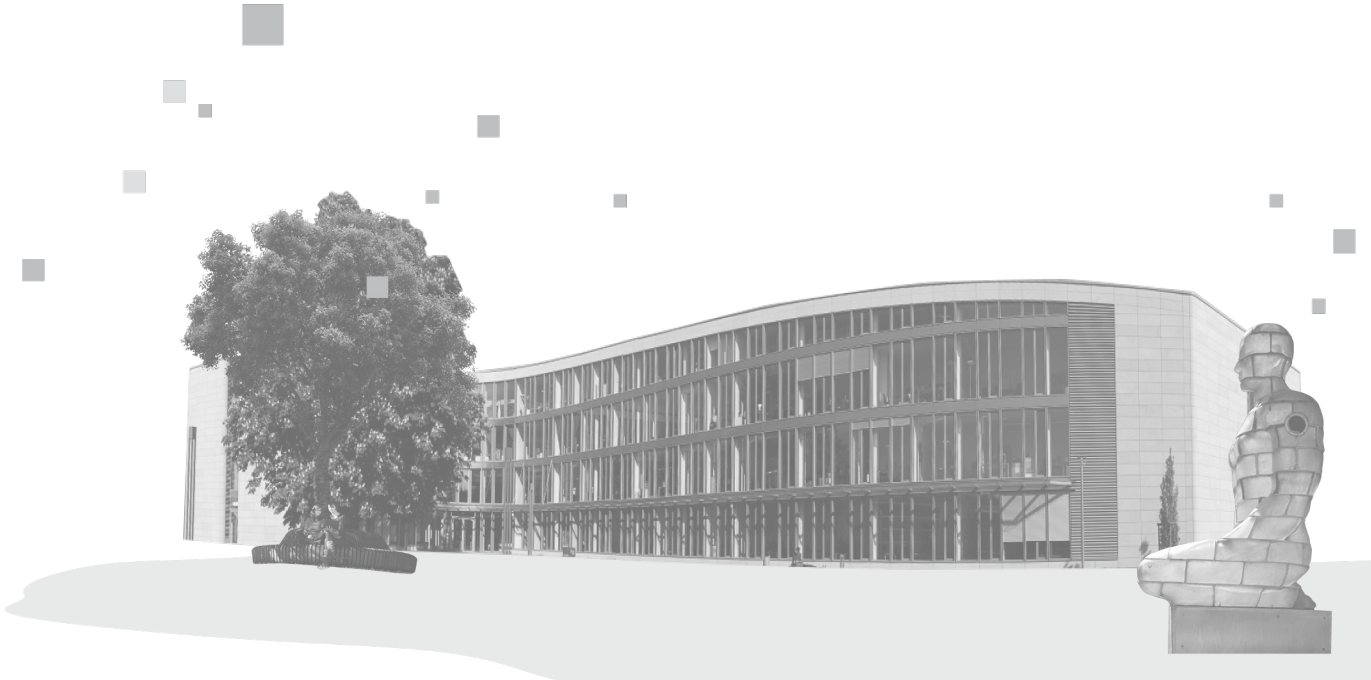
# Poodle: Seamlessly Scaling Down LLMs with Just-in-Time Model Replacement [Vision Paper]

**Nils Strassenburg**

Boris Glavic

Tilmann Rabl

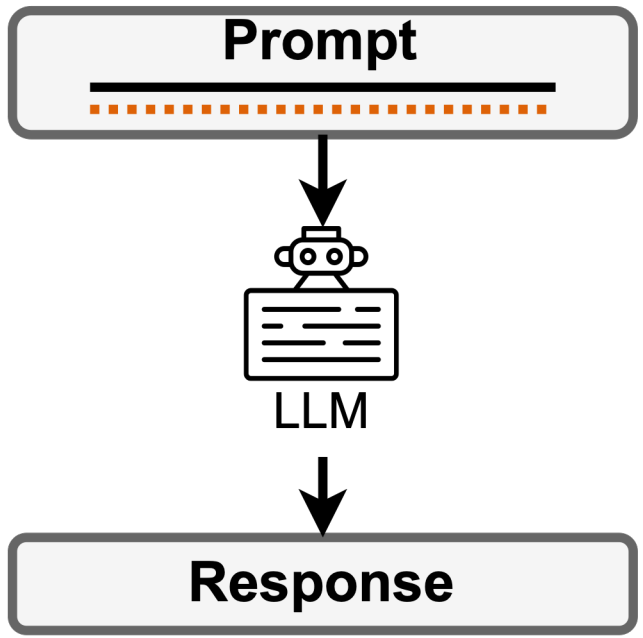


Design IT.
Create Knowledge.

www.hpi.de

# Just-in-Time Model Replacement

| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? → 1 |
| 2 | 1 | Broken after ... | ? → 0 |
| 3 | 1 | Super happy ... | ? → 1 |

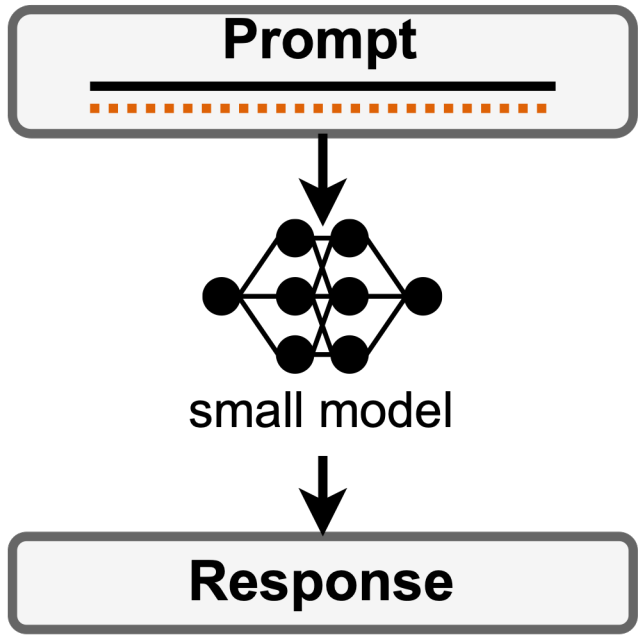



**Just-in-Time Model Replacement – JITR:**

- Simple, long-standing queries (e.g., sentiment classification) → use LLM

# Just-in-Time Model Replacement

| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? → 1 |
| 2 | 1 | Broken after ... | ? → 0 |
| 3 | 1 | Super happy ... | ? → 1 |

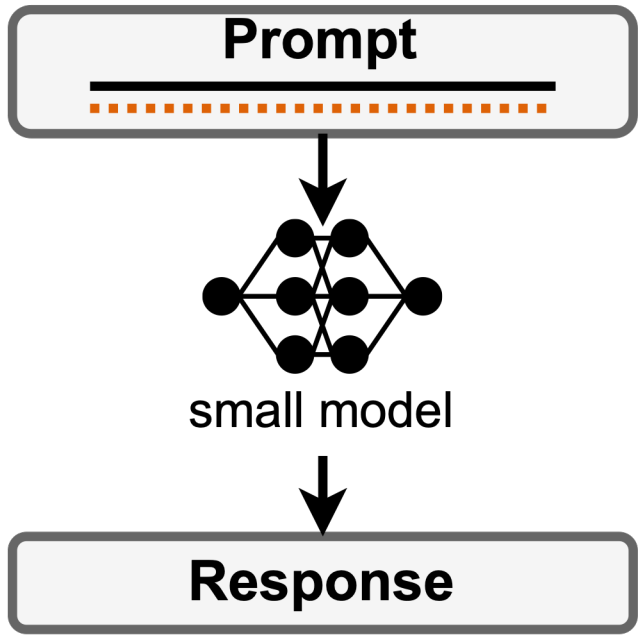



**Just-in-Time Model Replacement – JITR:**

- Simple, long-standing queries (e.g., sentiment classification) → use LLM
- Automatically replace LLM with smaller custom models *just-in-time*
- → Cost and resource savings without any overhead for the user

# Just-in-Time Model Replacement

| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? → 1 |
| 2 | 1 | Broken after ... | ? → 0 |
| 3 | 1 | Super happy ... | ? → 1 |

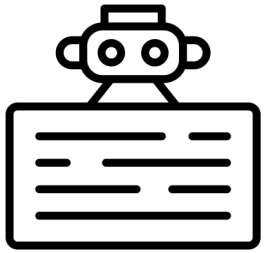



**Just-in-Time Model Replacement – JITR:**

- Simple, long-standing queries (e.g., sentiment classification) → use LLM
- Automatically replace LLM with smaller custom models *just-in-time*
- → Cost and resource savings without any overhead for the user

Why JITR → Prototype → Feasibility → Outlook

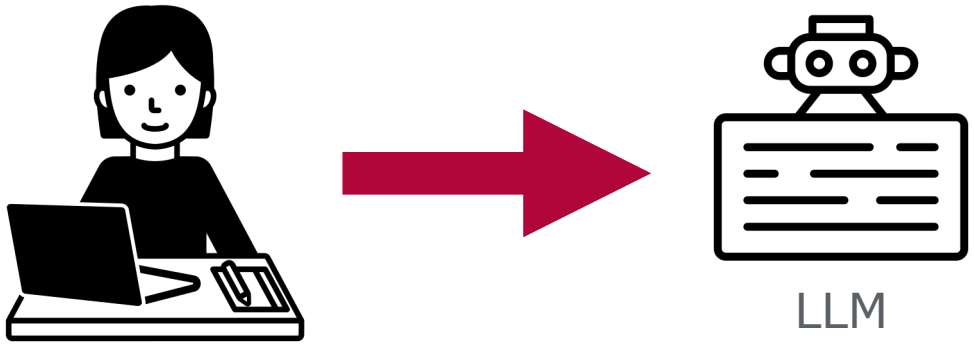
# LLMs



**LLMs (ML-as-a-Service)**

| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? → 1 |
| 2 | 1 | Broken after ... | ? → 0 |
| 3 | 1 | Super happy ... | ? → 1 |

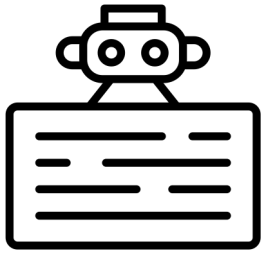
- State-of-the-art capabilities
- Simple API integration
- No AI expertise
- Zero model development cost
- No data collection
- No data labeling




**Offload simple recurring tasks to LLM**

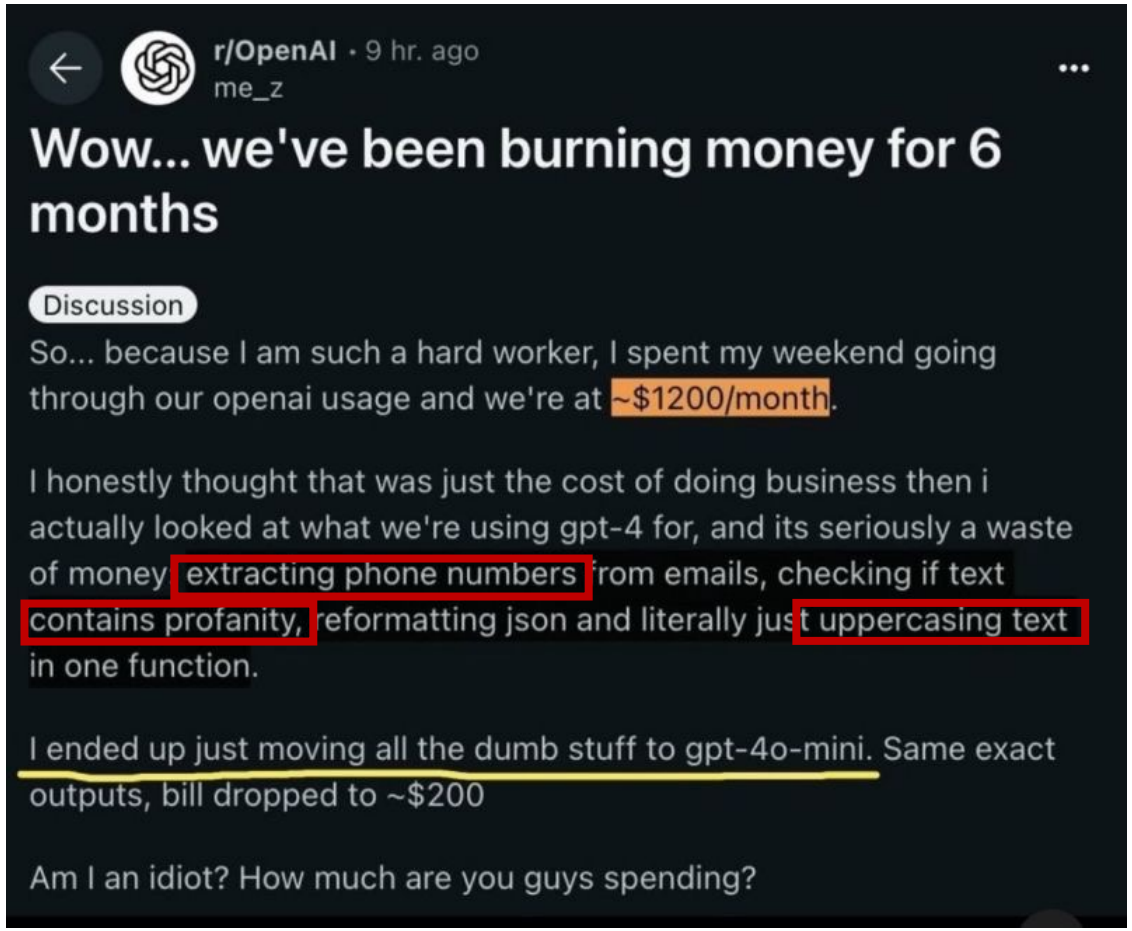
**As long as it runs → focus on something else**

