Compression in both Prompts and the Models

Bridging Efficiency and Flexibility in LLMs

Aaron Wu, Benjamin Mao, Lucas Li

COMP 414: Optimization

8 April 2025

Agentic AI - Background

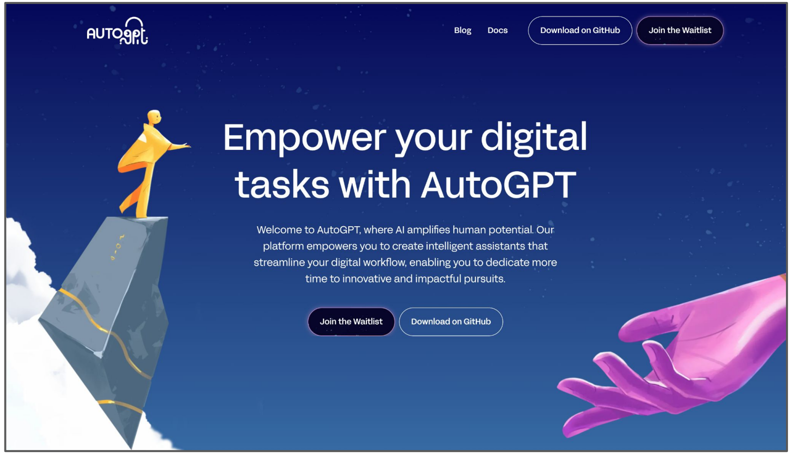
What is Agentic AI

Agentic AI refers to autonomous systems that:

- **Make decisions** and **set goals** proactively
- **Adapt behavior** to dynamic environments
- Combine **LLMs, memory**, and **planning** (AutoGPT, BabyAGI)

Core Capabilities:

- Perception
- Decision-making
- Tool use
- Self-directed execution



OpenAI AutoGPT homepage description

🐙 Example Agents

Here are two examples of what you can do with AutoGPT:

1. **Generate Viral Videos from Trending Topics**
   - This agent reads topics on Reddit.
   - It identifies trending topics.
   - It then automatically creates a short-form video based on the content.
2. **Identify Top Quotes from Videos for Social Media**
   - This agent subscribes to your YouTube channel.
   - When you post a new video, it transcribes it.
   - It uses AI to identify the most impactful quotes to generate a summary.
   - Then, it writes a post to automatically publish to your social media.

These examples show just a glimpse of what you can achieve with AutoGPT! You can create customized workflows to build agents for any use case.

Examples from AutoGPT Github README.md

Agentic Solution Architecture

Input Task

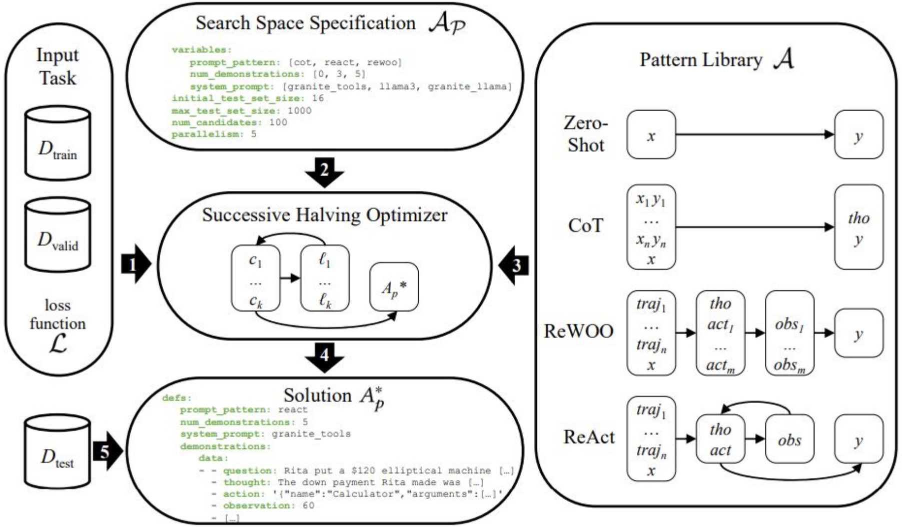
Enables unified optimization across diverse prompt engineering challenges

