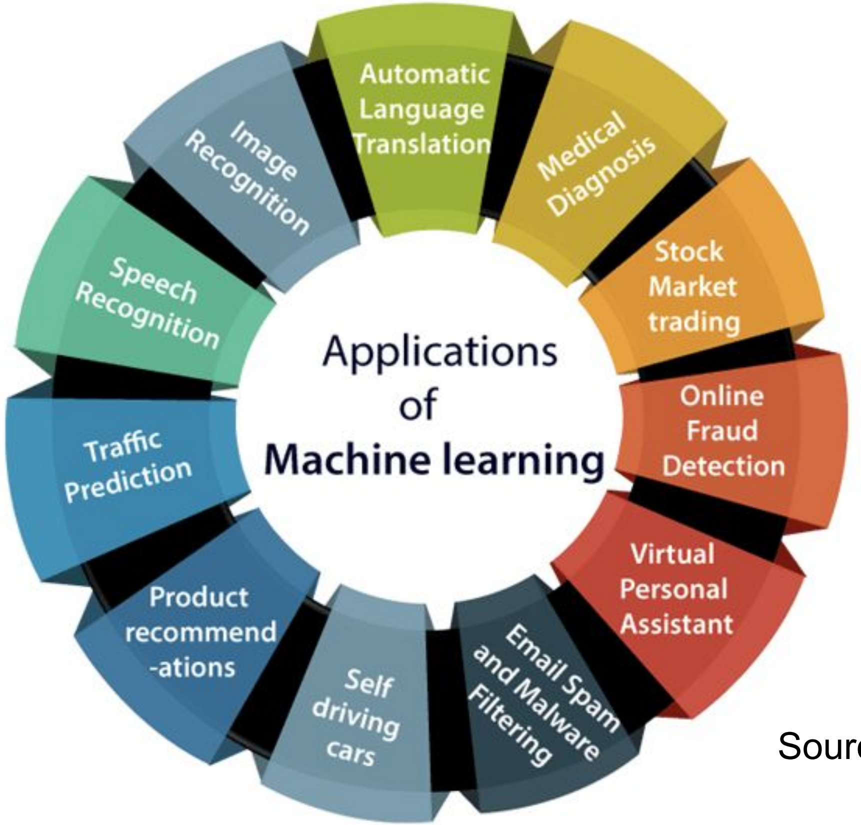
Optimisation of Inference Queries

Ziyu Li, TU Delft
with Marios Fragkoulis, Alessandro Bozzon, Asterios Katsifodimos

01.10.2021, DSDSD

Machine Learning is widely adopted

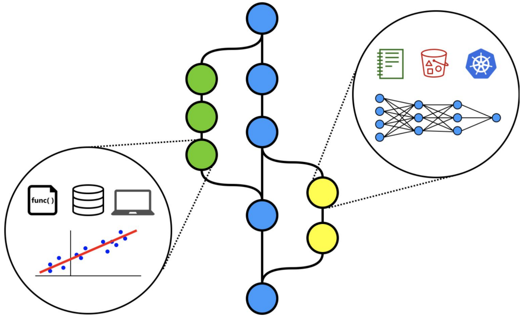


Source: Javapoint

TU Delft

Management for explosion of ML models

MLOps



ModelDB [2]