# LLMs



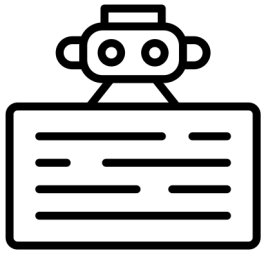
**LLMs (ML-as-a-Service)**

- ⊖ High inference costs
- ⊕ State-of-the-art capabilities
- ⊕ Simple API integration
- ⊕ No AI expertise
- ⊕ Zero model development cost
- ⊕ No data collection
- ⊕ No data labeling



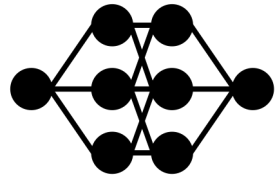

https://www.linkedin.com/posts/... (accessed 03.02.2026)

# LLMs vs Custom Models



**LLMs (ML-as-a-Service)**

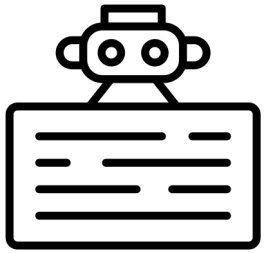
- ➖ High inference costs
- ➕ State-of-the-art capabilities
- ➕ Simple API integration
- ➕ No AI expertise
- ➕ Zero model development cost
- ➕ No data collection
- ➕ No data labeling



**Custom Models**

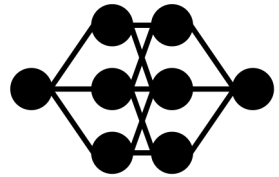
- ➕ Low inference costs

# LLMs vs Custom Models



**LLMs (ML-as-a-Service)**

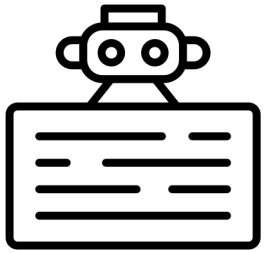
- ➖ High inference costs
- ➕ State-of-the-art capabilities
- ➕ Simple API integration
- ➕ No AI expertise
- ➕ Zero model development cost
- ➕ No data collection
- ➕ No data labeling



**Custom Models**

- ➕ Low inference costs
- ➖ (?) State-of-the-art capabilities
- ➖ (?) Simple API integration
- ➖ AI expertise needed
- ➖ High model development cost
- ➖ Data collection needed
- ➖ Data labeling needed

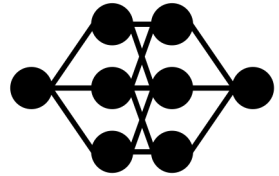
# LLMs vs Custom Models



**LLMs (ML-as-a-Service)**

- − High inference costs
- \+ State-of-the-art capabilities
- \+ Simple API integration
- \+ No AI expertise
- \+ Zero model development cost
- \+ No data collection
- \+ No data labeling

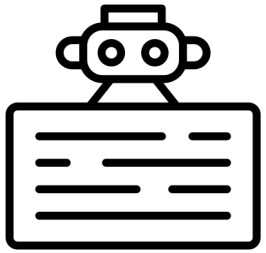
**Can we get the best of both worlds?**



**Custom Models**

- \+ Low inference costs
- − (?) State-of-the-art capabilities
- − (?) Simple API integration
- − AI expertise needed
- − High model development cost
- − Data collection needed
- − Data labeling needed

# LLMs vs Custom Models

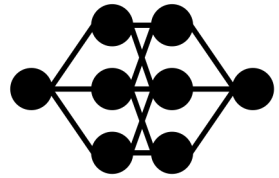


**LLMs (ML-as-a-Service)**

- − High inference costs
- \+ State-of-the-art capabilities
- \+ Simple API integration
- \+ No AI expertise
- \+ Zero model development cost
- \+ No data collection
- \+ No data labeling

**Just-in-Time Model Replacement**

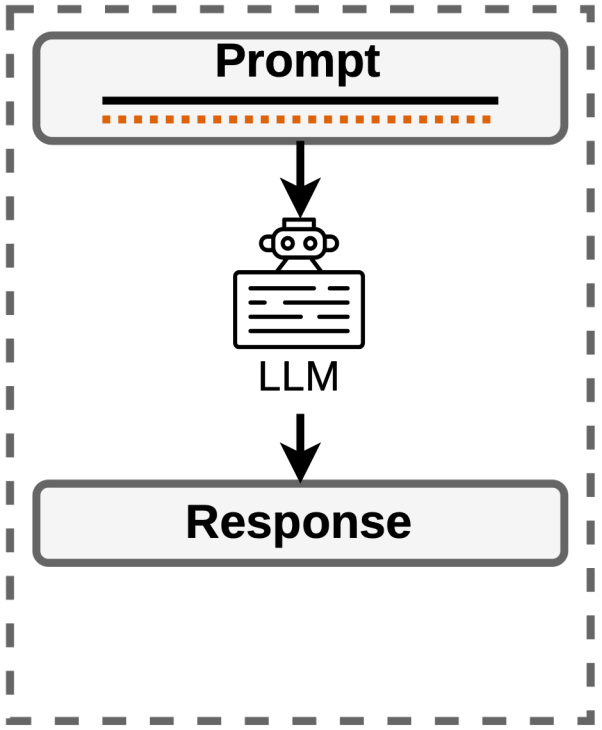
**Can we get the best of both worlds?**



**Custom Models**

- \+ Low inference costs
- − (?) State-of-the-art capabilities
- − (?) Simple API integration
- − AI expertise needed
- − High model development cost
- − Data collection needed
- − Data labeling needed

# Baseline LLM Usage

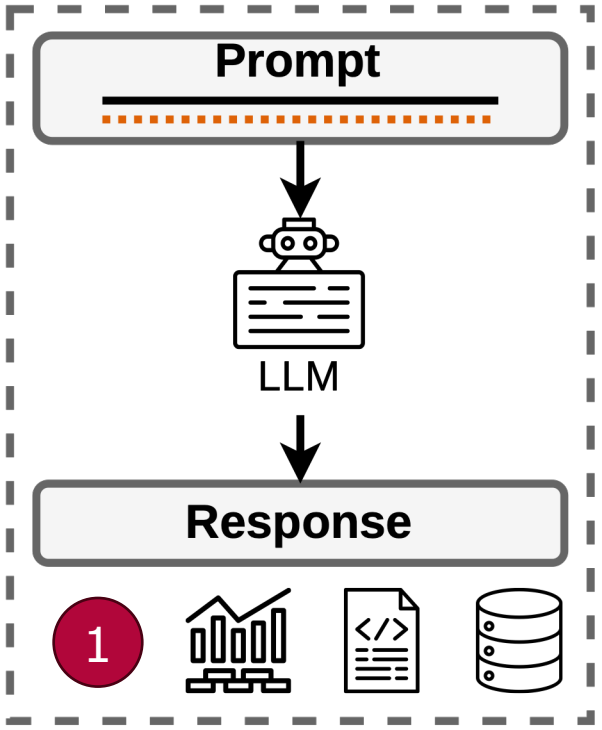



| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? ➔ 1 |
| 2 | 1 | Broken after ... | ? ➔ 0 |
| 3 | 1 | Super happy ... | ? ➔ 1 |

Baseline

- Embed relevant data in prompt
- Send to LLM
- Write response back to table

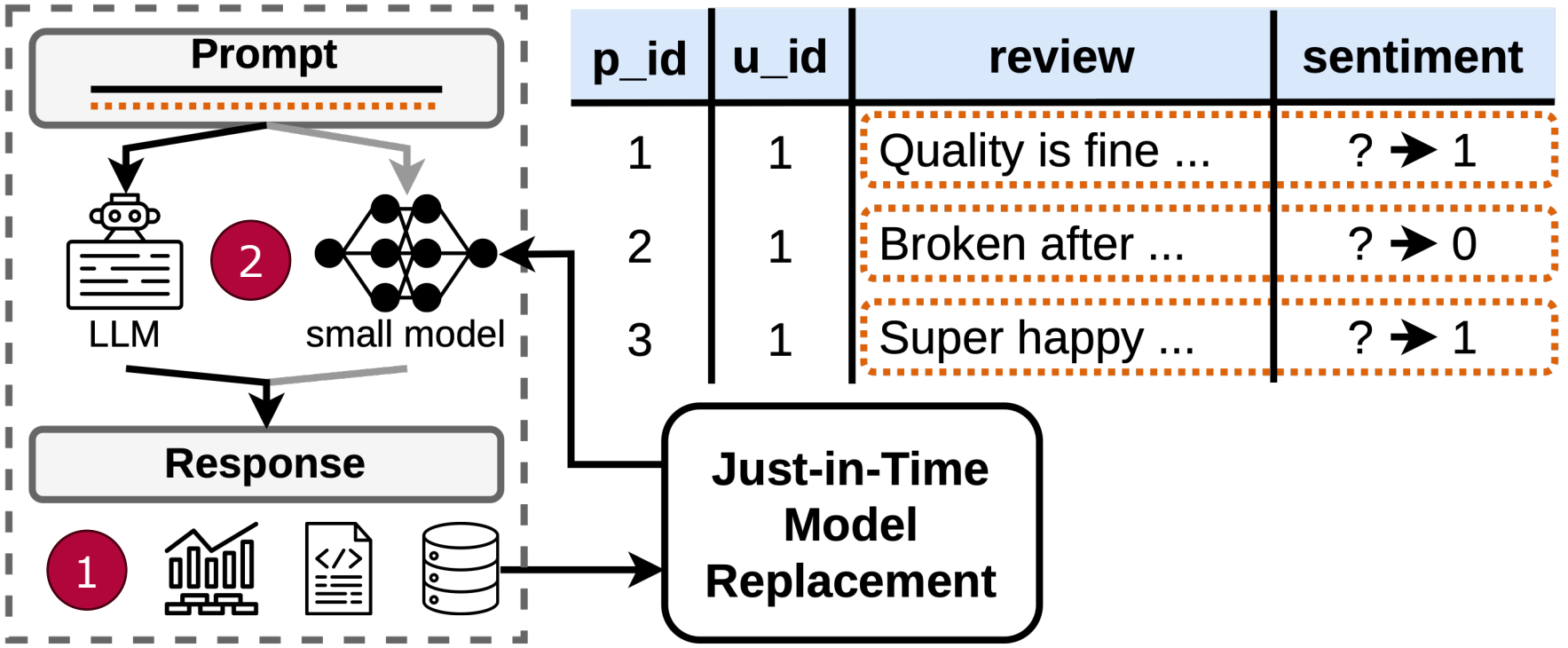
# Just-in-Time Model Replacement




| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? → 1 |
| 2 | 1 | Broken after ... | ? → 0 |
| 3 | 1 | Super happy ... | ? → 1 |

1 Collect LLM labels, task type, task metadata

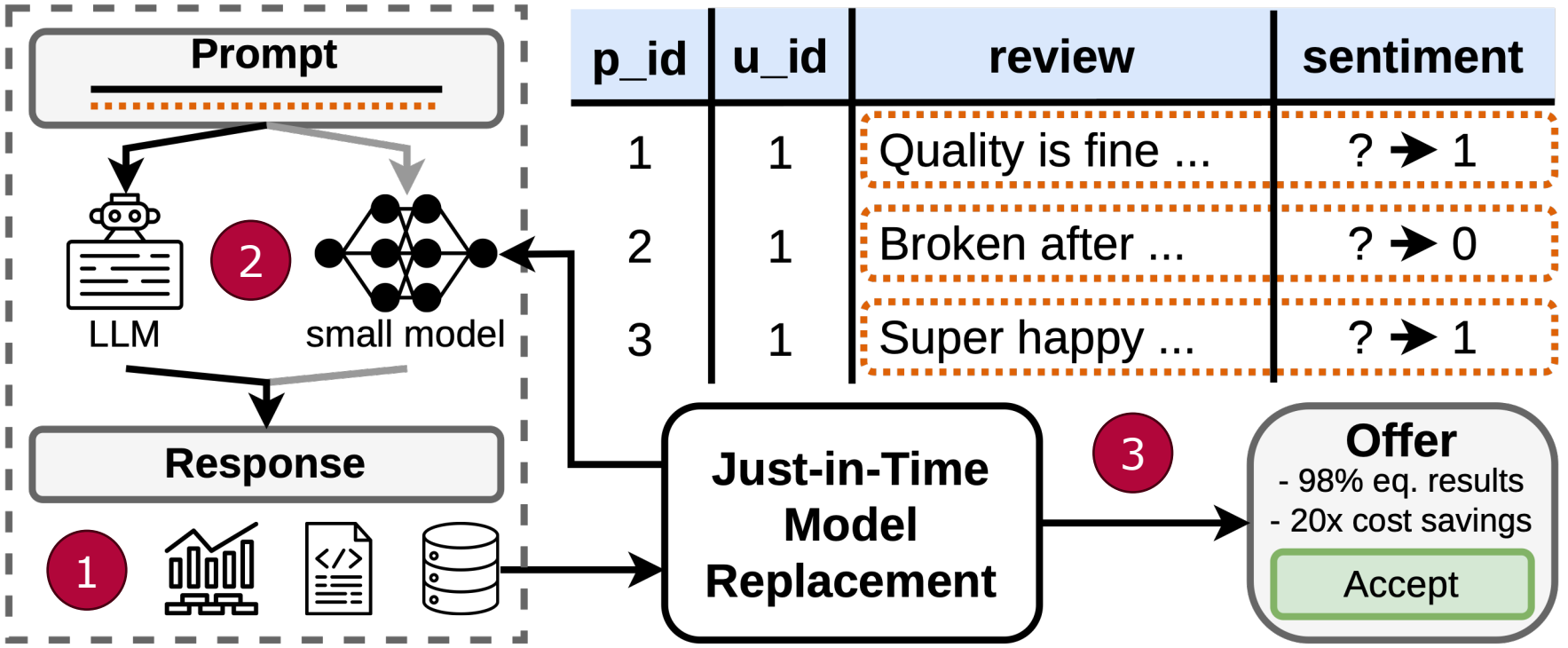
# Just-in-Time Model Replacement




| p_id | u_id | review | sentiment |
|---|---|---|---|
| 1 | 1 | Quality is fine ... | ? → 1 |
| 2 | 1 | Broken after ... | ? → 0 |
| 3 | 1 | Super happy ... | ? → 1 |

