Growing Transformers: Modular Composition and Layer-wise Expansion on a Frozen Substrate

Anonymous authors
Paper under double-blind review

Abstract

The prevailing paradigm for scaling large language models (LLMs) involves monolithic, end-to-end training, a resource-intensive process that lacks flexibility. This paper explores an alternative, constructive approach to model development, built upon the foundation of non-trainable, deterministic input embeddings. Building upon the recent finding that high-level semantic reasoning can emerge in Transformers using frozen embeddings derived from the visual structure of Unicode glyphs, we demonstrate that this fixed representational substrate acts as a universal "docking port," enabling two powerful and efficient scaling paradigms: seamless modular composition and progressive layer-wise growth. First, we show that specialist models trained on disparate datasets (e.g., Russian and Chinese text) can be merged into a single, more capable Mixture-of-Experts (MoE) model, post-training, with zero architectural modification. This is achieved by simply averaging their output logits. The resulting MoE model exhibits immediate performance improvements on reasoning benchmarks like MMLU, surpassing its constituent experts without catastrophic forgetting. Second, we introduce a layer-wise constructive training methodology, where a deep Transformer is "grown" by progressively stacking and training one layer at a time. This method demonstrates stable convergence and a clear correlation between model depth and the emergence of complex reasoning abilities, such as those required for SQuADv2. Our findings suggest a paradigm shift from monolithic optimization towards a more biological or constructive model of AI development, where complexity is built incrementally and modules can be composed freely. This opens new avenues for resource-efficient scaling, continual learning, and a more democratized ecosystem for building powerful AI systems. We release all code and models to facilitate further research.

1 INTRODUCTION

The pursuit of artificial general intelligence has led to the development of increasingly massive Large Language Models (LLMs). The dominant methodology for their creation is monolithic pre-training, where a model with a fixed, gargantuan architecture is trained end-to-end on vast datasets. While effective, this approach is computationally prohibitive, environmentally costly, and fundamentally inflexible. Once trained, modifying or extending such models without inducing catastrophic forgetting is a significant challenge.

Bochkov (2025) challenged a core assumption of this paradigm: the necessity of trainable input embeddings. It demonstrated that Transformers can achieve robust convergence and strong reasoning performance using a completely frozen embedding layer based on deterministic, non-semantic visual representations of Unicode characters. This finding established that semantic understanding is an emergent property of the Transformer's compositional architecture, not a feature of its input vectors.

This paper investigates the profound architectural implications of that discovery. If the input representation is a fixed, universal constant, can it serve as a standardized substrate for building models in a more modular and efficient manner? We answer this question affirmatively by proposing and validating a "constructive learning" framework. We hypothesize that a shared, frozen embedding space can act as a universal interface,

or a "docking port," allowing separately trained neural components to be combined and complex architectures to be grown incrementally.

We validate this hypothesis through two sets of experiments:

- Seamless Modular Composition: We demonstrate that specialized models, pre-trained on different languages, can be merged into a Mixture-of-Experts (MoE) model post-hoc. Because they share an identical representational input/output space, their learned knowledge can be combined by simply averaging their predictions, leading to a system that is more capable than its individual parts.
- Progressive Layer-Wise Growth: We introduce a method for "growing" a Transformer model from the ground up. We start with a single Transformer layer and train it to convergence. We then freeze it, stack a new layer on top, and repeat the process. This incremental, layer-by-layer construction mimics a more natural growth process and allows us to observe the emergence of capabilities as a direct function of model depth.

Our results show that these methods are not only viable but highly effective, offering a path towards more sustainable, adaptable, and interpretable model scaling.

Code and all artifacts are released to foster further progress in both science and industry[1].

2 RELATED WORK

Our work builds upon several lines of research in neural network training and modularity, but offers a distinct synthesis enabled by our frozen-embedding foundation.

2.1 Greedy Layer-Wise Training

The idea of training deep networks one layer at a time was pioneered by Hinton et al. (Hinton et al., 2006) for Deep Belief Nets and explored by Bengio et al. (Bengio et al., 2007) for deep autoencoders. These methods were primarily used for pre-training to find a good initialization for subsequent end-to-end fine-tuning. Our approach differs fundamentally: we use progressive layer-wise training not for initialization, but as the primary, constructive training process itself, building upon a fixed, non-trainable base.

