Hints-In-Browser: Benchmarking Language Models for Programming Feedback Generation

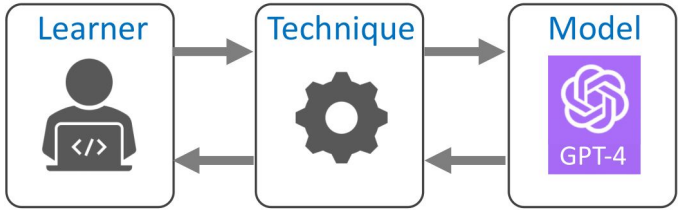
Nachiket Kotalwar, Alkis Gotovos, Adish Singla

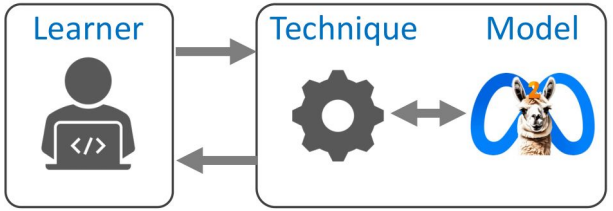
Max Planck Institute for Software Systems

MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

Existing Workflows for Programming Feedback Generation and Their Limitations

- Generative AI and LLMs are increasingly being used for programming feedback generation
- Current deployment workflows use external server-based deployments

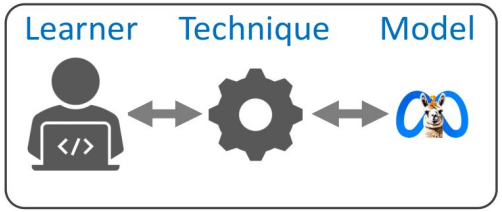




- Existing workflows have limitations in terms of running costs and privacy aspects

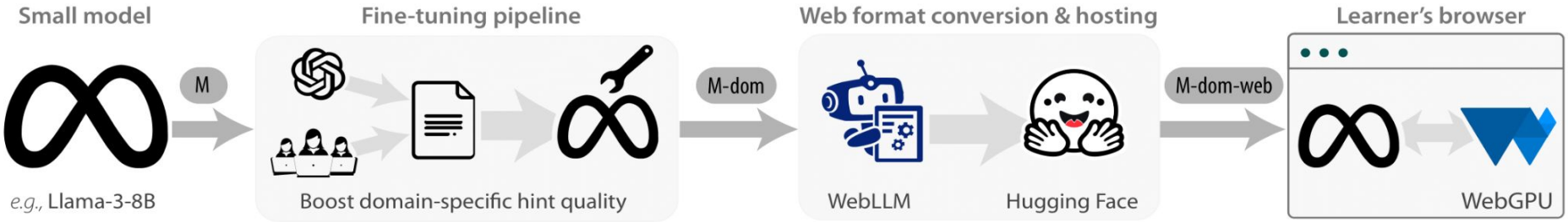
Our Proposed Workflow and Evaluation Metrics for Benchmarking

- Our proposed workflow **Hints-In-Browser** that uses local in-browser inference