Model Hub

HuggingFace [3]
17,144

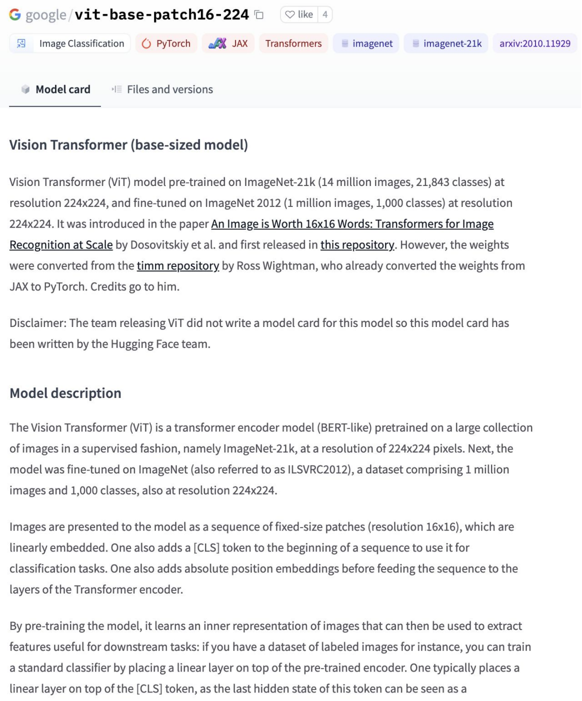
TensorFlow Hub
[4]
>1000

PYTORCH HUB
[5]
42+

[2] Vartak M, Subramanyam H, Lee W E, et al. ModelDB: a system for machine learning model management[C]//Proceedings of the Workshop on Human-In-the-Loop Data Analytics. 2016: 1-3.
[3] HuggingFace https://huggingface.co/models
[4] TensorFlow Hub https://tfhub.dev/
[5] PyTorch Hub https://pytorch.org/hub/

Example: HuggingFace

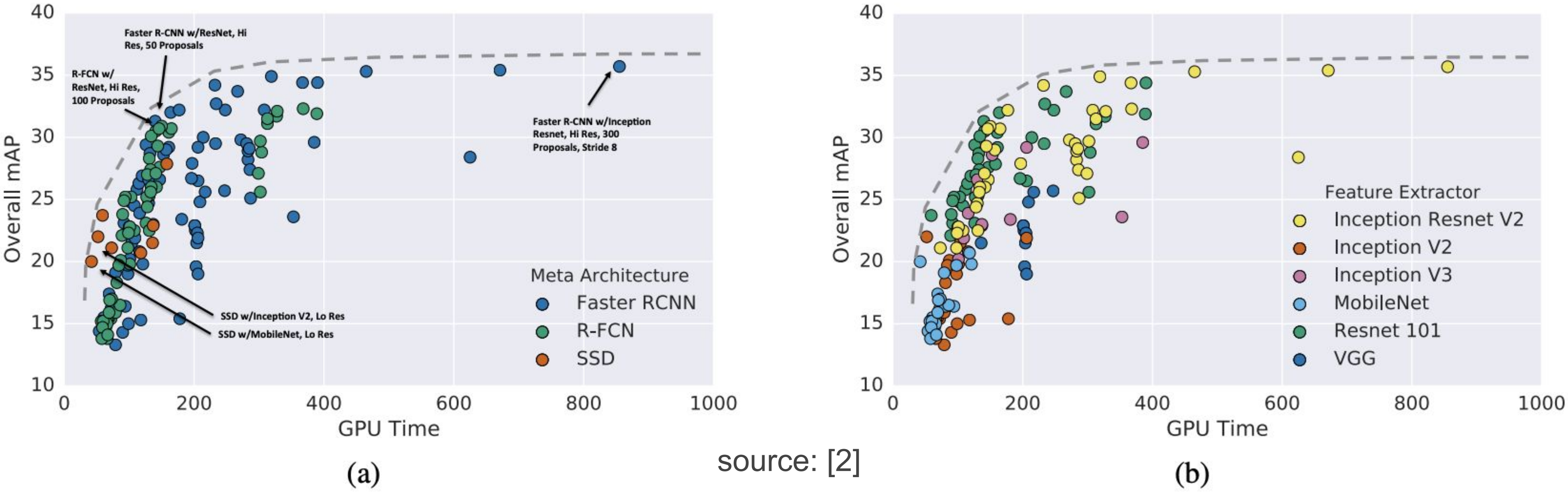
- Model description
- Intended uses & limitations
- Training procedure & data
- Evaluation results (accuracy)



Limitations

- Speed also matters
  - Lack of information regarding inference cost, e.g., FLOPs, execution time
- Lack of necessary metadata
  - Input & output
  - Performance across object classes

