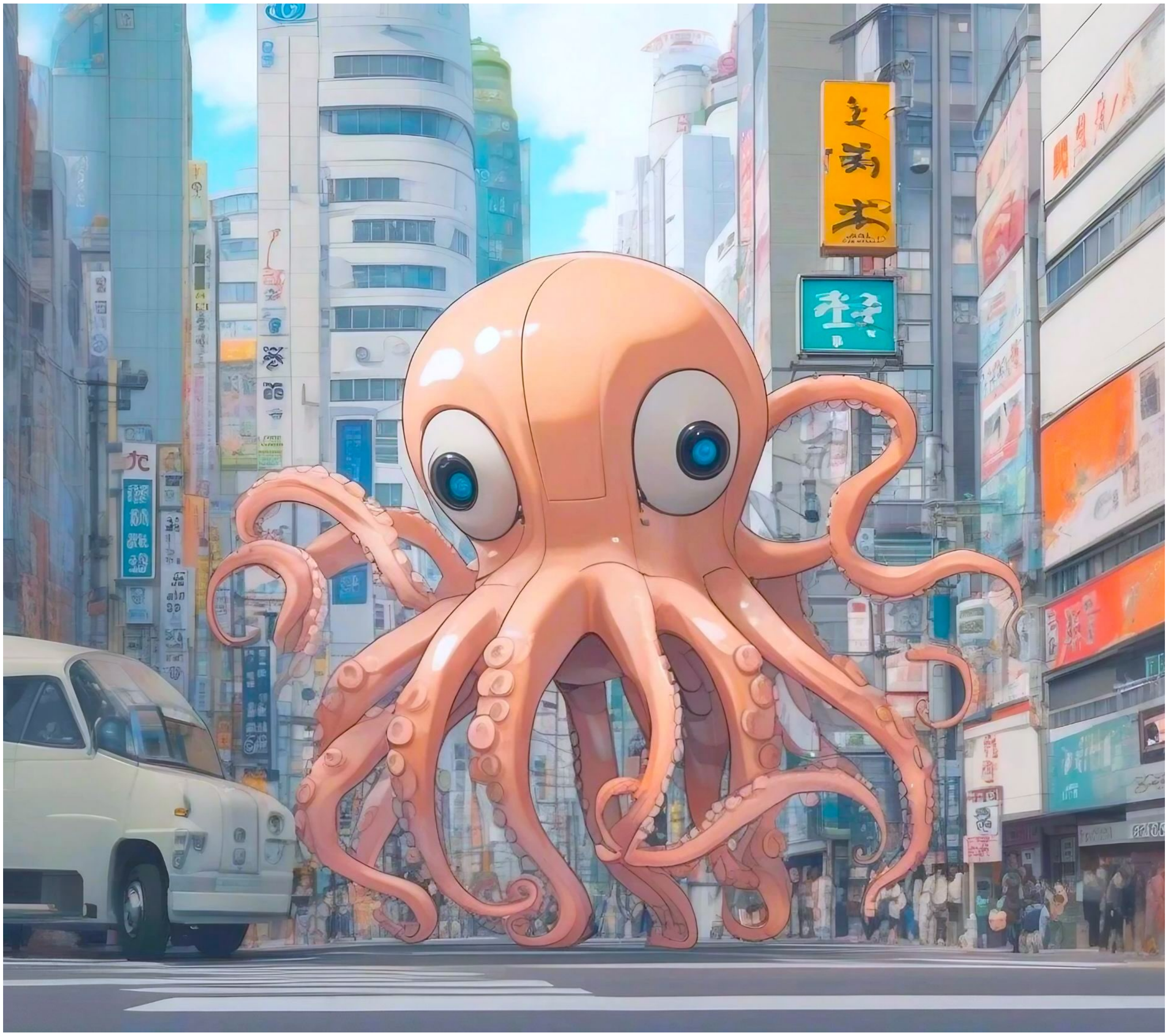
Transformer²: Self-Adaptive LLMs

sakana.ai/transformer-squared

January 15, 2025

sakana.ai



Summary

Adaptation is one of the most remarkable phenomena in nature. From the way an octopus can change their skin color to blend into its surroundings, to how the human brain rewires itself after an injury, allowing individuals to recover lost functions and adapt to new ways of

thinking or moving. Living organisms exhibit adaptability that allows life to flourish in diverse and ever-changing environments.

