamental principles of reproducibility. In Philosophical Transactions of the Royal Society 2021.

One more definition

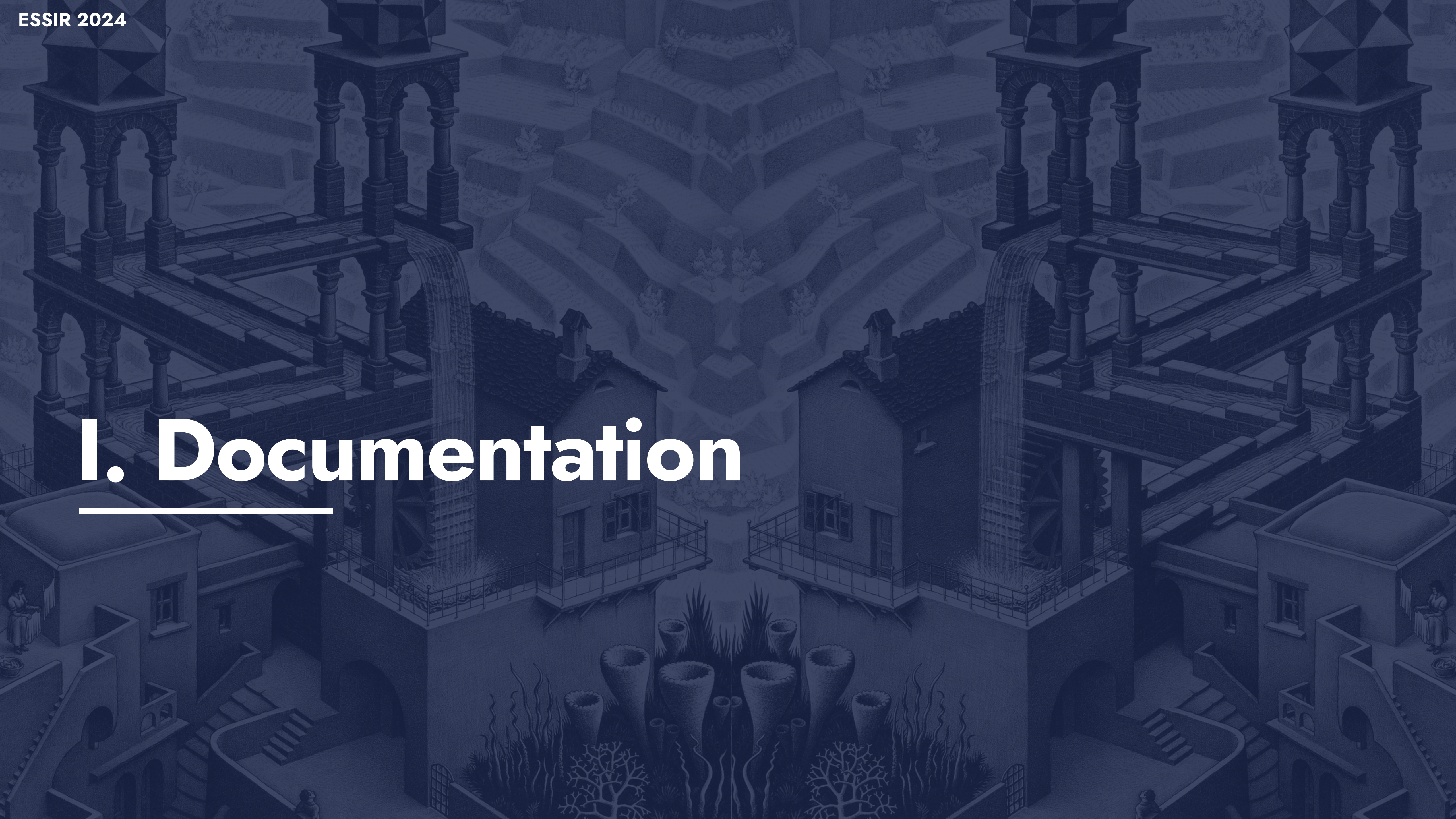
*“Reproducibility is the ability of **independent investigators** to **draw the same conclusions** from an experiment by **following the documentation** shared by the original investigators [1].”*

(agrees with the ACM definition)

What makes a paper irreproducible?



I. Documentation



Insufficient documentation

Gundersen and Kjensmo [1] surveyed 400 papers (2013 - 2016) and found that **documentation practices in AI render most reported research results irreproducible**, e.g.:

Method

Formulate problem statements (47%), objective (22%), or research questions (6%)

Results

Release train set (56%), test set (30%), or results (4%)

Experiments

Describe the setup (69%), hardware specs (27%), or release code (8%)

Disclaimer: The authors searched for explicit terms. Thus, the numbers are probably too low.

[1] Gundersen, Odd Erik, and Sigbjørn Kjensmo. State of the art: Reproducibility in artificial intelligence. In AAAI 2018.



II. Scientific method

Hypothesis testing

Only 47% of AI papers included a problem statement [1]. But let's be honest, we also often **start projects with experiments right away.**

That, however, can lead to:

- **Unclear research questions (RQs)**
- **Wrong conclusions**
- **Wasted time, effort, and computational power**

Formulate (at least an initial version) the RQs before starting experiments.

[1] Gundersen, Odd Erik, and Sigbjørn Kjensmo. State of the art: Reproducibility in artificial intelligence. In AAAI 2018.

Problematic hypothesis testing

Cherry-picking: Only report results that support your hypothesis.

P-Hacking: Analyze the results in different ways (e.g., including/excluding covariates) until you find a significant result.

