# Automated Representation Learning from Knowledge Graph


Dr. 姚权铭

清华大学电子系 – 助理教授

[前] 第四范式 – 研究组创始组长

E-mail: qyaoaa@tsinghua.edu.cn


2021.10.10

# Outline

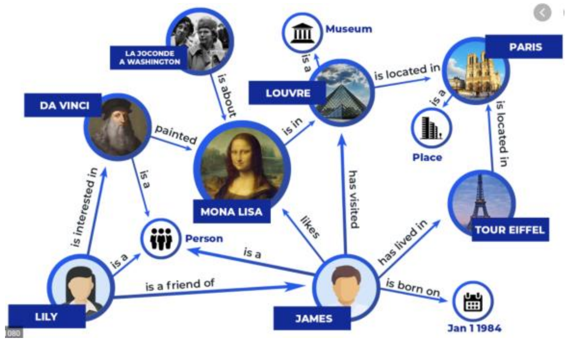
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
4. Future Works & Summary

# Knowledge Graph (KG)

A collection of interlinked descriptions of entities – objects, events or concepts

- Connect human understandings with computer computation power

Computer




KG

Human

# FROM SYSTEM 1 DEEP LEARNING TO SYSTEM 2 DEEP LEARNING

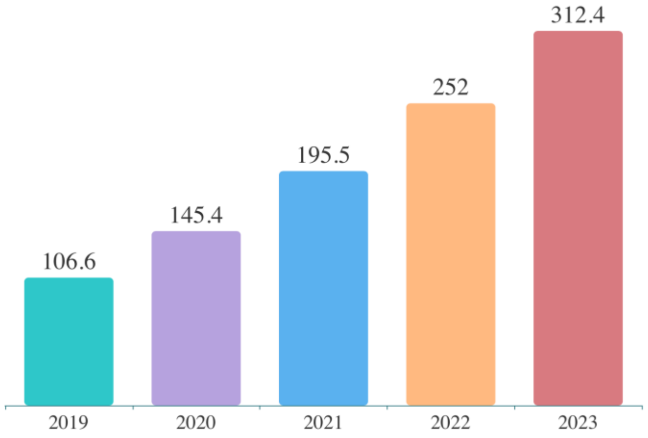
YOSHUA BENGIO

2018 Turing Award

NeurIPS’2019 Posner Lecture
December 11th, 2019, Vancouver BC

Academia: Cognitive Computing

de Montréal




Industry Market

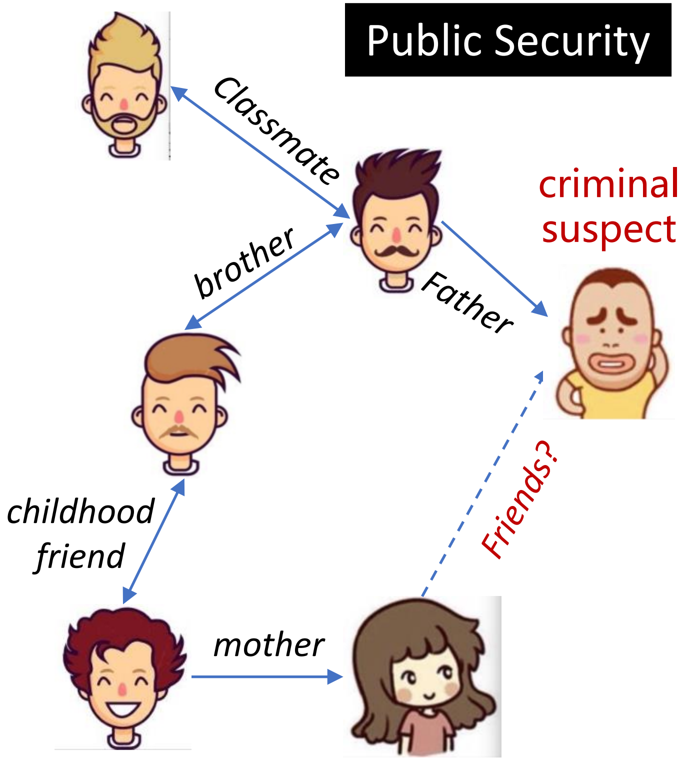
- Exceed 30 billion RMB in 2023
- Annual growth rate of 30.8%

Government Support

**1.3 认知计算基础理论与方法研究**

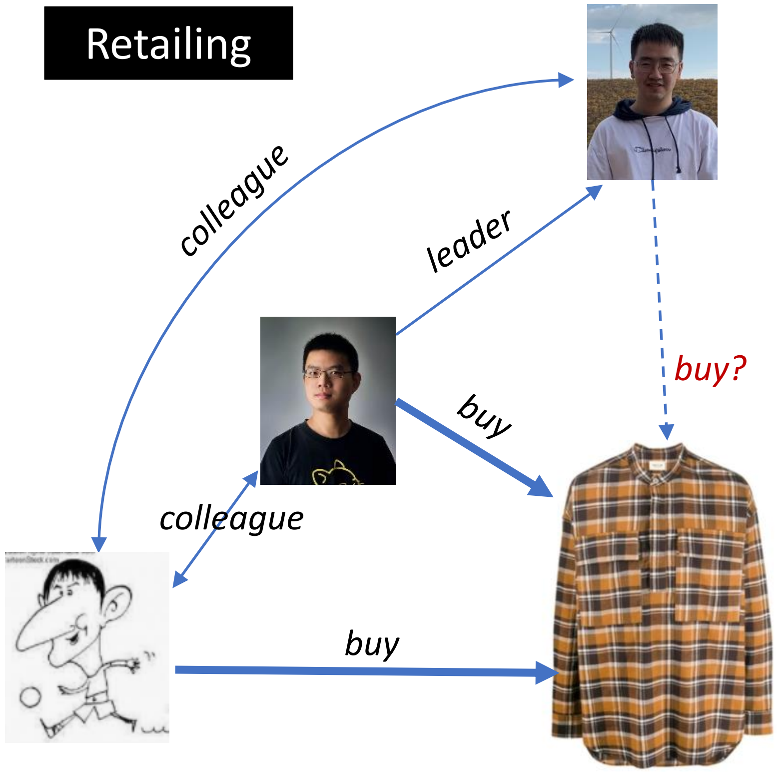
研究内容：聚焦开放、动态、真实环境下推理与决策重大问题，开展常识学习、直觉推理、自主演化、因果分析等理论和方法研究，重点突破刻画环境自适应、不完全推理、自主学习、对抗学习、智能体协同优化等特点的认知计算理论和算法，在跨媒体智能、自主智能、群体智能、人机混合或混合增强智能等智能形态方面实现应用验证。