In the field of AI, the concept of adaptation holds a similar allure. Imagine a machine learning system that could adjust its own weights dynamically to thrive in unfamiliar settings, essentially illustrating a system that evolves as it learns. Self-adaptiveness in AI promises greater efficiency and the potential for lifelong models ever aligned with the dynamic nature of the real world.

This vision of self-adaptive AI is at the heart of our latest **research paper**, **Transformer²** *('Transformer-squared')*, where we propose a machine learning system that dynamically adjusts its weights for various tasks. The name *Transformer²* reflects its two-step process: first, the model analyzes the incoming task to understand its requirements, and then it applies task-specific adaptations to generate optimal results. By selectively adjusting critical components of the model weights, our framework allows LLMs to dynamically adapt to new tasks in real time. Transformer² demonstrates significant advancements across various tasks (e.g., math, coding, reasoning, and visual understanding), outperforming traditional, static approaches like LoRA in efficiency and task-specific performance while requiring far fewer parameters.

Our research offers a glimpse into a future where AI models are no longer static. These systems will scale their compute dynamically at test-time to adapt to the complexity of tasks they encounter, embodying *living intelligence* capable of continuous change and lifelong learning. We believe self-adaptivity will not only transform AI research but also redefine how we interact with intelligent systems, creating a world where adaptability and intelligence go hand in hand.

Transformer² is a machine learning system that dynamically adjusts its weights for various tasks. Adaptation is a remarkable natural phenomenon, like how the octopus can blend its color in with its environment, or how the brain rewires itself after injury. We believe our new system paves the way for a new generation of adaptive AI models, modifying their own weights and architecture to adapt to the nature of the tasks they encounter, embodying living intelligence capable of continuous change and lifelong learning.

Dissecting the Brain of LLMs

Just as the human brain stores knowledge and processes information through interconnected neural pathways, LLMs store knowledge within their weight matrices. These matrices are the "brain" of an LLM, holding the essence of what it has learned from its training data.

Understanding this “brain” and ensuring that it can adapt effectively to new tasks requires a closer look at its inner structure. This is where Singular Value Decomposition (SVD) provides invaluable insights. Think of SVD as a surgeon performing a detailed operation on the brain of an LLM. This surgeon breaks down the vast, complex knowledge stored in the LLM into smaller, meaningful, and independent pieces (e.g., the different pathways or components for math, language understanding, etc).

SVD achieves this purpose by identifying the principal components of the LLM’s weight matrices. In our research, we found that enhancing the signal from a subset of these components while suppressing the others could improve an LLM’s performance on downstream tasks. By building on this foundation, Transformer² takes the next step toward dynamic, task-specific adaptation, enabling LLMs to excel in diverse and complex scenarios.

Introducing Transformer²

Transformer² is a novel approach pioneering the concept of self-adaptive LLMs with a two-step process that redefines how these powerful models tackle diverse tasks. At its core is the ability to dynamically adjust critical components of its weight matrices. At training time, we introduce Singular Value Finetuning (SVF), a method that employs reinforcement learning (RL) to enhance/suppress the signals from different “brain” components for various types of downstream tasks. At inference time, we employ three distinct strategies to detect the identity of the task and adapt the model’s weights accordingly. The figure below gives an overview of our method.