Fishing expeditions: Indiscriminately examine associations between variables without intending to test a priori hypothesis.

Hypothesizing After the Results are Known (HARKing): Find a significant result and construct your hypothesis retroactively. Note that this is not the same as an exploratory analysis.

[1] Andrade, Chittaranjan. HARKing, cherry-picking, p-hacking, fishing expeditions, and data dredging and mining as questionable research practices. The Journal of clinical psychiatry 2021.

Statistical testing

Comparing the means of two models is not enough to conclude model A is better than model B. Especially in ML, we often **obtain significant differences** by chance [1, 2].

Here are a few things to keep in mind:

- Compare model runs across seeds and datasets
- Formulate a null hypothesis per dataset
- Correct for multiple comparisons when comparing multiple models
- Be careful when using Wilcoxon, sign, or bootstrap-shift tests [3]
- **Report the used tests, the significance level, and add confidence intervals**

See [3, 4] for a good discussion of statistical testing in IR.

But does anybody know of a great hands-on Python tutorial?

[1] Reimers, Nils, and Iryna Gurevych. Why comparing single performance scores does not allow to draw conclusions about machine learning approaches. arXiv preprint arXiv:1803.09578 (2018).

[2] Dehghani, Mostafa, et al. "The benchmark lottery." arXiv preprint arXiv:2107.07002 (2021).

[3] Urbano, Julián, Harley Lima, and Alan Hanjalic. Statistical significance testing in information retrieval: an empirical analysis of type I, type II and type III errors. In SIGIR 2019.

[4] Smucker, Mark D., James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. In CIKM 2007.

III. Code & data

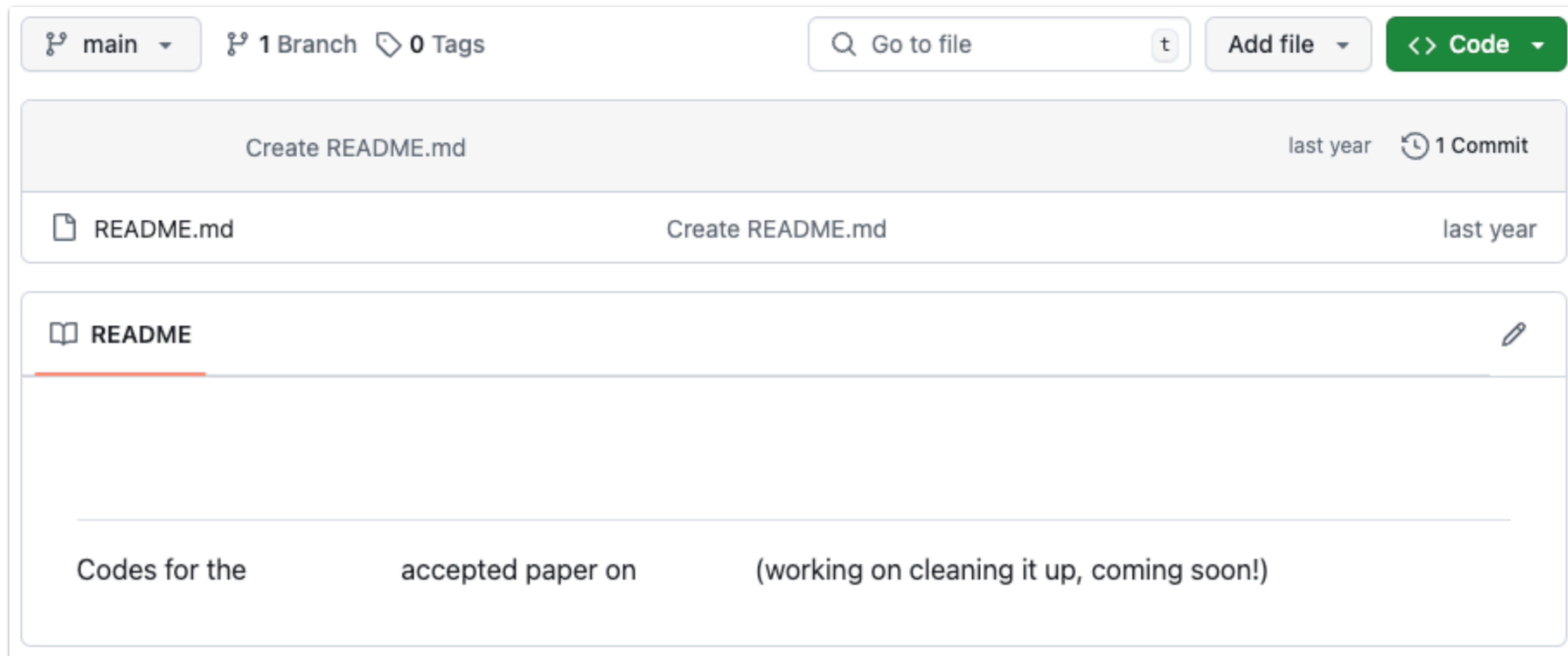


How to publish code

Implementation details. The source code to reproduce the findings from the paper is available at: <https://github.com/>

How NOT to publish code

Implementation details. The source code to reproduce the findings from the paper is available at: <https://github.com/>



The screenshot shows a GitHub repository interface. At the top, there are navigation elements: a dropdown menu for the current branch (set to 'main'), a link to '1 Branch', and a link to '0 Tags'. To the right, there is a search bar labeled 'Go to file', an 'Add file' button, and a green 'Code' button. Below these, a commit history table is visible, showing a single commit for 'Create README.md' made 'last year' with '1 Commit'. The file list below shows 'README.md' with a 'Create README.md' button and 'last year' timestamp. The README content area is mostly empty, with a red underline under the 'README' header. At the bottom of the README content, there is a line of text: 'Codes for the accepted paper on (working on cleaning it up, coming soon!)

