KAIST DeepAuto.ai

Agentic Predictor

Multi-View Encoders for Performance Prediction in LLM-Based Agentic Workflows

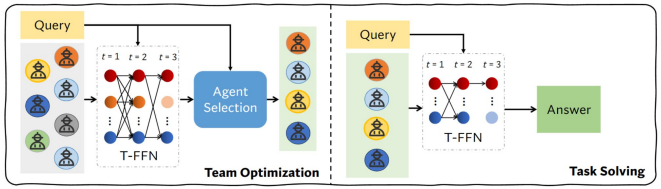
Patara Trirat[1] Wonyong Jeong[1] Sung Ju Hwang[1 2]

[1]DeepAuto.ai [2]KAIST

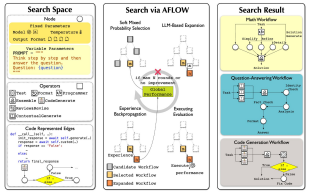
https://deepauto-ai.github.io/agentic-predictor

Why Predict Agentic Workflow Performance?

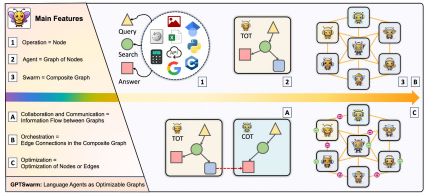
- Multi-agent systems powered by LLMs show great promise.



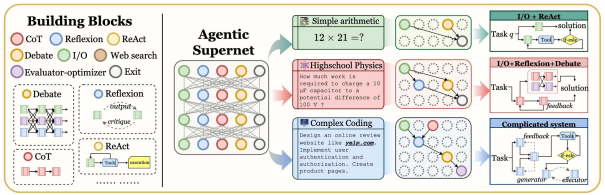
DyLAN
[COLM'24]



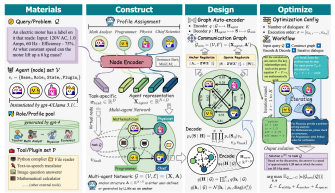
AFlow
[ICLR'25]



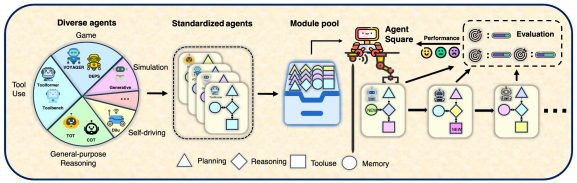
GPTSwarm
[ICML'24]



MaAS
[ICML'25]



G-Designer
[ICML'25]



AgentSquare
[ICLR'25]

Why Predict Agentic Workflow Performance?

- Current optimization frameworks rely on costly execution-based (runtime or LLM calls) evaluations.

