# LLM Modules: Knowledge Transfer from a Large to a Small Model using Enhanced Cross-Attention

Konstantin Kolomeitsev
Almaty, Kazakhstan
uol92kot@gmail.com

February 11, 2025

**Abstract**

In this paper, we propose an architecture of *LLM Modules* that enables the transfer of knowledge from a pre-trained large model to a smaller model using an Enhanced Cross-Attention mechanism. In the proposed scheme, the Qwen2-1.5B model is frozen, and its representations are passed through specially designed attention layers to the GPT-Neo-125M model, which is trained on limited computational resources. Experimental results obtained on the `Bespoke-Stratos-17k` dataset demonstrate that after 15 epochs of training, the combined model generates responses that are comparable in quality to models obtained by distillation. The paper discusses the advantages of the modular approach in detail, provides examples of input queries and their comparative analysis, and outlines prospects for further extension of the method.

## 1 Introduction

Large language models (LLMs) have demonstrated outstanding performance in natural language processing tasks; however, their training and deployment require significant computational resources. This has led to the need for methods that transfer knowledge from large pre-trained models to smaller models. Such approaches are especially relevant for applied tasks with limited computational resources.

In this work, we propose a modular LLM architecture in which a large model serves as a knowledge source, while a smaller model receives external representations via Enhanced Cross-Attention and generates responses. This method significantly reduces training costs while remaining effective for solving specific business tasks.

## 2 Related Work

Knowledge distillation and adaptive transfer of representations are widely discussed in the literature [3, 1]. For instance, distillation works such as DistilBERT [4] and TinyBERT [5] show that it is possible to significantly reduce model size without losing performance. Modern transfer learning approaches include the use of adapter modules [6] and methods like LoRA [7], which allow for the adaptation of large models with a minimal number of trainable parameters. AdapterFusion [8] demonstrates the potential for non-destructive task composition during knowledge transfer. Additionally, the T5 work [9] explores the limits of transfer learning by unifying multiple tasks into a single text-to-text format. Thus, the proposed methodology differs in that, rather than relying on classic distillation, it transfers representations via specialized Cross-Attention layers, thereby preserving a larger portion of the original model's information and handling longer input sequences (e.g., 128K tokens in Qwen2 versus 2K tokens in GPT-Neo).

# 3 Methodology

## 3.1 Core Idea of LLM Modules

The approach is based on two key components:

- **Knowledge Source:** A large pre-trained model (e.g., Qwen2-1.5B) is used to extract rich representations of the input query. The model's weights remain unchanged (frozen).
- **Generation Module:** A small model (e.g., GPT-Neo-125M) receives external representations via Enhanced Cross-Attention layers and, combining them with its own computations, generates a coherent response.

## 3.2 Enhanced Cross-Attention

The key component of the scheme is a modified Cross-Attention layer, which consists of:

1. Linear projections that convert the representation dimensions of the large model (e.g., 1536) to the dimensions required by the small model (e.g., 768).
2. An **Adapter Block** that provides an additional non-linear transformation to adapt the representations.
3. A **Gating Mechanism** that dynamically blends the original representations with the external knowledge.

This approach effectively "absorbs" knowledge from the source without the need to retrain the entire model.

# 4 Implementation

## 4.1 Model Architecture

The architecture comprises three interconnected modules:

1. **ModifiedQwenWithCrossAttention:**
   - Loads the pre-trained Qwen2-1.5B model.
   - Freezes all model parameters to preserve the pre-trained knowledge.
   - Adds linear projections and Enhanced Cross-Attention layers to transform the hidden representations.
2. **ModifiedGptNeo:**
   - Based on the GPT-Neo-125M model.
   - Freezes the embedding layers and replaces the final linear layer to align with the Qwen2 tokenizer.
3. **CombinedModel:**
   - Integrates the previous two modules.
   - Receives an input query that is tokenized using the Qwen2 tokenizer.
   - Passes the representations through the ModifiedQwenWithCrossAttention module, which are then processed by the ModifiedGptNeo model to generate the final response.

### 4.2 Training and Data Preparation Details

The training dataset used is `Bespoke-Stratos-17k`, with a pre-filtering step to limit the maximum sequence length to 4096 tokens. The data preparation procedure includes:

- Dynamic padding of sequences using a `collate_fn` function.
- Filtering out examples exceeding the specified length to avoid memory overflow errors.
- Splitting the dataset into training and validation sets using a random shuffling method.

Training is performed using the AdamW optimizer, and early stopping is applied to prevent overfitting.