Not publishing **all necessary** code & data

In [3], **12/21 papers linked a repository. In 2/12 cases**, that repository was **empty or non-existent**. Even if code is published, it is **often incomplete** [1, 2, 3]:

- **Datasets:** Including splits and preprocessing steps
- **Baselines:** Including code and hyperparameter tuning
- **Method:** All details, final hyperparameters, and random seeds
- **Evaluation protocol & visualizations**
- **Dependencies:** List of all dependencies with exact versions
- **Scripts:** All scripts used to orchestrate the project
- **Stale URLs:** Links for code and data stop working...

[1] Ferrari Dacrema, Maurizio, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In RecSys 2019.

[2] Ferrari Dacrema, Maurizio, et al. A troubling analysis of reproducibility and progress in recommender systems research. In TOIS 2021.

[3] Shehzad, Faisal, and Dietmar Jannach. Everyone's a winner! on hyperparameter tuning of recommendation models. In RecSys 2023.

Not polishing code

In most cases, papers defer to the code for exact details. However, **code quality impacts understanding and, thus, reproducibility**. Code readability can be impacted by, a.o:

- Inconsistent formatting
- Large amounts of commented code (e.g., commenting out different run options)
- Deeply nesting statements
- A high amount of redundancy
- Very long methods and complex file structures
- Missing comments for important deviations
- Being too modular (not everything has to be a library)
- ...

How to polish code

An incomplete list of things, I think, makes code easier to understand:

- Follow the Python style-guide for naming stuff
- Use (strict) formatters: black, autopep8, ruff
- Remove unused code: isort, autoflake
- Catch bugs early with linters: flake8, pylama
- Remove commented code, look up old code in git
- Use environment managers with reproducible environments: mamba, poetry
- Don't write files to disk unless absolutely necessary
- Write scripts that orchestrate your entire experiment
- Break up large projects into multiple repositories

...

I found these helpful for **reuse code** for future projects and to **understand code two years later**.

IV. Uncontrolled randomness



Randomness through design decisions

Design decisions that introduce stochasticity [1]:

- Random weight initialization
- Stochastic operations (dropout, noisy activations)
- Random feature selection (e.g., in random forest)
- Data splitting, shuffling, batch ordering
- Hyperparameter tuning procedure (e.g., Bayesian methods)
- **Sampled metrics [3]**

Fix and report random seeds [2], release code, and datasets!

[1] Gundersen, Odd Erik, et al. Sources of irreproducibility in machine learning: A review. In arXiv:2204.07610 2022.

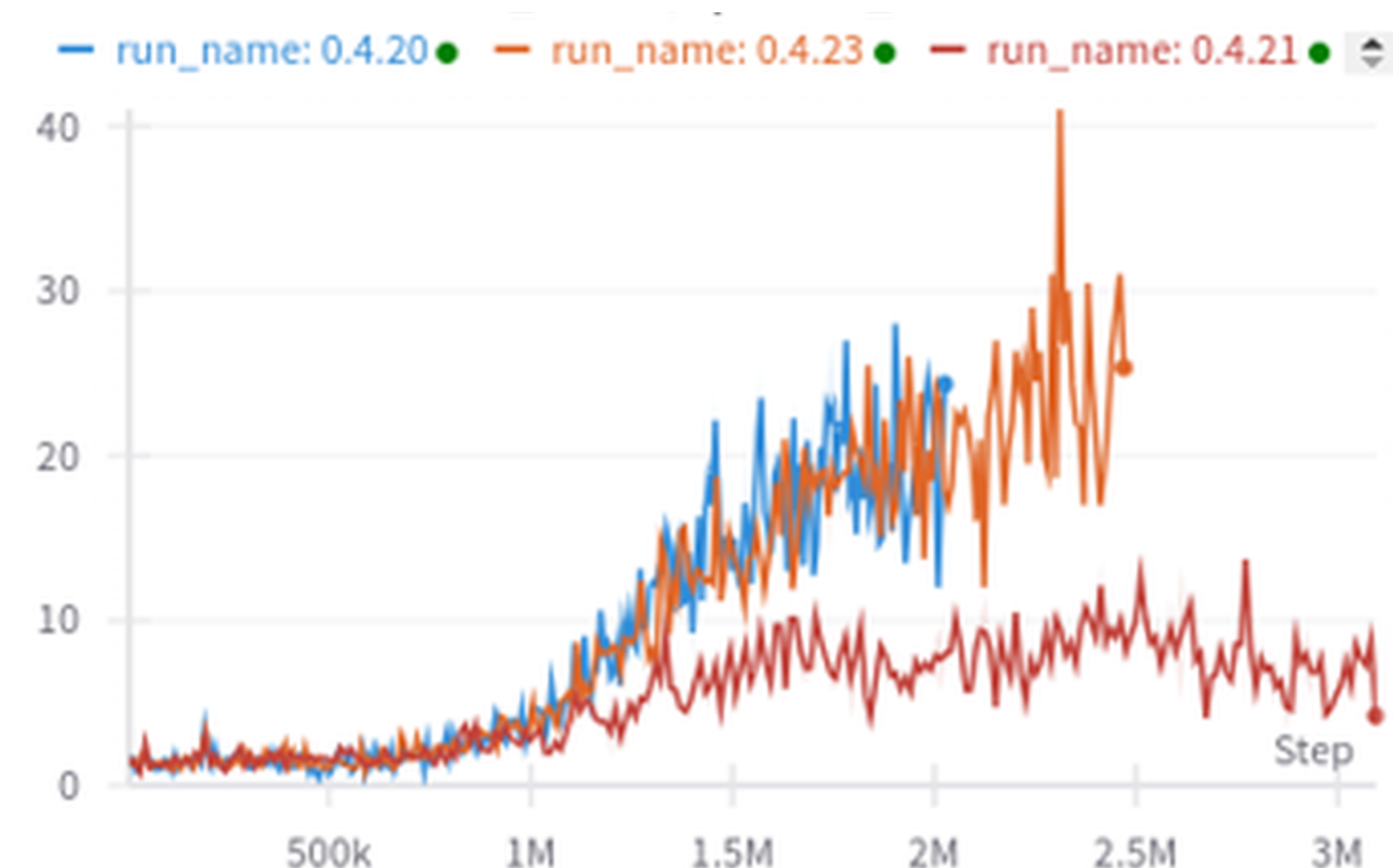
[2] E.g., see: <https://pytorch.org/docs/stable/notes/randomness.html>