2.2 Mixture of Experts (MoE) and Model Merging

MoE models, such as those by Shazeer et al. (Shazeer et al., 2017) and Fedus et al. (Fedus et al., 2022), use gating networks to dynamically route inputs to specialized subnetworks. However, these are typically trained jointly from the start. Model merging techniques, such as Model Soups (Wortsman et al., 2022), often average model weights, which requires careful alignment. Our approach is radically simpler: because our expert models share an identical, deterministic input/output vocabulary mapping via the frozen embeddings, their logits are directly comparable. This allows for a "zero-cost" merge by simple averaging, a capability not present in standard LLMs with independently trained embeddings.

2.3 Progressive and Modular Architectures

Progressive Neural Networks (PNNs) (Rusu et al., 2016) achieve continual learning by adding new network "columns" for each new task. Adapter-based methods like AdapterFusion (Pfeiffer et al., 2021) inject small, trainable modules into a frozen base model. Our layer-wise growth is a form of vertical, rather than horizontal, expansion. The use of Low-Rank Adaptation (LoRA) (Hu et al., 2022) in our later growth stages serves as an efficient tool for global readjustment, but the core principle remains constructive and layer-wise.

The novelty of our work lies in demonstrating that a frozen, non-semantic representational substrate makes these techniques drastically simpler and more powerful in the context of modern Transformers.

[1]The code to reproduce our experiments is provided in the supplementary materials.

3 CONSTRUCTIVE LEARNING METHODOLOGY

The cornerstone of this methodology is the frozen visual Unicode embedding layer, as detailed in (Bochkov, 2025). This provides a fixed, deterministic mapping from any token in a vocabulary to a d_{model} -dimensional vector. This shared (V, d_{model}) embedding matrix is the immutable foundation for all models in this study.

3.1 Seamless Model Composition via a Shared Substrate

Given two or more independently trained "expert" Transformer models, M_a, M_b, ..., which were all trained using the identical frozen embedding layer, we can compose them into an MoE model M_{moe}. Each expert M_i has its own trained Transformer blocks but shares the same input `nn.Embedding` layer and output projection shape.

For a given input sequence `idx`, each expert produces a logits tensor: $logits_a = M_a(\texttt{idx})$ and $logits_b = M_b(\texttt{idx})$. Since the vocabulary and its indexing are identical across models, the j-th element of the logit vector from any model corresponds to the same token. This enables a simple and effective merging strategy:

Logit Averaging: The combined logits are computed as $logits_{\text{moe}} = (logits_a + logits_b)/2$. This method is parameter-free and can be applied at inference time without any further training.

Adapter-based Fusion: For potentially higher performance, the logits can be concatenated and passed through a small, trainable adapter: $logits_{\text{moe}} = \text{adapter}(\text{concat}(logits_a, logits_b))$.

In our experiments, we primarily use logit averaging and demonstrate that even this simple approach yields significant gains, which can be further improved with minimal fine-tuning of the output layer.

3.2 Progressive Layer-Wise Growth

Instead of initializing a deep, N-layer Transformer and training it all at once, we "grow" it iteratively. The process, analogous to a locomotive pulling one wagon at a time, is as follows:

Initialization: Create a model M_1 with a single Transformer block ($n_{\text{layer}} = 1$, model: 'abs-bvv-1') on top of the frozen embedding layer.

Train Layer 1: Train M_1 on the full corpus until convergence.

Freeze and Stack: Freeze the weights of the trained block in M_1. Add a new, randomly initialized Transformer block on top, creating model M_2.

Train Layer 2: Train M_2 ('abs-bvv-2'), where only the weights of the newly added second layer are trainable.

Iterate: Repeat this process, adding one layer at a time (M_3, M_4, ...'), training only the newest layer at each step. This incremental process acts like a curriculum, with each new layer learning to compose the representations produced by the already-competent stack beneath it.

For deeper models (e.g., $n_{\text{layer}} >= 3$), we explore a variant where, upon adding a new layer, we use LoRA (Hu et al., 2022) to perform a low-rank update on the weights of all existing layers. This allows the entire network to adapt to the new depth with minimal trainable parameters, preventing ossification and promoting global coherence.

4 EXPERIMENTS AND RESULTS

4.1 Results: Seamless Model Composition (MoE)

We use the model families (`best_bvv`, `max_bvv`, `nemo_bvv`) from Bochkov (2025) for the MoE experiments. For the progressive growth experiment, we construct a new, larger model with $d_{\text{model}} = 4096$ and $n_{\text{head}} = 32$. All models use the frozen visual embeddings based on the `bvv241` tokenizer. Training data is a ≈9B token mix of Wikipedia and SFT datasets. Evaluation is performed on MMLU, ARC, CommonsenseQA, and SQuAD.