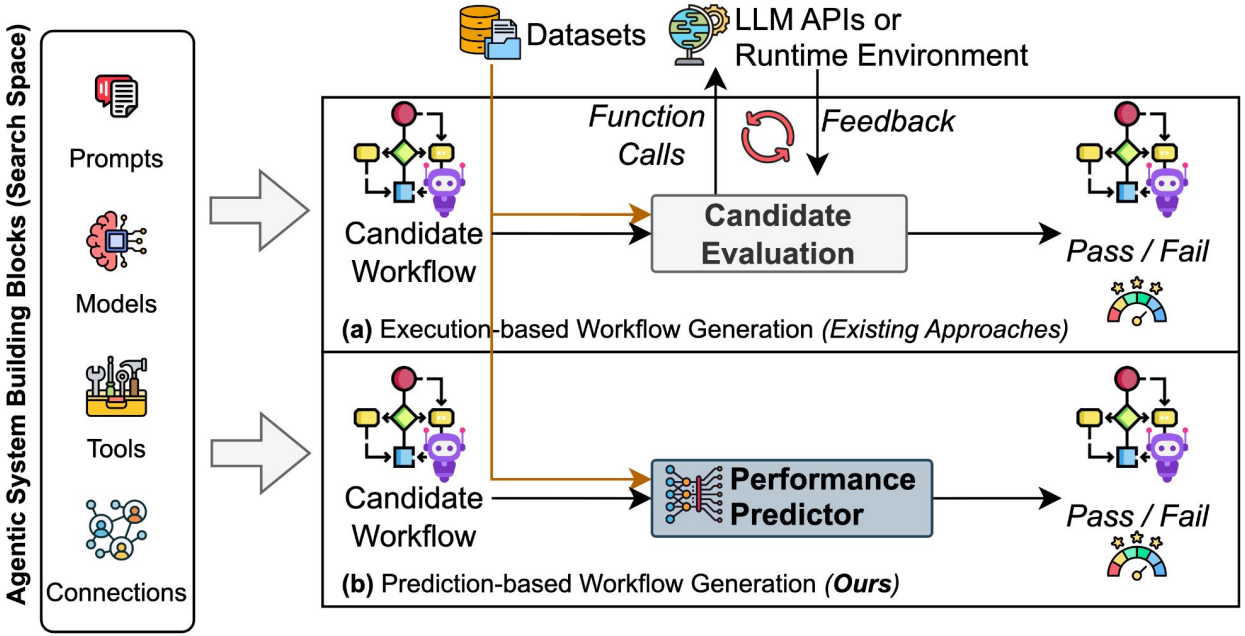
Table 3. Efficiency comparison between MaAS and state-of-the-art baselines on the MATH Benchmark. We shade the values of the lowest token/cost/wall-clock time and the highest performance.

Method	Training				Inference				Overall
	Prompt token	Completion token	Total cost ($)	Wall-clock time (min)	Prompt token	Completion token	Total cost ($)	Wall-clock time (min)	Acc. (%)
LLM-Debate	-	-	-	-	3,275,764	10,459,097	6.76$	92	48.54
DyLAN	22,152,407	16,147,052	13.01$	508	6,081,483	3,303,522	2.89$	39	48.63
MacNet	-	-	-	-	7,522,057	2,043,600	2.35$	47	45.18
GPTSwarm	21,325,266	6,369,884	7.02$	129	3,105,571	788,273	0.93$	30	47.88
AFlow	33,831,239	29,051,840	22.50$	184	2,505,944	2,151,931	1.66$	23	51.28
MaAS	3,052,159	2,380,505	3.38$	53	1,311,669	853,116	0.42$	19	51.82

Table extracted from MaAS [ICML'25]

Why Predict Agentic Workflow Performance?

- A predictive model can estimate the quality and viability of agentic workflows.