Tradeoff between accuracy and execution time



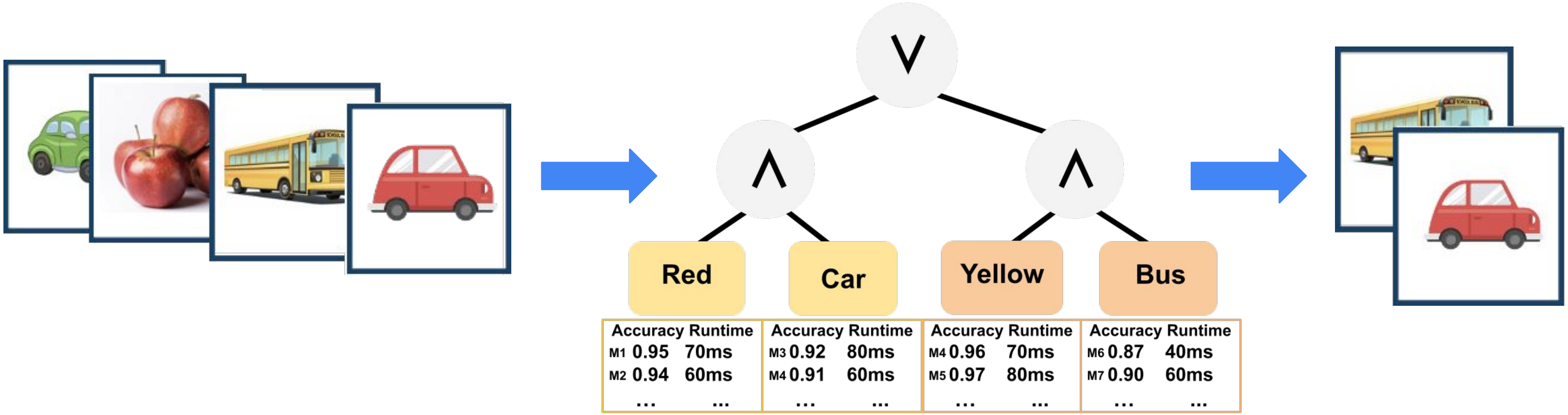
source: [2]

[2] Huang J, Rathod V, Sun C, et al. Speed/accuracy trade-offs for modern convolutional object detectors[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7310-7311.

Increasingly complex to select optimal ML models

For a specific inference task:

e.g., $(P_{car} \wedge P_{red}) \vee (P_{bus} \wedge P_{yellow})$



Formalizing model repository

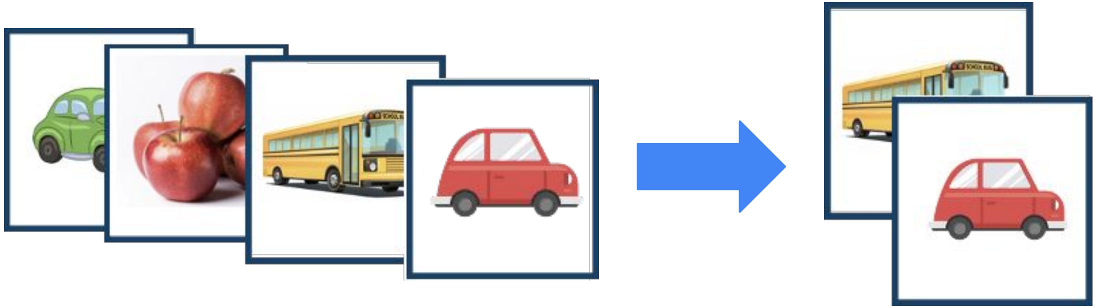
- Query
  - $(P_{car} \land P_{red}) \lor (P_{bus} \land P_{yellow})$
- Model repository
  - $\mathcal{R}(C\{M, P\}, A\{M, P\})$

Table 1: Example accuracy A of models in a repository.

	car	bus	red	yellow
Model 1	0.88	0	0	0
Model 2	0.98	0	0	0
Model 3	0	0.75	0	0
Model 4	0	0.95	0	0
Model 5	0	0	0.96	0.97
Model 6	0	0	0.97	0.98

Table 2: Example execution time C of models in a repository.