[2] Krichene, Walid, and Steffen Rendle. On sampled metrics for item recommendation. In KDD 2020.

Randomness through implementation

Implementation details that introduce stochasticity [1]:

- **Random seed** for pseudo-random number generator
- **Frameworks** (Jax, PyTorch, TensorFlow), different versions and operating systems
- **CPU/GPU model**, CUDA optimizations
- Parallel execution
- Compiler settings
- Hardware rounding errors...



**Reward of the same RL model
across three different Jax versions [2]**

Fix and report random seeds, software versions, and used hardware!

[1] Gundersen, Odd Erik, et al. Sources of irreproducibility in machine learning: A review. In arXiv:2204.07610 2022.

[2] Observation by Sami Jullien and Romain Deffayet



V. Baselines

Probably the number one complaint in IR reproducibility studies

Baselines

Unavailable baselines [1, 2]

E.g., copying results, parameters, or not including baseline code

Untuned baselines [1, 2, 4]

E.g., *we use the same parameters as X...*

Lack of simple baselines [1, 2, 3]

E.g., not comparing against sensible heuristics

Lack of strong baselines [1, 2, 4, 5, 6]

E.g., not comparing against strong non-neural methods

Incorrectly implemented baselines [4, 6]

E.g., different implementations of the same method can vary in performance

[1] Ferrari Dacrema, Maurizio, et al. A troubling analysis of reproducibility and progress in recommender systems research. In TOIS 2021.

[2] Shehzad, Faisal, and Dietmar Jannach. Everyone's a winner! On hyperparameter tuning of recommendation models. In RecSys 2023.

[3] Li, Ming, et al. A next basket recommendation reality check. In TOIS 2023.

[4] Petrov, Aleksandr, and Craig Macdonald. A systematic review and replicability study of bert4rec for sequential recommendation. In RecSys 2022.

[5] Lin, Jimmy. The neural hype and comparisons against weak baselines. In SIGIR Forum 2019.

[6] Qin, Zhen, et al. Are neural rankers still outperformed by gradient boosted decision trees?. In ICLR 2021.

Untuned baselines

Shehzad and Jannach [1] surveyed 21 recommender systems from KDD, RecSys, SIGIR, TheWebConf, and WSDM in 2022 and found:

- 6/21 papers contain **no information** about hyperparameters at all.
- 4/21 papers **copy parameters** from previous work.
- 4/21 papers use the **same parameters across datasets**.
- 7/21 papers list **parameter ranges** but **not the tuning method**.

Only two papers describe **parameter ranges, the final values, tuning methods, and tune across datasets**.

Only one of the two papers also released their code.

[1] Shehzad, Faisal, and Dietmar Jannach. Everyone's a winner! On hyperparameter tuning of recommendation models. In RecSys 2023.

Untuned baselines

The authors go on to demonstrate **the importance of tuning baselines:**

Even the worst-performing tuned model outperformed all other untuned methods!

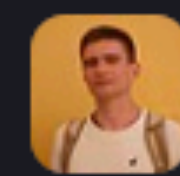
In short, **Everyone is a winner!**

Tuned models		
ML-1M		
<i>Model</i>	<i>nDCG@10</i>	<i>Model</i>
Mult-DAE	0,300	NeuMF
Mult-VAE	0,294	Mult-VAE
GMF	0,280	GMF
NeuMF	0,277	Mult-DAE
ONCF	0,225	<i>MostPop</i>
<i>MostPop</i>	0,162	ConvMF
ConvMF	0,160	NGCF
NGCF	0,100	ONCF
Non-tuned models		>
Mult-DAE	0,071	Mult-DAE
ONCF	0,037	Mult-VAE
ConvMF	0,022	ConvMF
NeuMF	0,021	GMF
GMF	0,016	NGCF
NGCF	0,013	ONCF
Mult-VAE	0,006	NeuMF

Comparison of tuned and untuned models on ML-1M [1]

The importance of simple baselines

One fateful morning after working on a project for over two months:



Romain Deffayet 8:30 AM

what do you think of that perf ?