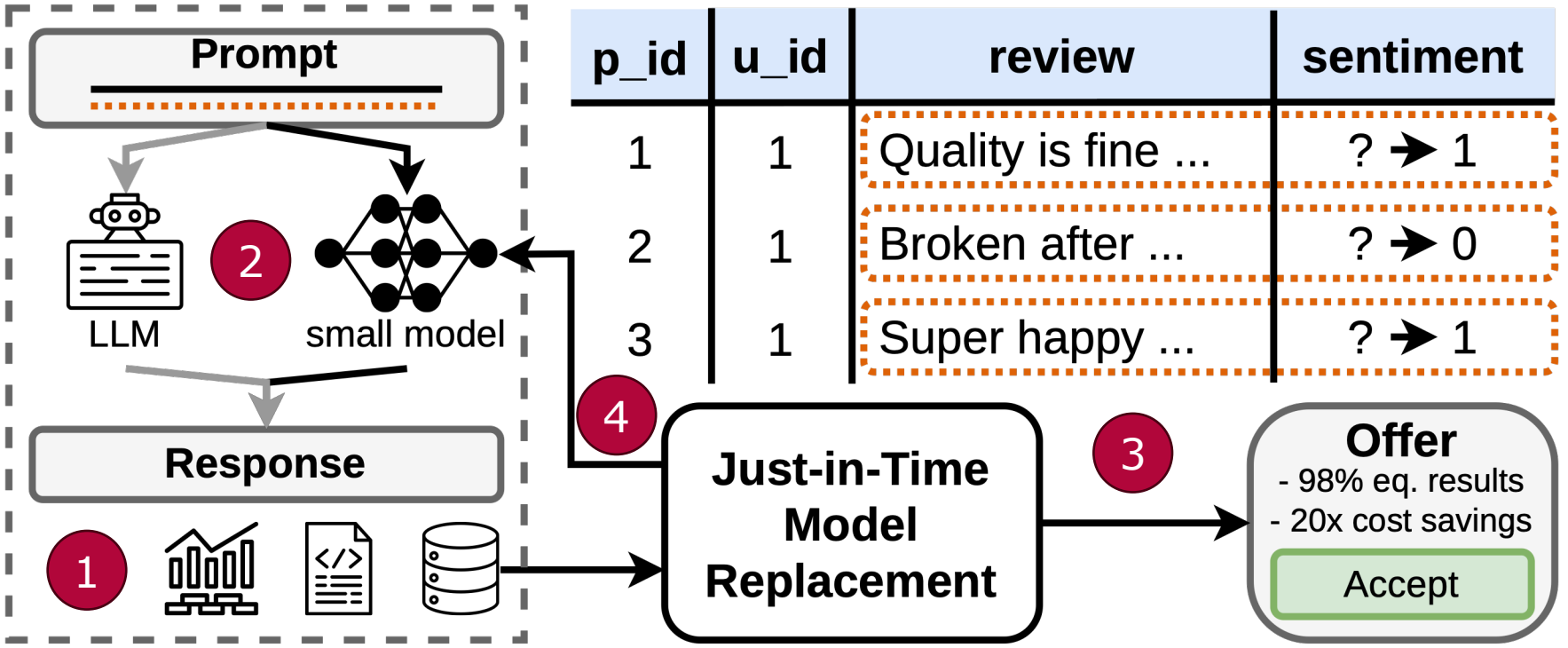
1. Collect LLM labels, task type, task metadata
2. Develop custom model → monitor model

# Just-in-Time Model Replacement




1. Collect LLM labels, task type, task metadata
2. Develop custom model → monitor model
3. (Ask user to accept switch)

# Just-in-Time Model Replacement




1. Collect LLM labels, task type, task metadata
2. Develop custom model → monitor model
3. (Ask user to accept switch)
4. Replace LLM with custom model

# Conditions to Meet For JITR to Be Effective

1) Tasks can be identified

# Conditions to Meet For JITR to Be Effective

1) Tasks can be identified

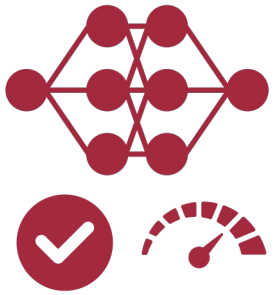
2) Logged requests and LLM responses provide sufficient training data

# Conditions to Meet For JITR to Be Effective

1) Tasks can be identified

2) Logged requests and LLM responses provide sufficient training data

3) Surrogate model has satisfactory performance



# Conditions to Meet For JITR to Be Effective

1) Tasks can be identified

2) Logged requests and LLM responses provide sufficient training data

3) Surrogate model has satisfactory performance

4) Monitoring can detect performance degradation

# Conditions to Meet For JITR to Be Effective

1) Tasks can be identified

2) Logged requests and LLM responses provide sufficient training data

3) Surrogate model has satisfactory performance

4) Monitoring can detect performance degradation

5) Cost of task identification, model development, and monitoring can be amortized

# Conditions to Meet For JITR to Be Effective

1) Tasks can be identified

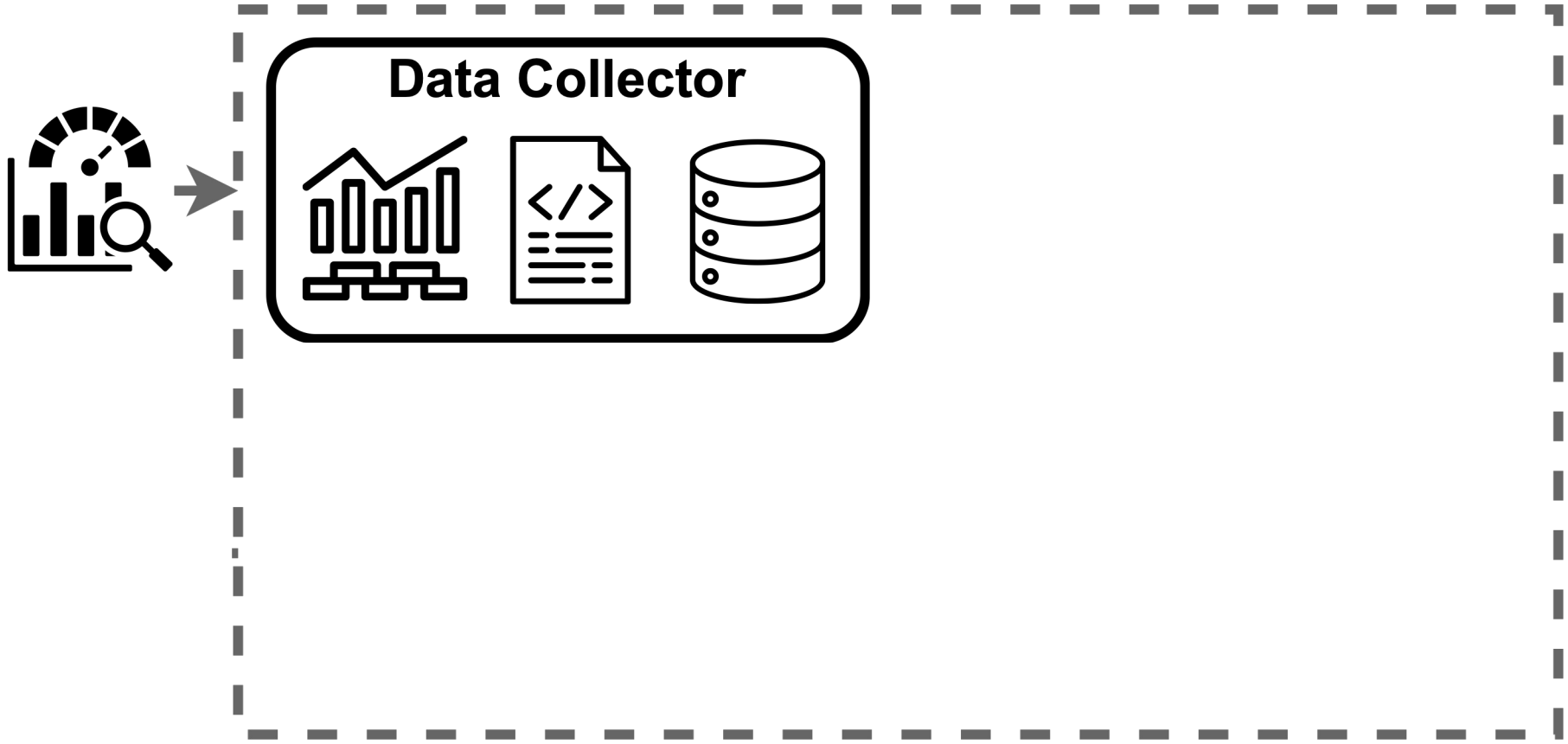
2) Logged requests and LLM responses provide sufficient training data

3) Surrogate model has satisfactory performance

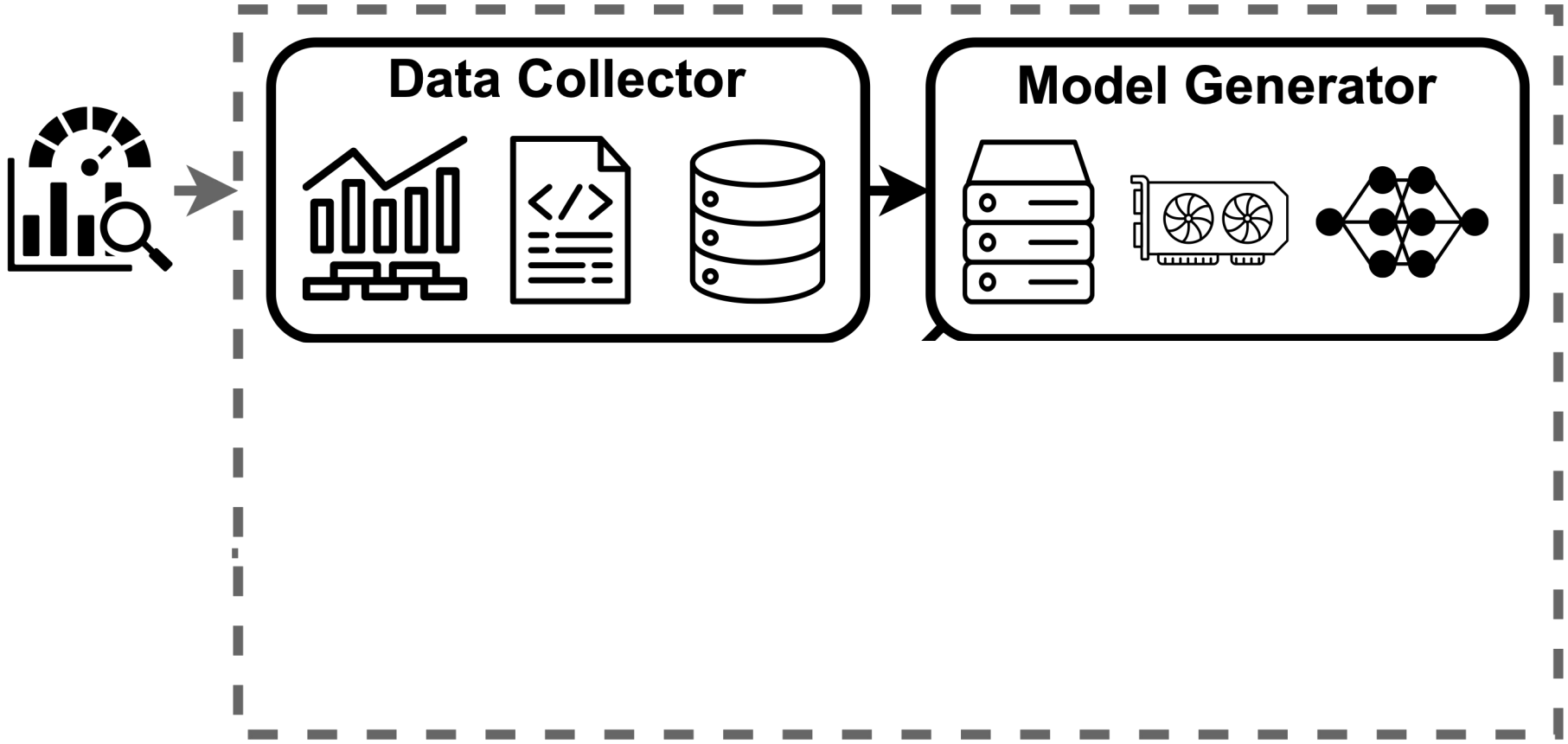
4) Monitoring can detect performance degradation