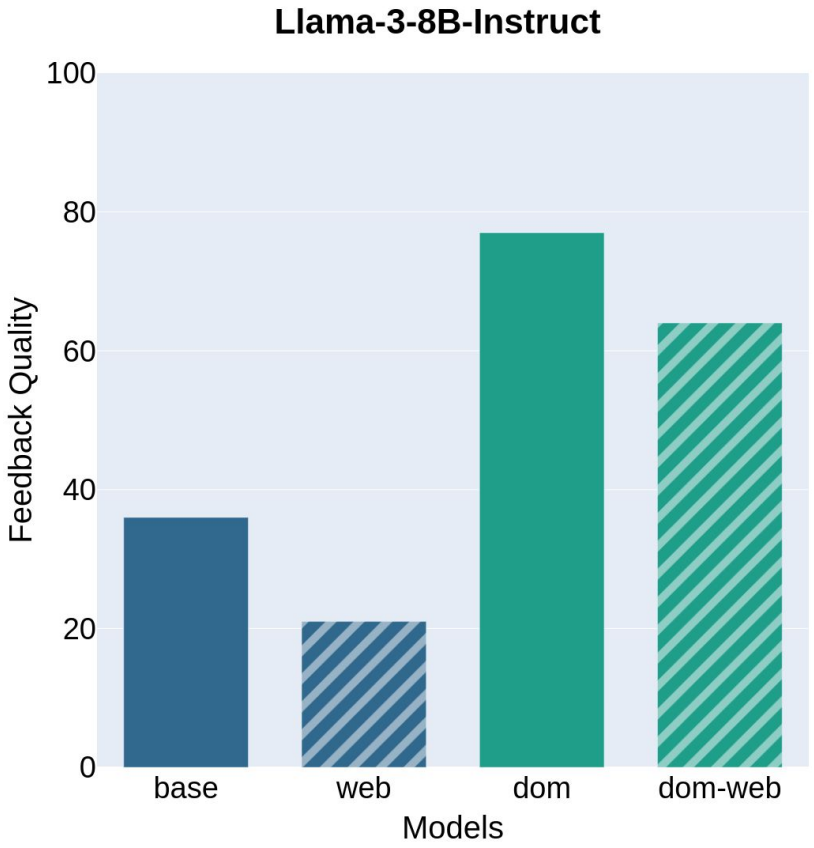


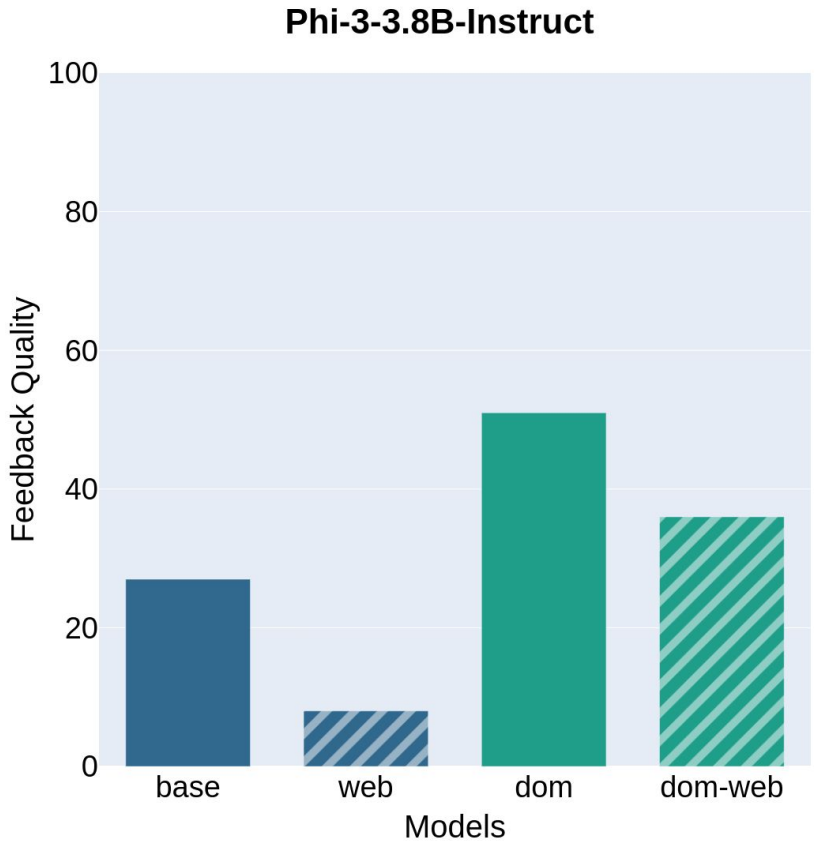
- Evaluation metrics for benchmarking different feedback generation workflows
  - Quality
  - Cost
  - Time
  - Data privacy

Fine-Tuning Small Models and Leveraging In-Browser Inference



Our Fine-tuning Pipeline Significantly Boosts the Feedback Quality of Web Models





Benchmarking Performance for Different Workflows: Time, Cost, and Data Privacy

- We benchmark the web models across multiple hardware configurations
- The domain-specific fine-tuned and quantized (`dom-web`) models
  - deliver competitive inference times on capable systems
  - are virtually free and ensure complete data privacy

Model	Inference (s)	Cost (USD)	Privacy
GPT-4-Turbo	34	5.7×10^{-2}	External Org
GPT-4o-mini	15	1.0×10^{-3}	External Org
Llama-3-8B-dom	31	2.1×10^{-3}	External Server
Llama-3-8B-dom-web	56	n/a	User local
Phi-3-3.8B-dom-web	34	n/a	User local

Hints-In-Browser Web App: *hints-in-browser.netlify.app*