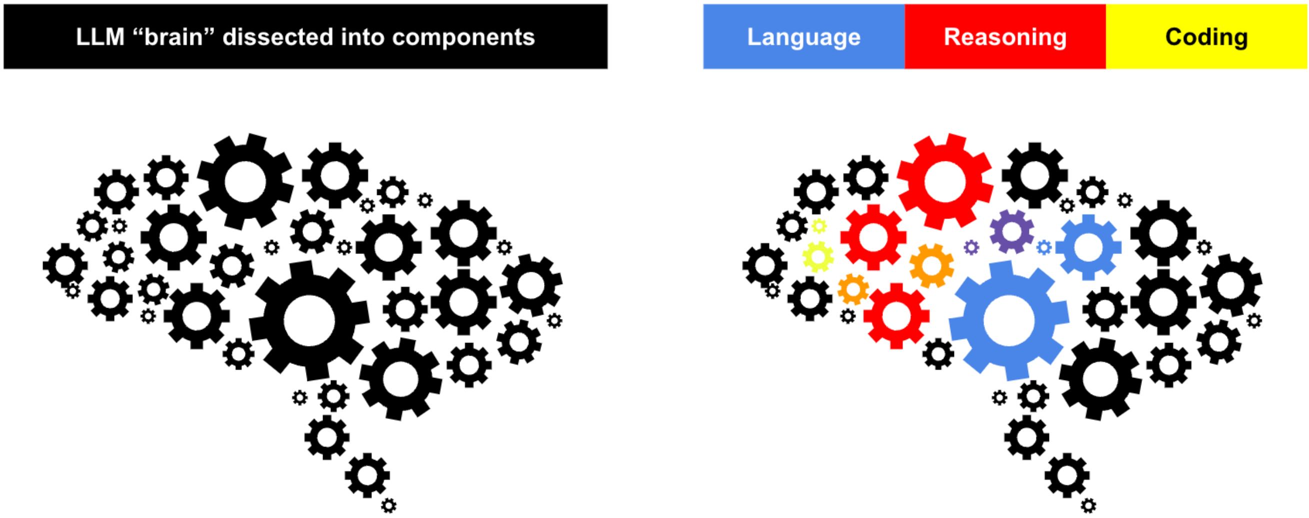


Illustration of our method.
<u>Left</u>: We decompose an LLM’s “brain” (i.e., weight matrices) into several independent components using SVD.

Right: We employ RL to train the combination of these components for various tasks. Components may be shared among different tasks. E.g., in the figure above, purple cogs are shared by language understanding and reasoning. At inference time, we identify the task type and then adjust the combination of the components dynamically.

Training with SVF and RL

At training time, SVF learns a set of *z-vectors*, one for each downstream task. Each z-vector, which can be regarded as an expert on a task, is a compact representation that specifies the desired strength of each component in the weight matrix, acting as a set of “amplifiers” or “dampeners” to modulate the influence of different components on the model’s behavior.

For example, suppose SVD decomposes a weight matrix into five components [A, B, C, D, E]. For a math task, the learned z-vector might be [1, 0.8, 0, 0.3, 0.5], meaning that component A is critical for math while component C hardly affects its performance. For a language understanding task, the z-vector could be [0.1, 0.3, 1, 0.7, 0.5], highlighting that component C is essential for this task despite being less useful for math.

SVF employs RL to learn these z-vectors on a pre-defined set of downstream tasks. The learned z-vectors enable Transformer² to adapt to various new downstream tasks while introducing only a minimal number of additional parameters (i.e., the z-vectors).

Self-Adaptation

At inference time, we devise a two-pass adaptation strategy for our framework that effectively combines the set of task-specific z-vectors. In the first inference pass, given a task or an individual input prompt, Transformer² analyzes its test-time conditions using one of the three adaptation methods below. In the second pass, Transformer² then modulates the weights accordingly by combining the z-vectors, producing a final response most relevant for its new settings.

We summarize the three methods for task detection/adaptation in the following:

1. **Prompt-based adaptation.** A specifically designed adaptation prompt classifies the task (e.g., math, coding) and selects a pre-trained z-vector.
2. **Classifier-based adaptation.** A task classifier trained with SVF identifies the task during inference and selects the appropriate z-vector.

3. **Few-shot adaptation.** Combines multiple pre-trained z-vectors through weighted interpolation. A simple optimization algorithm tunes these weights based on performance on a few-shot evaluation set.