**5) Cost** of task identification, model development, and monitoring **can be amortized**

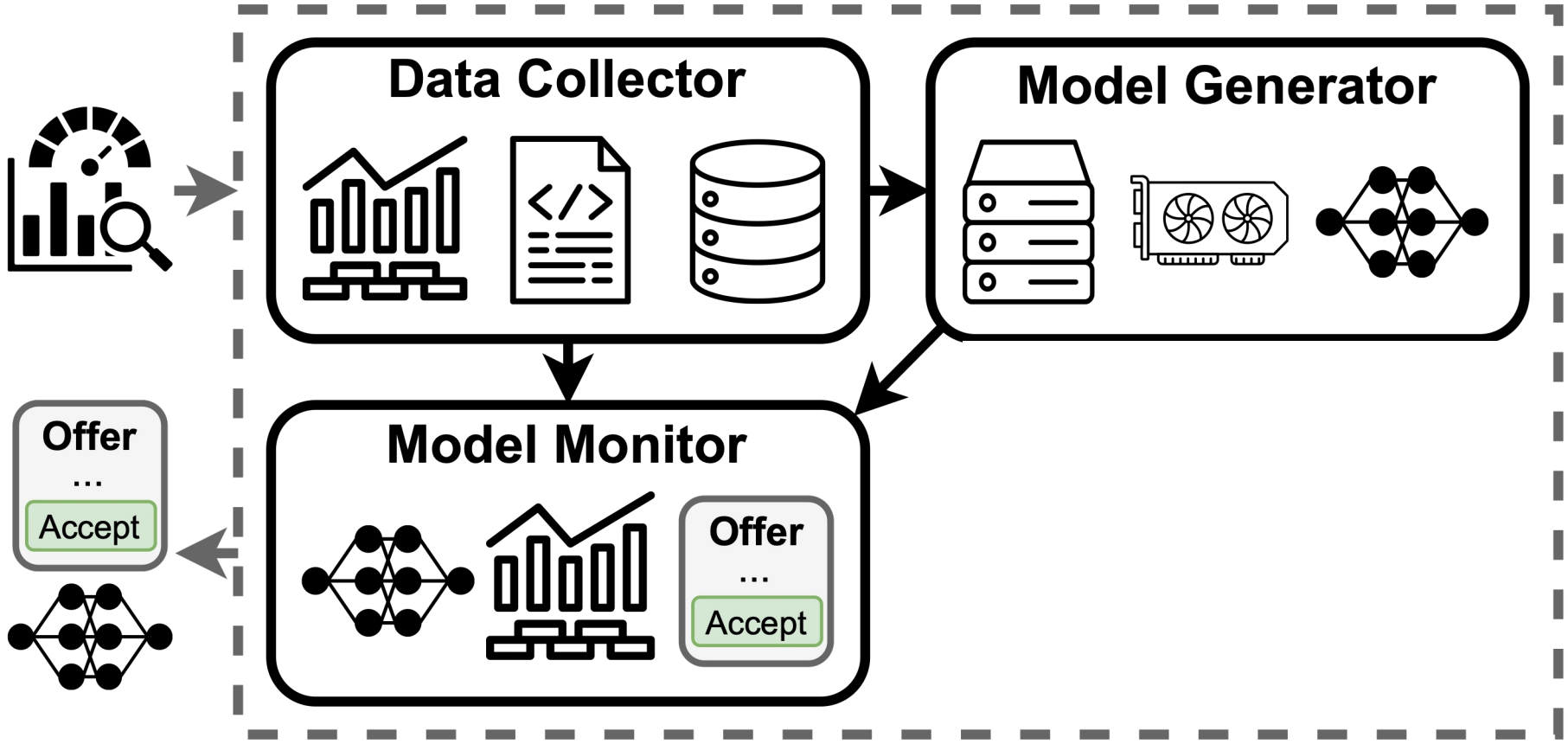
# Poodle: Just-in-Time Model Replacement Prototype



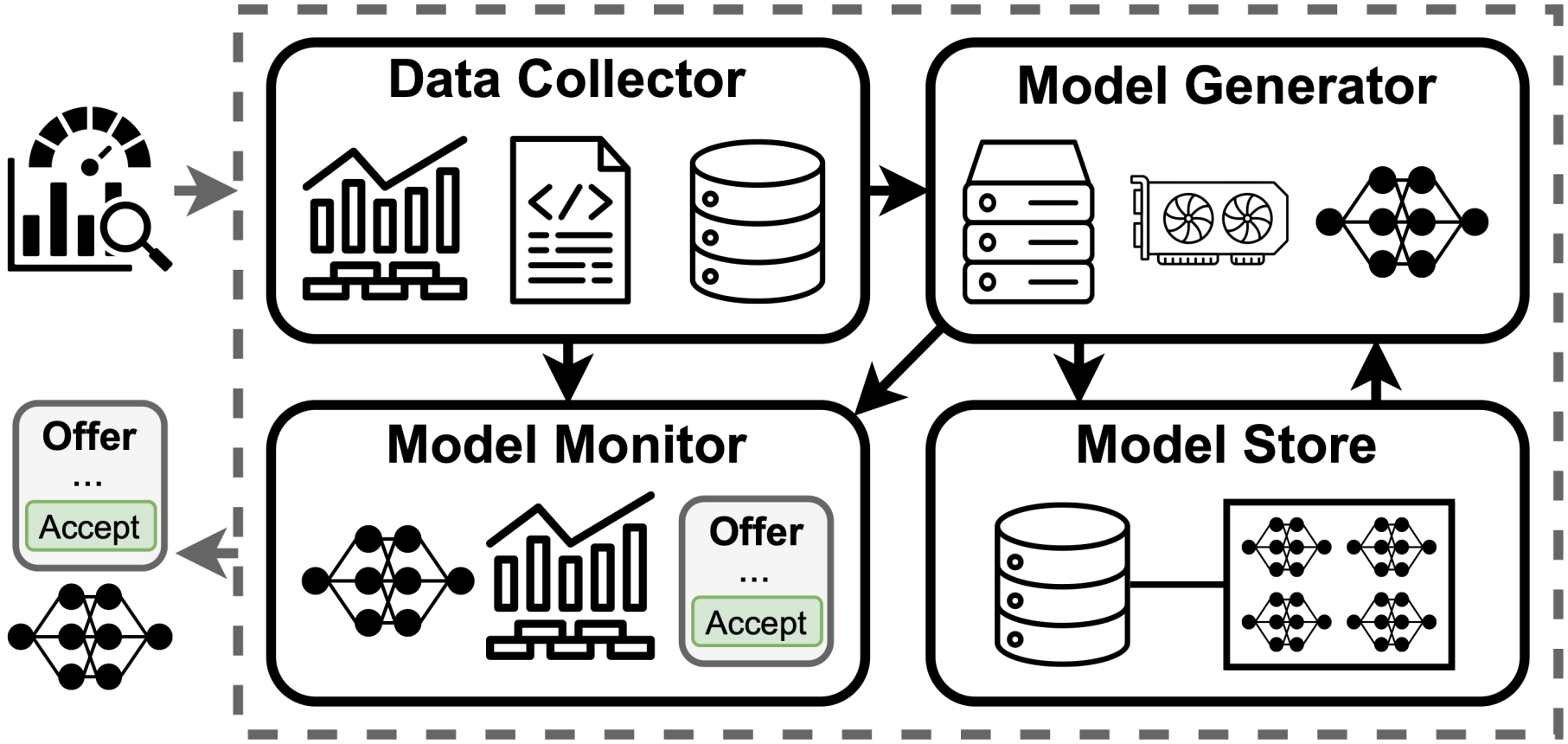
# Poodle: Just-in-Time Model Replacement Prototype



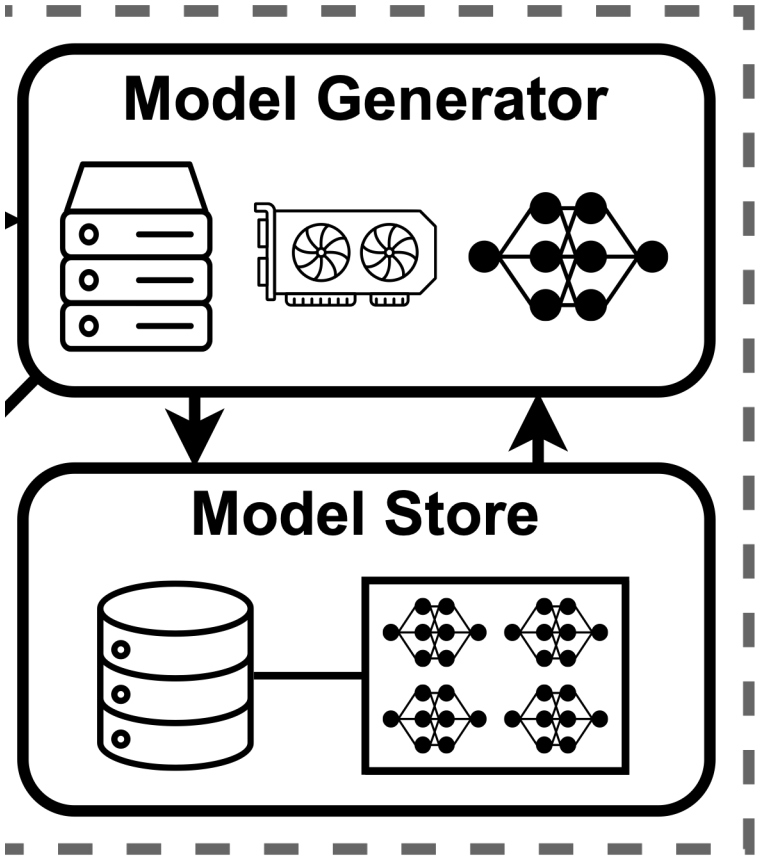
# Poodle: Just-in-Time Model Replacement Prototype



# Poodle: Just-in-Time Model Replacement Prototype

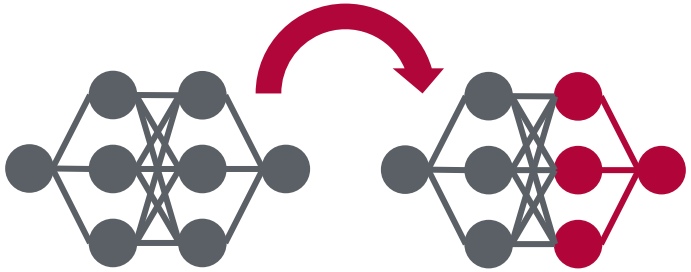


# Poodle: Just-in-Time Model Replacement Prototype



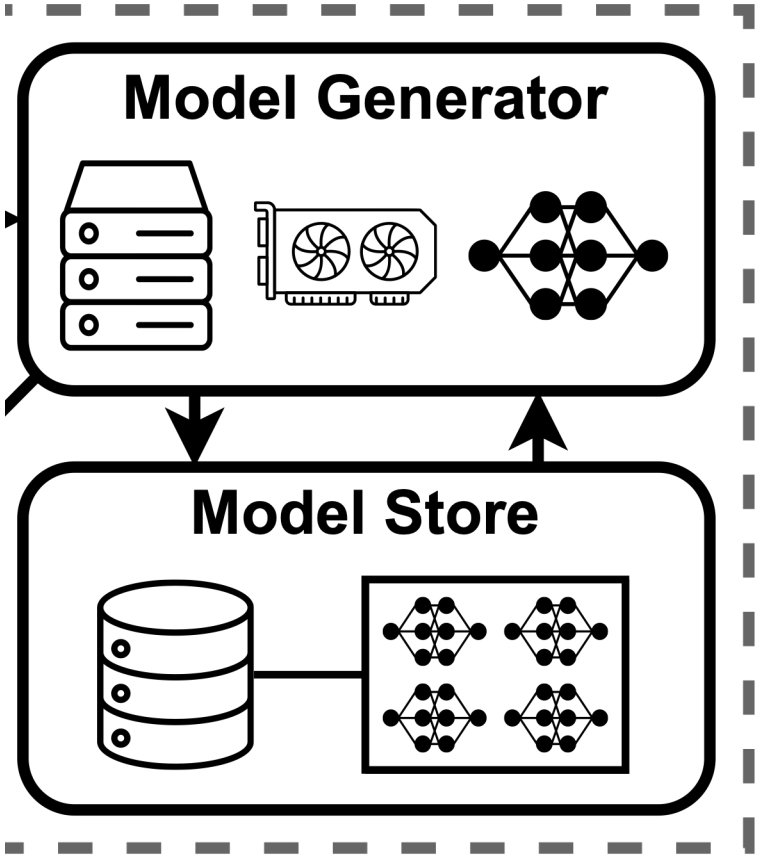

**Baseline Model Development**

- Label or collect training data
- Fine-tune a “standard” model



Transfer Learning
(fine-tuning)

# Poodle: Just-in-Time Model Replacement Prototype

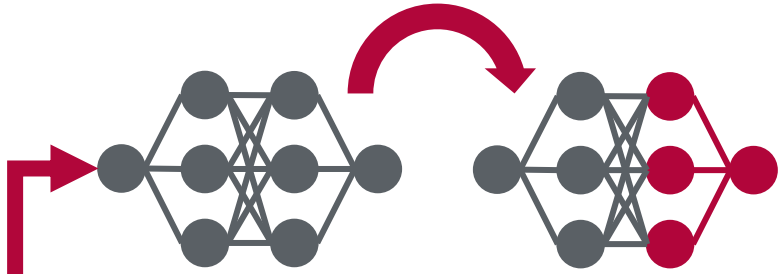



**Baseline Model Development**

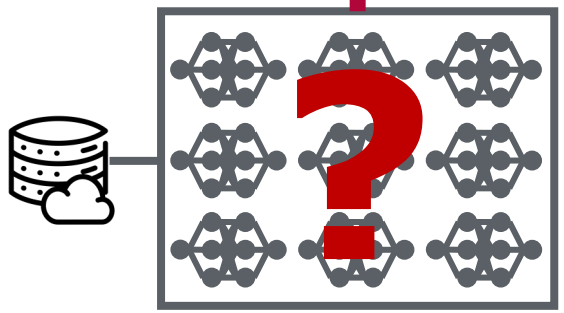
- Label or collect training data
- Fine-tune a “standard” model
- **Model Search**