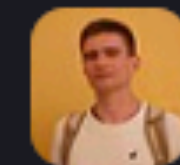
image.png ▾

mrr@10	dcg@01	dcg@03	dcg@05
0.5156	1.2347	2.5978	3.6720



Philipp Hager 8:31 AM

Rather mediocre I'd say to what we've been seeing lately



Romain Deffayet 8:32 AM

Ok and now that ?

image.png ▾

mrr@10	dcg@01	dcg@03	dcg@05
0.5838	1.5257	3.1245	4.2864

This is the fancy AI model that gave rise to the second one:

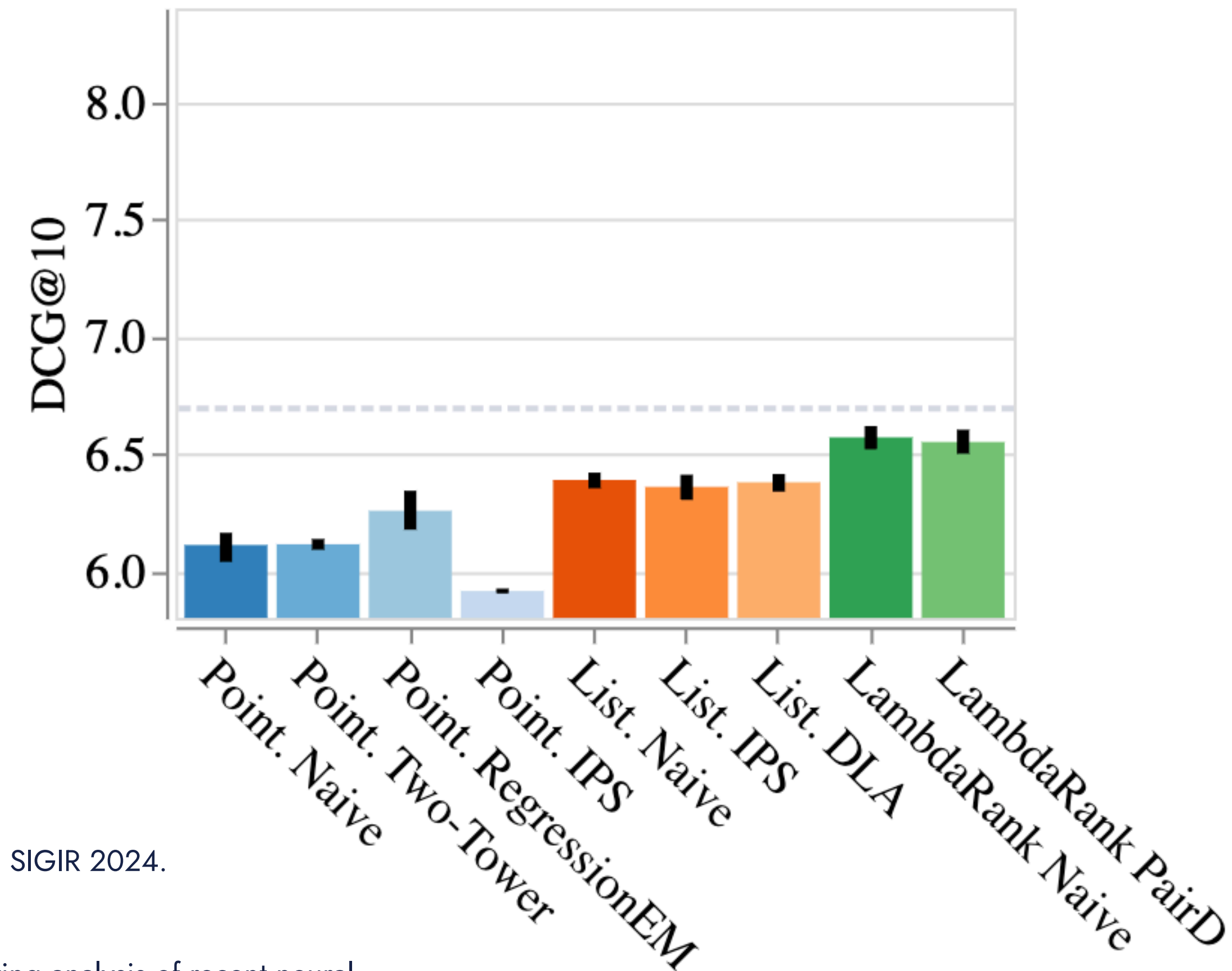
```
y_predict = jax.numpy.asarray(np.random.uniform(size=batch["label"].shape))
```

Simple baselines

Simple baselines can **uncover fundamental flaws** and help to **judge performance gains**, e.g.:

- Random ranking [1]
- Frequency/popularity baseline [2]
- Heuristics such as BM25 and query likelihood [3]
- KNN-based method [4]
- Random-walk-based methods [5]

All reproduced baseline models perform worse than random [1].



[1] Hager, Philipp, et al. Unbiased Learning to Rank Meets Reality: Lessons from Baidu's Large-Scale Search Dataset. In SIGIR 2024.

[2] Li, Ming, et al. A next basket recommendation reality check. In TOIS 2023.

[3] Lin, Jimmy. The neural hype and comparisons against weak baselines. In SIGIR Forum 2019.

[4] Ferrari Dacrema, Maurizio, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In RecSys 2019.

[5] Ferrari Dacrema, Maurizio, et al. A troubling analysis of reproducibility and progress in recommender systems research. In TOIS 2021.

Strong baselines

To claim state-of-the-art, a comparison against strong baselines is necessary.

The best baselines dependent on the task, sometimes even on the implementation.
Some examples from this talk:

- **Top-n recommendation [1, 2]:** Linear models (SLIM, EASER^R)
- **Rating prediction [3, 4]:** Matrix factorization (SVD++, iALS)
- **Learning to rank [5]:** Gradient boosting (LightGBM LambdaMART)

Reproducibility papers are good starting points for identifying strong baselines.