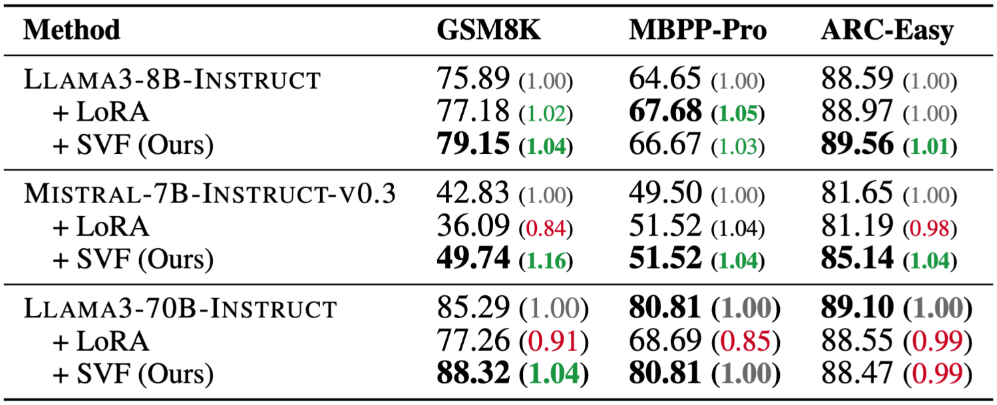
These three methods collectively ensure that Transformer² achieves robust and efficient task adaptation, paving the way for remarkable performance across diverse scenarios. Please refer to our paper for details.

Main Results

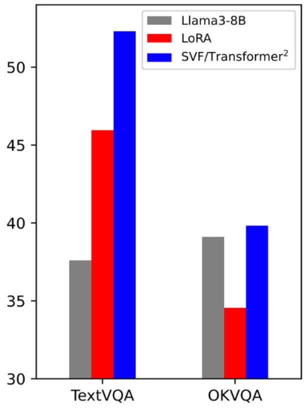
We apply our methods to both the Llama and Mistral LLMs across a broad range of tasks, including math (GSM8K, MATH), code (MBPP-Pro, HumanEval), reasoning (ARC-Easy, ARC-Challenge), and visual question answering (TextVQA, OKVQA).

We first set out to obtain the z-vectors by SVF on these tasks, and compare it with LoRA. Our results in the table below show that SVF outperforms LoRA on text-based tasks, with particularly strong gains on GSM8K. This can be attributed to our RL training objective, which does not require "perfect solutions" for each question, unlike LoRA's fine-tuning approach. The histogram on the right also illustrates SVF's amazing capacity in the vision domain.

Evaluation of SVF on broad tasks.
We split each task into train, validation, and test sets. We report test set performance using pass@1 for MBPP-Pro and accuracy for all other tasks as evaluation metrics. Left: SVF on language tasks. Normalized scores are in parentheses. Right: SVF on VQA tasks.



Method	GSM8K	MBPP-Pro	ARC-Easy
LLAMA3-8B-INSTRUCT	75.89 (1.00)	64.65 (1.00)	88.59 (1.00)
+ LoRA	77.18 (1.02)	**67.68 (1.05)**	88.97 (1.00)
+ SVF (Ours)	**79.15 (1.04)**	66.67 (1.03)	**89.56 (1.01)**
MISTRAL-7B-INSTRUCT-V0.3	42.83 (1.00)	49.50 (1.00)	81.65 (1.00)
+ LoRA	36.09 (0.84)	51.52 (1.04)	81.19 (0.98)
+ SVF (Ours)	**49.74 (1.16)**	**51.52 (1.04)**	**85.14 (1.04)**
LLAMA3-70B-INSTRUCT	85.29 (1.00)	**80.81 (1.00)**	**89.10 (1.00)**
+ LoRA	77.26 (0.91)	68.69 (0.85)	88.55 (0.99)
+ SVF (Ours)	**88.32 (1.04)**	**80.81 (1.00)**	88.47 (0.99)