→ **Fine-Tuning [25, 26]**

- ⊕ Less training data
- ⊕ Faster convergence
- ⊕ Higher accuracy



Transfer Learning
(fine-tuning)



Models in a model store

# Poodle – Preliminary Results

**Costs** 

**Inference Time** 

**Accuracy** 

**Model Development** 

# Poodle – Preliminary Results

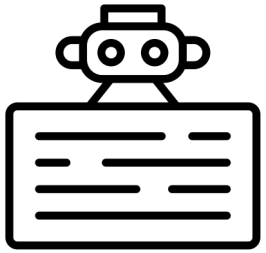
**Costs** 

- JITR amortizes overhead
- Significant cost savings

# Poodle – Preliminary Results

**Costs**

- JITR amortizes overhead
- Significant cost savings



**VS**

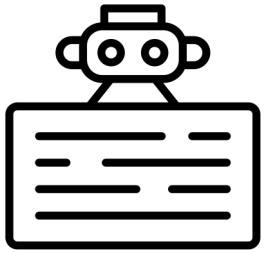
**Just-in-Time Model Replacement**

- Wrap requests in an additional prompt → LLM

# Poodle – Preliminary Results

**Costs** $

- JITR amortizes overhead
- Significant cost savings



**VS**

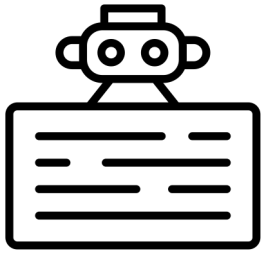
**Just-in-Time Model Replacement**

- Wrap requests in an additional prompt → LLM
- Collect 5,000 labels
- Training of a small model
- Use a small model instead

# Poodle – Preliminary Results

**Costs** $

- JITR amortizes overhead
- Significant cost savings



**VS**

**Just-in-Time Model Replacement**

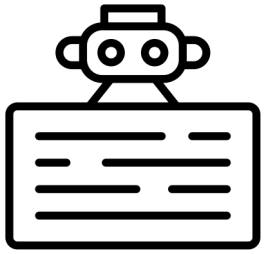
- Wrap requests in an additional prompt → LLM
- Collect 5,000 labels
- Training of a small model
- Use a small model instead

| Model | Input | Output | Provider |
|---|---|---|---|
| GPT-4.1 | $2.00 | $8.00 | OpenAI |
| GPT-4.1-nano | $0.10 | $0.40 | OpenAI |
| Llama 405B Turbo | $3.50 | $3.50 | TogetherAI |
| Llama 8B | $0.20 | $0.20 | TogetherAI |
| BERT 80M | $0.01 | $0.01 | TogetherAI |

# Poodle – Preliminary Results

**Costs** $

- JITR amortizes overhead
- Significant cost savings



**VS**

**Just-in-Time Model Replacement**

- Wrap requests in an additional prompt → LLM
- Collect 5,000 labels
- Training of a small model
- Use a small model instead

| Model | Input | Output | Provider |
|---|---|---|---|
| GPT-4.1 | $2.00 | $8.00 | OpenAI |
| GPT-4.1-nano | $0.10 | $0.40 | OpenAI |
| Llama 405B Turbo | $3.50 | $3.50 | TogetherAI |
| Llama 8B | $0.20 | $0.20 | TogetherAI |
| BERT 80M | $0.01 | $0.01 | TogetherAI |

**GPT-4.1 nano**

**GPT-4.1**

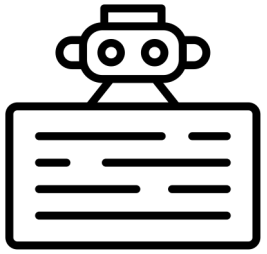
**Break-even**

**Savings 1M requests**

# Poodle – Preliminary Results

**Costs** $

- JITR amortizes overhead
- Significant cost savings

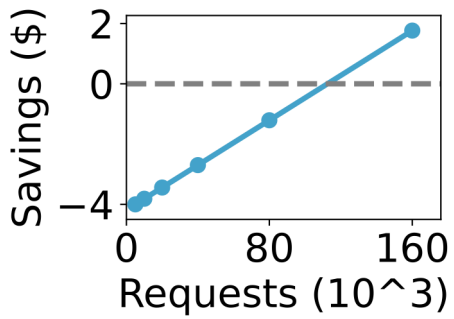


**VS**

**Just-in-Time Model Replacement**

- Wrap requests in an additional prompt → LLM
- Collect 5,000 labels
- Training of a small model
- Use a small model instead

| Model | Input | Output | Provider |
|---|---|---|---|
| GPT-4.1 | $2.00 | $8.00 | OpenAI |
| GPT-4.1-nano | $0.10 | $0.40 | OpenAI |
| Llama 405B Turbo | $3.50 | $3.50 | TogetherAI |
| Llama 8B | $0.20 | $0.20 | TogetherAI |
| BERT 80M | $0.01 | $0.01 | TogetherAI |

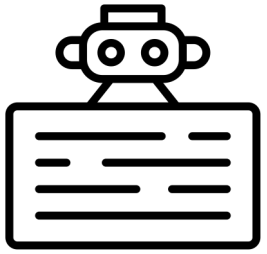



| | GPT-4.1 nano | GPT-4.1 |
|---|---|---|
| **Break-even** | 100,000 | |
| **Savings 1M requests** | $33 | |

# Poodle – Preliminary Results

**Costs** $

- JITR amortizes overhead
- Significant cost savings

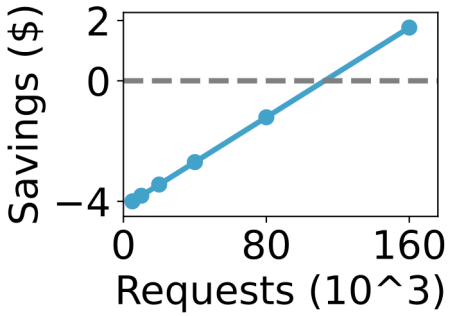


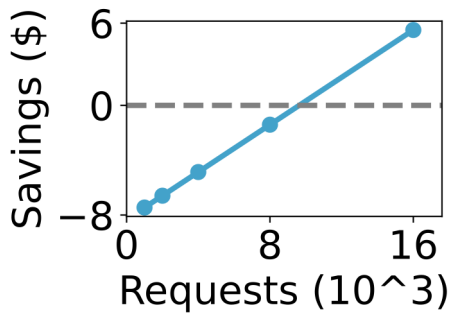
**VS**

**Just-in-Time Model Replacement**

- Wrap requests in an additional prompt → LLM
- Collect 5,000 labels
- Training of a small model
- Use a small model instead

| Model | Input | Output | Provider |
|---|---|---|---|
| GPT-4.1 | $2.00 | $8.00 | OpenAI |
| GPT-4.1-nano | $0.10 | $0.40 | OpenAI |
| Llama 405B Turbo | $3.50 | $3.50 | TogetherAI |
| Llama 8B | $0.20 | $0.20 | TogetherAI |
| BERT 80M | $0.01 | $0.01 | TogetherAI |




| | GPT-4.1 nano | GPT-4.1 |
|---|---|---|
| **Break-even** | 100,000 | <10,000 |
| **Savings 1M requests** | $33 | $850 |

# Poodle – Preliminary Results

**Costs**

- JITR amortizes overhead
- Significant cost savings

**Inference Time**

- Even for small LLMs JITR reduces inference time

**Setup**

Evaluate break-even (number of requests):
**LLM** (Llama-2-7B) VS **JITR** with Poodle (BERT)

IMDB dataset; NVIDIA RTX A5000 GPU; switch after 1K requests

(1) **LLM** and **JITR** break-even in less than 100,000 requests.

(2) BERT processes 2.4x more items per second.

(3) For 1M requests Poodle takes 1.5x less time; For 2M requests more than 1hr.

(4) Real LLMs orders of magnitude larger than Llama-2-7B

# Poodle – Preliminary Results

**Costs**

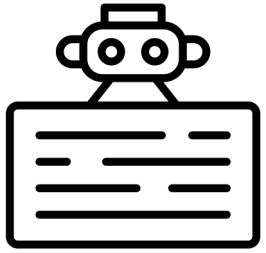
- JITR amortizes overhead
- Significant cost savings

**Inference Time**

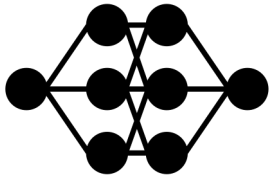
- Even for small LLMs JITR reduces inference time

**Accuracy**

- Small, specialized models reach competitive accuracy



Llama 2 7B
Accuracy: 0.93
1024 Tokens



BERT, no tuning
Accuracy: 0.89 (0.92)
256 Tokens

| # Items | Ground Truth | | LLM-Generated | |
|---|---|---|---|---|
| | Accuracy | Epochs | Accuracy | Epochs |
| 500 | 0.86 | 9 | 0.88 | 6 |
| 1,000 | 0.88 | 5 | 0.88 | 7 |
| 2,000 | 0.89 | 3 | 0.88 | 5 |
| 5,000 | 0.90 | 3 | 0.90 | 2 |

Related work [6, 7, 10, 12, 23]
Across multiple studies, fine-tuned small models outperform LLMs for text classification tasks

# Poodle – Preliminary Results

**Costs**

- JITR amortizes overhead
- Significant cost savings

**Inference Time**

- Even for small LLMs JITR reduces inference time

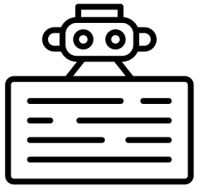
**Accuracy**

- Small, specialized models reach competitive accuracy

**Model Development**

- Model Search → Fine-Tuning outperforms alternative approaches in dev. time, accuracy, and required data

# Poodle – Preliminary Results



Llama 2 7B
Accuracy: 0.93

**Costs**

- JITR amortizes overhead
- Significant cost savings

**Inference Time**

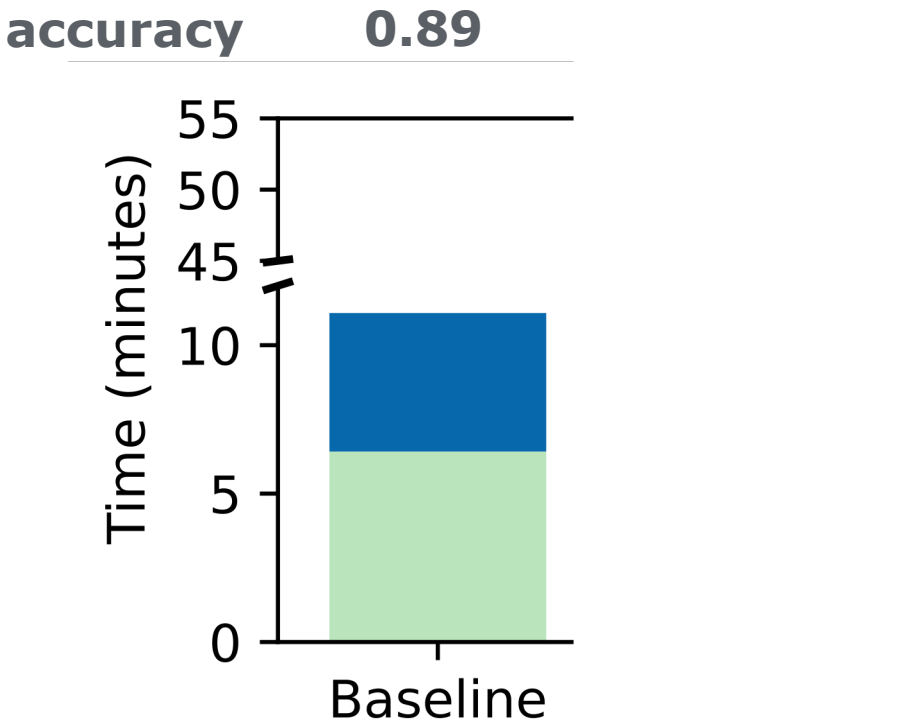
- Even for small LLMs JITR reduces inference time

**Accuracy**

- Small, specialized models reach competitive accuracy

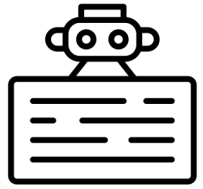
**Model Development**

- Model Search → Fine-Tuning outperforms alternative approaches in dev. time, accuracy, and required data