We merged language-specific experts (EN + Russian and EN + Chinese) into MoE models. Critically, the training process for the MoE model did not start from a high-loss state. For instance, 'best_bvv_moe' 's initial validation loss was already low (2.7), close to the final loss of the experts, before converging further to 2.044 (Figure 1). This confirms that the composition immediately leverages the combined competence of the experts without catastrophic interference. Figure 2 shows a similar trend for 'max_bvv_moe'. Figure 3 visualizes the performance gains of the merged models over their constituent experts.

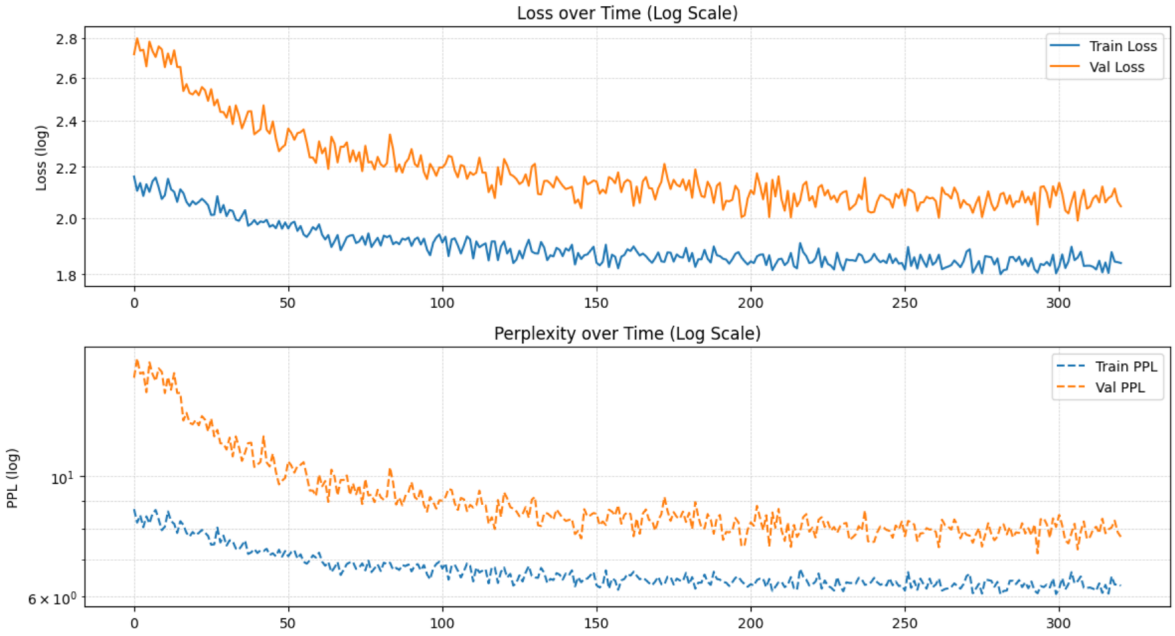

Figure 1: Training dynamics for the 'best_bvv_moe' model. The low starting loss indicates successful knowledge transfer from expert models, followed by further optimization.