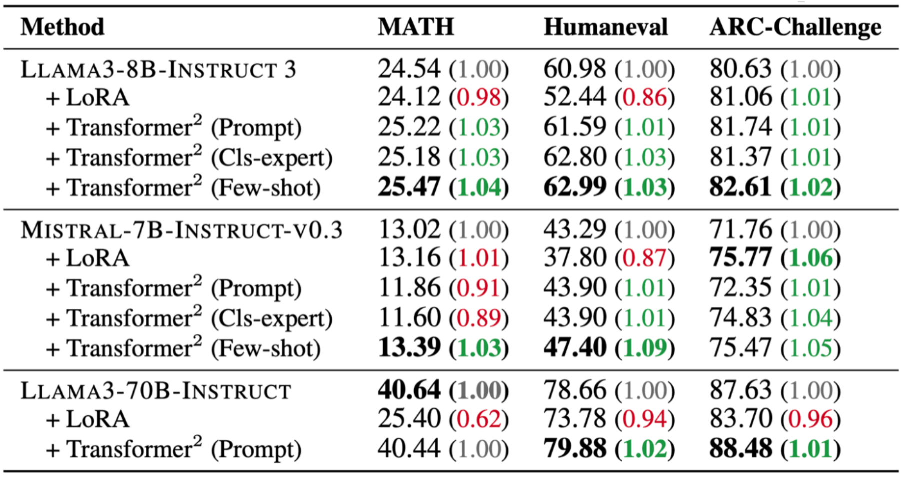


We then evaluate our adaptation framework against LoRA on unseen tasks, specifically MATH, HumanEval, and ARC-Challenge. The left table below demonstrates that our strategies achieve increasing performance gains as method complexity increases across all the tasks.

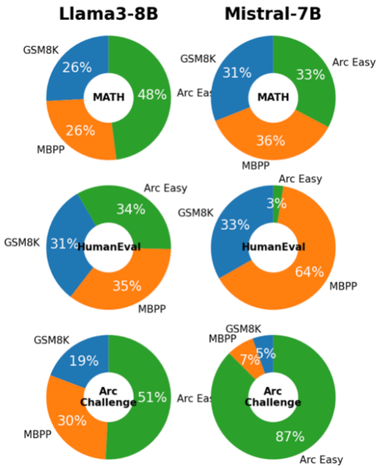
A particularly intriguing finding comes from analyzing how few-shot learning combines different z-vectors to tackle tasks, as shown in the right figure. When solving MATH problems, contrary to expectations, the model does not rely exclusively on its GSM8K (math) specialized z-vectors. This suggests that complex mathematical reasoning benefits from combining mathematical, programmatic, and logical reasoning capabilities. We observe similar unexpected combinations across other tasks and models, highlighting the framework's ability to synthesize diverse types of expertise for optimal performance.

Evaluation of Transformer².

We directly report the test set performance on the unseen tasks. Left: Self-adaptation on unseen tasks. Right: Learned z-vectors interpolation weights.



Method	MATH	Humaneval	ARC-Challenge
LLAMA3-8B-INSTRUCT 3	24.54 (1.00)	60.98 (1.00)	80.63 (1.00)
+ LoRA	24.12 (0.98)	52.44 (0.86)	81.06 (1.01)
+ Transformer^2 (Prompt)	25.22 (1.03)	61.59 (1.01)	81.74 (1.01)
+ Transformer^2 (Cls-expert)	25.18 (1.03)	62.80 (1.03)	81.37 (1.01)
+ Transformer^2 (Few-shot)	**25.47 (1.04)**	**62.99 (1.03)**	**82.61 (1.02)**
MISTRAL-7B-INSTRUCT-V0.3	13.02 (1.00)	43.29 (1.00)	71.76 (1.00)
+ LoRA	13.16 (1.01)	37.80 (0.87)	**75.77 (1.06)**
+ Transformer^2 (Prompt)	11.86 (0.91)	43.90 (1.01)	72.35 (1.01)
+ Transformer^2 (Cls-expert)	11.60 (0.89)	43.90 (1.01)	74.83 (1.04)
+ Transformer^2 (Few-shot)	**13.39 (1.03)**	**47.40 (1.09)**	75.47 (1.05)
LLAMA3-70B-INSTRUCT	**40.64 (1.00)**	78.66 (1.00)	87.63 (1.00)
+ LoRA	25.40 (0.62)	73.78 (0.94)	83.70 (0.96)
+ Transformer^2 (Prompt)	40.44 (1.00)	**79.88 (1.02)**	**88.48 (1.01)**