Challenges in Agentic Workflow Prediction

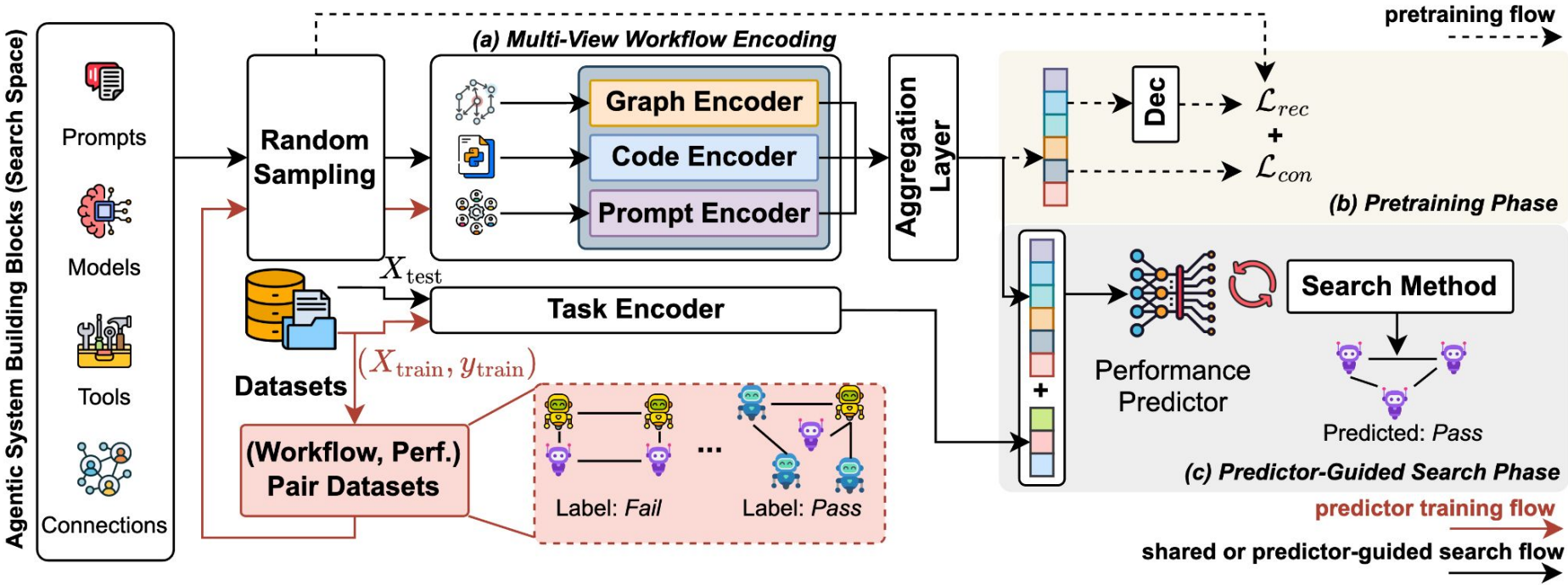
- **Heterogeneity**
  - Workflows vary widely in communication, code, and prompts.
- **Scarcity of Labeled Data**
  - Only few labels available due to expensive evaluations.

Our Solution: *Agentic Predictor*

- **Multi-View Encoding** integrates graph, code, and prompt views.
- **Cross-Domain Unsupervised Pretraining** uses unlabeled data from diverse tasks.

- **Lightweight Predictor** quickly estimates workflow performance, guiding efficient search.

Architecture Overview



Multi-View Workflow Encoding

- **Graph View** explicitly captures structural dependencies and direct interactions among agents, emphasizing interagent communication channels.
- **Code View** implicitly encodes complex semantic structures, logical sequences, computational complexities, and patterns of tool usage inherent in workflow implementations.
- **Prompt View** provides semantic embeddings that encapsulate nuanced agent roles, behavioral descriptions, and broader contextual guidance embedded within system and instruction prompts.

Multi-View Workflow Encoding: *Encoder Network*

- **(Multi-)Graph View**

aggregates importance scores across different views

captures inter-graph importance

$$\mathbf{Z}_{\mathcal{G}} = G_{\text{pool}}(\text{ViewAttnPool}(\text{CrossGraphAttn}(\mathbf{X})))$$

- **Code View**

$$\mathbf{Z}_{\mathcal{C}} = \text{MLP}_{\mathcal{C}}(\mathcal{C})$$

- **Prompt View**

$$\mathbf{Z}_{\mathcal{P}} = \text{MLP}_{\mathcal{P}}(\mathcal{P})$$

- **Aggregation Layer**

$$\mathbf{Z} = \text{MLP}([\mathbf{Z}_{\mathcal{G}}, \mathbf{Z}_{\mathcal{C}}, \mathbf{Z}_{\mathcal{P}}])$$