# Know. Graph – Application examples



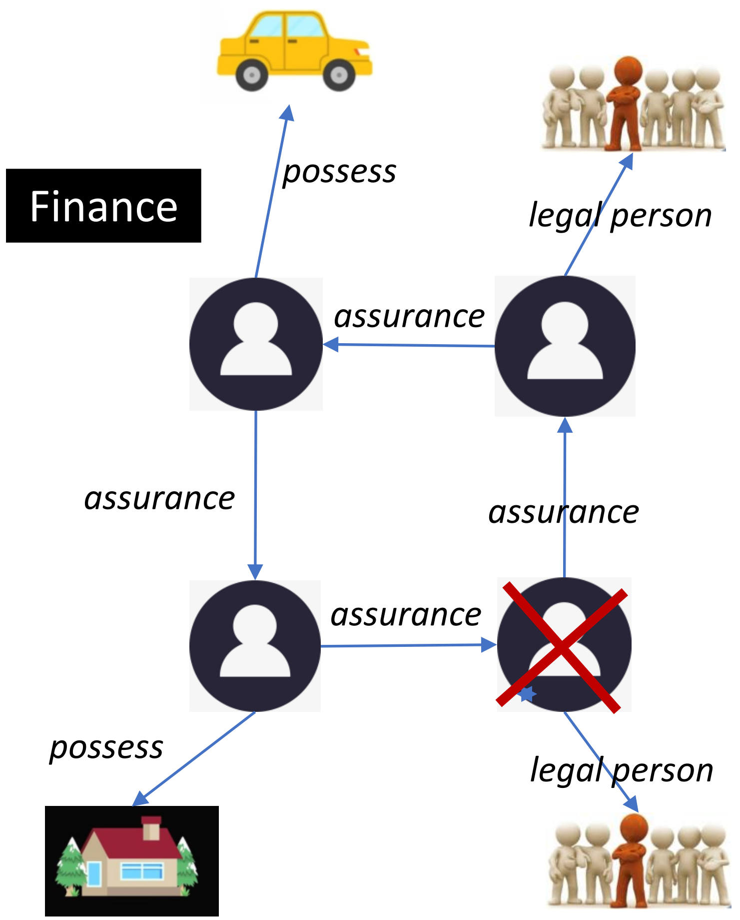

## Person of Interest

Find contact




## Recommendation

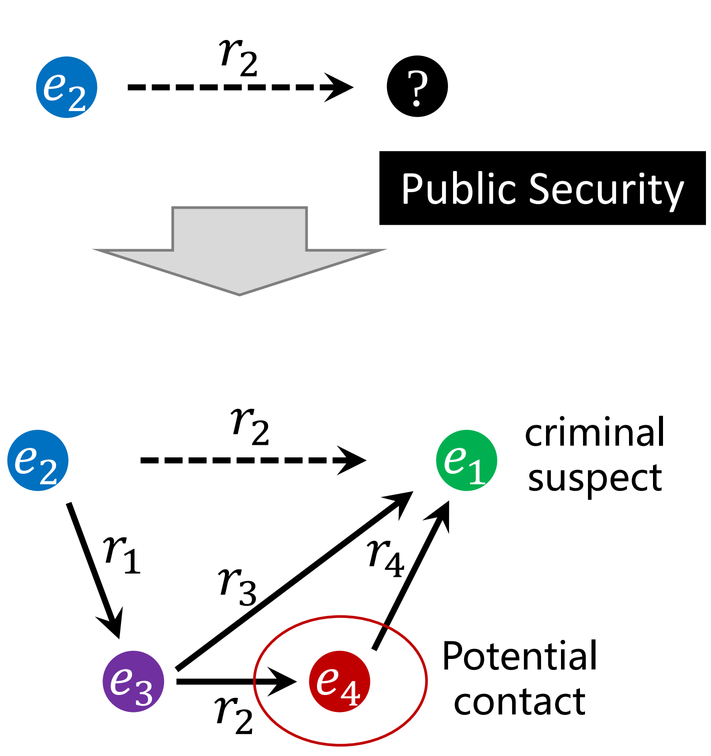
Track preference




## Bank Credits

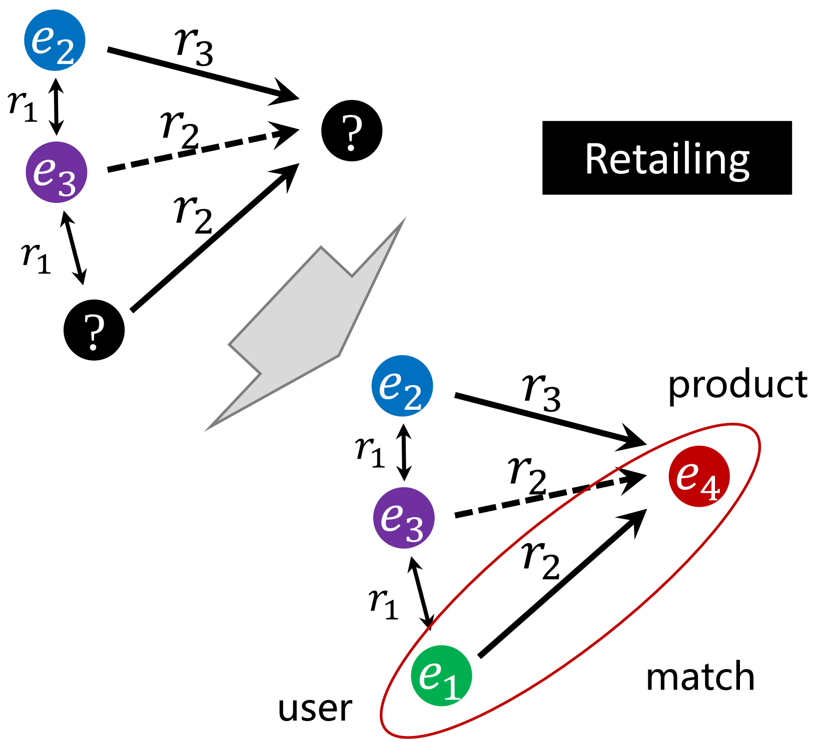
Money chain

# Know. Graph – Learning tasks



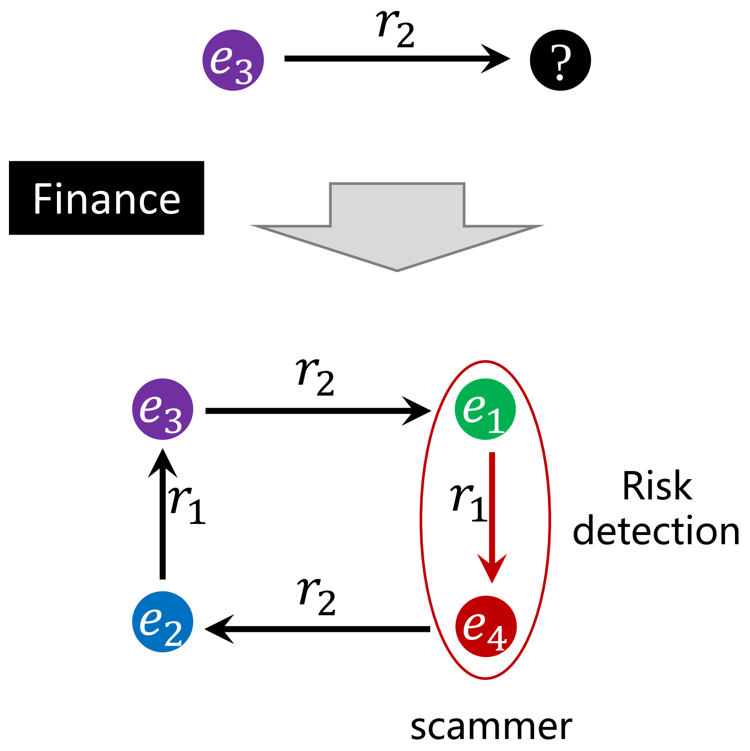

## Person of Interest

Find contact




## Recommendation

Track preference




## Bank Credits

Money chain

# Know. Graph – Core issues

Knowledge Graph = Knowledge + Graph

Semantics: Symmetric, inverse, asymmetric, composition…

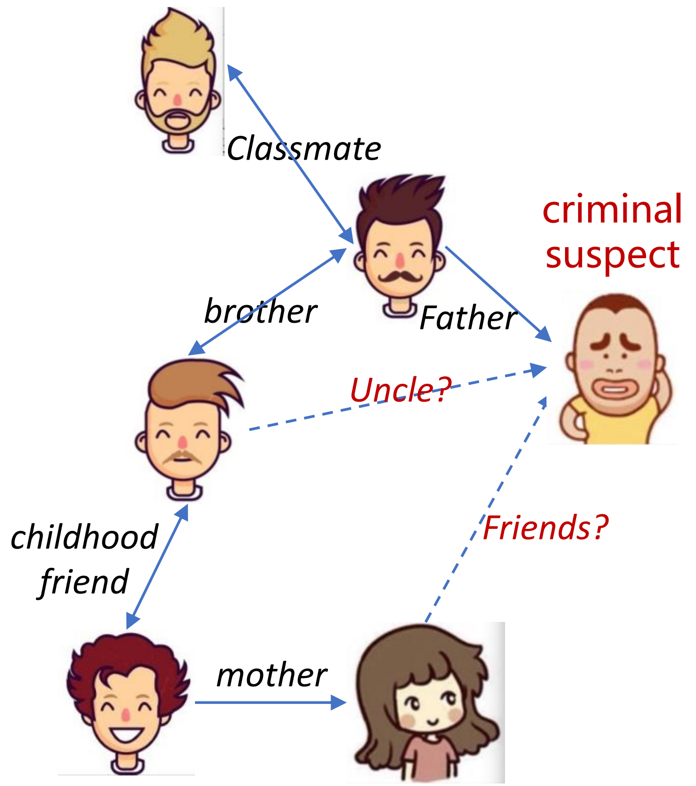
- (A, *isBrotherOf*, B) ∧ (B, *isFatherOf*, C) ⇒ (A, *isUncleOf*, C)
- (A, *spouse*, B) ⟺ (B, *spouse*, A)
- (A, *older*, B) ⟺ (B, *younger*, A)
- (A, *location*, USA)

Topology: A directed multi-relational graph

A graph-structured representation

Whole graph/subgraph as input

**How to exploit semantic and topological information?**



# Outline

1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
4. Future Works & Summary

# What is Machine Learning (ML)?

Applications

Search Engine
Recommender Systems
Loss Assessment

Image Classification

Predict the class of the object

Security Monitoring
Bio-payment
Flow Statistics

Face Recognition

Who is the person

COVID Simulation
Petroleum Exploration
Drug Discovery


Drug Design

Learn to make decisions

**Better Performance**

**Higher Efficiency**

---

Definition

Structure Samples

$$\min_x F(\, x \,;\, D \,)$$

Parameters

(iterative) optimization → Prediction Accuracy ↑

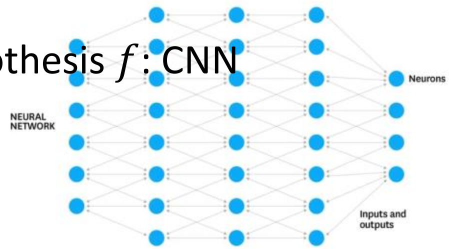
[1]. Machine Learning, Tom Mitchell, McGraw Hill, 1997.
[2]. 周志华 著. 机器学习, 北京: 清华大学出版社, 2016年

# ML = Data + Knowledge

Image Classification

Optimization ⇩

Hypothesis $f$: CNN




Generalization ⇩

Accuracy

Design a **hypothesis (function) $f$** to perform the learning task

| Target | Concept | Issue |
|---|---|---|
| Better Performance (more important) → | Generalization → | What kind of $f$ should we use? |
| Higher Efficiency → | Optimization → | How can we find such $f$? |

Not everything can be learnt

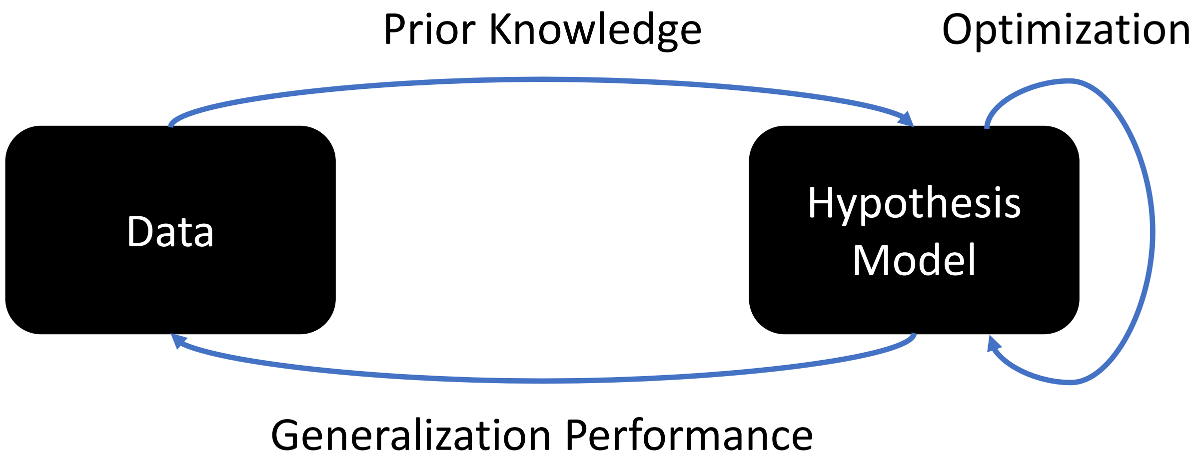
**PAC-Learning** (Definition 2.3 in [1]): What kind of problems can be solved in polynomial time

**No Free Lunch Theorem** (Appendix B [2]): No single algorithm can be good on all problems

[1]. M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of machine learning. 2018
[2]. O. Bousquet, et.al. Introduction to Statistical Learning Theory. 2016

清華大学电子工程系
Department of Electronic Engineering, Tsinghua University

# How to use ML Well?

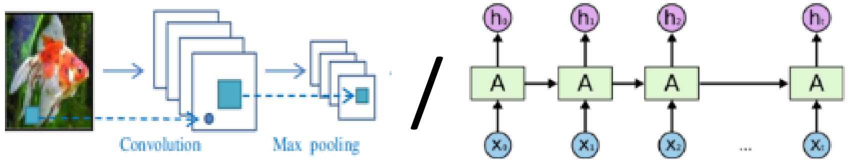



The Advancement of Learning

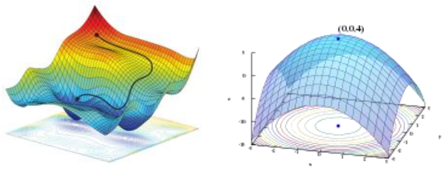
- An iteration between theory and practice

- A feedback loop

CNN RNN




Generalization: What kind of $f$ should we use?



SGD v.s. Adagrad[1]

Optimization: How can we find such $f$?

*Prior knowledge*

"All models are wrong, but some are useful"[2]

**Better understanding of prior knowledge → Better hypothesis → Better generalization performance**

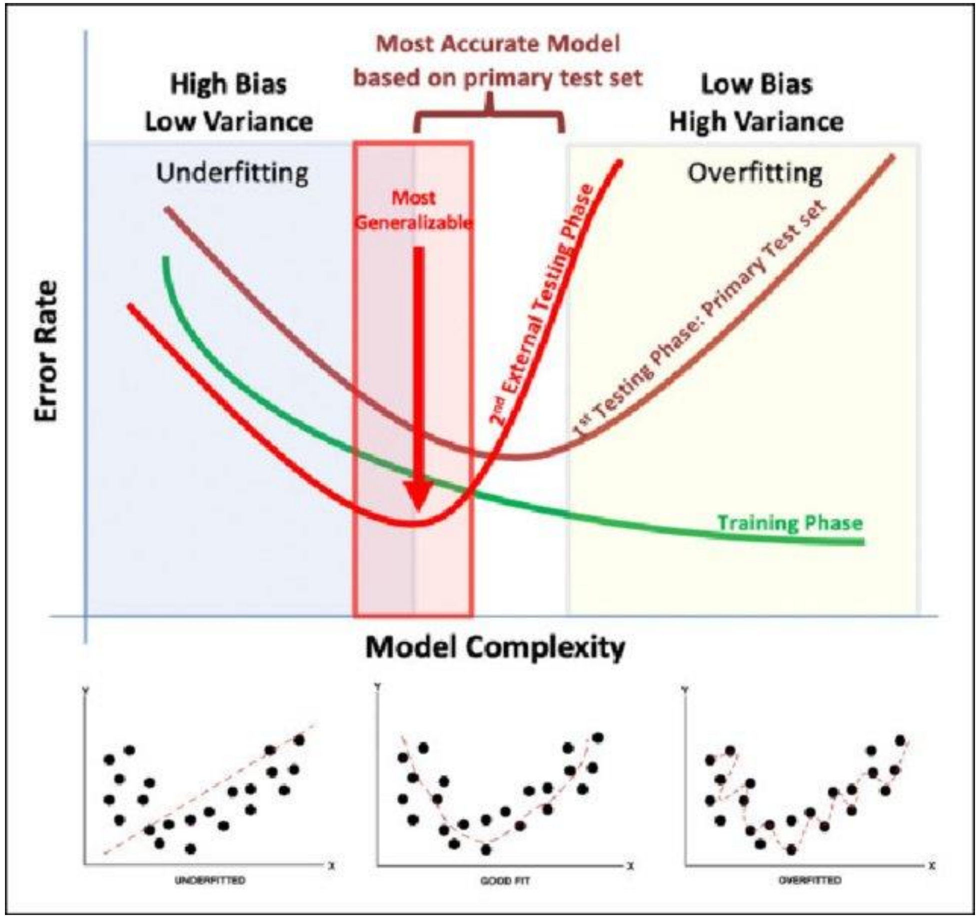
[1]. Image Source: A. Amini et al. "[Spatial Uncertainty Sampling for End-to-End Control]()". NeurIPS Bayesian Deep Learning 2018
[2] G. Box, Science and statistics, JASA 1976

# Simple AutoML Example – Tune hyper-parameter

Bi-level optimization

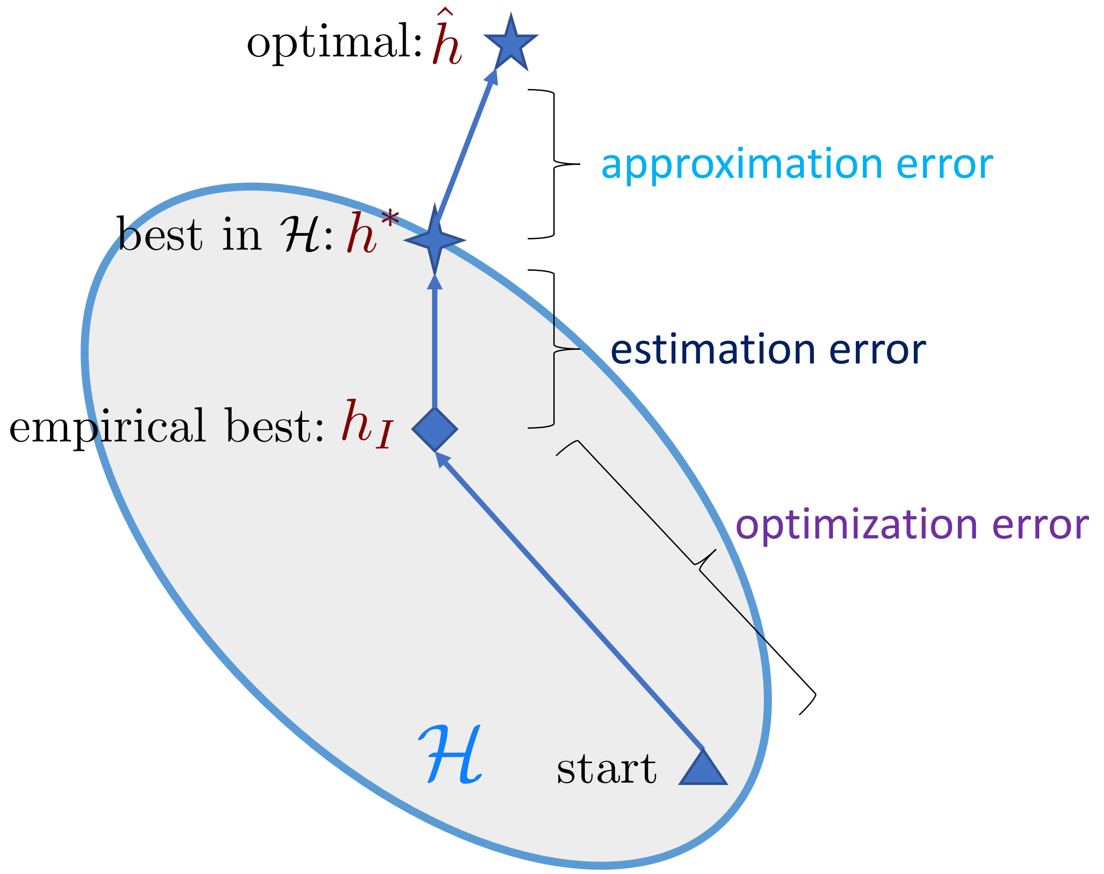
$$\underbrace{\max_{\lambda} \sum_j h(x_j; w^*)}_{\text{Validation Performance}} \quad \text{s.t.} \quad w^* = \underbrace{\min_{w} \sum_i f(x_i; w) + \overset{\text{Hyper-parameter}}{\lambda} \|w\|_1}_{\text{Training objective}}$$

- Large $\lambda$ leads to sparse $w^*$
- Grid search: enumerating $\lambda \in \{1,2,4,8,\dots\}$




[1]. Image source: Artificial Intelligence and Machine Learning in Pathology: The Present Landscape of Supervised Methods.