	car	bus	red	yellow
Model 1	15	∞	∞	∞
Model 2	30	∞	∞	∞
Model 3	∞	20	∞	∞
Model 4	∞	35	∞	∞
Model 5	∞	∞	5	5
Model 6	∞	∞	10	10



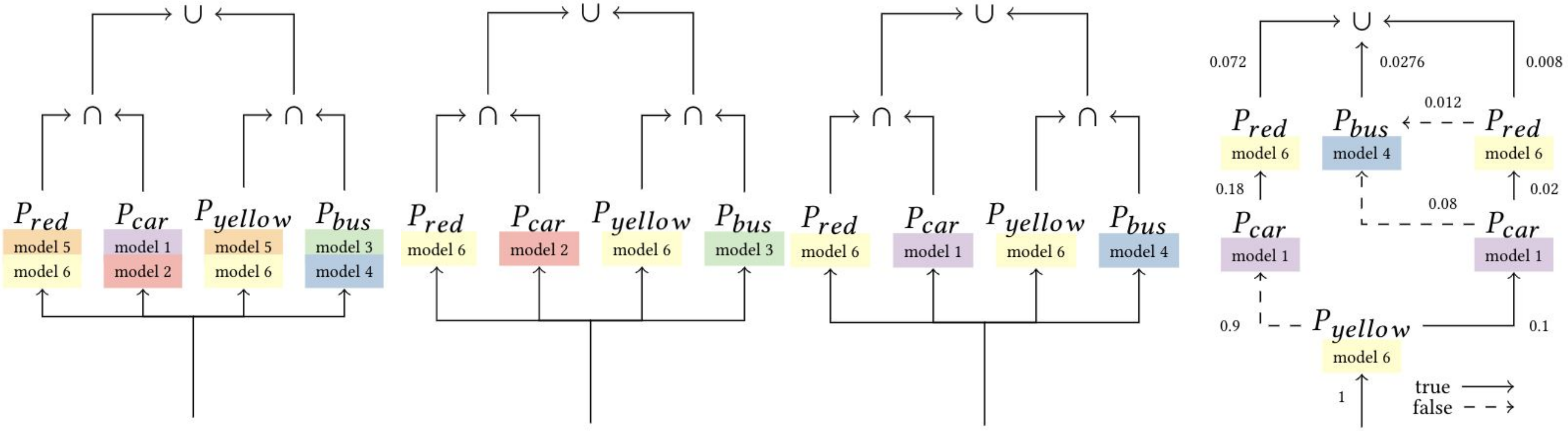
TU Delft

Goal

- Generate query plans for inference queries defined on model repositories
  - Tackle the problem of optimal **model selection** and **predicate ordering** *under accuracy and execution time constraints*

We devise three approaches

- Greedy (model selection)
- Model optimizer (model selection)
- Order optimizer (model selection + predicate ordering)



(a) Logical query plan (b) Greedy query plan (c) Model optimal query plan (d) Order optimal query plan

We devise three approaches

- Greedy (model selection)
- Model optimizer (model selection)
- Order optimizer (model selection + predicate ordering)

1. Select Pareto-optimal models
2. Loop over predicates and greedily select model with most accurate / least execution time

We devise three approaches

- Greedy (model selection)
- Model optimizer (model selection)
- Order optimizer (model selection + predicate ordering)

Apply Mixed Integer Programming:

- Model the accuracy of the query
- Model the execution time of the models
- Maximize accuracy / Minimize execution time

	car	bus	red	yellow
Model 1	$x_{m1,car}$	$x_{m1,bus}$	$x_{m1,red}$	$x_{m1,yellow}$
Model 2				
Model 3	...	...	...	...
Model 4				
Model 5				
Model 6	...	...	...	...
	$\sum = 1$	$\sum = 1$	$\sum = 1$	$\sum = 1$

Cost model

$$\left.\begin{aligned} B_{m1} &= \{0,1\} \cdot c_{m1} \\ B_{m2} &= \{0,1\} \cdot c_{m2} \\ B_{m3} &= \{0,1\} \cdot c_{m3} \\ B_{m4} &= \{0,1\} \cdot c_{m4} \\ B_{m5} &= \{0,1\} \cdot c_{m5} \\ B_{m6} &= \{0,1\} \cdot c_{m6} \end{aligned}\right\} \sum$$