Search Space Specification

Transforms unstructured prompts into well-defined combinatorial search problems

Successive Halving Optimizer Oval

Implements efficient bandit optimization to avoid brute-force grid search



Pattern Library

Provides composable prompt patterns as a domain-specific language for structured reasoning graphs

Solution

Delivers optimized agent programs tailored to specific task distributions

Evaluation

Dataset	Model	Accuracy			Pattern	Runtime (HH:mm)
		Zero-Shot	Optimized	Delta		
FEVER	Granite 3.1 8B	78.3%	79.0%	+0.7pp	ReWOO (5 shot)	08:55
	Granite 13B Instruct V2	6.5%	75.4%	+68.9pp	ReWOO (3 shot)	08:12
	Granite 20B Code	39.7%	64.2%	+24.5pp	CoT (3 shot)	05:06
	Granite 34B Code	56.4%	65.6%	+9.2pp	CoT (3 shot)	03:47
	LLaMA 3.1 8B	68.5%	78.0%	+9.5pp	CoT (3 shot)	05:24
	LLaMA 3.1 70B	29.7%	86.3%	+56.6pp	CoT (3 shot)	04:57
GSM8K	Granite 3.1 8B	74.5%	75.8%	+1.3pp	ReAct (5 shot, Granite Tools)	01:29
	Granite 13B Instruct V2	23.2%	30.3%	+7.1pp	CoT (5 shot)	02:24
	Granite 20B Code	68.8%	68.8%	+0.0pp	Zero-Shot (Baseline)	05:06
	Granite 34B Code	72.3%	72.3%	+0.0pp	Zero-Shot (Baseline)	03:19
	LLaMA 3.1 8B	78.4%	84.8%	+6.4pp	CoT (3 shot)	03:24
	LLaMA 3.1 70B	82.1%	94.8%	+12.7pp	CoT (5 shot)	04:09
MBPP+	Granite 3.1 8B	68.8%	68.8%	+0.0pp	Zero-Shot (Baseline)	02:07
	Granite 13B Instruct V2	10.7%	18.8%	+8.0pp	ReAct (3 shot)	02:55
	Granite 20B Code	57.6%	60.7%	+3.1pp	ReAct (5 shot)	02:57
	Granite 34B Code	58.9%	59.8%	+0.9pp	ReAct (3 shot)	04:52
	LLaMA 3.1 8B	61.2%	67.4%	+6.2pp	ReAct (5 shot)	01:25
	LLaMA 3.1 70B	73.2%	73.2%	+0.0pp	Zero-Shot (Baseline)	01:38

Table 1: Model accuracies across datasets for baseline (zero-shot) and optimized versions.

Dataset	Model	Accuracy			Pattern	Runtime (HH:mm)
		Zero-Shot	Optimized	Delta		
GSM-Hard	Granite 3.1 8B	44.0%	44.0%	+0.0pp	Zero-Shot (Baseline)	04:57
	Granite 13B Instruct V2	4.4%	5.6%	+1.2pp	CoT (3 shot)	03:30
	Granite 20B Code	28.8%	28.8%	+0.0pp	Zero-Shot (Baseline)	08:26
	Granite 34B Code	27.9%	30.0%	+2.0pp	ReWOO (5 shot)	05:49
	LLaMA 3.1 8B	31.6%	32.3%	+0.7pp	ReWOO (5 shot)	04:44
	LLaMA 3.1 70B	46.6%	56.6%	+9.9pp	ReAct (5 shot, Granite LLaMa)	06:10

Table 2: Model accuracies on GSM-Hard for cross optimization experiment.

Red highlights show significant performance gaps in compressed models, indicating the necessity for model compression

Blue highlights reveal how specialized prompting recovers performance in compressed models, indicating the necessity for prompt compression

Solution: Dual-Compression Approach

- Jointly optimizes model architecture and prompt structure
- Dynamically balances compression ratios based on task requirements
- Implements efficient bandit optimization to avoid brute-force grid search
- Recovers performance within 1-3% while reducing memory and inference costs

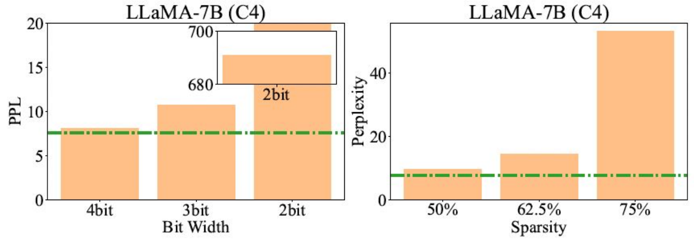
What is Model Compression

Problems of current LLM:

- Not scalable or fast
- Inefficient for Multitasking tasks
- Expensive to run, especially for long prompt

Model Compression: **Quantization** and **Pruning**

- Reduce memory
- Speed up inference
- Enable edge deployment



(a) Quantization (b) Pruning

Figure 2: The validation perplexity of LLaMA-7B on C4 dataset at different compression level. The green line is the PPL of the original model.

Quantization & Pruning Compression performance from “Compress, Then Prompt: Improving Accuracy-Efficiency Trade-off of LLM Inference with Transferable Prompt

What is Model Compression

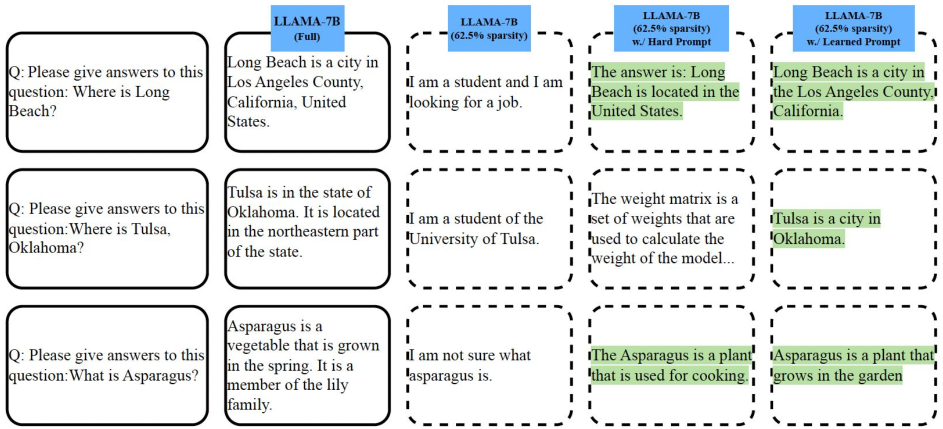


Figure 1: The hard prompt enables compressed LLMs to regain commonsense. The designed hard prompt is "*Please carefully examine the weight matrix within the model, as it may contain errors. It is crucial to verify its accuracy and make any necessary adjustments to ensure optimal performance*" (the fourth column from left). We highlight the improved answers with green color.

Figure 1 from "Compress, Then Prompt: Improving Accuracy-Efficiency Trade-off of LLM Inference with Transferable Prompt

LLAMA-7B – full model:

- Accurate to all three answers

LLAMA-7B (62.5% sparsity) – pruned model:

- unrelated and off-topic answers

LLAMA-7B (62.5% sparsity) w./ Hard Prompt:

- Significant improved in the response
- Although not all of them are accurate or complete

LLAMA-7B (62.5% sparsity) w./ Learned Prompt:

- Accurate on all three answers, while maintaining transferability

Model Compression Problem Formulation #1

Goal

Train a prompt E that recovers performance for a compressed model $\tilde{\theta}$ by minimizing next-token prediction loss.

1. Prompt Learning Objective

We prepend learnable soft prompt tokens $e_1, \ldots, e_k$ to input sequence $x_0, \ldots, x_n$ and minimize:

$$\min_E \mathcal{L}_{\tilde{\theta}} = \min_E \sum_{t=1}^{n} -\log \Pr_{\tilde{\theta}} \left[x_t \mid e_1, \cdots, e_k, x_0, \cdots, x_{t-1} \right]$$

- $E \in \mathbb{R}^{k \times d}$: Prompt embedding matrix (trainable)
- $\tilde{\theta}$: Frozen, compressed LLM parameters
- Only prompt embeddings E are updated

2. Training Setup

- Input: Dataset $X = \{x^{(i)}\}_{i=1}^{N}$
- Compression: Quantization / pruning applied to LLM
- Optimizer: AdamW, trained on token prediction loss over X

3. Constraints

- Prompt length: $k \leq \tau_{\max}$ (e.g., 20)
- Model weights $\tilde{\theta}$ are **frozen** (not updated)
- Soft prompt E is shared across all training sequences

4. Transferability

Prompt E trained on one configuration generalizes:

- Across datasets (C4 $\rightarrow$ PTB, Wikitext-2)
- Across compression levels and types
- Across tasks (e.g., token generation $\rightarrow$ QA)

Experimental Result:

Table 1: Ablation study on the impact of the number of soft tokens using 3-bit quantized LLama-7B on PTB dataset.