# Behind Hyper-para. – Error decomposition




Total error in machine learning

- Approximation error
  - Which classifier to be used
  - What are their hyper-parameters
  - Distribution changes
- Estimation error
  - Finite samples
  - Regularization hyper-parameter

Reduce

$$\min_w \sum_i f(x_i; w) + \lambda \|w\|_1$$

- Optimization error
  - Which algorithm to be used
  - How to tune its step-size

Image is from Y. Wang, et. al. Generalizing from a Few Examples: A Survey on Few-Shot Learning. CSUR 2020

# Look Inside Error Decomposition

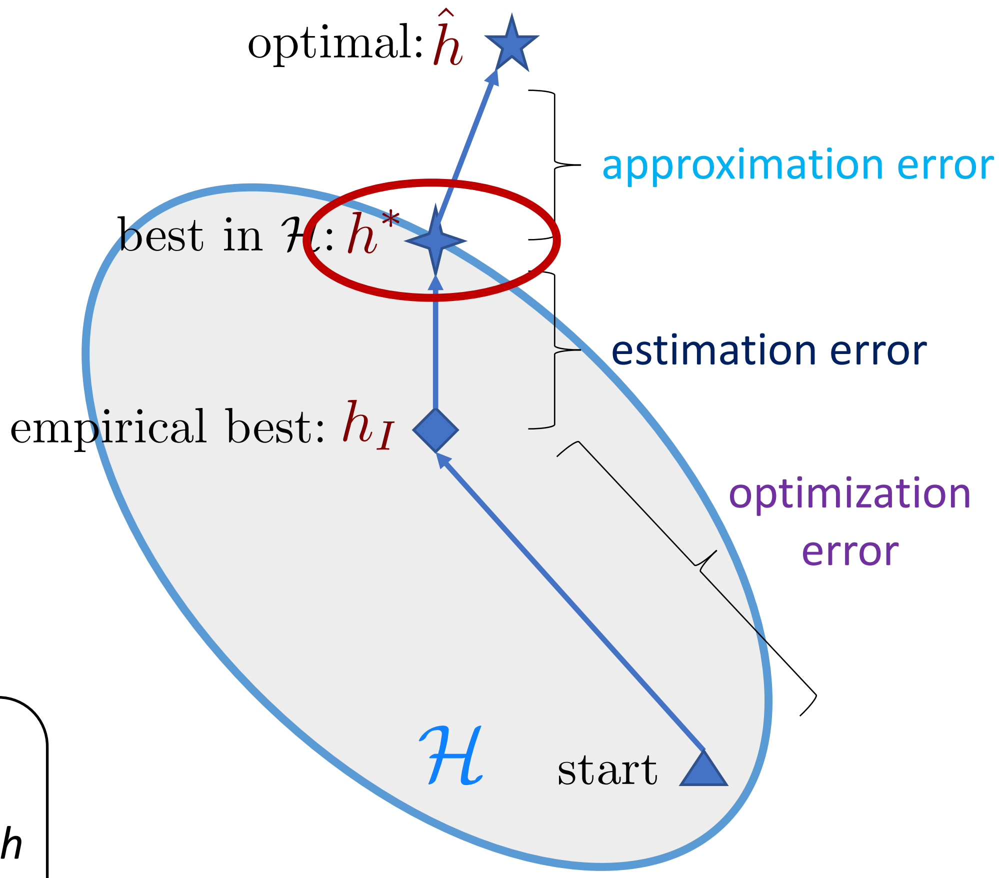
Automatically find $h^*$ by bi-level optimization

$$\underbrace{\max_{\lambda} \sum_j h(x_j; w^*)}_{\text{Validation Performance}} \quad \text{s.t.} \quad w^* = \underbrace{\min_{w} \sum_i f(x_i; w) + \lambda \|w\|_1}_{\text{Training objective}}$$

How to further improve the performance in an automatic manner (i.e., reduce the approximation error)?

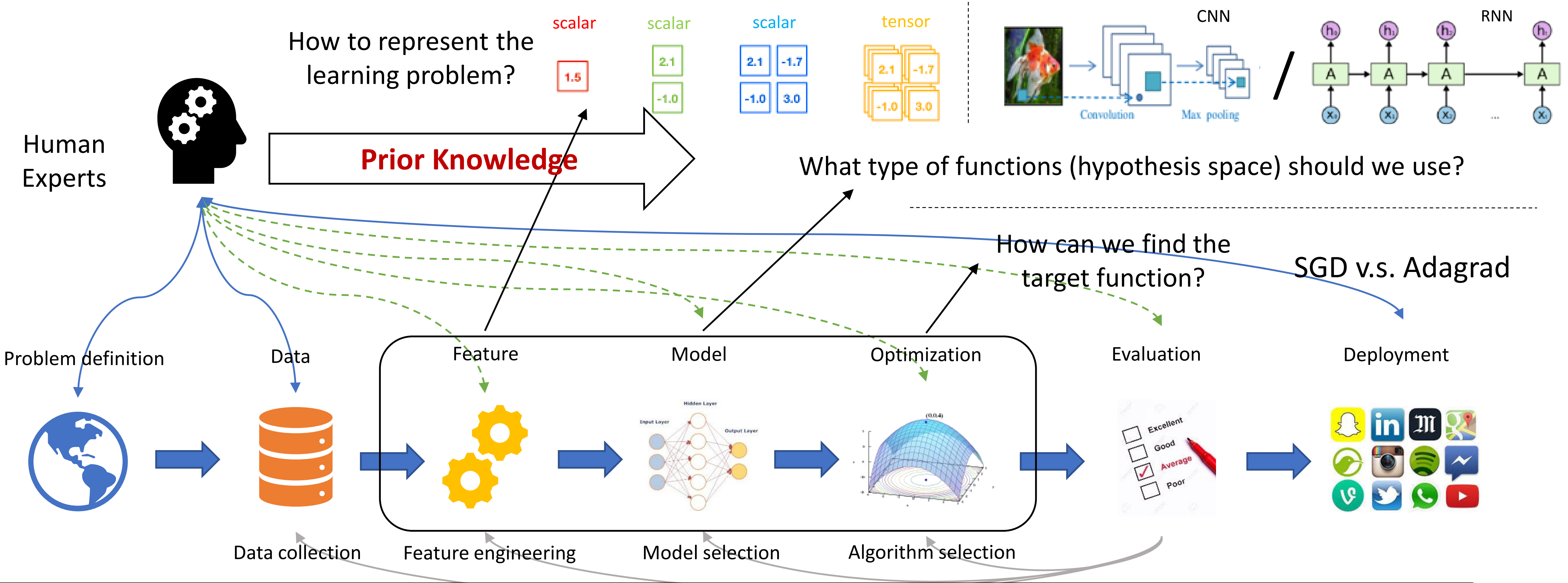
- Feature can be weak → *Automatic feature engineering*
- Linear predictor can be too restrictive → *Neural architecture search*
- Grid search can be slow → *Search in a supernet*

AutoML



清華大学电子工程系
Department of Electronic Engineering, Tsinghua University

# What is AutoML – Practical Viewpoint




**Parameterize (low-level) prior knowledge in the usage and design of machine learning**

As a consequence
- Human participations can be naturally replaced by computation power
- total error of machine learning can be reduced (generalization can be improved)

Figure 4.1 & 4.2. M. Mohri, A. Rostamizadeh, A. Talwalkar. Foundations of machine learning. 2018
[1]. P. Battaglia. et.al. Relational inductive biases, deep learning, and graph networks. arXiv 2018.

清華大学电子工程系
Department of Electronic Engineering, Tsinghua University

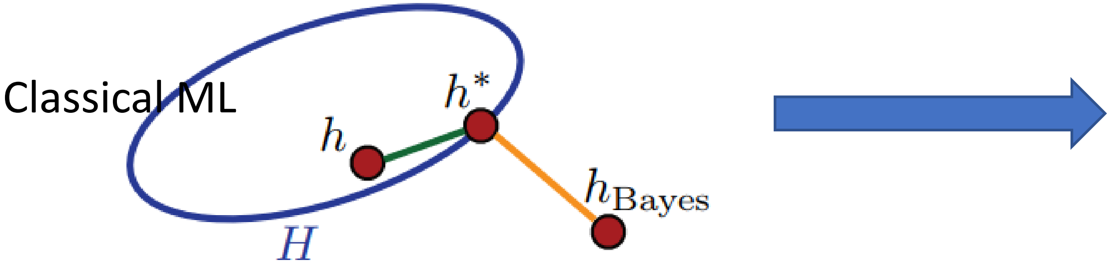
# What is AutoML – Generalization Viewpoint

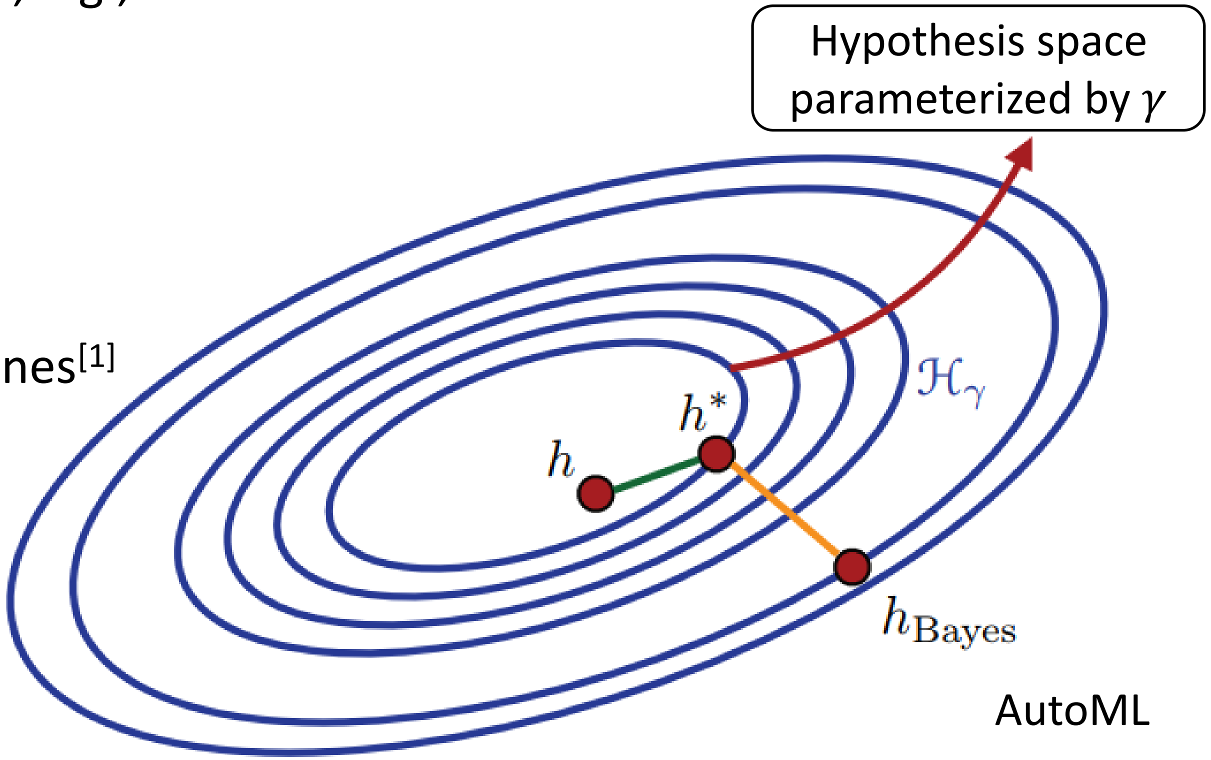
Parameterized the prior knowledge of learning methods, e.g.,

- minimize the total error
- reduce parameter numbers

Perform efficient search in the designed (new) space

- combinatorial generalize new models from existing ones[1]



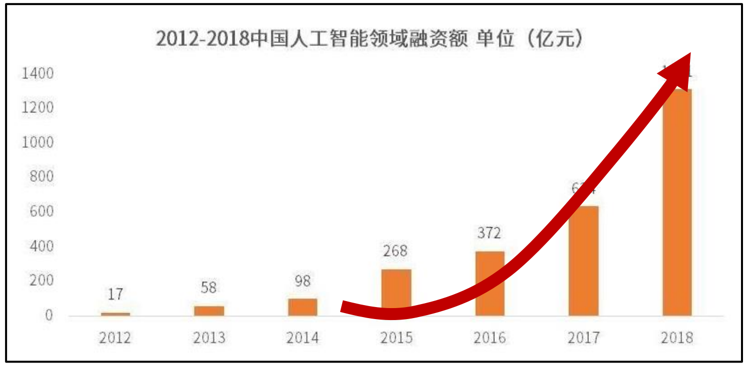




**Parameterize (low-level) prior knowledge in the usage and design of machine learning**

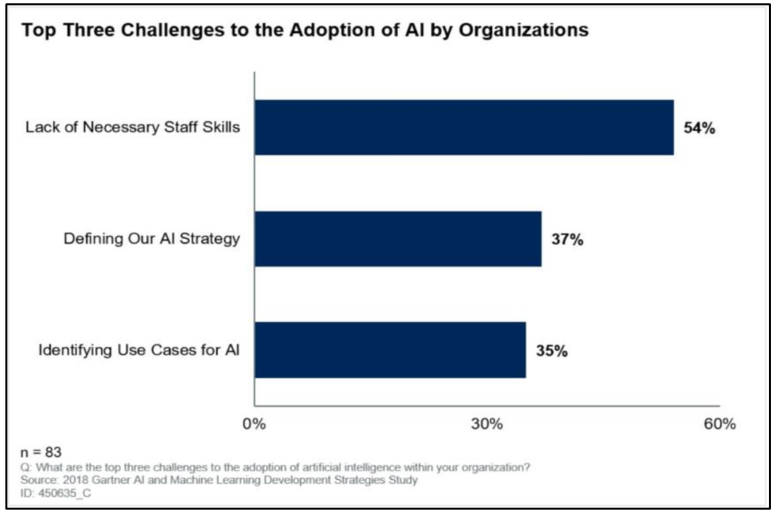
As a consequence

- Human participations can be naturally replaced by computation power
- total error of machine learning can be reduced (generalization can be improved)