[1] Ferrari Dacrema, Maurizio, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In RecSys 2019.

[2] Ferrari Dacrema, Maurizio, et al. A troubling analysis of reproducibility and progress in recommender systems research. In TOIS 2021.

[3] Rendle, Steffen, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. arXiv preprint arXiv:1905.01395 (2019).

[4] Rendle, Steffen, et al. Revisiting the performance of iALS on item recommendation benchmarks. In RecSys 2022.

[5] Qin, Zhen, et al. Are neural rankers still outperformed by gradient boosted decision trees?. In ICLR 2021.

Back to the idealism part

But, but, ... you said at the start that **we should not chase wins!**

Indeed, I echoed Sculley et al.'s sentiment that research should be **about developing insight and understanding rather than winning [1]**.

So, we should not claim wins, if we didn't achieve them honestly.

We should **investigate a good idea and report the results (also negative ones)**.

Or we chase different goals, **goals beyond accuracy [2]**.

And maybe, the next time your transformer model is crushed by BM25 (like mine was [3]), we need to **acknowledge that real progress is very hard**.

[1] Sculley, David, et al. Winner's curse? On pace, progress, and empirical rigor. In ICLR workshop 2018.

[2] Kaminskas, Marius, and Derek Bridge. Diversity, serendipity, novelty, and coverage:

A survey and empirical analysis of beyond-accuracy objectives in recommender systems. In ACM TIIS 2016.

[3] Hager, Philipp, et al. Unbiased Learning to Rank Meets Reality: Lessons from Baidu's Large-Scale Search Dataset. In SIGIR 2024.

V. Communication



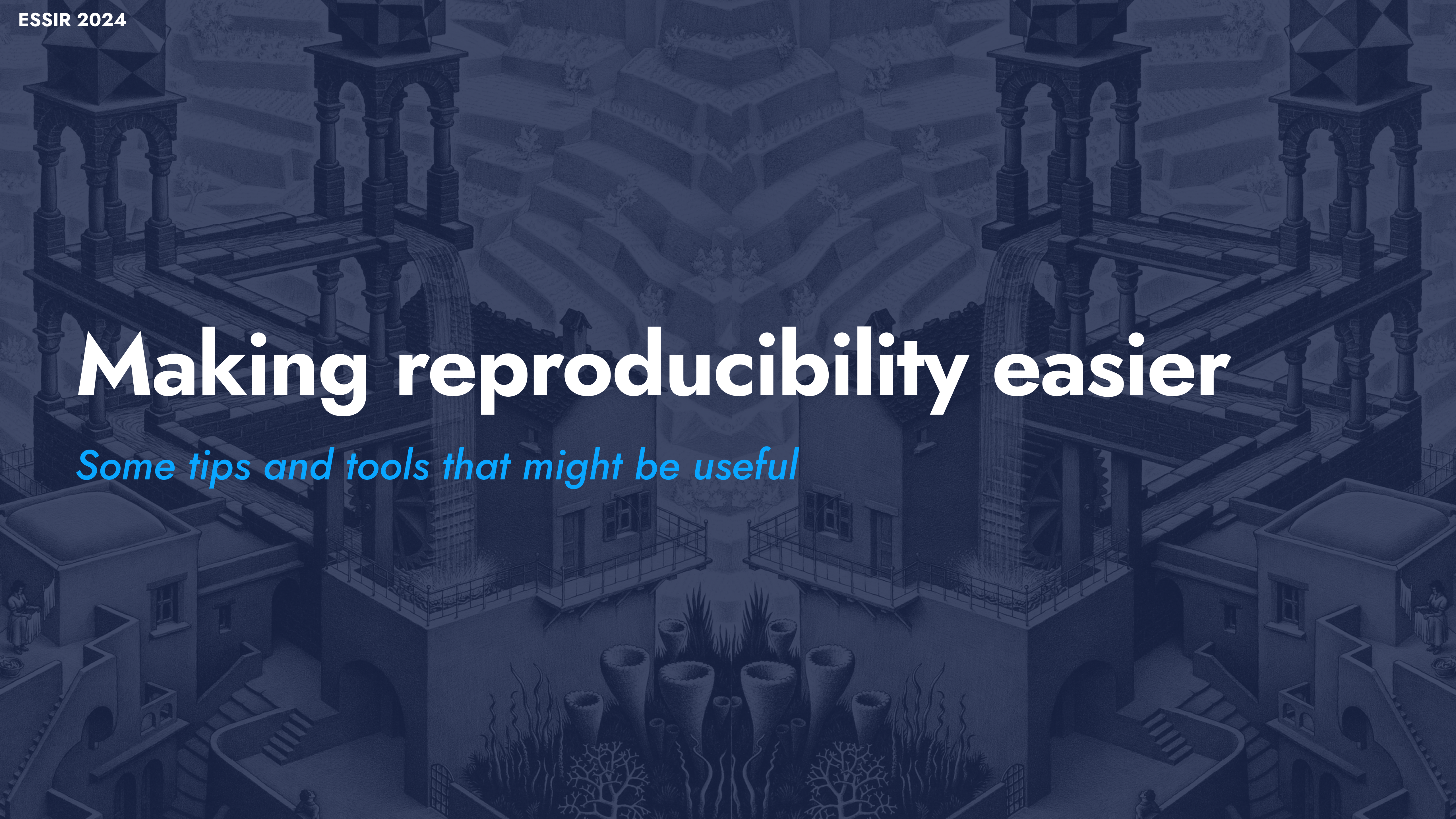
Communication

Communication: Dacrema et al. [1] contacted authors without public code. **Only 4/14 authors responded** to their inquiries within 30 days.