**Baseline:** Fine-tune BERT

# Poodle – Preliminary Results



Llama 2 7B
Accuracy: 0.93

**Costs**

- JITR amortizes overhead
- Significant cost savings

**Inference Time**

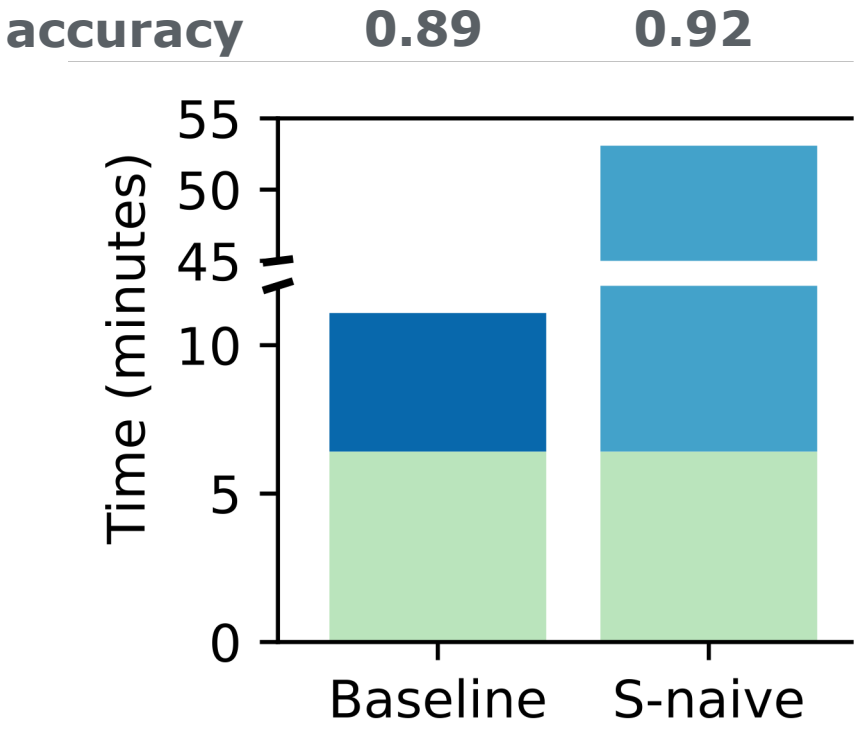
- Even for small LLMs JITR reduces inference time

**Accuracy**

- Small, specialized models reach competitive accuracy

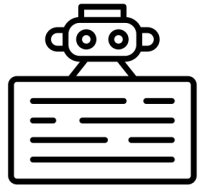
**Model Development**

- Model Search → Fine-Tuning outperforms alternative approaches in dev. time, accuracy, and required data




**Baseline:** Fine-tune BERT
**S-naïve:** Exhaustive search over 10 hand-selected models from HuggingFace

# Poodle – Preliminary Results



Llama 2 7B
Accuracy: 0.93

**Costs**

- JITR amortizes overhead
- Significant cost savings

**Inference Time**

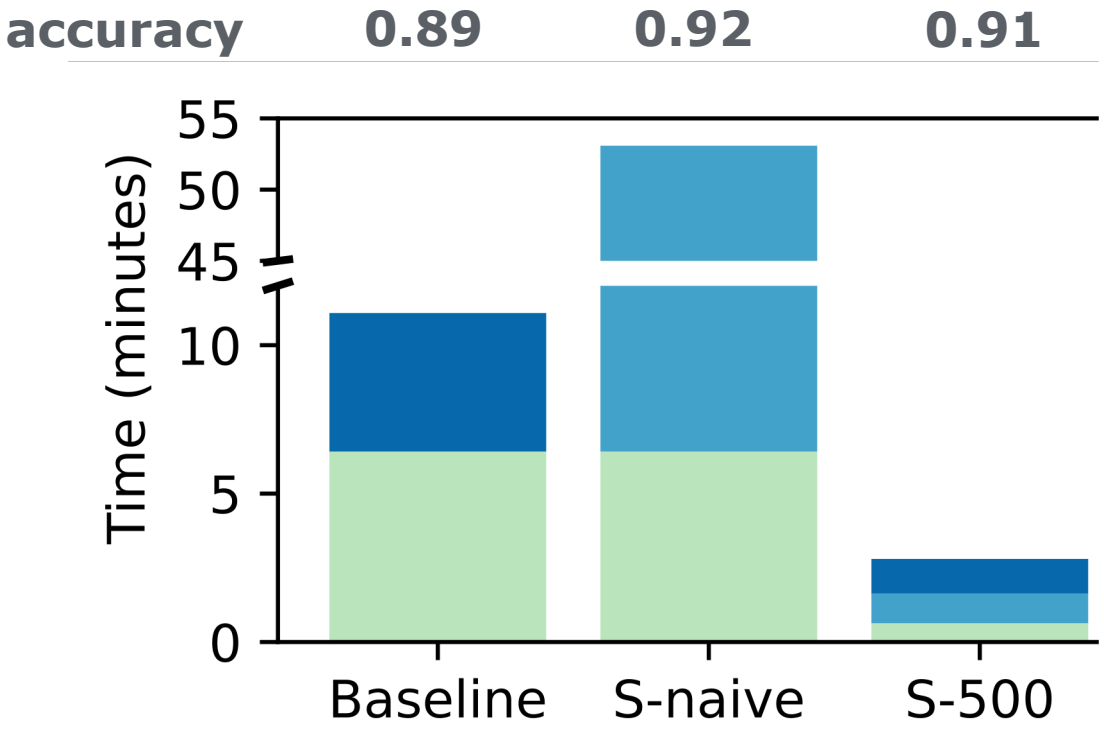
- Even for small LLMs JITR reduces inference time

**Accuracy**

- Small, specialized models reach competitive accuracy

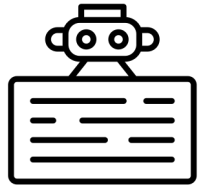
**Model Development**

- Model Search → Fine-Tuning outperforms alternative approaches in dev. time, accuracy, and required data




**Baseline:** Fine-tune BERT
**S-naïve:** Exhaustive search over 10 hand-selected models from HuggingFace
**S-500:** Search + fine-tune on 500 items

# Poodle – Preliminary Results



Llama 2 7B
Accuracy: 0.93

**Costs**

- JITR amortizes overhead
- Significant cost savings

**Inference Time**

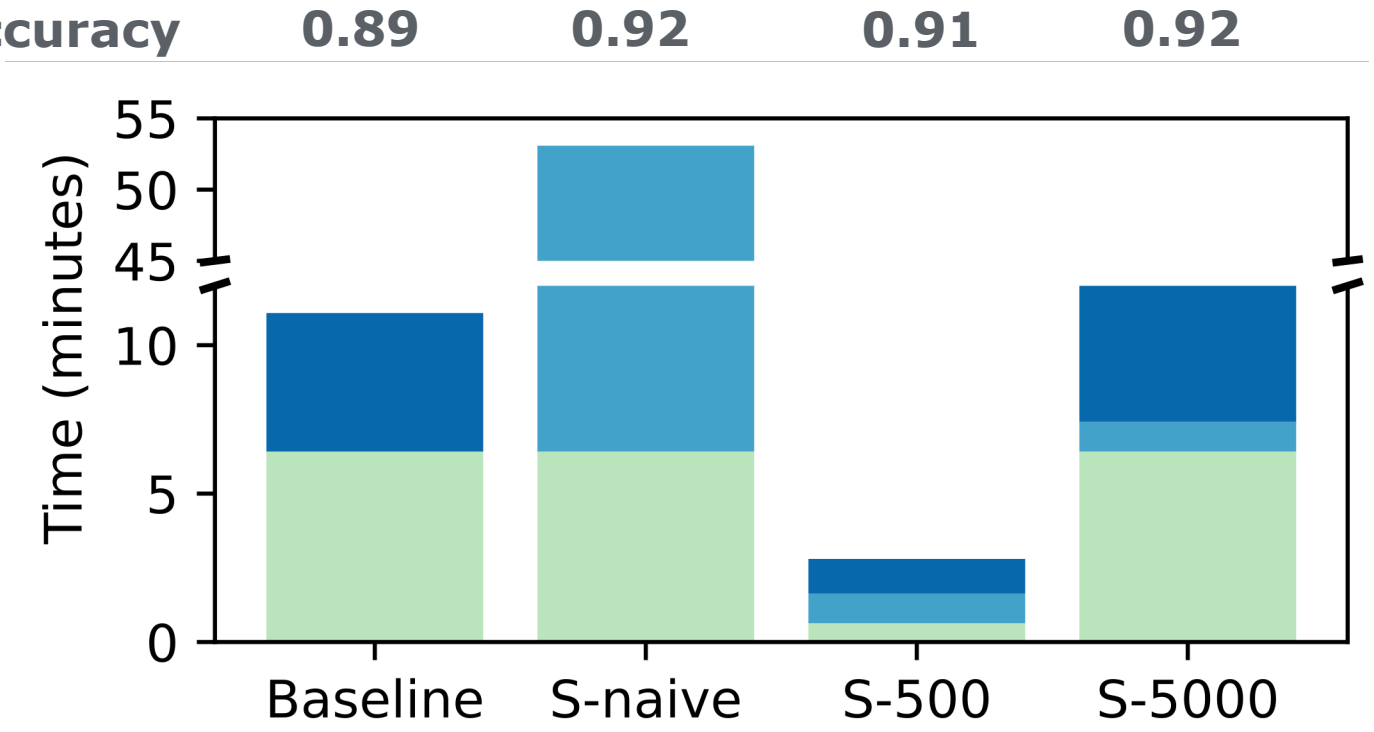
- Even for small LLMs JITR reduces inference time

**Accuracy**

- Small, specialized models reach competitive accuracy

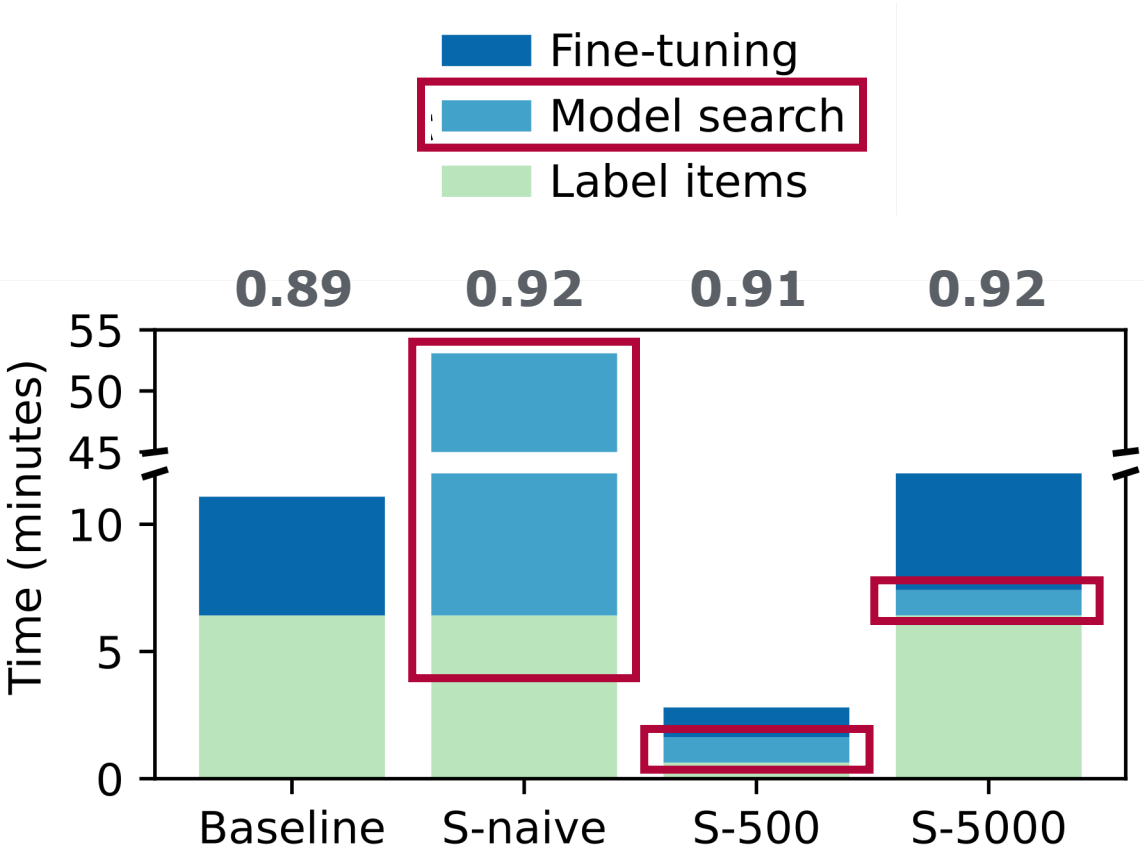
**Model Development**

- Model Search → Fine-Tuning outperforms alternative approaches in dev. time, accuracy, and required data