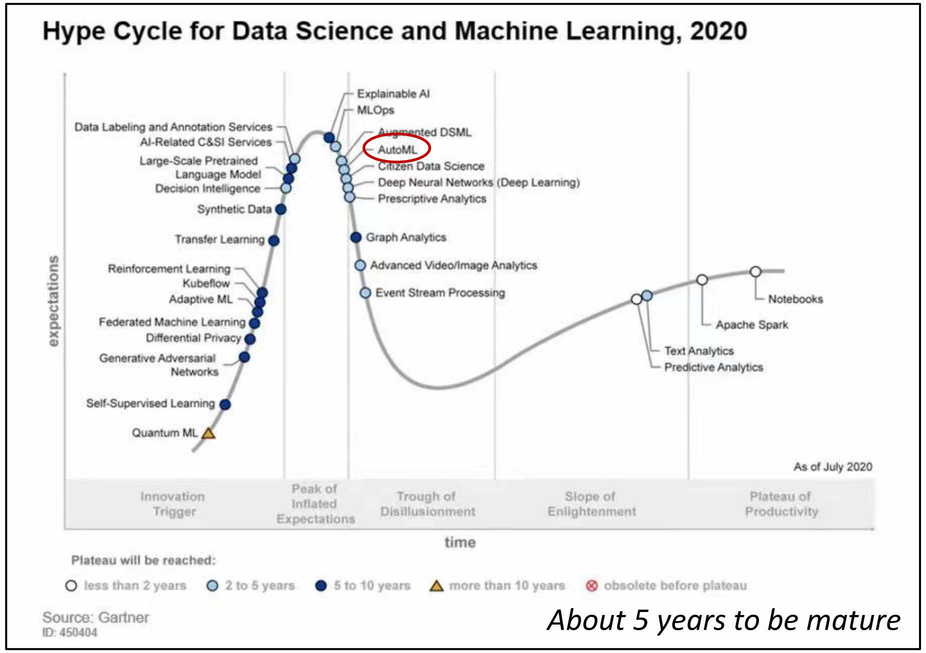
# Why We need AutoML?




Investment in AI industry




Practical needs




Technical trends

- **Industry** – reduce the expense, increase usage coverage – huge market value [1]
- **Academy** – understanding data science on a higher level – great intelligence value [2,3]

[1]. Gartner: https://www.forbes.com/sites/janakirammsv/2020/03/02/key-takeaways-from-the-gartner-magic-quadrant-for-ai-developer-services/#a95b99ee3e5e
[2]. Y. Bengio: From System 1 Deep Learning to System 2 Deep Learning | NeurIPS 2019
[3]. F Hutter, L Kotthoff, J Vanschoren. Automated machine learning: methods, systems, challenges. Book 2019

# Related Areas

## Sub-areas

- Neural architecture search
- Hyper-parameter search
- Automated feature engineering
- Algorithms selection
- Model selection

## Related areas

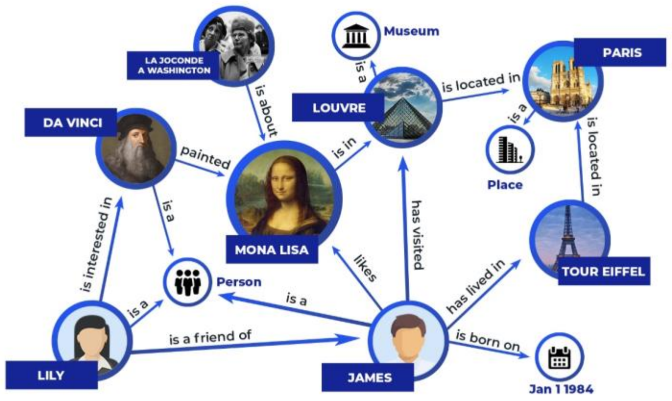
- Bi-level / Derivative-free optimization
  - Focus more on algorithm design
  - AutoML objective is one kind of objective where these algorithms can be applied
- Meta-learning
  - Focus on parameterize task distributions
  - Another kind of bi-level objective
  - Do not use validation set to update hyper-parameters

# Outline

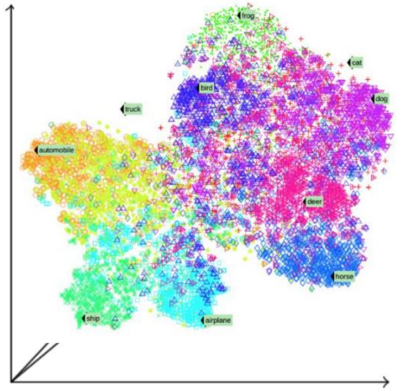
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
    - Overview of Ideas
    - Search to Capture Semantics
    - Search to Exploit Graph Topology
4. Future Works & Summary

# KG Representation Learning

Encode entities and relations in KG into low-dimensional vector spaces, while capturing nodes' and edges' connection & semantic properties.




input



output

Advantages:

- Inject into downstream ML pipelines.
- Provide efficient similarity search.
- Discover latent properties in missing links.

Scoring functions (SFs) $f(\mathbf{h}, \mathbf{r}, \mathbf{t})$:

- measure the plausibility of triplets $\{(h, r, t)\}$ in KG.

Observed triplet $S^+$: increase score

Unobserved triplet $S^-$: decrease score

# Our Work – Overview

Using AutoML techniques

- Explore **core issue** in data-specific / task-aware manner

Knowledge Graph = Knowledge + Graph

Semantic information (Knowledge)

Topological information (Graph)

Progress of our works

- ICDE 2020: Capture semantic information in relation on triplet level
- NeurIPS 2020: Capture topological information in path level

*Covered this time*

- ICDE 2021: Capture topological information in graph level
- KDD 2021: Capture semantic and topological information in graph level

# Our Work – Best on KG in OGB

## Open Graph Benchmark

**Benchmark datasets, data loaders and evaluators for graph machine learning**

GET STARTED | KDD CUP 2021 (NEW!)

### Leaderboard for ogbl-biokg

April 2021

The MRR score on the test and validation sets. The higher, the better.

Package: >=1.2.0

| Rank | Method | Test MRR | Validation MRR | Contact | References | #Params | Hardware | Date |
|---|---|---|---|---|---|---|---|---|
| 1 | **AutoSF** | 0.8309 ± 0.0008 | 0.8317 ± 0.0007 | Yongqi Zhang (4Paradigm) | Paper, Code | 93,824,000 | GeForce RTX 2080 (11GB GPU) | Apr 2, 2021 |
| 2 | **PairRE** | 0.8164 ± 0.0005 | 0.8172 ± 0.0005 | LinlinChao (AntGroup KG&NLP) | Paper, Code | 187,750,000 | Tesla P100 (16GB GPU) | Nov 9, 2020 |
| 3 | ComplEx | 0.8095 ± 0.0007 | 0.8105 ± 0.0001 | Hongyu Ren – OGB team | Paper, Code | 187,648,000 | GeForce RTX 2080 (11GB GPU) | Jun 10, 2020 |
| 4 | DistMult | 0.8043 ± 0.0003 | 0.8055 ± 0.0003 | Hongyu Ren – OGB team | Paper, Code | 187,648,000 | GeForce RTX 2080 (11GB GPU) | Jun 10, 2020 |
| 5 | RotatE | 0.7989 ± 0.0004 | 0.7997 ± 0.0002 | Hongyu Ren – OGB team | Paper, Code | 187,597,000 | GeForce RTX 2080 (11GB GPU) | Jun 10, 2020 |
| 6 | TransE | 0.7452 ± 0.0004 | 0.7456 ± 0.0003 | Hongyu Ren – OGB team | Paper, Code | 187,648,000 | GeForce RTX 2080 (11GB GPU) | Jun 10, 2020 |

### Leaderboard for ogbl-wikikg2

April 2021

The MRR score on the test and validation sets. The higher, the better.

Package: >=1.2.4

Deprecated `ogbl-wikikg` leaderboard can be found here.

| Rank | Method | Test MRR | Validation MRR | Contact | References | #Params | Hardware | Date |
|---|---|---|---|---|---|---|---|---|
| 1 | **AutoSF** | 0.5458 ± 0.0052 | 0.5510 ± 0.0063 | Yongqi Zhang (4Paradigm) | Paper, Code | 500,227,800 | Quadro RTX 8000 (45GB GPU) | Apr 2, 2021 |
| 2 | **PairRE (200dim)** | 0.5208 ± 0.0027 | 0.5423 ± 0.0020 | Linlin Chao | Paper, Code | 500,334,800 | Tesla P100 (16GB GPU) | Jan 28, 2021 |
| 3 | RotatE (250dim) | 0.4332 ± 0.0025 | 0.4353 ± 0.0028 | Hongyu Ren – OGB team | Paper, Code | 1,250,435,750 | Quadro RTX 8000 (45GB GPU) | Jan 23, 2021 |
| 4 | TransE (500dim) | 0.4256 ± 0.0030 | 0.4272 ± 0.0030 | Hongyu Ren – OGB team | Paper, Code | 1,250,569,500 | Quadro RTX 8000 (45GB GPU) | Jan 23, 2021 |
| 5 | ComplEx (250dim) | 0.4027 ± 0.0027 | 0.3759 ± 0.0016 | Hongyu Ren – OGB team | Paper, Code | 1,250,569,500 | Quadro RTX 8000 (45GB GPU) | Jan 23, 2021 |

Solution is available at: https://github.com/AutoML-Research/AutoSF

- Based on ICDE 2020 / NeurIPS 2020

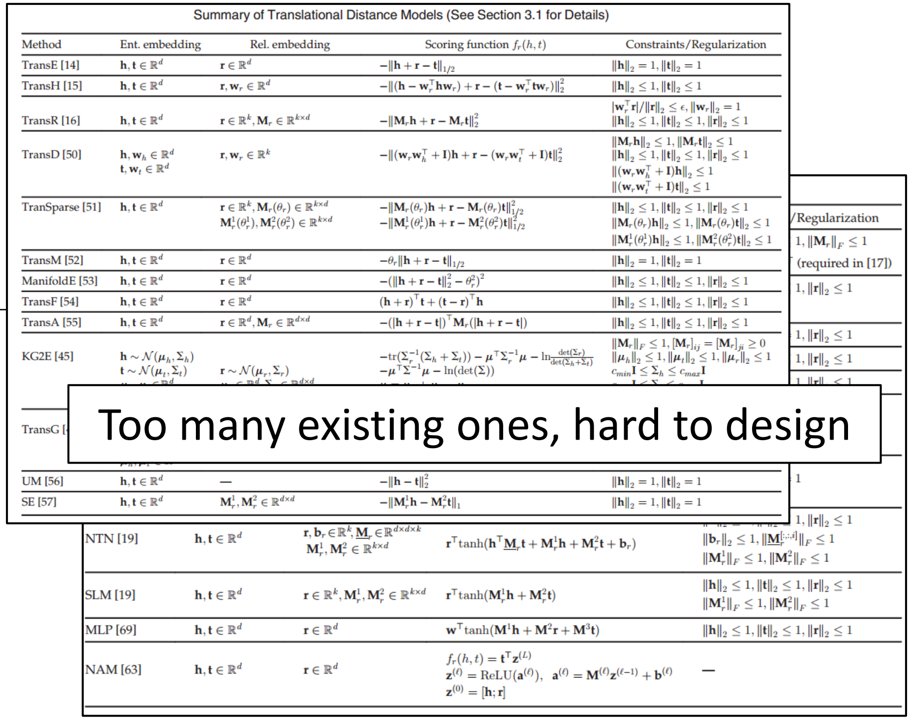
# Outline

1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
   - Overview of Ideas
   - Search to Capture Semantics
   - Search to Exploit Graph Topology
4. Future Works & Summary

# Scoring Function (SF) – Example

Design principles

- Encode entity and relation into some space to measure the plausibility.
- Capture important semantic properties:
  - symmetric, anti-symmetric, inverse, asymmetric…



Summary of Translational Distance Models (See Section 3.1 for Details)

| Method | Ent. embedding | Rel. embedding | Scoring function $f_r(h,t)$ | Constraints/Regularization |
|---|---|---|---|---|
| TransE [14] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $-\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_{1/2}$ | $\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$ |
| TransH [15] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$ | $-\lVert(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1$<br>$\lvert\mathbf{w}_r^\top\mathbf{r}\rvert/\lVert\mathbf{r}\rVert_2\le\epsilon, \lVert\mathbf{w}_r\rVert_2=1$ |
| TransR [16] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$ | $-\lVert\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$<br>$\lVert\mathbf{M}_r\mathbf{h}\rVert_2\le1, \lVert\mathbf{M}_r\mathbf{t}\rVert_2\le1$ |
| TransD [50] | $\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$<br>$\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$ | $\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$ | $-\lVert(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$<br>$\lVert(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\rVert_2\le1$<br>$\lVert(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\rVert_2\le1$ |
| TranSparse [51] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$<br>$\mathbf{M}_r^1(\theta_r^1), \mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$ | $-\lVert\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\rVert_{1/2}^2$<br>$-\lVert\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\rVert_{1/2}^2$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$<br>$\lVert\mathbf{M}_r(\theta_r)\mathbf{h}\rVert_2\le1, \lVert\mathbf{M}_r(\theta_r)\mathbf{t}\rVert_2\le1$<br>$\lVert\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\rVert_2\le1, \lVert\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\rVert_2\le1$ |
| TransM [52] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $-\theta_r\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_{1/2}$ | $\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$ |
| ManifoldE [53] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $-(\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_2^2-\theta_r^2)^2$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$ |
| TransF [54] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$ |
| TransA [55] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$ | $-(\lvert\mathbf{h}+\mathbf{r}-\mathbf{t}\rvert)^\top\mathbf{M}_r(\lvert\mathbf{h}+\mathbf{r}-\mathbf{t}\rvert)$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$<br>$\lVert\mathbf{M}_r\rVert_F\le1, [\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge0$ |
| KG2E [45] | $\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$<br>$\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ | $\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$ | $-\mathrm{tr}(\boldsymbol{\Sigma}_r^{-1}(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t))-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\boldsymbol{\Sigma}_r)}{\det(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t)}$<br>$-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\ln(\det(\boldsymbol{\Sigma}))$ | $\lVert\boldsymbol{\mu}_h\rVert_2\le1, \lVert\boldsymbol{\mu}_t\rVert_2\le1, \lVert\boldsymbol{\mu}_r\rVert_2\le1$<br>$c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_h\le c_{max}\mathbf{I}$ |
| TransG [ | | | | |
| UM [56] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | — | $-\lVert\mathbf{h}-\mathbf{t}\rVert_2^2$ | $\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$ |
| SE [57] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{d\times d}$ | $-\lVert\mathbf{M}_r^1\mathbf{h}-\mathbf{M}_r^2\mathbf{t}\rVert_1$ | $\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$ |