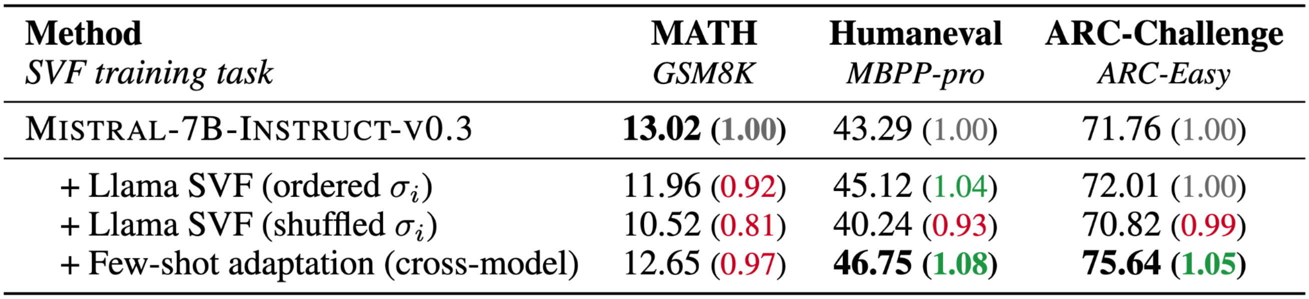


Finally, we explored an intriguing question that challenges conventional wisdom in AI development: Can we transfer the knowledge from one model to another? To our excitement, when taking the learned z-vectors from Llama to Mistral, we observe positive effects with the latter showing improved performance on most tasks. See table below for detailed results.

While these findings are promising, we should note that both models share similar architectures, which might explain their compatibility. Whether this knowledge-sharing works between more diverse AI models remains an open question. Still, these results suggest exciting possibilities for opening the doors to disentangling and recycling task-specific skills for newer/larger models.

Cross-model z-vector transfer.

Results from transferring the "experts" trained on Llama3-8B-Instruct to Mistral-7B-Instruct-v0.3 with few-shot adaptation.



Method *SVF training task*	MATH *GSM8K*	Humaneval *MBPP-pro*	ARC-Challenge *ARC-Easy*
MISTRAL-7B-INSTRUCT-V0.3	**13.02 (1.00)**	43.29 (1.00)	71.76 (1.00)
+ Llama SVF (ordered σ_i)	11.96 (0.92)	45.12 (1.04)	72.01 (1.00)
+ Llama SVF (shuffled σ_i)	10.52 (0.81)	40.24 (0.93)	70.82 (0.99)
+ Few-shot adaptation (cross-model)	12.65 (0.97)	**46.75 (1.08)**	**75.64 (1.05)**

The Future: From Static Models to Living Intelligence

Transformer² represents a significant milestone in the evolution of AI systems. Its ability to dynamically adapt to unseen tasks in real-time with enhanced compositionality demonstrates the potential of self-adaptive LLMs to revolutionize AI research and applications alike.

But this is just the beginning. Transformer² offers a glimpse into a future where AI systems are no longer static entities trained for fixed tasks. Instead, they will embody “living intelligence”, models that continually learn, evolve and adapt over time. Imagine an AI capable of seamlessly integrating new knowledge or adapting its behavior in real-world environments without retraining, much like how humans adjust to new challenges.

The path forward lies in building models that dynamically adapt and collaborate with other systems, combining specialized capabilities to solve complex, multi-domain problems. Self-adaptive systems like Transformer² bridge the gap between static AI and living intelligence, paving the way for efficient, personalized, and fully integrated AI tools that drive progress across industries and our daily lives.

Sakana AI

Interested in joining us? Please see our career opportunities for more information.