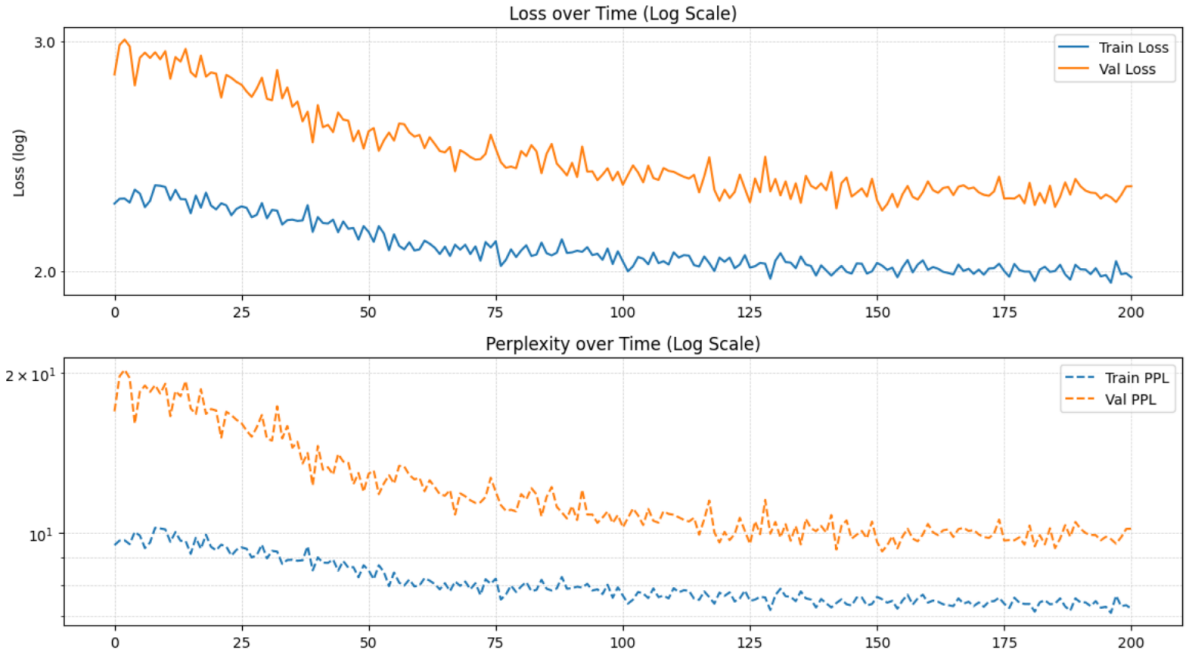

Figure 2: Training dynamics for the ‘max_bvv_moe‘ model, mirroring the successful convergence pattern seen in ‘best_bvv_moe‘.

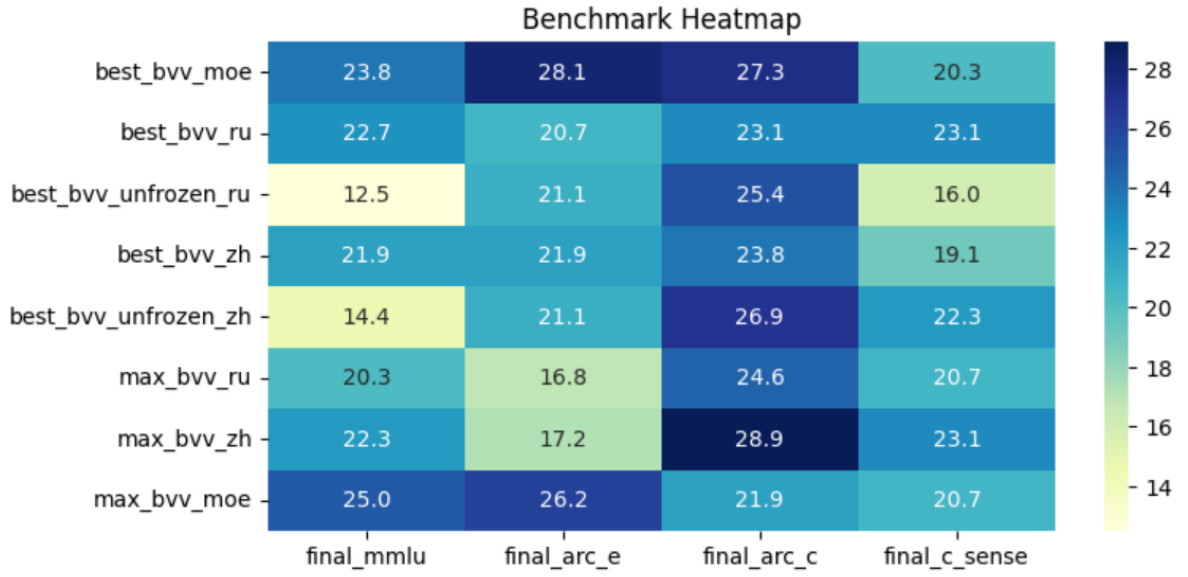

Figure 3: Performance comparison: The merged MoE model (best_bvv_moe) demonstrates synergistic capabilities, achieving superior performance on key reasoning benchmarks like MMLU compared to its individual expert counterparts.

4.2 Results: Progressive Layer-Wise Growth

We grew a model from 1 (‘abs-bvv-1‘) to 6 layers (‘abs-bvv-6‘). Figure 4 illustrates the training process. Each sharp spike in the loss corresponds to the addition of a new, untrained layer, followed by rapid convergence, demonstrating the stability of the method.