Providing follow-up help, clarifying questions, or answering GitHub issues are all part of enabling reproducible research.

One problem is that current **publishing structures do not incentivize follow-up support** of already published work, emphasizing the importance of reviewers [1].

[1] Ferrari Dacrema, Maurizio, et al. A troubling analysis of reproducibility and progress in recommender systems research. In TOIS 2021.



Making reproducibility easier

Some tips and tools that might be useful

Useful Tools and Libraries

- Config management
- Parameter tuning
- Managing experiments
- Data management
- Documentation
- Seeds, dependencies, ...

ML Reproducibility Tools and Best Practices

Koustuv Sinha, Jessica Zosa Forde

Aug 5, 2020 · 12 min read

A recurrent challenge in machine learning research is to ensure that the presented and published results are reliable, robust, and reproducible [4,5,6,7].

Reproducibility, obtaining similar results as presented in a paper using the same code and data, is necessary to verify the reliability of research findings. Reproducibility is also an important step to promote open and accessible research, thereby allowing the scientific community to quickly integrate new findings and convert ideas to practice. Reproducibility also promotes the use of robust experimental workflows, which potentially reduce unintentional errors.

In this blog post, we will share commonly used tools and explain 12 basic practices that you can use in your research to ensure reproducible science.

An overview from the organizers of the ML Reproducibility Challenge (MLRC) [1]

[1] Koustuv Sinha, Robert Stojnic - ML Reproducibility Tools and Best Practices: https://koustuvsinha.com//practices_for_reproducibility/

Useful Tools and Libraries

- **Config management**
- Parameter tuning
- Managing experiments
- Data management
- Documentation
- Seeds, dependencies, ...

Typical argparse at the end of a project [1]:

```
python train.py \  
  --hidden_dim 100 \  
  --batch_size 32 \  
  --num_tasks 10 \  
  --dropout 0.2 \  
  --with_mask \  
  --log_interval 100 \  
  --learning_rate 0.001 \  
  --optimizer sgd \  
  --scheduler plateau \  
  --scheduler_gamma 0.9 \  
  --weight_decay 0.9 \  
  ...
```

[1] Koustuv Sinha, Robert Stojnic - ML Reproducibility Tools and Best Practices: https://koustuvsinha.com//practices_for_reproducibility/

Useful Tools and Libraries

- **Config management**

- Parameter tuning
- Managing experiments
- Data management
- Documentation
- Seeds, dependencies, ...

Instead, store your configs in files [1]:

- JSON, YAML, CSV, etc.
- Config tools like Hydra [2] allow composing files, e.g.,: **load different params per model.**
- Some tools support **parameter tuning** and **SLURM execution** [2].

```
# config.yaml
general: # for generic args
  batch_size: 32
  num_tasks: 10
  with_mask: False
  log_interval: 100
optim: # for optimizer args
  learning_rate: 0.001
  optimizer: sgd
  scheduler: plateau
  scheduler_gamma: 0.9
  weight_decay: 0.9
model:
  hidden_dim: 100
```

[1] Koustuv Sinha, Robert Stojnic - ML Reproducibility Tools and Best Practices: https://koustuvsinha.com//practices_for_reproducibility/

[2] Hydra: <https://hydra.cc/>

Useful Tools and Libraries

- Config management
- **Parameter tuning**
- Managing experiments
- Data management
- Documentation
- Seeds, dependencies, ...

Google's Deep Learning Tuning Playbook [1]

- Start simple and make **incremental improvements**.
- First, **explore your parameter space** through random or grid search.
- Learn about scientific, nuisance, and fixed hyperparameters to know what to tune each round.
- **Maximize performance** with black-box optimizers only when you understand your parameters well (e.g., using Optuna, Nevergrad, or Ax).

[1] https://github.com/google-research/tuning_playbook?tab=readme-ov-file#a-scientific-approach-to-improving-model-performance

Useful Tools and Libraries

- Config management
- Parameter tuning
- **Managing experiments**
- Data management
- Documentation
- Seeds, dependencies, ...

Many tools can make experimentation easier, e.g. [1]:

- Track experiments (names, parameters, versions, etc.)
- Plot metrics in real-time
- Checkpoint models and data artifacts
- Integrate hyperparameter tuning libraries
- Share results with collaborators

Tools: Weights & Biases, MLFlow, Comet.ML, Neptune.ai, Aim, TensorBoard, PyTorch Lightning

Useful Tools and Libraries

- Config management
- Parameter tuning
- Managing experiments
- **Data management**
- Documentation
- Seeds, dependencies, ...

Use established libraries:

`ir_datasets` [1], HuggingFace (HF) datasets, RecBole

Publish your datasets in permanent locations:

- Your institution?
- HF datasets (up to 300GB), DVC (unlimited)

Document your datasets:

Datasheets [2], HF dataset cards, Google data cards

[1] MacAvaney, Sean, et al. Simplified data wrangling with `ir_datasets`. In SIGIR 2021.

[2] Gebru, Timnit, et al. Datasheets for datasets. Communications of the ACM 2021.

Useful Tools and Libraries

- Config management
- Parameter tuning
- Managing experiments
- Data management
- **Documentation**
- Seeds, dependencies, ...

Document your model [1]:

- Authors, license, funding
- Model architecture, training, evaluation
- Risks, limitations, biases
- Carbon emissions [3]
- Usage examples
- Citation

See [2] for a comprehensive overview of documentation tools.