Accuracy model

$$\begin{aligned} a = &(a_{red} \cdot a_{car}) + (a_{yellow} \cdot a_{bus}) \\ &- (a_{red} \cdot a_{car}) \cdot (a_{yellow} \cdot a_{bus}) \end{aligned}$$

We devise three approaches

- Greedy (model selection)
- Model optimizer (model selection)
- Order optimizer (model selection + predicate ordering)

Apply Mixed Integer Programming:

- **Model the order of predicates**
- Model the accuracy of the query
- **Model the execution time of the models while taking into account of selectivity**
- Maximize accuracy / Minimize execution time

Predicate ordering

	1	**2**	**3**	**4**
red	1	0	0	0
car	0	1	0	0
yellow	0	0	1	0
bus	0	0	0	1

Model selection

	car	bus	red	yellow
Model 1	0	0	0	0
Model 2	1	0	0	0
Model 3	...	...	...	...
Model 4				
Model 5				
Model 6	...	...	...	...

Formalizing model repository

- Query
  - $(P_{car} \wedge P_{red}) \vee (P_{bus} \wedge P_{yellow})$
- Model repository
  - $\mathcal{R}(C\{M, P\}, A\{M, P\})$
  - ***Selectivity*(*P*)**

Table 1: Example accuracy A of models in a repository.

	car	bus	red	yellow
Model 1	0.88	0	0	0
Model 2	0.98	0	0	0
Model 3	0	0.75	0	0
Model 4	0	0.95	0	0
Model 5	0	0	0.96	0.97
Model 6	0	0	0.97	0.98

Table 2: Example execution time C of models in a repository.

	car	bus	red	yellow
Model 1	15	∞	∞	∞
Model 2	30	∞	∞	∞
Model 3	∞	20	∞	∞
Model 4	∞	35	∞	∞
Model 5	∞	∞	5	5
Model 6	∞	∞	10	10