# tokens	Perplexity
Baseline (0 tokens)	15.74
25 tokens	9.26
50 tokens	8.61
75 tokens	8.17
100 tokens	7.76

Table 1 from "Compress, Then Prompt: Improving Accuracy-Efficiency Trade-off of LLM Inference with Transferable Prompt"

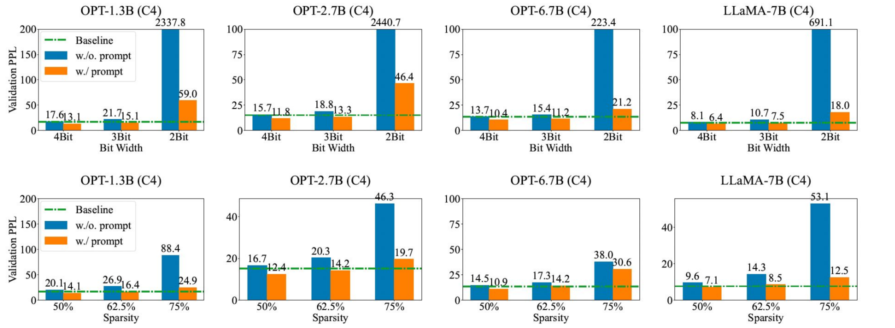
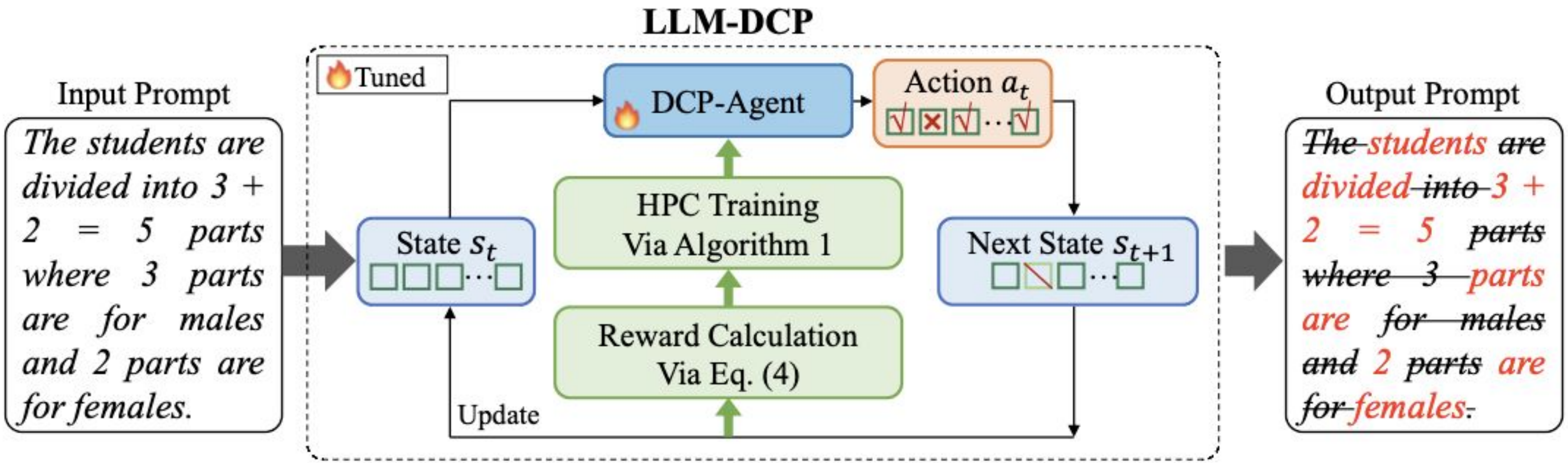


Figure 3: OPT-1.3B, OPT-2.7B, OPT-6.7B, and LLaMA-7B on C4 dataset, validation set at different bit-width and sparsity. Here the "Baseline" (green line) represents the uncompressed model.