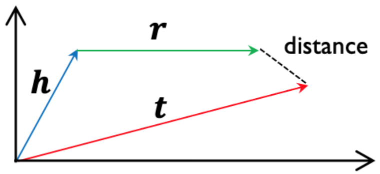
| | | | | /Regularization |
|---|---|---|---|---|
| NTN [19] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r},\mathbf{b}_r\in\mathbb{R}^k, \underline{\mathbf{M}}_r\in\mathbb{R}^{d\times d\times k}$<br>$\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$ | $\mathbf{r}^\top\tanh(\mathbf{h}^\top\underline{\mathbf{M}}_r\mathbf{t}+\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t}+\mathbf{b}_r)$ | $\lVert\mathbf{b}_r\rVert_2\le1, \lVert\underline{\mathbf{M}}_r^{[:,:,i]}\rVert_F\le1$<br>$\lVert\mathbf{M}_r^1\rVert_F\le1, \lVert\mathbf{M}_r^2\rVert_F\le1$ |
| SLM [19] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{k\times d}$ | $\mathbf{r}^\top\tanh(\mathbf{M}_r^1\mathbf{h}+\mathbf{M}_r^2\mathbf{t})$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$<br>$\lVert\mathbf{M}_r^1\rVert_F\le1, \lVert\mathbf{M}_r^2\rVert_F\le1$ |
| MLP [69] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $\mathbf{w}^\top\tanh(\mathbf{M}^1\mathbf{h}+\mathbf{M}^2\mathbf{r}+\mathbf{M}^3\mathbf{t})$ | $\lVert\mathbf{h}\rVert_2\le1, \lVert\mathbf{t}\rVert_2\le1, \lVert\mathbf{r}\rVert_2\le1$ |
| NAM [63] | $\mathbf{h},\mathbf{t}\in\mathbb{R}^d$ | $\mathbf{r}\in\mathbb{R}^d$ | $f_r(h,t)=\mathbf{t}^\top\mathbf{z}^{(L)}$<br>$\mathbf{z}^{(\ell)}=\mathrm{ReLU}(\mathbf{a}^{(\ell)}), \ \mathbf{a}^{(\ell)}=\mathbf{M}^{(\ell)}\mathbf{z}^{(\ell-1)}+\mathbf{b}^{(\ell)}$<br>$\mathbf{z}^{(0)}=[\mathbf{h};\mathbf{r}]$ | — |

Too many existing ones, hard to design

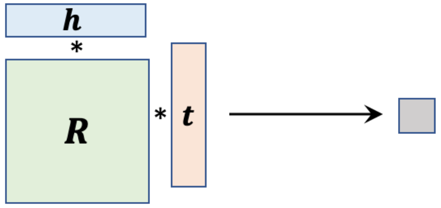
Wang et.al. Knowledge graph embedding: A survey of approaches and applications. TKDE 2017

Examples:

1. Translation Distance Models (TDMs)
   - TransE, TransH, RotatE, etc
   - less expressive

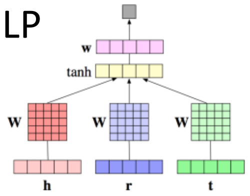



2. BiLinear Models (BLMs)
   - DistMult, ComplEx, Analogy, SimplE, etc
   - state-of-the-art and fully expressive

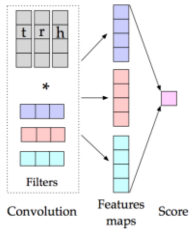



3. Neural Network Models (NNMs)
   - MLP, ConvE, RSN, etc
   - complex and difficult to train

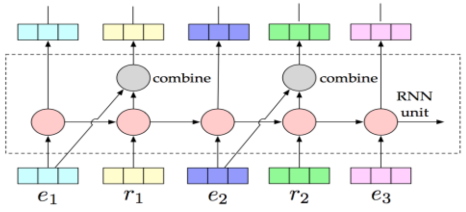
MLP




ConvE




RSN



# Contribution – Search to capture semantics

1. There is no absolute winner among them since KGs exhibit distinct patterns. Even the fully expressive models do not definitely perform the best
2. KG is sparse, thus regularization (i.e., prior on semantics) is important
3. Designing novel and universal SFs becomes harder

Our solutions:

- Adaptively search to regularize the BLMs for different KG tasks
- Design novel and task-aware scoring functions

AutoSF: Searching Scoring Functions for Knowledge Graph Embedding. ICDE 2020

DistMult: Embedding entities and relations for learning and inference in knowledge bases. ICLR 2015
ComplEx: Knowledge graph completion via complex tensor factorization. JMLR 2017
Analogy: Analogical inference for multirelational embeddings. ICML 2018
SimplE embedding for link prediction in knowledge graphs. NeurIPS 2018

# Revisit Bilinear SFs

The BLMs can be written as $f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \mathbf{h}^{\mathrm{T}}\mathbf{R}\mathbf{t}$, with different form of $\mathbf{R}$, a square matrix of $\mathbf{r}$

For unified representation, we evenly split the embedding into 4 parts, e.g. $\boldsymbol{r} = [\boldsymbol{r}_1; \boldsymbol{r}_2; \boldsymbol{r}_3; \boldsymbol{r}_4]$

Denote $\mathbf{D}_i^r = \mathrm{diag}(\boldsymbol{r}_i)$ as the corresponding diagonal matrix

---

DistMult: $f(h, r, t) = \langle \mathbf{h}, \mathbf{r}, \mathbf{t} \rangle$

$$\begin{array}{|c|c|c|c|} \hline \mathbf{D}_1^r & & & \\ \hline & \mathbf{D}_2^r & & \\ \hline & & \mathbf{D}_3^r & \\ \hline & & & \mathbf{D}_4^r \\ \hline \end{array}$$

| | |
|---|---|
| symmetric | √ |
| anti-symmetric | × |
| asymmetric | × |
| inverse | × |

ComplEx: $f(h, r, t) = \mathrm{Re}(\langle \mathbf{h}, \mathbf{r}, \mathrm{conj}(\mathbf{t}) \rangle)$

$$\begin{array}{|c|c|c|c|} \hline \mathbf{D}_1^r & & \mathbf{D}_3^r & \\ \hline & \mathbf{D}_2^r & & \mathbf{D}_4^r \\ \hline -\mathbf{D}_3^r & & \mathbf{D}_1^r & \\ \hline & -\mathbf{D}_4^r & & \mathbf{D}_2^r \\ \hline \end{array}$$

| | |
|---|---|
| symmetric | √ |
| anti-symmetric | √ |
| asymmetric | √ |
| inverse | √ |

Analogy: $f(h, r, t) = \langle \hat{\mathbf{h}}, \hat{\mathbf{r}}, \hat{\mathbf{t}} \rangle + \mathrm{Re}\left(\langle \check{\mathbf{h}}, \check{\mathbf{r}}, \mathrm{conj}(\check{\mathbf{t}}) \rangle\right)$

$$\begin{array}{|c|c|c|c|} \hline \mathbf{D}_1^r & & & \\ \hline & \mathbf{D}_2^r & & \\ \hline & & \mathbf{D}_3^r & \mathbf{D}_4^r \\ \hline & & -\mathbf{D}_4^r & \mathbf{D}_3^r \\ \hline \end{array}$$

| | |
|---|---|
| symmetric | √ |
| anti-symmetric | √ |
| asymmetric | √ |
| inverse | √ |

SimplE: $f(h, r, t) = \langle \hat{\mathbf{h}}, \hat{\mathbf{r}}, \check{\mathbf{t}} \rangle + \langle \check{\mathbf{h}}, \check{\mathbf{r}}, \hat{\mathbf{t}} \rangle$

$$\begin{array}{|c|c|c|c|} \hline & & \mathbf{D}_1^r & \\ \hline & & & \mathbf{D}_2^r \\ \hline \mathbf{D}_3^r & & & \\ \hline & \mathbf{D}_4^r & & \\ \hline \end{array}$$

| | |
|---|---|
| symmetric | √ |
| anti-symmetric | √ |
| asymmetric | √ |
| inverse | √ |

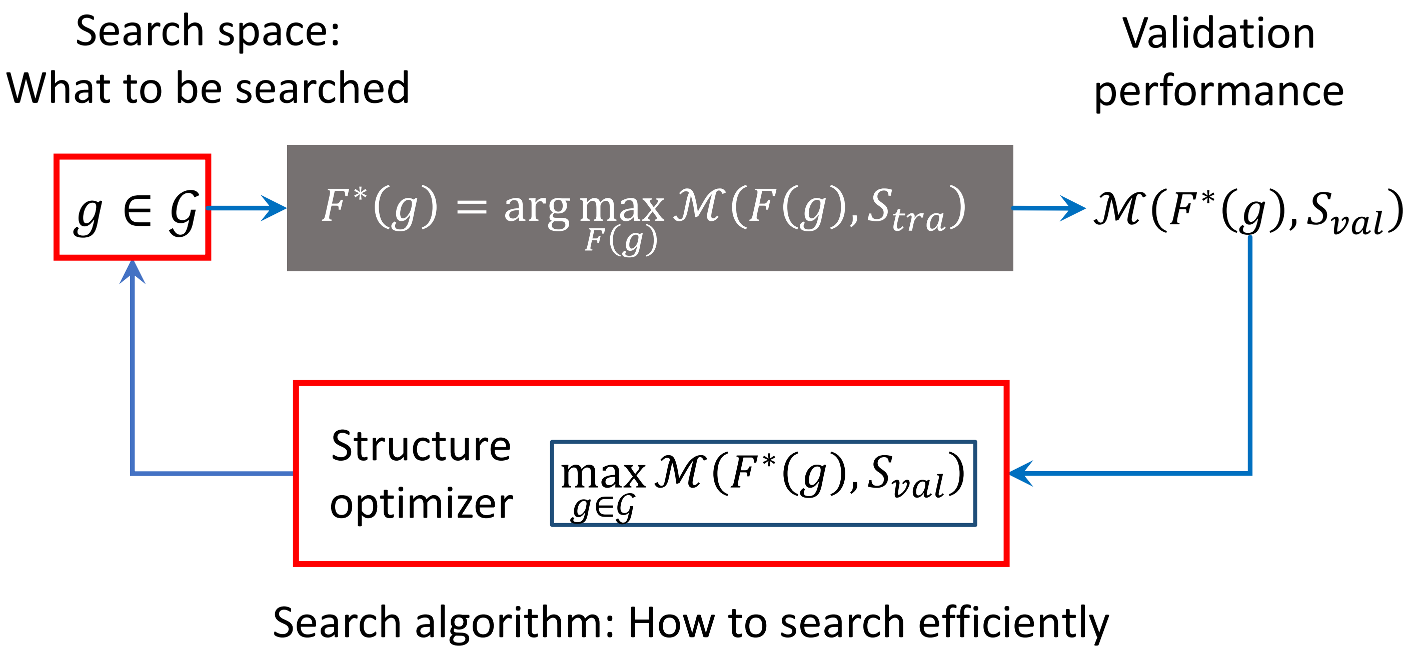
# AutoSF – Search to regularize bilinear SFs

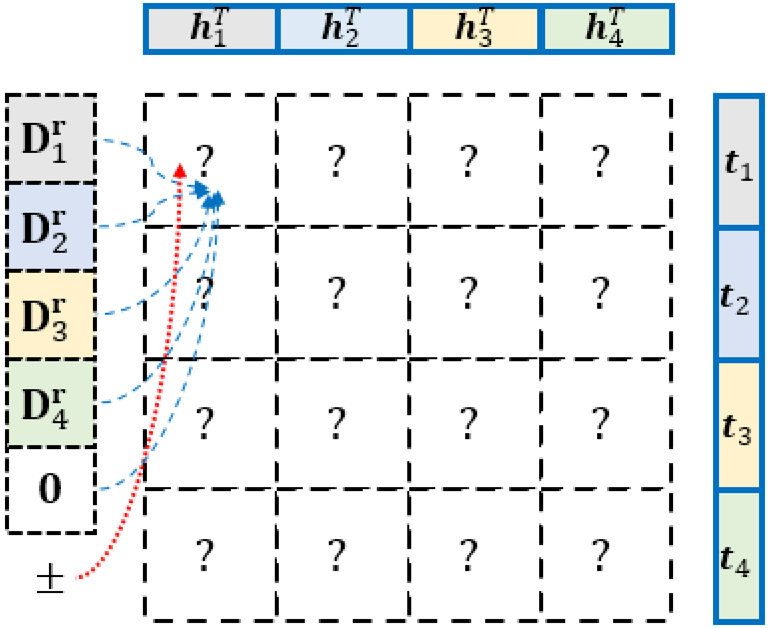
**Definition 3** (S2R Problem). *Let $F(\boldsymbol{P}; g)$ be a KG embedding model (with indexed embeddings $\boldsymbol{P} = \{\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}\}$ and architecture $g$), $M(F, \mathcal{S})$ measures the performance of a KG embedding model $F$ on a set of triplets $\mathcal{S}$ (the higher the better). The problem of S2R is formulated as:*

$$g^* \in \arg\max_{g \in \mathcal{G}} M\left(F(\boldsymbol{P}^*; g), \mathcal{S}_{val}\right) \tag{4}$$

$$\text{s.t.}\ \boldsymbol{P}^* = \arg\max_{\boldsymbol{P}} M(F(\boldsymbol{P}; g), \mathcal{S}_{tra}), \tag{5}$$