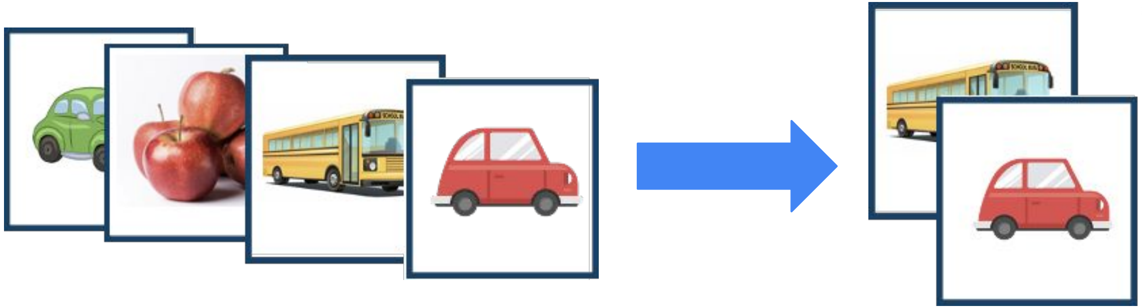


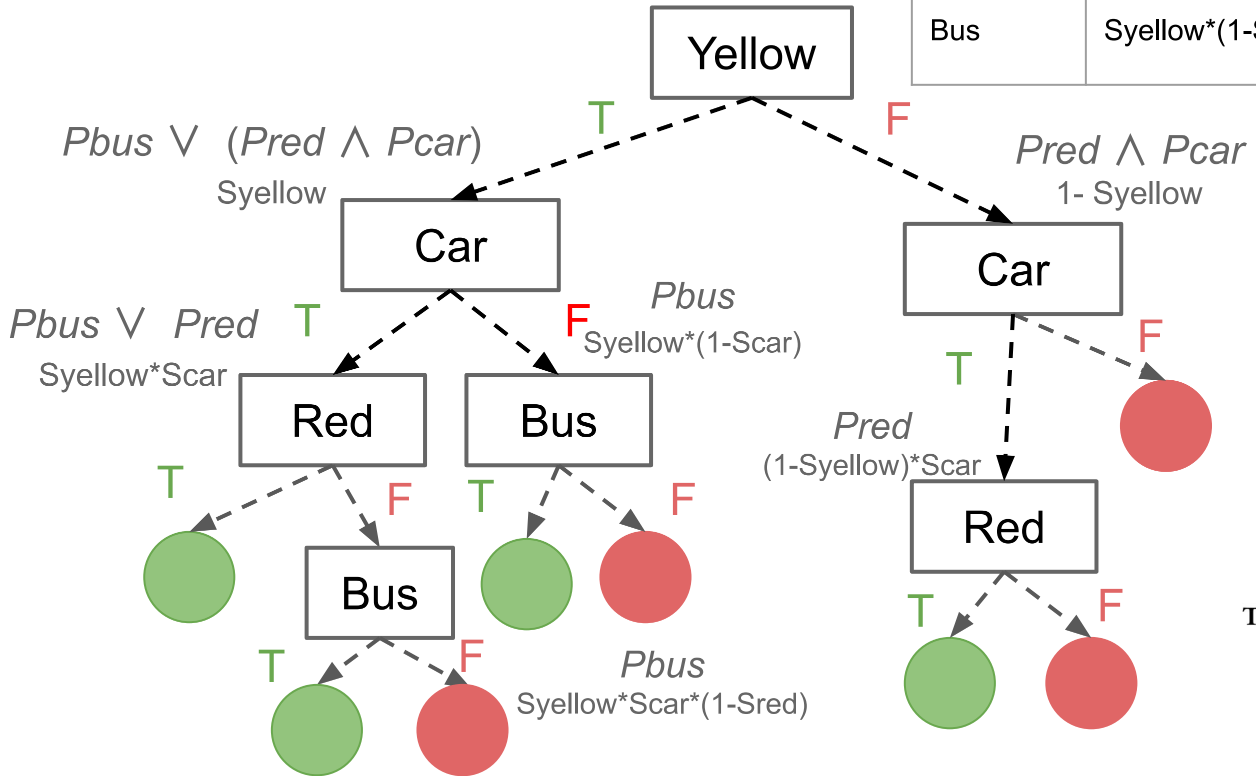
Table 3: Example selectivity of predicates in a dataset.

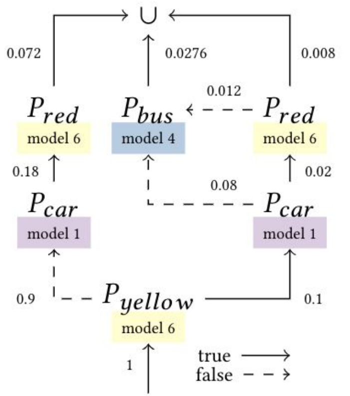
	car	bus	red	yellow
selectivity	0.2	0.3	0.4	0.1

$(Pred \land Pcar) \lor (Pyellow \land Pbus)$

Yellow -> Car -> Red -> Bus

Predicate	Data Proportion	Condition
Yellow	1	‘’
Car	1	‘’
Red	Scar	*Pcar = 1*
Bus	Syellow*(1-Scar*Sred)	*Pyellow*=1 and (*Pcar*=1 or *Pred*=1)





(d) Order optimal query plan

Table 3: Example selectivity of predicates in a dataset.

	car	bus	red	yellow
selectivity	0.2	0.3	0.4	0.1

Preliminary results: accuracy vs execution time

Test on query generated from COCO classes and evaluate on validation set.
125 model variants generated from YOLOv3 and YOLOv5.