The results, summarized in Figure 5, reveal a clear pattern. General reasoning ability on MMLU increases steadily with depth, from 18.08% with one layer to 21.63% with six. More strikingly, the ability to perform complex extractive question-answering (SQuAD) is virtually non-existent in shallow models (1.21% at $n_{\text{layer}} = 1$). A significant signal appears only at ‘n_layer=3‘ (3.75%), and reaching its peak performance in our experiments at $n_{\text{layer}} = 6$ (5.55%). This compellingly demonstrates that complex capabilities are an emergent property of model depth. Figure 6 shows how performance on various MMLU subjects evolves as layers are added.

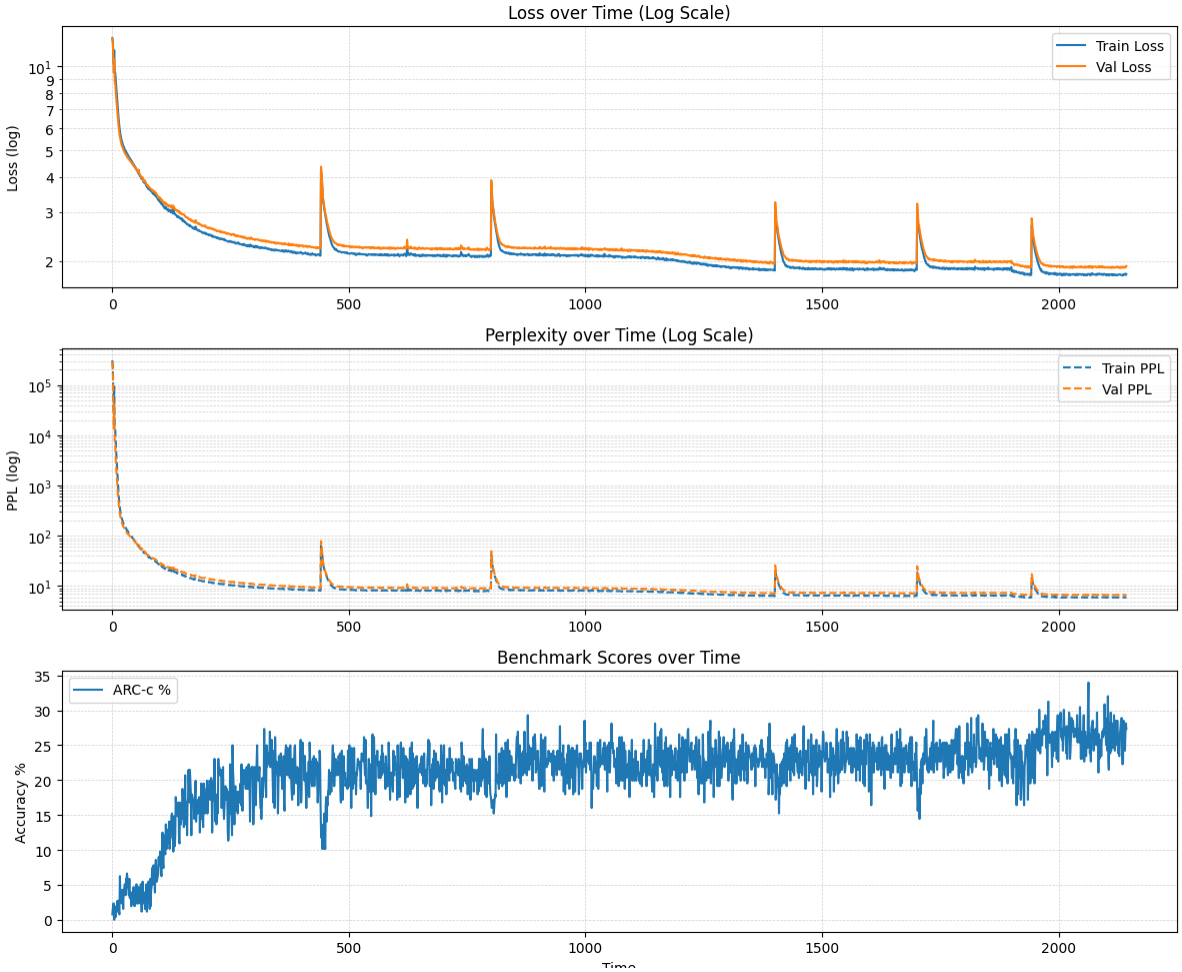

Figure 4: Training dynamics during progressive layer-wise growth. Each loss spike marks the stacking of a new layer, followed by rapid convergence. The ARC-c metric shows a corresponding increase in capability.