[1] Mitchell, Margaret, et al. Model cards for model reporting. In FAccT 2019.

[2] <https://huggingface.co/docs/hub/model-card-landscape-analysis#summary-of-ml-documentation-tools>

[3] <https://mlco2.github.io/impact/>

Writing reproducibility papers



Those in glass houses ...

A reproducibility paper should be reproducible [1, 2, 3]:

[−] Reproduction hinting add valid flaws, but repeats similar mistakes

ML Reproducibility Challenge 2022

A review for a reproducibility paper at MLRC 2022.

Not including all code/data in a reproducibility paper is a **reason for desk rejection** at some conferences [2].

[1] SIGIR 24: https://sigir-2024.github.io/call_for_res_rep_papers.html

[2] RecSys 24: <https://recsys.acm.org/recsys24/call/#content-tab-1-1-tab>

[3] ECIR24: <https://www.ecir2024.org/2023/07/10/call-for-reproducibility-papers/>

New and important lessons

*“We are particularly interested in **reproducibility papers** (different team, different experimental setup) **rather than replicability papers** (different team, same experimental setup). The emphasis is [...] on generating new research insights with existing approaches [1].”*

Key points to consider [1, 2, 3, 4]:

- **Novelty:** Are your findings and your setup novel?
- **Generalizability:** Which lessons from prior work hold up?
- **Impact:** Are your conclusions important for the IR community?

[1] SIGIR 24: https://sigir-2024.github.io/call_for_res_rep_papers.html

[2] RecSys 24: <https://recsys.acm.org/recsys24/call/#content-tab-1-1-tab>

[3] ECIR24: <https://www.ecir2024.org/2023/07/10/call-for-reproducibility-papers/>

[4] MLRC 23: https://reproml.org/call_for_papers/

How novel?

Disclaimer: Even more personal opinion

Reproducibility papers walk a **fine line between:**

- **lack of novelty** (e.g., plainly replicating results or the findings are known)
- **and too much novelty** (e.g., proposing a new method based on your findings).

The **main novelty** of your work should be the **lessons learned** from the reproduction.

If you propose too many novel methods, **consider writing a follow-up paper.**

And if you're unsure, it might help to read a few influential reproducibility papers*.

*There are plenty linked in the introduction.

Everybody makes mistakes

Involve the original authors in the process

Ask for code, ask questions, discuss findings, send the final manuscript, and plan adequate response times (e.g., 30 days).

The golden rule

Write the paper as if somebody else writes about your work.

Hanlon's razor [1]

Never attribute to malice that which can be adequately explained by neglect, ignorance, or incompetence.*

[1] Arthur Bloch. Murphy's Law Book Two: More Reasons Why Things Go Wrong! p. 52. ISBN 9780417064505, 1980.

* the original quote just states: "[...] adequately explained by stupidity", but I think the version above is more useful.

A tradition in the IR community



Reproducibility Efforts in IR

Cranfield Collections (1958 - 1967)

The first systematic benchmarks to compare library indexing systems using topics of interest, documents, and relevance judgments [1].

Text REtrieval Conference, TREC (1992 - now)

Scaled up the Cranfield paradigm and introduced pooling: Teams submit the top-n documents for relevance judgments. Over 150 test collections for dozens of search tasks, domains, feedback ...

NII Test Collection for IR Systems, NTCIR (1997 - now)

Evaluation of IR systems with a focus on East Asian languages.

[1] Cleverdon, Cyril W. The significance of the Cranfield tests on index languages. In SIGIR 1991.

Reproducibility Efforts in IR

Conference and Labs of the **E**valuation **F**orum, **CLEF (2000 - now)**

Emphasizes multilingual and multimodal information retrieval with a set of very diverse tasks.

Forum for **I**nformation **R**etrieval **E**valuation, **FIRE (2008 - now)**

Evaluation of IR systems focusing on the Indian languages.

CLEF, NTCIR, TREC REproducibility, **CENTRE (2018 - now)**

Joint effort to reproduce the most interesting systems in previous CLEF/NTCIR/TREC editions.

Leaderboards & competitions

The crisis in Ad-Hoc retrieval between 1998 - 2008 happened despite of TREC datasets [1].
Leaderboards and competitions **unify datasets** and also **enforce baselines and evaluation procedures:**

Netflix Prize (2006 - 2009)

Netflix offered 1M dollars for outperforming their system by 10% on 100M movie ratings.
Over 5K teams ended up submitting runs, with the winners outperforming Netflix by 10.10% in RMSE.

Microsoft MAchine Reading COmprehension, MS MARCO (2016 - 2023)

Leaderboards for passage reranking, question answering, and NLP tasks.

BEenchmarking IR, BEIR (2021 - now)

Benchmark datasets for zero-shot evaluation of IR models across different domain/task combinations.

[1] Armstrong, Timothy G., et al. Improvements that don't add up: ad-hoc retrieval results since 1998. In CIKM 2009.

Conclusion



Concluding

- Reproducibility is at the **heart of scientific progress** and has a **long tradition in IR.**
- Producing truly reproducible work is much **more complex than just publishing code.**
- Select **simple and strong baselines** and **tune them with care.**
- Tools can make reproducibility easier, but ultimately, it comes down to continually striving to **publish clear, open, and detailed research in exchange with our peers.**
- When conducting reproducibility work, focus on **novelty, generalizability, and impact** of your work, and try to **involve the original authors.**

Rant over.

Thank you for listening!

Any questions?