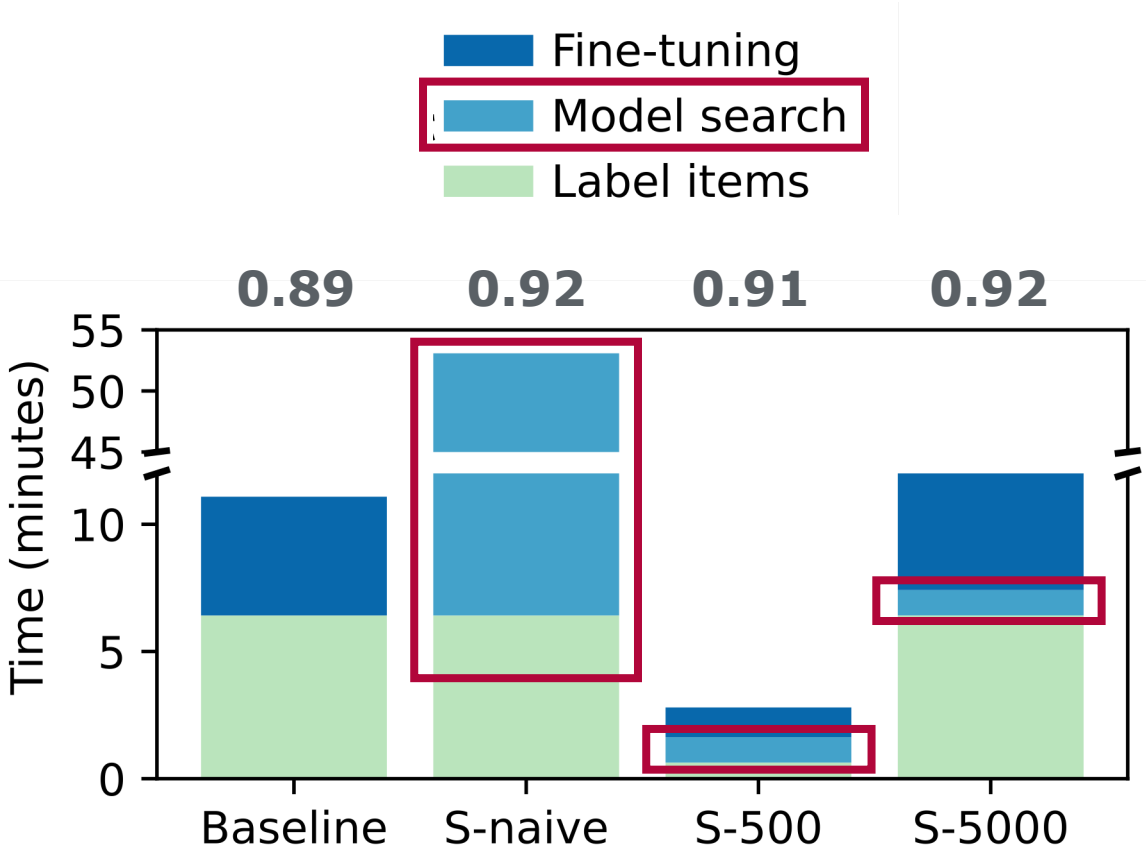

**Baseline:** Fine-tune BERT
**S-naïve:** Exhaustive search over 10 hand-selected models from HuggingFace
**S-500:** Search + fine-tune on 500 items
**S-5000:** Search on 500 items
+ fine-tune on 5,000 items

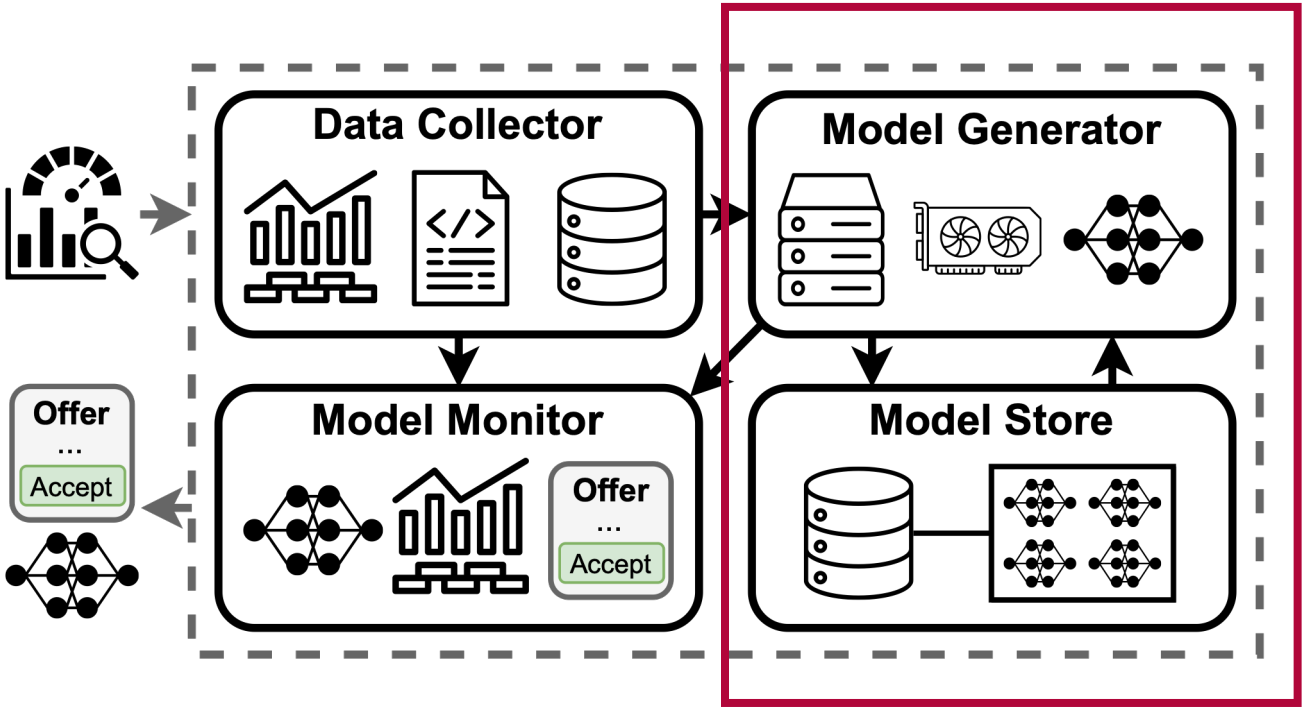
# The Road Ahead




**Baseline:** Fine-tune BERT
**S-naïve:** Exhaustive search over **10 hand-selected models** from HuggingFace
**S-500:** Search + fine-tune on 500 items
**S-5000:** Search on 500 items
+ fine-tune on 5,000 items

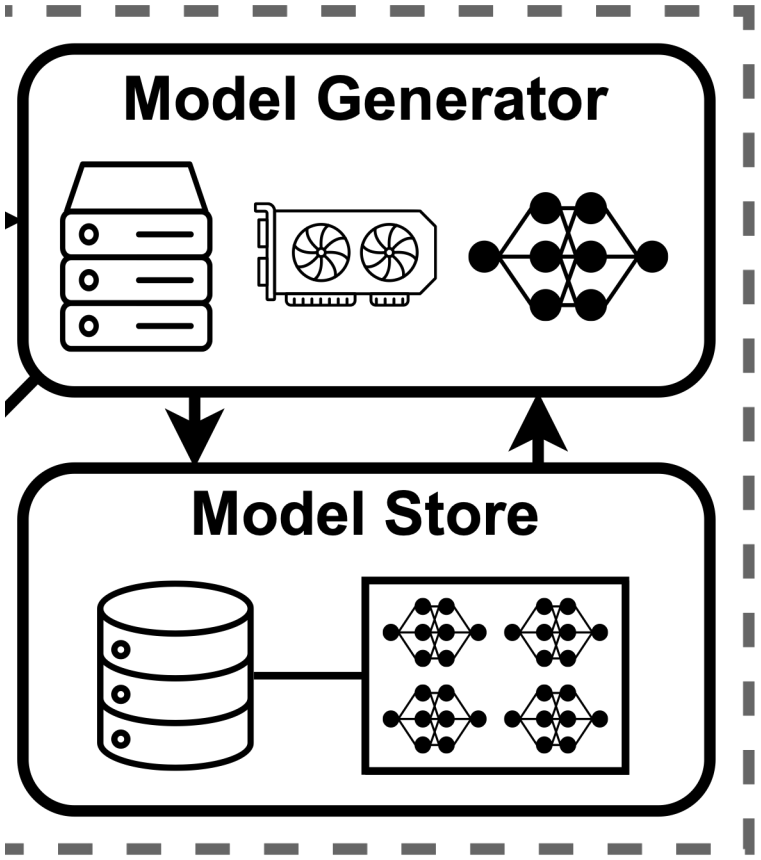
# The Road Ahead




**Baseline:** Fine-tune BERT
**S-naïve:** Exhaustive search over **10 hand-selected models** from HuggingFace
**S-500:** Search + fine-tune on 500 items
**S-5000:** Search on 500 items
+ fine-tune on 5,000 items

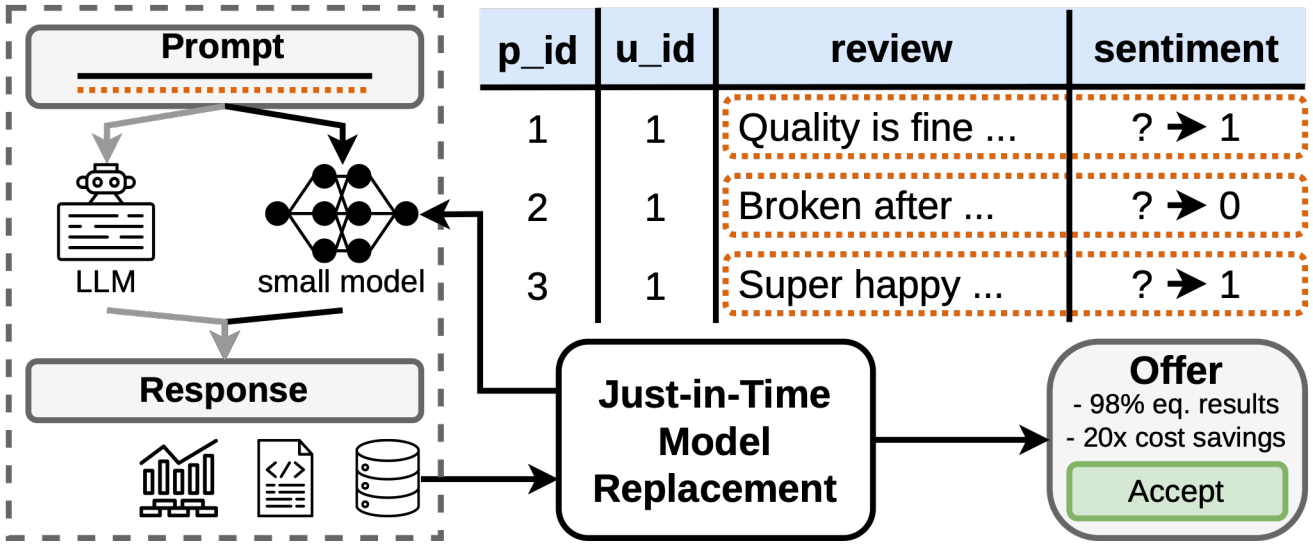
# Co-Design Model Search and Model Store