### 4.3 Code Example

Below is a simplified code example for creating the combined model:

```
% Initialize configurations
qwen_config = AutoConfig.from_pretrained("Qwen/Qwen2-1.5B")
gptneo_config = AutoConfig.from_pretrained("EleutherAI/gpt-neo-125M")

% Load the tokenizer
qwen_tokenizer = AutoTokenizer.from_pretrained("Qwen/Qwen2-1.5B")
if qwen_tokenizer.pad_token is None:
    qwen_tokenizer.pad_token = qwen_tokenizer.eos_token

% Create the combined model
model = CombinedModel(qwen_config, gptneo_config, qwen_tokenizer)

% Initialize the optimizer to update only the trainable parameters
optimizer = AdamW([
    {'params': model.qwen.cross_attention_layers.parameters(), 'lr': 1e-4},
    {'params': model.qwen.intermediate_layers.parameters(), 'lr': 1e-4},
    {'params': model.gptneo.parameters(), 'lr': 5e-5}
])
```

## 5 Experimental Study

### 5.1 Comparative Analysis of Models

To evaluate the effectiveness of the proposed approach, the following models were compared:

1. **DeepSeek R1 671B** — a large model with high generation quality.
2. **DeepSeek-R1-Distill-Qwen-1.5B-GGUF 32FP** — a distilled model.
3. **Qwen2-1.5B-Instruct-GGUF FP16** — the knowledge source model.
4. **GPT-Neo-125M** — the original small model.
5. **GPT-Neo-125M-fine-tuned** — GPT-Neo-125M further trained (with a 2048-token limit).
6. **GPT-Neo-125M-clean** — a model trained from scratch.
7. **CombinedModel** — the proposed paired model.

### 5.2 Examples of Input Queries and Results

To assess generation quality, the following test examples were used:

1. **Arithmetic Task:** `sum of 5 and 5`.
   All small models produced a set of incoherent text. Large pre-trained models generated the correct answer; however, the Qwen2 model did not provide any "reasoning" even when instructed. The CombinedModel demonstrated detailed reasoning comparable to large "reasoning" models.

2. **Arithmetic Task:** `find the remainder by dividing 7 by 4`.
   The CombinedModel generated a response with a step-by-step explanation of the calculation, indicating the presence of a "reasoning" component.

### 5.3 Detailed Experimental Analysis

During the experiments, the following observations were made:

- After 15 epochs of training, the training loss and validation loss decreased significantly, from 13.8 to 2.3 in the first epoch and to 1.1 in subsequent epochs, confirming the successful convergence of the model.
- Despite the limited amount of training data, the CombinedModel demonstrated a significant improvement in quality compared to the original small models.
- Analysis of the generated responses (see Table 1) indicates that transferring knowledge via Enhanced Cross-Attention enables the combined model to produce more structured and logically coherent outputs.

Table 1: Comparison of Responses from Various Models

| Model | Response | Note |
|---|---|---|
| DeepSeek R1 671B | [Detailed reasoning] [Brief structured answer] | High computational complexity |
| DeepSeek R1 Distill Qwen 1.5B | [Detailed reasoning] [Brief structured answer] | Pre-trained model obtained via distillation |
| Qwen2 1.5B | [Brief structured answer, without reasoning] | Pre-trained model |
| GPT-Neo-125M-clean | [Lack of reasoning and coherent answer] | Trained from scratch |
| GPT-Neo-125M-fine-tuned | [Lack of reasoning and coherent answer] | Improved via fine-tuning |
| CombinedModel | [Detailed reasoning] [Brief structured answer] | Optimized through knowledge transfer |

## 6 Discussion and Future Work

The advantages of the proposed approach include:

- A significant reduction in computational costs compared to full fine-tuning of large models.
- The ability to adapt the model to specific tasks by training it on only the necessary subset of data.

- The potential for exponential improvement in generation quality through the combination of representations from various architectures.

Future research directions include:

- Expanding the architecture by integrating other types of models (e.g., a combination of CNN and LLM).
- Investigating the impact of different configurations of Cross-Attention layers on generation quality.
- Applying the method to specialized business tasks to enhance adaptability and control over the generated responses.

# 7 Conclusion

This work presents a novel approach to designing language models based on the use of *LLM Modules* and an Enhanced Cross-Attention mechanism for transferring knowledge from a large pre-trained model to a smaller model. Experimental results demonstrate that even with limited computational resources, significant improvements in generation quality can be achieved, as evidenced by the reduction in training and validation loss and enhanced coherence of the generated responses. Further research in combining various architectures promises new opportunities for adapting models to specific tasks.

# Code and Data Availability

The source code, trained weights, and full outputs are available at: `https://huggingface.co/kkolomeitsev/llm-modules` (DOI: 10.57967/hf/4462).

# References


[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, & Polosukhin, I. (2017). *Attention is All You Need*. In *Advances in Neural Information Processing Systems*.

[2] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & others. (2020). *Transformers: State-of-the-art Natural Language Processing*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38–45).

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). *Language Models are Few-Shot Learners. arXiv preprint* arXiv:2005.14165.

[4] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. arXiv preprint arXiv:1910.01108.

[5] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q., & Liu, X. (2019). *TinyBERT: Distilling BERT for Natural Language Understanding*. arXiv preprint arXiv:1909.10351.

[6] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., & Gelly, S. (2019). *Parameter-Efficient Transfer Learning for NLP*. In *International Conference on Machine Learning (ICML)*.

[7] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv preprint arXiv:2106.09685.

[8] Pfeiffer, J., Houlsby, N., Tracey, M., Bolukbasi, M., & Ruder, S. (2020). *AdapterFusion: Non-Destructive Task Composition for Transfer Learning*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

[9] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. Journal of Machine Learning Research.