*where $\mathcal{G}$ contains all possible choices of $g$, $\mathcal{S}_{tra}$ is the training set, and $\mathcal{S}_{val}$ is the validation set.*




**Definition 2** (Search space). *Let $g(\mathbf{r})$ return a $4 \times 4$ block matrix, of which the elements in each block is given by $[g(\mathbf{r})]_{ij} = diag(\mathbf{a}_{ij})$ where $\mathbf{a}_{ij} \in \{\mathbf{0}, \pm\mathbf{r}_1, \pm\mathbf{r}_2, \pm\mathbf{r}_3, \pm\mathbf{r}_4\}$ for $i, j \in \{1, 2, 3, 4\}$. Then, SFs can be represented by $f_{unified}(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \sum_{i,j} \langle \mathbf{h}_i, \mathbf{a}_{ij}, \mathbf{t}_j \rangle = \mathbf{h}^\top g(\mathbf{r})\, \mathbf{t}$.*

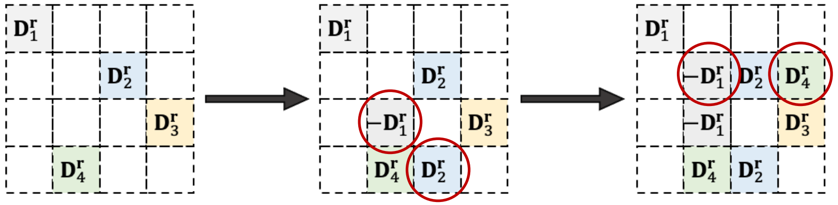
The location of a block matrix $\mathbf{D}_i^r$ represents a multiplicative term.

# AutoSF – Search algorithm

**Challenges**

1. Size of search space is very large: $9^{16}$.
2. Cost of training and evaluating a specific model structure is expensive.
3. How to capture important properties like symmetric, asymmetric?

Greedy search: progressively evaluate from few blocks to more blocks.




Genetic search can also be used, check

- https://arxiv.org/abs/2107.00184

Not all scoring functions / structures need to be trained.

For $f^6$, reduces from $2 \times 10^9$ to $3 \times 10^4$.

- ➢ Filter: remove bad and equivalent SFs.
  - Bad: there are zero/repeated rows/columns.
  - Equivalent: have the same expressive ability after permutation or slipping signs.

Select better SFs based on matrix structure to train and evaluate.

For $f^4$, reduces from 9216 to 5.

- ➢ Predictor: select promising SFs based on matrix structures.
  - The predictor learns a mapping from structure to performance.

# Experiments – Effectiveness

| | model | WN18 | | | FB15k | | | WN18RR | | | FB15k237 | | | YAGO3-10 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 | MRR | H@1 | H@10 |
| (TDM) | TransH | 0.521 | — | 94.5 | 0.452 | — | 76.6 | 0.186 | — | 45.1 | 0.233 | — | 40.1 | — | — | — |
| | RotatE | 0.949 | 94.4 | 95.9 | 0.797 | 74.6 | 88.4 | 0.476 | 42.8 | 57.1 | 0.338 | 24.1 | 53.3 | 0.488 | 39.6 | 66.3 |
| (NNM) | ConvE | 0.942 | 93.5 | 95.5 | 0.745 | 67.0 | 87.3 | 0.46 | 39. | 48. | 0.316 | 23.9 | 49.1 | 0.52 | 45. | 66. |
| | RSN | 0.94 | 92.2 | 95.3 | — | — | — | — | — | — | 0.28 | 20.2 | 45.3 | — | — | — |
| | CompGCN | — | — | — | — | — | — | 0.479 | 44.3 | 54.6 | 0.355 | 26.4 | 53.5 | — | — | — |
| (BLM) | TuckER | **0.953** | **94.9** | 95.8 | 0.795 | 74.1 | 89.2 | 0.470 | 44.3 | 52.6 | 0.358 | 26.6 | 54.4 | — | — | — |
| | DistMult | 0.821 | 71.7 | 95.2 | 0.775 | 71.4 | 87.2 | 0.443 | 40.4 | 50.7 | 0.352 | 25.9 | 54.6 | 0.552 | 47.1 | 68.9 |
| | SimplE/CP | 0.950 | 94.5 | 95.9 | 0.826 | 79.4 | 90.1 | 0.462 | 42.4 | 55.1 | 0.350 | 26.0 | 54.4 | 0.565 | 49.1 | 71.0 |
| | HolE/ComplEx | 0.951 | 94.5 | 95.7 | 0.831 | 79.6 | 90.5 | 0.471 | 43.0 | 55.1 | 0.345 | 25.3 | 54.1 | 0.563 | 49.0 | 70.7 |
| | Analogy | 0.950 | 94.6 | 95.7 | 0.816 | 78.0 | 89.8 | 0.467 | 42.9 | 55.4 | 0.348 | 25.6 | 54.7 | 0.557 | 48.5 | 70.4 |
| | QuatE | 0.950 | 94.5 | 95.9 | 0.782 | 71.1 | 90.0 | 0.488 | 43.8 | **58.2** | 0.348 | 24.8 | 55.0 | 0.556 | 47.4 | 70.4 |
| | AutoBLM | 0.952 | 94.7 | **96.1** | 0.853 | 82.1 | 91.0 | 0.490 | 45.1 | 56.7 | 0.360 | 26.7 | 55.2 | 0.571 | 50.1 | **71.5** |
| | AutoBLM+ | 0.952 | 94.7 | **96.1** | **0.861** | **83.2** | **91.3** | **0.492** | **45.2** | 56.7 | **0.364** | **27.0** | **55.3** | **0.577** | **50.2** | **71.5** |

## Measurements

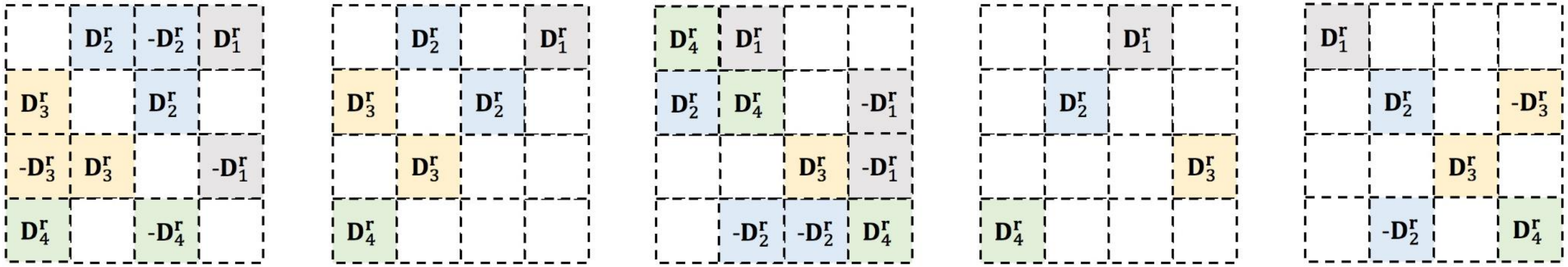
- Given a triplet $(h, r, t)$;
- Compute the score of $(h', r, t), \forall h' \in \mathcal{E}$;
- Get the rank of $h$ among all $h'$

## Metrics

- MRR: $\frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{\text{rank}_i}$
- Hit@k: $\frac{1}{|S|}\sum_{i=1}^{|S|}\mathbb{I}(\text{rank}_i < 10)$

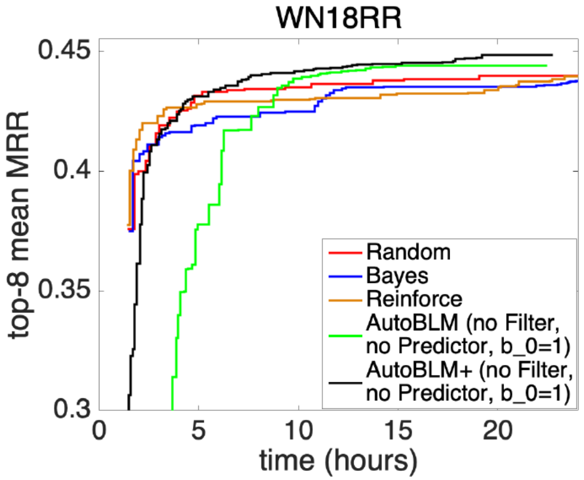
- BLMs are better than the other types and rule-based models
- There is no absolute winner among the BLMs
- Compared with human-designed ones, the SFs searched by AutoSF always lead the performance

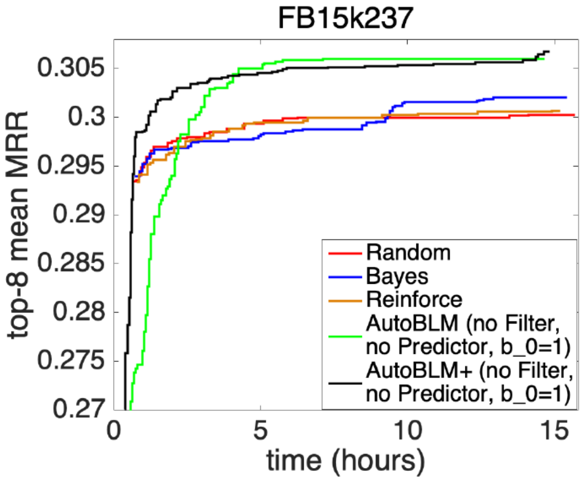
# Experiments – Efficiency




(a) WN18. (b) FB15k. (c) WN18RR. (d) FB15k237. (e) YAGO3-10.

The searched SFs are KG dependent and novel to the literature.



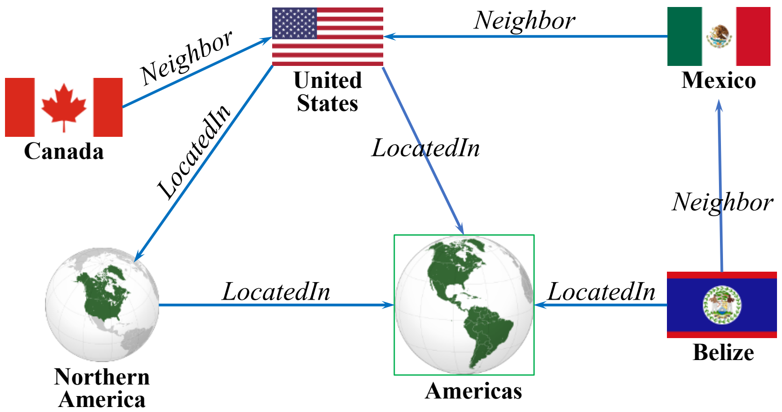




- Random: totally random for SF generation
- Bayes: Tree Parzen Estimator (TPE) algorithm
- AutoSF (AutoBLM): domain-specific search algorithm

# Outline

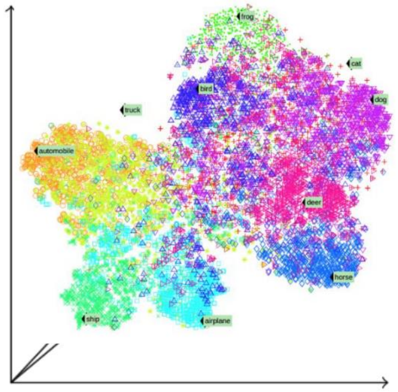
1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
    - Overview of Ideas
    - Search to Capture Semantics
    - Search to Exploit Graph Topology
4. Future Works & Summary

# Relational Path in KG




vectorize

entities & relations



Triplets: $(s, r, o)$;

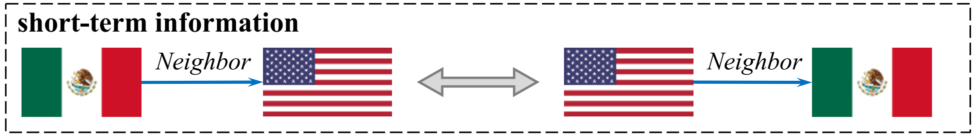
Relational path [1,2]:

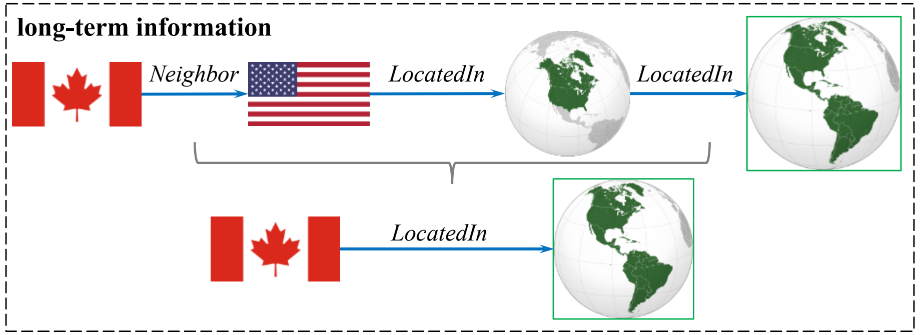
$$s_1 \xrightarrow{r_1} s_2 \xrightarrow{r_2} s_3 \cdots s_L \xrightarrow{r_L} s_{L+1}$$

Short-term information inside triplets.

Composition of relations.

Long-term information across triplets.




[1]. Guu et.al. Traversing knowledge graphs in vector space. EMNLP, 2015
[2]. Sadeghian et.al. DRUM: End-to-end differentiable rule mining on knowledge graphs. NeurIPS 2019

# Contribution – Search to exploit topology

1. The relational path contains several mixed information.
2. Link prediction task emphasizes on the short-term semantic information, while entity alignment task requires to model the long-term information.
3. How to properly encode such prior knowledge into the model design?