Cross-Domain Unsupervised Pretraining

- In real-world scenarios, the availability of labeled performance data for agentic workflows may be highly limited due to the costly and time-consuming evaluation process.
- To address this challenge and enable data-efficient predictor training, we *optionally* adopt a two-phase strategy.
- **Cross-Domain Multi-Task Pretraining**

$$\mathcal{L}_{rec} = \frac{1}{M}\sum_{i=1}^{M} \|\mathcal{G}_i - \hat{\mathcal{G}}_i\|^2 + \|\mathcal{C}_i - \hat{\mathcal{C}}_i\|^2 + \|\mathcal{P}_i - \hat{\mathcal{P}}_i\|^2$$

$$+$$

$$\mathcal{L}_{con} = \frac{1}{M}\sum_{i=1}^{M} -\log \frac{\exp(\text{sim}(\mathbf{Z}_i, \mathbf{Z}_j^+)/\tau)}{\sum_{k=1}^{M} \exp(\text{sim}(\mathbf{Z}_i, \mathbf{Z}_k)/\tau)}$$

Positives are the index-aligned embeddings of the configuration across two different views, while *negatives* are all other configurations within the batch.

This learning objective encourages the encoder to capture meaningful signals related to both structure and performance.

Performance Predictor

- **Input Features**
  - The **learned workflow embeddings** from the previous stage.
  - The **task-specific semantic embeddings** from natural language task descriptions. These embeddings, derived from pretrained language models (e.g., T5 or BERT).
- **Loss Function:** binary cross-entropy loss for *fail* vs. *pass* predictions

$$\mathcal{L}_{\text{pred}} = -\frac{1}{N}\sum_{i=1}^{N}[e_i \log \hat{e}_i + (1 - e_i)\log(1 - \hat{e}_i)]$$

Experiments

- **(Q1)** How does Agentic Predictor perform as a predictor of agentic workflow performance compared to relevant baselines?
- **(Q2)** How do different design choices of Agentic Predictor affect its predictive accuracy?
- **(Q3)** Is the pretraining phase helpful for maintaining prediction quality under varying numbers of labels?
- **(Q4)** Does Agentic Predictor maintain strong predictive performance under out-of-distribution shifts?
- **(Q5)** How does Agentic Predictor compare against few-shot LLM-based workflow performance predictors?

Experiments: *Setup*

- **Benchmark Dataset**: FLORA-Bench (Zhang et al., 2025b), including ***five*** representative datasets across ***three*** core domains in the agentic workflow literature:
  - **code generation** (HumanEval, MBPP),
  - **mathematics problem solving** (GSM8K, MATH), and
  - **reasoning** (MMLU).

Table 2: Summary of benchmark statistics.

Domains	Code Generation (GD/AF)	Math (GD/AF)	Reasoning (GD/AF)
# workflows	739 / 38	300 / 42	189 / 30
Avg. # nodes	5.96 / 6.11	6.06 / 5.49	5.97 / 6.58
# tasks	233 / 233	782 / 782	2,400 / 2,400
# samples	30,683 / 7,362	12,561 / 4,059	453,600 / 72,000

Experiments: *Setup*

- **Evaluation Metrics**:
  - **Accuracy** quantifies how well a model predicts agentic workflow performance.
  - **Utility** evaluates the *consistency between the workflow rankings predicted by the model and the ground-truth rankings*, emphasizing the model's ability to determine the relative order of different workflows.

Experiments: *Results* (*Prediction Accuracy (Q1)*)

Table 3: Performance comparison between Agentic Predictor and baselines. The best and second-best results are highlighted in **bold** and underlined, respectively. GD is G-Designer, and AF is AFlow.