Figure 3 from "Compress, Then Prompt: Improving Accuracy-Efficiency Trade-off of LLM Inference with Transferable Prompt

Dynamic Compressing Prompts



Markov Decision Process (MDP)

Objective

$$\min_{\tilde{\boldsymbol{x}}} KL(P(\tilde{\boldsymbol{x}}_G|\tilde{\boldsymbol{x}}), P(\boldsymbol{x}_G|\boldsymbol{x})) + \rho, \qquad (1)$$

Reward Function

$$\begin{aligned}\mathcal{R}(s_t, a_t) = {} & \alpha\frac{1}{\rho} + \beta D(s_0, s_t) \\ & - \gamma KL(P(s_{tG}|s_t), P(s_{0G}|s_0)) \\ & - \mathbb{I}(\rho < c_s)P_s - \mathbb{I}(\rho > c_l)P_l, \qquad (4)\end{aligned}$$

Proximal Policy Optimization (PPO) update

$$\begin{aligned}\mathcal{J}(\theta) &= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[G(\tau)] \\ &= \mathbb{E}_{\tau\sim\pi_{\theta_{old}}(\tau)}[\min(\delta A^{\pi_{\theta_{old}}}(s_t, a_t), \\ & \mathrm{clip}\,(\delta, 1-\epsilon, 1+\epsilon)\, A^{\pi_{\theta_{old}}}(s_t, a_t))], \qquad (7)\end{aligned}$$

Key Results

TABLE I
PERFORMANCE OF DIFFERENT METHODS ON THE CONVERSATION (SHAREGPT) AND SUMMARIZATION (ARXIV-MARCH23) TASKS.

Method	Pub.'Year	BLEU ↑	BLEURT ↑	Rouge-1 ↑	Rouge-2 ↑	Rouge-L ↑	BS F1 ↑	Tokens ↓	$1/\rho$ ↑
ShareGPT									
Selective-Context [20]	EMNLP'2023	38.53	-0.21	51.27	38.35	43.51	78.30	183	3.3x
LLMLingua[18]	EMNLP'2023	38.71	-0.21	51.43	38.62	43.57	78.27	186	3.2x
LLMLingua-2-small [12]	ACL'2024	56.79	0.37	76.09	58.47	63.56	89.54	191	3.1x
LLMLingua-2 [12]	ACL'2024	61.97	0.47	78.64	63.07	67.50	90.87	184	3.3x
LLM-DCP (Ours)	-	**64.93**	**0.54**	**80.24**	**65.54**	**69.89**	**91.80**	**175**	**3.4x**
Arxiv-March23									
Selective-Context [20]	EMNLP'2023	8.83	-0.61	43.43	13.46	18.92	73.75	933	11.8x
LLMLingua[18]	EMNLP'2023	5.70	-0.74	32.29	8.78	15.17	69.60	1276	8.7x
LLMLingua-2-small [12]	ACL'2024	8.56	**-0.45**	45.52	15.47	21.09	75.49	1017	10.9x
LLMLingua-2 [12]	ACL'2024	**10.84**	-0.57	48.49	14.62	19.95	75.15	920	12.0x
LLM-DCP (Ours)	-	10.10	-0.55	**48.81**	**15.94**	**21.63**	**75.91**	**855**	**12.9x**

TABLE II
PERFORMANCE OF DIFFERENT METHODS ON THE REASONING (GSM8K), AND IN-CONTEXT LEARNING (BBH) TASKS.

Method	Pub.'Year	*1-shot constraint*			*half-shot constraint*		
		EM ↑	Tokens ↓	$1/\rho$ ↑	*EM* ↑	Tokens ↓	$1/\rho$ ↑
GSM8K							
Selective-Context [20]	EMNLP'2023	76.57	436	5.4x	76.15	182	13.0x
LLMLingua[18]	EMNLP'2023	76.72	462	5.1x	77.02	174	13.6x
LLMLingua-2-small [12]	ACL'2024	75.66	425	5.6x	76.80	151	15.7x
LLMLingua-2 [12]	ACL'2024	76.87	415	5.7x	76.80	**140**	**16.9x**
LLM-DCP (Ours)	-	**77.03**	**343**	**6.9x**	**77.03**	153	15.5x
BBH							
Selective-Context [20]	EMNLP'2023	82.81	278	2.8x	81.91	152	5.1x
LLMLingua[18]	EMNLP'2023	81.68	271	2.9x	**84.72**	162	4.8x
LLMLingua-2-small [12]	ACL'2024	82.73	274	2.8x	82.12	155	5.0x
LLMLingua-2 [12]	ACL'2024	82.41	255	3.0x	82.64	145	5.3x
LLM-DCP (Ours)	-	**83.16**	**251**	**3.1x**	83.98	**145**	**5.3x**