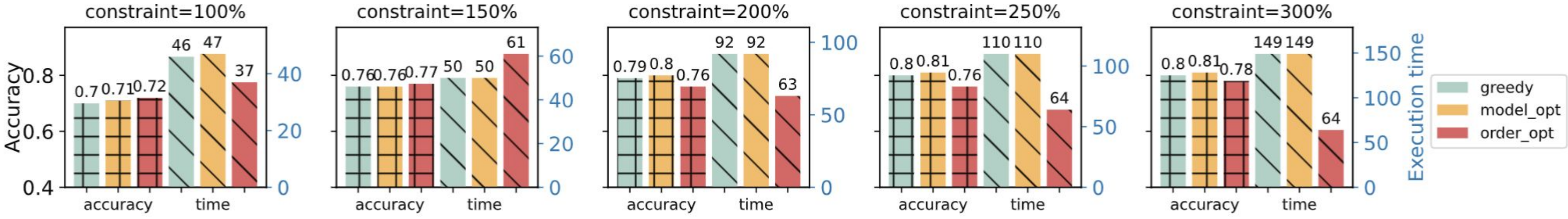
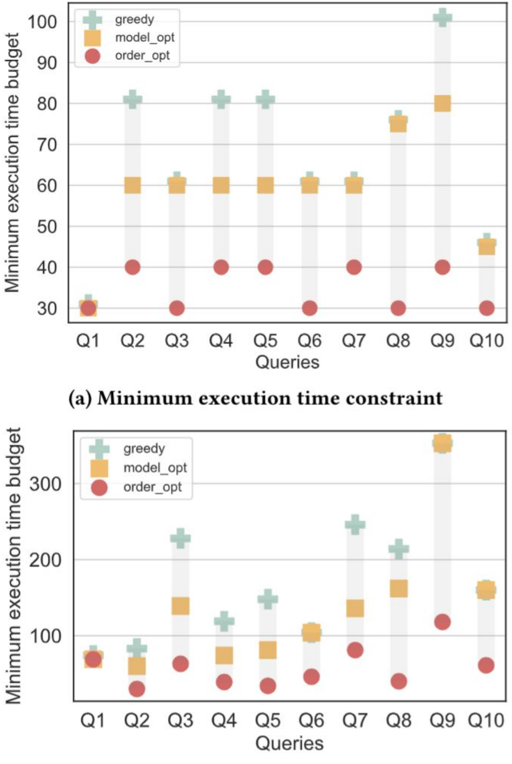


Figure 6: Performance of Query 4

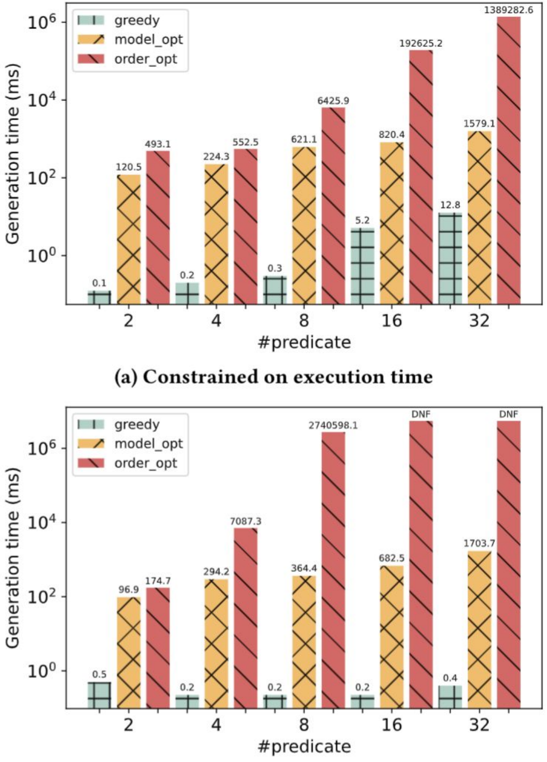
Preliminary results



(a) Minimum execution time constraint

(b) Maximum accuracy constraint

Figure 7: Comparison on the execution time budget with the boundary constraint



(a) Constrained on execution time

(b) Constrained on accuracy

Figure 8: Generation time of a query plan varying number of predicates

Takeaways

- We motivate the problem by highlighting the emergence of repositories of ML models
  - Available models along with their metadata descriptions.
- We propose three query optimization strategies
  - We evaluate them on a model repository that we construct from real models using queries defined over the COCO datase
- Our greedy optimizer is the fastest in generating query plans, but our order optimizer produces substantially better query plans when a tight constraint is encountered