Domain	CodeGD		CodeAF		MathGD		MathAF		ReasonGD		ReasonAF		Average	
Model	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility
MLP	83.88 (±0.04)	76.16 (±0.03)	78.02 (±0.59)	73.94 (±1.35)	63.22 (±0.30)	64.13 (±0.44)	73.73 (±0.31)	69.64 (±0.29)	71.54 (±0.09)	62.41 (±1.67)	78.45 (±0.08)	88.48 (±0.63)	74.81 (±0.24)	72.46 (±0.74)
GCN	84.23 (±0.04)	79.31 (±0.10)	84.35 (±0.34)	72.73 (±1.18)	64.12 (±0.17)	63.03 (±0.59)	76.19 (±0.42)	66.52 (±1.66)	72.22 (±0.03)	59.18 (±0.85)	87.12 (±0.14)	**91.82** (±0.46)	78.04 (±0.19)	72.10 (±0.81)
GAT	85.14 (±0.25)	79.50 (±0.14)	84.49 (±0.56)	76.46 (±0.91)	64.84 (±0.96)	62.32 (±0.93)	76.44 (±0.61)	66.51 (±1.28)	72.16 (±0.03)	59.44 (±1.06)	87.07 (±0.08)	89.40 (±0.68)	78.36 (±0.42)	72.27 (±0.83)
GCN-II	83.81 (±0.07)	78.45 (±0.74)	83.72 (±0.40)	77.75 (±0.98)	63.56 (±0.74)	66.02 (±0.10)	75.04 (±0.31)	64.33 (±0.47)	72.29 (±0.09)	59.10 (±1.02)	87.28 (±0.14)	89.92 (±1.90)	77.62 (±0.29)	72.60 (±0.87)
Graph Transformer	85.24 (±0.19)	80.20 (±0.64)	84.71 (±0.45)	74.09 (±0.35)	63.25 (±0.70)	64.97 (±0.36)	75.45 (±0.23)	66.48 (±0.96)	72.26 (±0.08)	60.92 (±1.79)	86.93 (±0.27)	90.60 (±1.97)	77.97 (±0.32)	72.88 (±1.01)
Dir-GNN	84.85 (±0.11)	79.81 (±0.69)	83.45 (±0.41)	76.08 (±0.92)	63.01 (±0.54)	64.68 (±1.66)	76.11 (±0.65)	67.97 (±0.16)	74.25 (±0.12)	62.64 (±0.92)	86.66 (±0.13)	90.07 (±1.68)	78.05 (±0.33)	73.54 (±1.01)
One For All	83.74 (±0.09)	75.93 (±0.12)	81.05 (±0.34)	73.42 (±1.39)	63.17 (±0.21)	66.65 (±0.82)	75.21 (±0.23)	69.08 (±0.64)	72.29 (±0.12)	60.35 (±1.25)	82.52 (±0.13)	87.64 (±1.98)	76.33 (±0.19)	72.18 (±1.03)
Agentic Predictor	**85.33** (±0.05)	**81.42** (±0.26)	**85.62** (±0.47)	**80.08** (±0.46)	**66.20** (±0.17)	**67.88** (±0.21)	**79.56** (±0.25)	**74.08** (±0.47)	**75.13** (±0.01)	**63.06** (±0.45)	**87.96** (±0.02)	91.47 (±0.44)	**79.97** (±0.16)	**76.33** (±0.38)
Δ vs. best baseline (% Improvement)	+0.09 (0.11%)	+1.22 (1.52%)	+0.91 (1.07%)	+2.33 (3.00%)	+1.36 (2.09%)	+1.23 (1.85%)	+3.12 (4.08%)	+4.44 (6.38%)	+0.88 (1.19%)	+0.42 (0.67%)	+0.68 (0.78%)	-0.35 -(0.38%)	+1.61 (2.05%)	+2.79 (3.79%)

Experiments: *Results* (Ablation Study (Q2))

Table 4: Results of ablation study on different input view variations.

Variations			Code Generation		Math Problem		Common Reasoning		Average	
Code	Graph	Text	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility
✓			82.04	75.66	75.70	68.52	83.19	**91.51**	80.31	78.56
	✓		84.44	77.22	79.14	67.99	87.00	91.03	83.53	78.75
		✓	79.87	70.34	76.60	68.45	68.06	71.04	74.84	69.94
✓	✓		83.72	73.97	75.86	70.18	86.88	86.14	82.15	76.76
✓		✓	82.27	77.28	76.03	66.66	54.17	53.21	70.82	65.72
	✓	✓	82.45	74.64	75.70	67.83	69.47	70.55	75.87	71.01
✓	✓	✓	**85.62**	**80.08**	**79.56**	**74.08**	**87.96**	91.47	**84.38**	**81.88**

Table 5: Results of ablation study on different input graph variations.