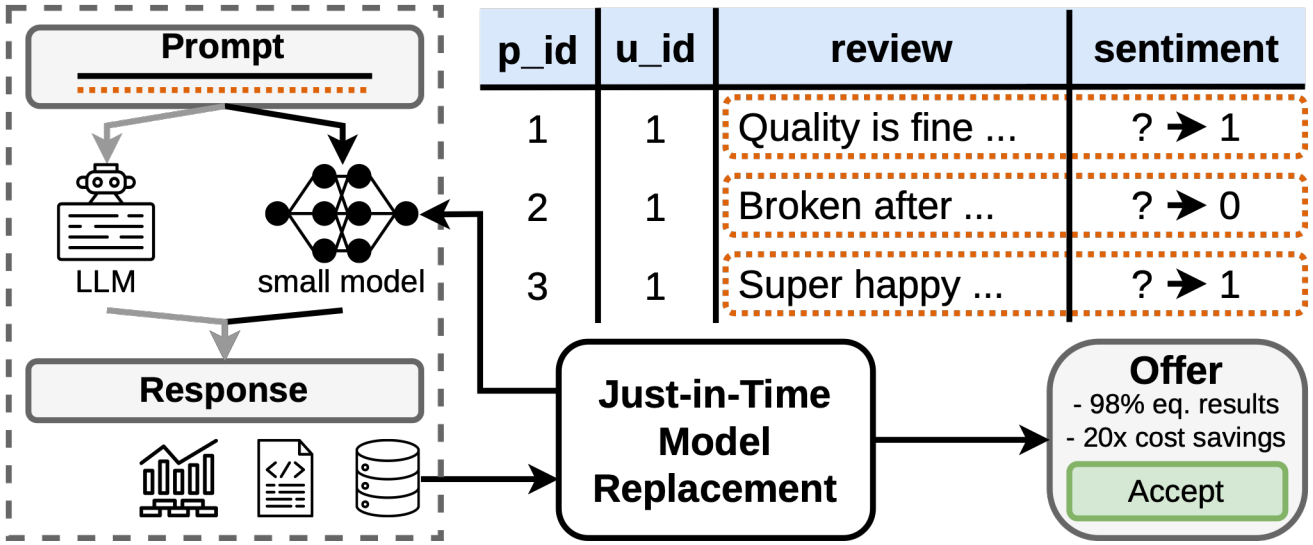


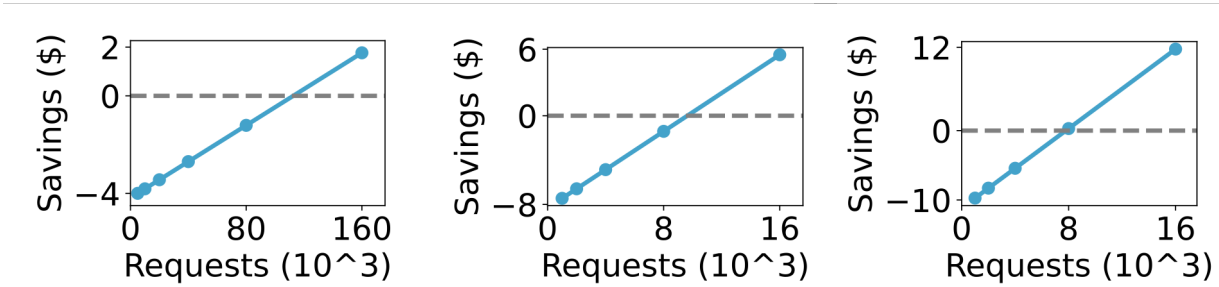

Two main directions

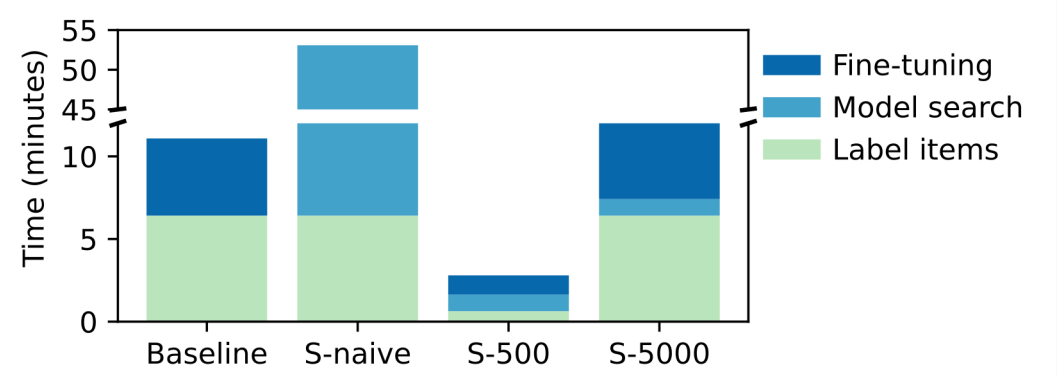
(1) Approximate model search

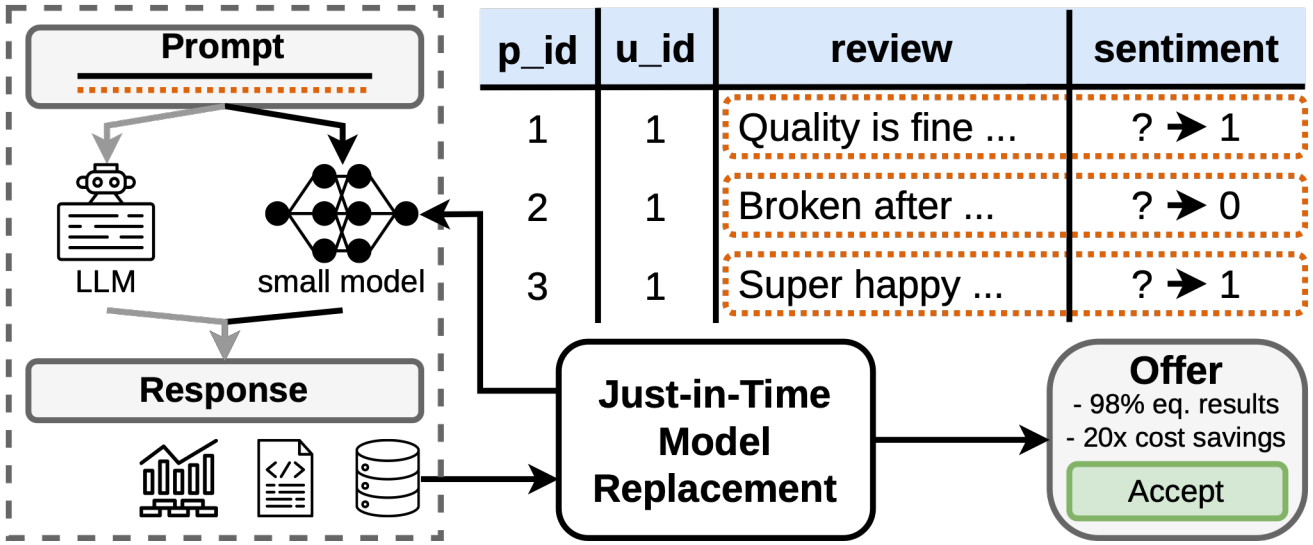
(2) Co-design model store and model search techniques

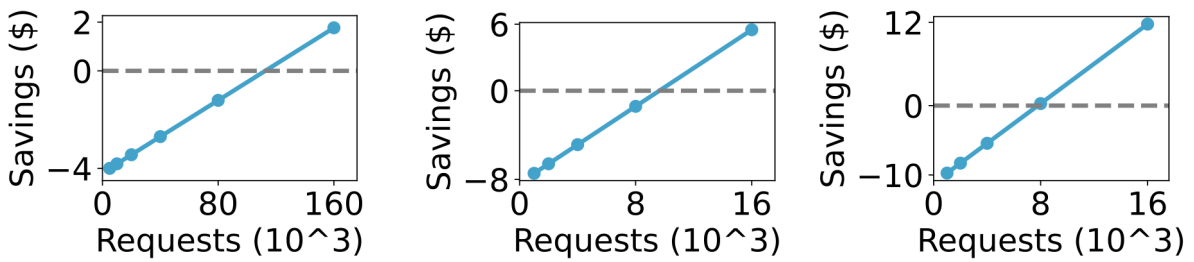

Prompt
LLM
small model
Response
p_id
u_id
review
sentiment
1
1
Quality is fine ...
? ➔ 1
2
1
Broken after ...
? ➔ 0
3
1
Super happy ...
? ➔ 1
Just-in-Time Model Replacement
Offer
- 98% eq. results
- 20x cost savings
Accept


Prompt
LLM
small model
Response
p_id
u_id
review
sentiment
1
1
Quality is fine ...
? → 1
2
1
Broken after ...
? → 0
3
1
Super happy ...
? → 1
Just-in-Time Model Replacement
Offer
- 98% eq. results
- 20x cost savings
Accept


Savings ($)
Requests (10^3)
2
0
−4
0
80
160
6
−8
8
16
12
−10


Time (minutes)
55
50
45
10
5
0
Baseline
S-naive
S-500
S-5000
Fine-tuning
Model search
Label items



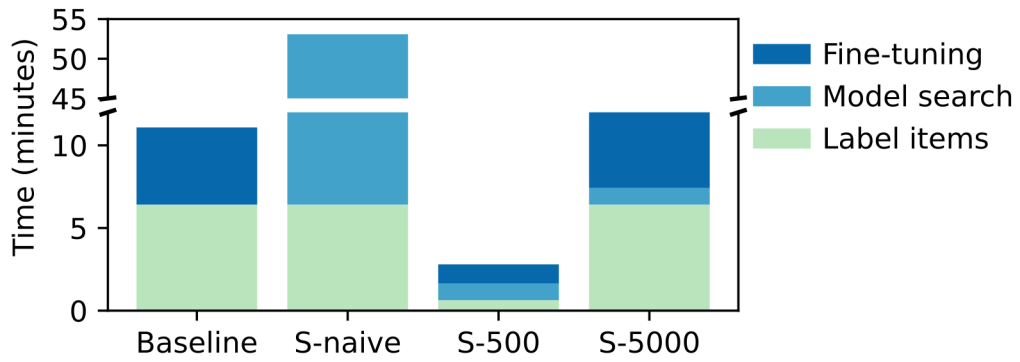




Image created with Gemini AI, Google.

**Poodle**: Seamlessly Scaling Down LLMs with Just-in-Time Model Replacement

# References


[1] 2025. API Pricing — OpenAI. https://platform.openai.com/docs/pricing. Accessed:
2025-07-31.

[2] 2025. Llama 3 8B — Together AI Model. https://www.together.ai/models/llama-3-8b. Accessed: 2025-07-31.

[3] 2025. m2-bert-80M-32k-retrieval — Together AI Model. https://api.together.ai/models/togethercomputer/m2-bert-80M-32k-retrieval. Accessed: 2025-07-31.

[4] 2025. Meta-Llama-3.1-405B-Instruct-Turbo — Together AI Model. https://api.together.ai/models/meta-llama/Meta-Llama-3.1-405B-Instruct-Turbo. Accessed: 2025-07-31.

[5] Alessandro Achille, Michael Lam, Rahul Tewari, Avinash Ravichandran, Subhransu Maji, Charless C Fowlkes, Stefano Soatto, and Pietro Perona. 2019. Task2vec: Task embedding for meta-learning. In ICCV. 6430–6439.

[6] Mitchell Bosley, Musashi Jacobs-Harukawa, Hauke Licht, and Alexander Hoyle. 2023. Do we still need BERT in the age of GPT? Comparing the benefits of domain-adaptation and in-context-learning approaches to using LLMs for Political Science Research. In 2023 Annual Meeting of the Midwest Political Science Association (MPSA).

[7] Martin Juan José Bucher and Marco Martini. 2024. Fine-tuned small LLMs (still) significantly outperform zero-shot generative AI models in text classification. arXiv preprint arXiv:2406.08660 (2024).

[8] Tianle Cai, Xuezhi Wang, Tengyu Ma, Xinyun Chen, and Denny Zhou. 2024. Large Language Models as Tool Makers. In ICLR.

[9] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. arXiv preprint arXiv:2305.05176 (2023).

[10] Timothy Dai, Austin Peters, Jonah B Gelbach, David Freeman Engstrom, and Daniel Kang. 2024. tailwiz: Empowering Domain Experts with Easy-to-Use, Task-Specific Natural Language Processing Models. In Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning. 12–22.

[11] Clément Delangue. 2023. Hugging Face just crossed 1,000,000 free public models.
https://x.com/ClementDelangue/status/1839375655688884305?s=20. Accessed: 2025-11-27.

[12] Aleksandra Edwards and Jose Camacho-Collados. 2024. Language Models for Text Classification: Is In-Context Learning Enough? arXiv:2403.17661 [cs.CL] https://arxiv.org/abs/2403.17661

[13] Google-BERT. 2018. bert-base-uncased. https://huggingface.co/google-bert/bert-
base-uncased. Accessed: 2025-11-26.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML] https://arxiv.org/abs/1503.02531

# References


[15] Long-Kai Huang, Junzhou Huang, Yu Rong, Qiang Yang, and Ying Wei. 2022. Frustratingly Easy Transferability Estimation. In ICML, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.), Vol. 162. 9201–9225.

[16] Hugging Face. 2025. Hugging Face: Machine Learning Platform. https://huggingface.co/

[17] Hao Li, Charless Fowlkes, Hao Yang, Onkar Dabeer, Zhuowen Tu, and Stefano Soatto. 2023. Guided recommendation for model fine-tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 3633–3642.

[18] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In NAACL-HLT. 142–150.

[19] Hui Miao, Ang Li, Larry S Davis, and Amol Deshpande. 2017. Towards unified data and lifecycle management for deep learning. In ICDE. 571–582.

[20] NousResearch. 2023. Llama-2-7b-chat-hf. Hugging Face model repository. https://huggingface.co/NousResearch/Llama-2-7b-chat-hf Accessed: 2025-11-26.

[21] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E.
Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to Route LLMs with Preference Data. arXiv:2406.18665 [cs.LG] https://arxiv.org/abs/2406.18665

[22] Koyena Pal, David Bau, and Renée J Miller. 2025. Model Lakes. In EDBT 2025. (2025).

[23] Nicholas Pangakis and Samuel Wolken. 2024. Knowledge Distillation in Automated Annotation: Supervised Text Classification with LLM-Generated Training Labels. arXiv:2406.17633 [cs.CL] https://arxiv.org/abs/2406.17633

[24] Cheng Qian, Chi Han, Yi Fung, Yujia Qin, Zhiyuan Liu, and Heng Ji. 2023. CREATOR: Tool Creation for Disentangling Abstract and Concrete Reasoning of Large Language Models. In The 2023 Conference on Empirical Methods in Natural Language Processing.

[25] Cedric Renggli, André Susano Pinto, Luka Rimanic, Joan Puigcerver, Carlos Riquelme, Ce Zhang, and Mario Lučić. 2022. Which model to transfer? finding the needle in the growing haystack. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 9205–9214.

[26] Cedric Renggli, Xiaozhe Yao, Luka Kolar, Luka Rimanic, Ana Klimovic, and Ce Zhang. 2022. SHiFT: an efficient, flexible search engine for transfer learning. Proceedings of the VLDB Endowment 16, 2 (2022), 304–316. Publisher: VLDB Endowment.

[27] sanj.dev. 2025. The Real Cost of AI: OpenAI's $13.5B Loss Explained. https://sanj.dev/post/real-cost-of-ai-openai-financials. sanj.dev (3 Oct 2025). Accessed: 2025-11-26.

# References

[28] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. Advances in Neural Information Processing Systems 36 (2023), 38154–38180.

[29] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning,
Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (Eds.). Association for Computational Linguistics, Seattle, Washington, USA, 1631–1642. https://aclanthology.org/D13-1170/

[30] Nils Strassenburg, Boris Glavic, and Tilmann Rabl. 2025. Alsatian: Optimizing Model Search for Deep Transfer Learning. Proc. ACM Manag. Data 3, 3 (2025), 127:1–127:27. doi:10.1145/3725264

[31] Nils Strassenburg, Ilin Tolovski, and Tilmann Rabl. 2022. Efficiently Managing Deep Learning Models in a Distributed Environment. In EDBT.

[32] TapTwice Digital. 2025. 8 OpenAI Statistics (2025): Revenue, Valuation, Profit, Funding. TapTwice Digital (18 May 2025). https://taptwicedigital.com/stats/openai Accessed: 2025-11-26.

[33] Manasi Vartak, Joana M F. da Trindade, Samuel Madden, and Matei Zaharia. 2018.
Mistique: A system to store and query model intermediates for model diagnosis. In SIGMOD. 1285–1300.

[34] Siqi Xiang, Sheng Wang, Xiaokui Xiao, Cong Yue, Zhanhao Zhao, and Beng Chin
Ooi. 2025. NeurStore: Efficient In-database Deep Learning Model Management System. Proceedings of the ACM on Management of Data 3, 6 (2025), 1–26.

[35] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. 2023. Dive into
Deep Learning. Cambridge University Press.

[36] Huayi Zhang, Binwei Yan, Lei Cao, Samuel Madden, and Elke Rundensteiner. 2024. MetaStore: Analyzing Deep Learning Meta-Data at Scale. Proceedings of the VLDB Endowment 17, 6 (Feb. 2024), 1446–1459. doi:10.14778/3648160.3648182

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009