Our solutions:

- Search to adaptively learn the mixed information in relational path.
- A novel hybrid-search algorithm for efficient search.

Interstellar: Searching Recurrent Architecture for Knowledge Graph Embedding. NeurIPS 2020

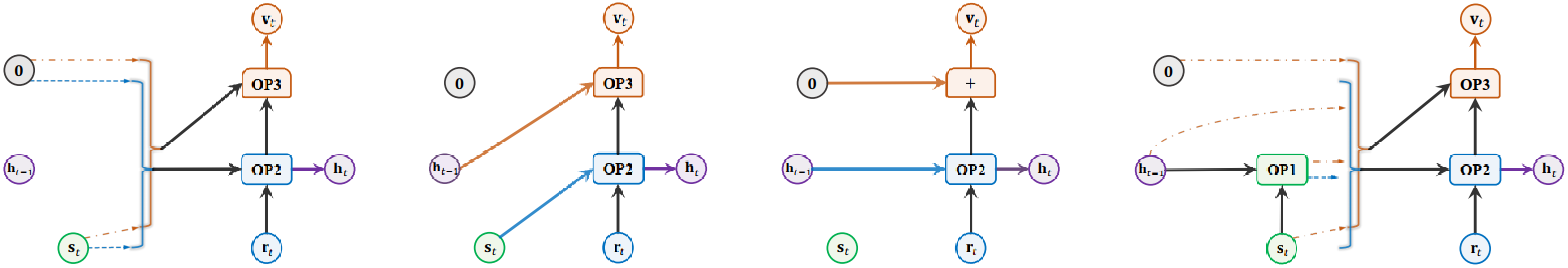
# Recurrent Structure – Case study

| data | tasks |
|---|---|
| S1 | neighbor ∧ locatedin → locatedin<br>locatedin ∧ locatedin → locatedin |
| S2 | neighbor ∧ locatedin → locatedin |
| S3 | neighbor ∧ locatedin ∧ locatedin → locatedin |

harder | longer

Table 3: Performance on Countries dataset.

| | S1 | S2 | S3 |
|---|---|---|---|
| P1 | 0.998±0.001 | 0.997±0.002 | 0.933±0.031 |
| P2 | **1.000±0.000** | 0.999±0.001 | 0.952±0.023 |
| P3 | 0.992±0.001 | **1.000±0.000** | 0.961±0.016 |
| P4 | 0.977±0.028 | 0.984±0.010 | **0.964±0.015** |
| Interstellar | **1.000±0.000** | **1.000±0.000** | **0.968± 0.007** |

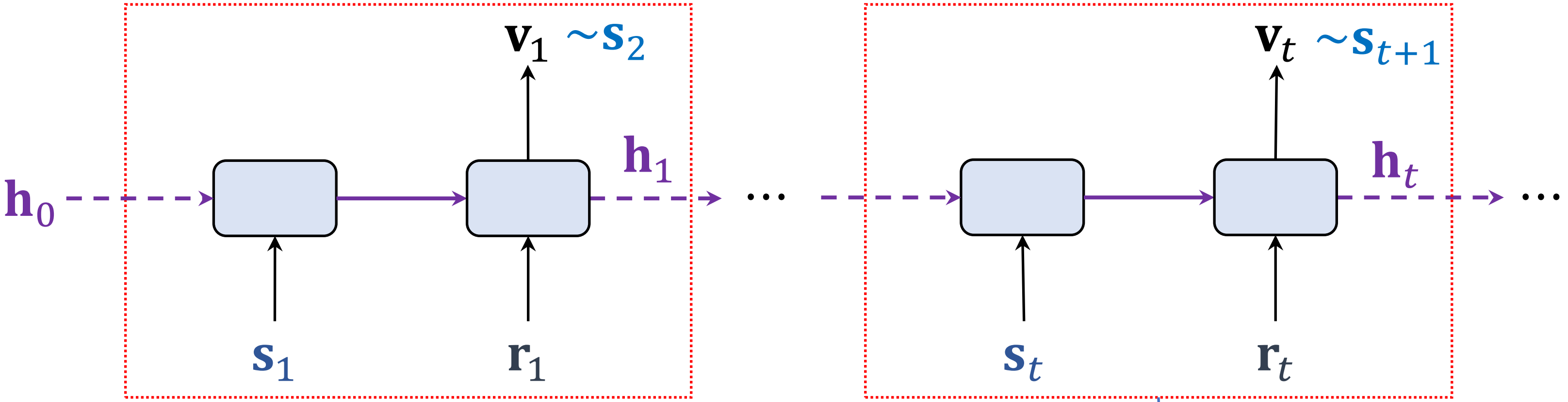



P1: single triplet P2: two triples P3: relation composition P4: fully recurrent

Model design should be data-specific. Search to leverage proper prior knowledge.

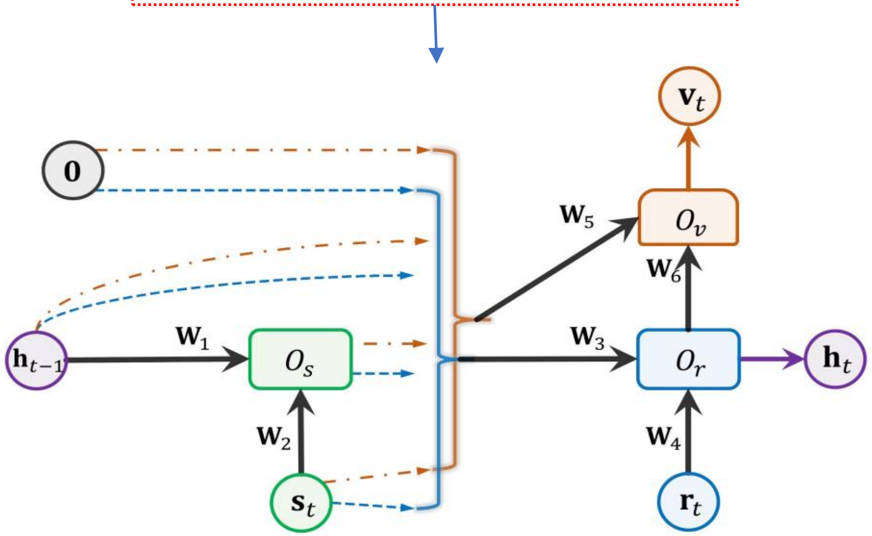
# Interstellar – Searching recurrent structure

Recurrently process the path by $[\mathbf{v}_t, \mathbf{h}_t] = f(\mathbf{s}_t, \mathbf{r}_t, \mathbf{h}_{t-1}), \ \ \forall t = 1 \dots L$




Searching !

| | | |
|---|---|---|
| macro-level | connections | $\mathbf{h}_{t-1}, O_s, \mathbf{0}, \mathbf{s}_t$ |
| $\hat{\boldsymbol{\alpha}} \in \hat{\mathcal{A}}$ | combinators | $+, \odot, \otimes$, gated |
| micro-level | activation | identity, tanh, sigmoid |
| $\check{\boldsymbol{\alpha}} \in \check{\mathcal{A}}$ | weight matrix | $\{\mathbf{W}_i\}_{i=1}^{6}, \mathbf{I}$ |



# Hybrid Search Algorithm

Search appropriate $\alpha \in \mathcal{A}$ that maximize the validation performance

$$\boldsymbol{\alpha}^* = \arg\max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{M}\left(f(\boldsymbol{F}^*;\boldsymbol{\alpha}), \mathcal{G}_{\text{val}}\right), \quad \text{s.t.} \quad \boldsymbol{F}^* = \arg\min_{\boldsymbol{F}} \mathcal{L}\left(f(\boldsymbol{F};\boldsymbol{\alpha}), \mathcal{G}_{\text{tra}}\right)$$

**Stand-alone approach:**

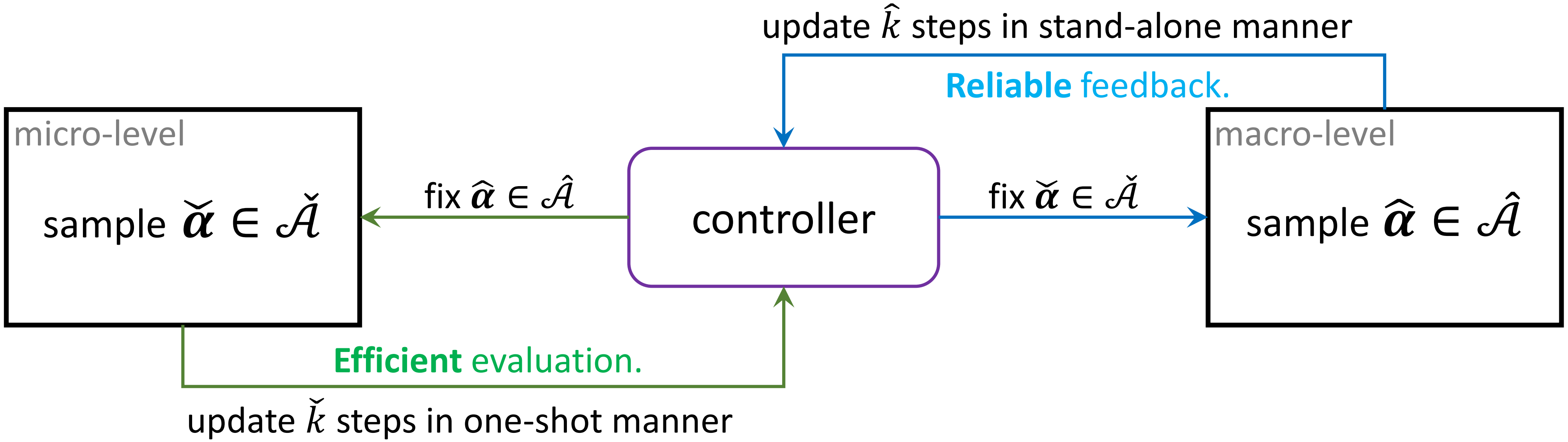
- $\mathcal{M}$ is accurate;
- $\boldsymbol{F}^*$ needs high cost.

[Zoph and Le 2017]

**One-shot approach:**

- $\boldsymbol{F}^*$ is shared and efficient;
- $\mathcal{M}$ is not always reliable.

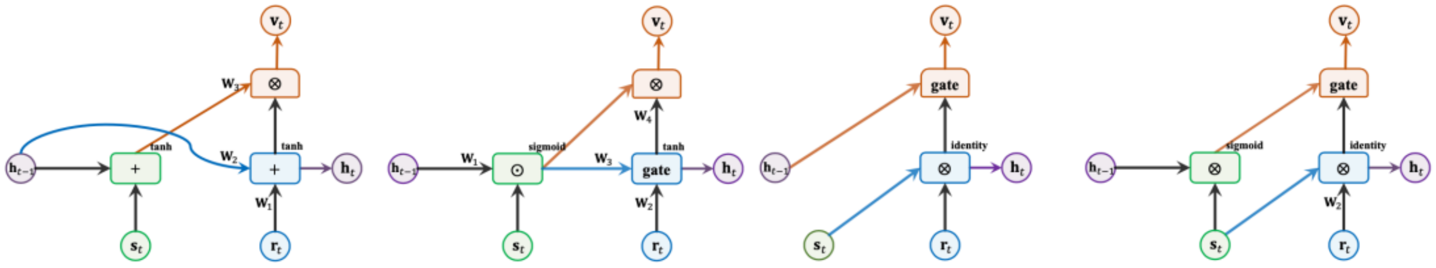
[Pham et al. 2018, Liu et al. 2019]



# Experiments – Effectiveness

## Entity alignment task

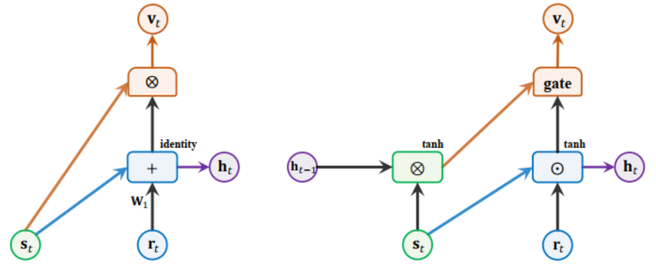
| models | | DBP-WD H@1 | DBP-WD H@10 | DBP-WD MRR | DBP-YG H@1 | DBP-YG H@10 | DBP-YG MRR | EN-FR H@1 | EN-FR H@10 | EN-FR MRR | EN-DE H@1 | EN-DE H@10 | EN-DE MRR |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| triplet | TransE | 18.5 | 42.1 | 0.27 | 9.2 | 24.8 | 0.15 | 16.2 | 39.0 | 0.24 | 20.7 | 44.7 | 0.29 |
| | TransD* | 27.7 | 57.2 | 0.37 | 17.3 | 41.6 | 0.26 | 21.1 | 47.9 | 0.30 | 24.4 | 50.0 | 0.33 |
| | BootEA* | 32.3 | 63.1 | 0.42 | 31.3 | 62.5 | 0.42 | 31.3 | 62.9 | 0.42 | 44.2 | 70.1 | 0.53 |
| GCN | GCN-Align | 17.7 | 37.8 | 0.25 | 19.3 | 41.5 | 0.27 | 15.5 | 34.5 | 0.22 | 25.3 | 46.4 | 0.22 |
| | VR-GCN | 19.4 | 55.5 | 0.32 | 20.9 | 55.7 | 0.32 | 16.0 | 50.8 | 0.27 | 24.4 | 61.2 | 0.36 |
| | R-GCN | 8.6 | 31.4 | 0.16 | 13.3 | 42.4 | 0.23 | 7.3 | 31.2 | 0.15 | 18.4 | 44.8 | 0.27 |
| path | PTransE | 16.7 | 40.2 | 0.25 | 7.4 | 14.7 | 0.10 | 7.3 | 19.7 | 0.12 | 27.0 | 51.8 | 0.35 |
| | IPTransE* | 23.1 | 51.7 | 0.33 | 22.7 | 50.0 | 0.32 | 25.5 | 55.7 | 0.36 | 31.3 | 59.2 | 0.41 |
| | Chains | 32.2 | 60.0 | 0.42 | 35.3 | 64.0 | 0.45 | 31.4 | 60.1 | 0.41 | 41.3 | 68.9 | 0.51 |
| | RSN* | 38.8 | 65.7 | 0.49 | 40.0 | 67.5 | 0.50 | 34.7 | 63.1 | 0.44 | 48.7 | 72.0 | 0.57 |
| | **SRAP** | **40.7** | **71.2** | **0.51** | **40.2** | **72.0** | **0.51** | **35.5** | **67.9** | **0.46** | **50.1** | **75.6** | **0.59** |