Variations		Code Generation		Math Problem		Commong Reasoning		Average	
Single View	Multi View	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility
✓		82.58	**78.52**	78.57	67.51	86.95	90.14	82.70	78.72
	✓	**84.44**	77.22	**79.14**	**67.99**	**87.00**	**91.03**	**83.53**	**78.75**

Experiments: *Results* (*Effects of Pretraining Phase (Q3)*)

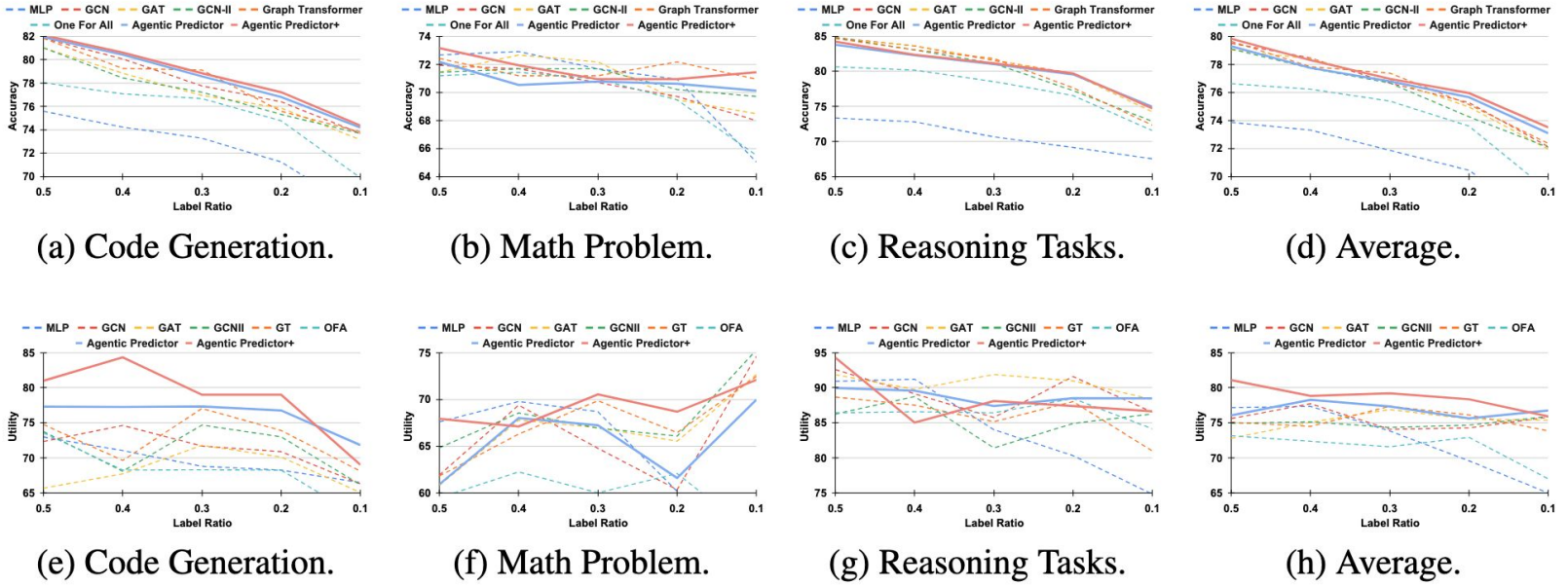


Figure 4: Comparison of accuracy (upper) and utility (lower) between Agentic Predictor and the baselines across varying label ratios.

Experiments: *Results* (OOD Performance (Q4))

Table 10: Results when train on AFlow and test on G-Designer.