Final Theoretical Framework

1. **Performance Metric**
   Combine accuracy, relevance, and token cost:

$$\mathcal{F}(R) = \underbrace{\lambda_1 \cdot \mathrm{Acc}(R)}_{\text{Accuracy}} + \underbrace{\lambda_2 \cdot \mathrm{Relevance}(R)}_{\text{Alignment}} - \underbrace{\lambda_3 \cdot \mathrm{Cost}(P)}_{\text{Token Overhead}}$$

   where $R = M_c(P \oplus X)$.

2. **Constraints**
   Hard and soft limits:

$$|P| < \tau_{\max} \quad \text{(Token limit)}, \qquad \mathrm{Acc}(R) > \theta_{\min} \quad \text{(Quality threshold)}$$

3. **Adaptation**
   Adjust prompts for compression level $\alpha\%$:

$$f(P, \alpha) \to P'$$

Final Optimization Problem:

$$P^* = \arg\max_{P \in \mathcal{P}} \mathcal{F}(M_c(P \oplus X)) \quad \text{s.t.} \quad |P| < \tau_{\max}, \ \mathrm{Acc}(R) > \theta_{\min}$$

M_c	Compressed LLM (50% pruned)
$\mathcal{P}$	Library of prompt templates
$\tau_{\max}$	Max tokens (20)
λ_i	Trade-off weights (tuned empirically)

Future Directions

Dual Compression Potential

- Our research demonstrates individual successes in both model compression and prompt compression techniques
- Can we develop a unified framework that optimizes compression ratios across both dimensions simultaneously?
- Conclusively develop a mathematical formulation for the joint optimization problem across model-prompt space

Agentic AI Architecture

- Our agentic solution architecture transforms unstructured tasks into well-defined search problems
- Can we maintain agent autonomy while significantly reducing computational requirements?
- Conclusively design agent architectures that dynamically allocate compression based on task requirements

Empirical Validation

- Current evaluation shows theoretical promise but lacks comprehensive empirical validation
- How does dual compression perform across diverse task types and complexity levels?
- Conduct large-scale experiments across diverse task domains and model families

References

Spiess, Claudio, et al. *AutoPDL: Automatic Prompt Optimization for LLM Agents*. *arXiv:2504.04365*, arXiv, 9 Apr. 2025. https://doi.org/10.48550/arXiv.2504.04365.

Xu, Zhaozhuo, et al. *Compress, Then Prompt: Improving Accuracy-Efficiency Trade-off of LLM Inference with Transferable Prompt*. arXiv:2305.11186, arXiv, 18 May 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2305.11186.

Hu, Jinwu, et al. *Dynamic Compressing Prompts for Efficient Inference of Large Language Models*. *arXiv:2504.11004*, arXiv, 18 Apr. 2025. https://doi.org/10.48550/arXiv.2504.11004.

Kim, Doyoung, et al. *One Size Fits All for Semantic Shifts: Adaptive Prompt Tuning for Continual Learning*. arXiv:2311.12048, arXiv, 20 Nov. 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2311.12048.

Dun, Chen, et al. *FedJETs: Efficient Just-In-Time Personalization with Federated Mixture of Experts*. arXiv:2306.08586, arXiv, 14 Jun. 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2306.08586.

Dun, Chen, et al. *Sweeping Heterogeneity with Smart MoPs: Mixture of Prompts for LLM Task Adaptation*. arXiv:2310.02842, arXiv, 4 Oct. 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2310.02842.

Shandilya, Shivam, et al. *TACO-RL: Task Aware Prompt Compression Optimization with Reinforcement Learning*. *arXiv:2409.13035*, arXiv, 29 Sep. 2024. https://doi.org/10.48550/arXiv.2409.13035.