(a) DBP-WD (b) DBP-YG (c) EN-FR (d) EN-DE

## Link prediction task

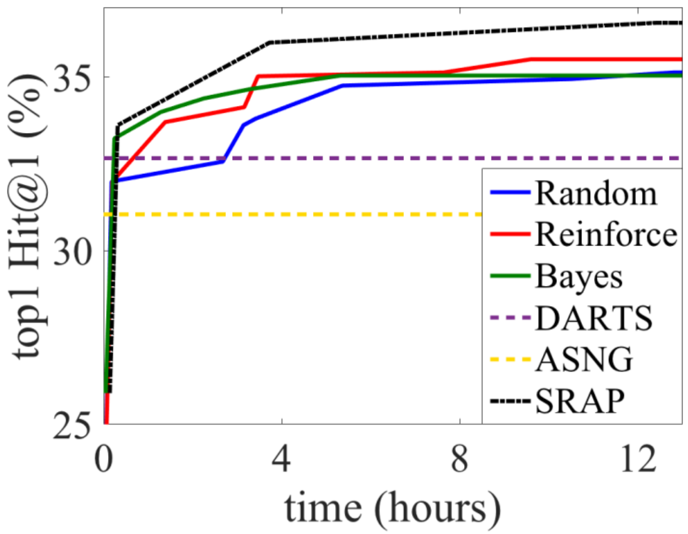
| models | WN18-RR H@1 | WN18-RR H@10 | WN18-RR MRR | FB15k-237 H@1 | FB15k-237 H@10 | FB15k-237 MRR |
|---|---|---|---|---|---|---|
| TransE | 12.5 | 44.5 | 0.18 | 17.3 | 37.9 | 0.24 |
| ComplEx | 41.4 | 49.0 | 0.44 | 22.7 | 49.5 | 0.31 |
| RotatE | 43.6 | 54.2 | 0.47 | **23.3** | 50.4 | **0.32** |
| R-GCN | - | - | - | 15.1 | 41.7 | 0.24 |
| PTransE | 27.2 | 46.4 | 0.34 | 20.3 | 45.1 | 0.29 |
| RSN | 38.0 | 44.8 | 0.40 | 19.2 | 41.8 | 0.27 |
| SRAP | **44.0** | **54.8** | **0.48** | **23.3** | **50.8** | **0.32** |



WN18RR FB15k-237

# Experiments – Efficiency

entity alignment




link prediction

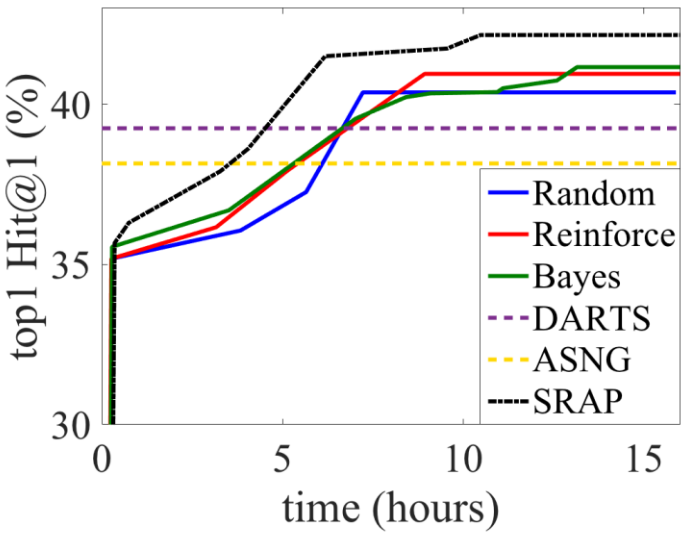


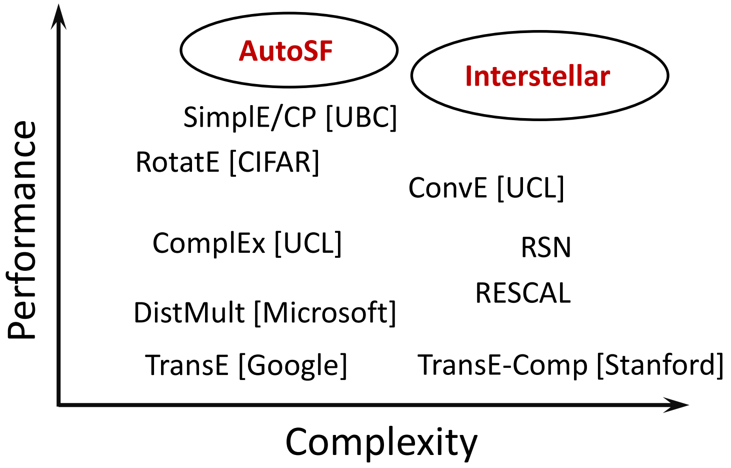

Table 6: Comparison of searching and fine-tuning time (in hours) in Algorithm 1.

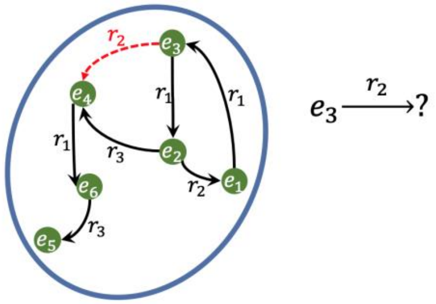
| procedure | | entity alignment | | link prediction | |
|---|---|---|---|---|---|
| | | Normal | Dense | WN18-RR | FB15k-237 |
| search | macro-level (line 2-3) | 9.9±1.5 | 14.9±0.3 | 11.7±1.9 | 23.2±3.4 |
| | micro-level (line 4-5) | 4.2±0.2 | 7.5±0.6 | 6.3± 0.9 | 5.6±0.4 |
| fine-tune (line 7) | | 11.6±1.6 | 16.2±2.1 | 44.3±2.3 | 67.6±4.5 |

# Summary

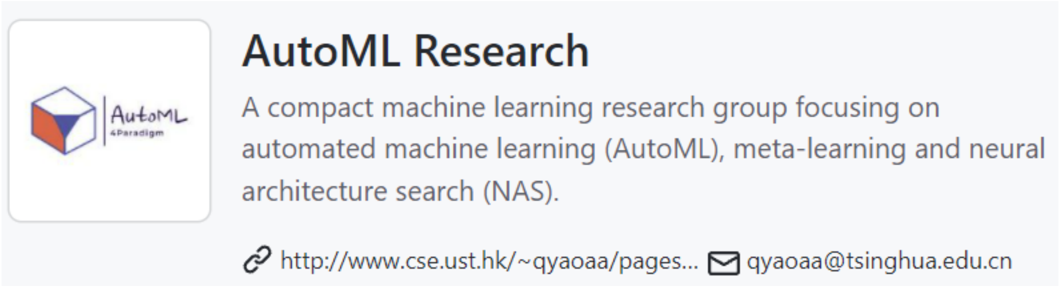
Design data-specific KG learning methods by AutoML

- Better explore semantics and topology
- Adapt to different application needs




Code: https://github.com/AutoML-Research/AutoSF



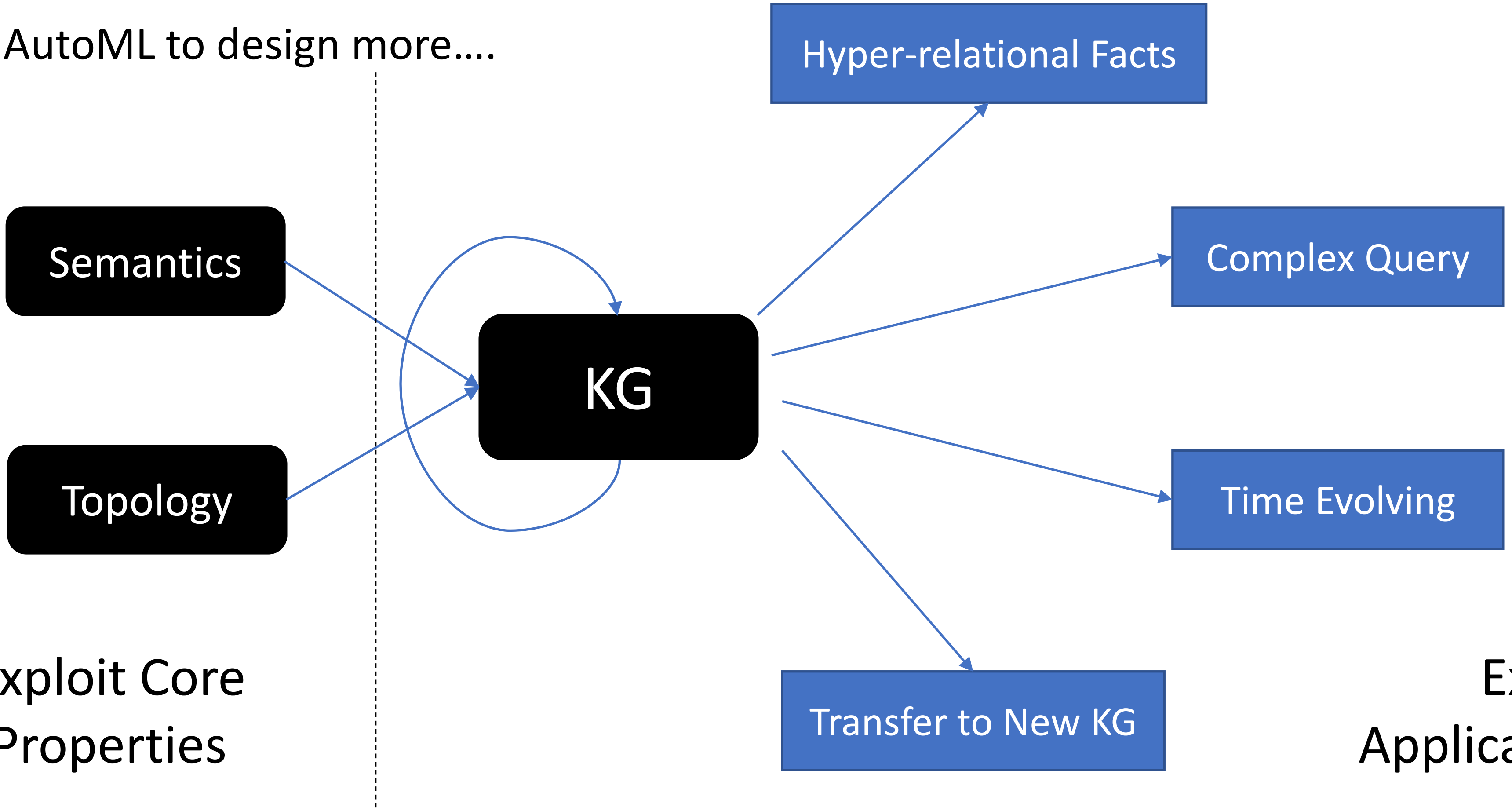

Winning solution in OGB

# Outline

1. What is Knowledge Graph (KG)?
2. What is Automated Machine Learning (AutoML)?
3. Attacking Core Issues in KG by AutoML
4. Future Works & Summary

# AutoML in KG – Problem level

Using AutoML to design more….




Exploit Core Properties

Explore Application Needs

# AutoML in KG – Benchmark level

As a learning problem, the KG embedding problem contains the following important components:

$$\underbrace{\min_{\boldsymbol{w}}}_{\text{optimization}} \overbrace{\mathcal{L}}^{\text{loss function}} \Big( \underbrace{S_r}_{\text{scoring function}} \big( \overbrace{f(h), f(\boldsymbol{r}), f(t)}^{\text{encoder}} \big), \underbrace{\mathcal{S}^{+}, \mathcal{S}^{-}}_{\text{negative sampling}} \Big) + \underbrace{r(\boldsymbol{w})}_{\text{regularization}}$$

# Collaborators on This Topic

- James Kwok. *Professor*. HKUST
- Tong Zhang. *Professor*. HKUST
- Yong Li. *Associate Professor*. Tsinghua
- Huan Zhao. *Senior Researcher*. 4Paradigm
- Yongqi Zhang. *Researcher*. 4Paradigm
- Fengli Xu. *Post-Doc*. University of Chicago
- Shimin Di. *Ph.D Student*. HKUST
- Yu Liu. *Ph.D Student*. Tsinghua
- Yuhui Ding. *M.Phi*. HKUST

# Related Publication

1. Automorphic Equivalence-aware Graph Neural Network. NeurIPS 2021
2. Efficient, Simple and Automated Negative Sampling for Knowledge Graph Embedding. VLDBJ 2021
3. DiffMG: Differentiable Meta Graph Search for Heterogeneous Graph Neural Networks. KDD 2021
4. Searching to Sparsify Tensor Decomposition for N-ary Relational Data. WebConf 2021
5. Role-Aware Modeling for N-ary Relational Knowledge Bases. WebConf 2021
6. Efficient Relation-aware Scoring Function Search for Knowledge Graph Embedding. ICDE 2021
7. Search to aggregate neighborhood for graph neural network. ICDE 2021
8. Interstellar: Searching Recurrent Architecture for Knowledge Graph Embedding. NeurIPS 2020
9. Generalizing Tensor Decomposition for N-ary Relational Knowledge Bases. WebConf 2020
10. AutoSF: Searching Scoring Functions for Knowledge Graph Embedding. ICDE 2020

# Thanks!

# 恳请各位批评&指正!