Domain	Code Generation		Math Problem		Common Reasoning		Average	
Model	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility
GCN	56.76	54.29	49.64	51.92	61.37	54.13	55.92	53.45
GAT	57.25	56.05	48.29	48.71	57.03	53.12	54.19	52.63
GCN-II	64.16	62.67	48.85	50.56	65.55	52.76	59.52	55.33
Graph Transformer	60.83	58.39	47.73	46.65	55.88	48.87	54.81	51.30
One For All	58.97	53.25	50.60	51.02	63.84	55.22	57.80	53.16
Agentic Predictor	**65.02**	**64.91**	**53.62**	**52.83**	**67.51**	**57.74**	**62.05**	**58.49**

Table 11: Results when train on G-Designer and test on AFlow.

Domain	Code Generation		Math Problem		Common Reasoning		Average	
Model	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility
GCN	58.21	57.33	67.57	54.63	57.51	53.37	61.10	55.11
GAT	59.29	59.68	66.34	52.70	56.07	50.38	60.57	54.25
GCN-II	58.75	61.17	67.32	52.96	55.93	52.19	60.67	55.44
Graph Transformer	60.52	**61.44**	58.97	57.49	56.50	54.86	58.66	57.93
One For All	**62.01**	54.57	58.72	61.23	**59.40**	54.17	60.04	56.66
Agentic Predictor	60.94	59.75	**69.11**	**63.02**	58.56	**56.73**	**62.87**	**59.83**

Table 12: Results on cross-domain OOD test.

Domain	Code-Math		Code-Reason		Math-Reason		Math-Code		Reason-Code		Reason-Math		Average	
Model	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility	Accuracy	Utility
GCN	48.89	54.07	52.61	53.29	49.38	46.69	50.07	48.75	32.56	50.53	33.42	50.57	44.49	50.65
GAT	45.95	49.42	53.71	57.90	46.83	38.90	51.02	47.40	33.79	52.62	33.42	51.10	44.12	49.56
GCN-II	56.02	44.49	53.44	45.93	50.38	47.36	39.48	51.55	38.13	51.35	36.61	**57.93**	45.68	49.77
Graph Transformer	47.67	56.18	53.71	57.95	47.90	43.63	54.00	56.20	60.92	52.37	41.77	52.91	51.00	53.21
One For All	36.61	**61.11**	50.33	39.82	44.92	45.88	**65.40**	56.24	**63.36**	50.60	38.08	45.27	49.78	49.82
Agentic Predictor	**57.17**	61.03	**54.22**	**62.99**	**53.86**	**61.75**	59.88	**60.25**	61.60	**54.52**	**62.90**	52.69	**58.27**	**58.87**

Experiments: *Results* (vs. LLM Predictors (Q5))

Table 9: Comparison between Agentic Predictor and LLM-based few-show classification.

Domain	Code Generation		Math Problem		Common Reasoning		Average	
Model	**Accuracy**	**Utility**	**Accuracy**	**Utility**	**Accuracy**	**Utility**	**Accuracy**	**Utility**
GPT-4.1 (∼$59)	62.42	57.00	67.08	52.97	59.10	66.79	62.86	58.92
Claude 4 Sonnet (∼$202)	56.72	51.65	64.62	57.32	44.50	41.25	55.28	50.07
Gemini 2.5 Flash (∼$21)	60.52	58.94	51.60	55.21	59.20	63.17	57.10	59.11
Agentic Predictor	**84.40**	**78.84**	**80.10**	**77.61**	**90.40**	**87.67**	**84.97**	**81.37**

Conclusions

- We propose **Agentic Predictor**, a lightweight, predictive framework to estimate the success of agentic workflows using multi-view representation learning and unsupervised pretraining.
- We use the multi-view encoding to capture **workflow heterogeneity** from
  - Graph Structure (agent interaction),
  - Code Semantics (logic & tool use), and
  - Instruction Prompts (roles & behaviors).
- We introduce cross-domain unsupervised pretraining to lessen the **label scarcity** problem by training the encoder on unlabeled workflows from various domain.

Thank you!

GitHub: github.com/deepauto-ai/agentic-predictor
Email: patara@deepauto.ai