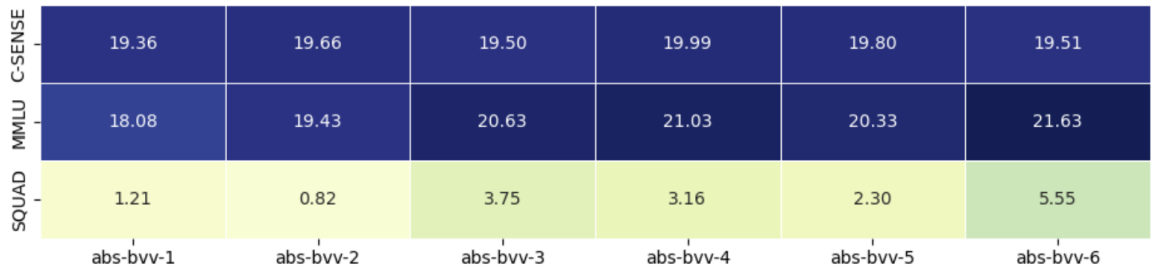

Figure 5: Benchmark performance as a function of model depth. Note the significant jump in SQuAD score at ‘n_layer=3‘, indicating the emergence of complex reasoning.

Figure 6: MMLU performance on select subjects as a function of model depth, illustrating how different reasoning capabilities strengthen as the model grows.

5 DISCUSSION

Our findings present a compelling case for a constructive approach to building LLMs, moving away from the monolithic paradigm.

5.1 The Frozen Embedding as a Universal Docking Port

The key enabler for our methods is the shared, frozen representational substrate. It acts as a universal standard, an "API" or "docking port" for neural components. When all models speak the same fundamental input/output language of visual forms, their higher-level learned knowledge (the transformations within their layers) becomes interoperable. This resolves a major hurdle in model merging and modularity, which is typically plagued by incompatible, independently learned embedding spaces.

5.2 From Monolithic Forging to Constructive Growth

Standard LLM training is like trying to forge a complex machine from a single, molten block of metal—immensely difficult and inflexible. Our progressive growth method is more akin to building a skyscraper floor by floor, or a living organism growing cell by cell. Each new layer builds upon a stable, functional foundation. This incremental process is more computationally tractable, more interpretable, and inherently more adaptable. The emergence of complex abilities like SQuAD performance only at significant depth is analogous to the development of higher-order cognitive functions in a growing brain, which require a sufficient hierarchy of neural processing.

5.3 Implications for a Modular AI Ecosystem

This paradigm has significant implications for the future of AI development:

- Resource Efficiency and Specialization: Organizations could train smaller, expert models on proprietary or specialized data. These experts could then be merged or sold as compatible "plugins" for larger, general-purpose models.
- Continual Learning: New knowledge and skills can be added by training and adding new expert modules or by extending the core model with new layers, drastically reducing the risk of catastrophic forgetting.
- Democratization: This approach lowers the barrier to entry. Instead of requiring the resources to train a 100 B+ parameter model from scratch, researchers could contribute to a larger ecosystem by developing and sharing smaller, compatible modules.

6 CONCLUSION

Building on the foundation of frozen visual embeddings, we have demonstrated two novel and efficient paradigms for scaling Transformer models: seamless post-hoc composition of expert models and progressive layer-wise growth. Our experiments show that specialist models can be merged into a more capable whole without costly retraining, and that deep models can be "grown" incrementally, with complex reasoning abilities emerging as a direct function of depth.

This work reframes the challenge of scaling AI from a monolithic endeavor to a constructive and modular one. By establishing a fixed, universal representational substrate, we unlock a new design space for creating powerful, flexible, and efficient AI systems. This "constructive learning" paradigm offers a promising path toward a more sustainable and collaborative future for artificial intelligence.

References

Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems 19*, pp. 153–160, 2007.

A. Bochkov. Emergent semantics beyond token embeddings: Transformer lms with frozen visual unicode representations, 2025. URL https://arxiv.org/abs/2507.04886.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Adapterfusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 686–700, 2021.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Hadsell, and Felix Heß. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Mitchell Wortsman, Gabriel Ilharco, Hojin Kim, Rémi Gontijo-Lopes, Ali Farhadi, Hannaneh Hajishirzi, and Ludwig Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy. In *International Conference on Machine Learning*, pp. 24144–24164. PMLR, 2022.