Mid-scale RI-1 (M1:IP), National Deep Inference Facility for Hundred-Billion-Parameter Language Models (NDIF)

PI: David Bau; Co-PI: Byron C. Wallace; Co-PI: Arjun Guha; Co-PI: Jonathan Bell; and Raphael Schroter, Northeastern University

This preliminary proposal for the **implementation of a mid-scale RI-1** outlines a plan to design, build, and deploy **computing infrastructure for deep inference**, to enable and propel American academic research to advance scientific understanding of state-of-the-art very large language models, which now dominate artificial intelligence (AI). The **NDIF** proposal consists of:

1. Development of open-source software that will enable deep inference research on very large language models, a critical research capability not widely available to U.S. academics today.
2. The creation and testing of a computing cluster that will deploy the software to provide a national deep inference service that will be made available to academic researchers.
3. Finally, training for PhD students and outreach and support for U.S. researchers to use this facility to advance understanding of very large neural network models.

This computing infrastructure will be developed at Northeastern University, building on our existing organizational structure, facilities, and experience in research computing. The hardware cluster will be deployed at the Massachusetts Green High Performance Computing Center.

1 Introduction

One of the great challenges to continued progress in research on large-scale machine learning is maintaining scientific transparency, collaboration, and innovation as AI enters into an era dominated by very large "pre-trained" models. The development of massive neural language models such as GPT-3 [1] and ChatGPT [2] has yielded blistering progress with respect to AI capabilities. But researchers interested in analyzing *how* they work by probing the internal workings and pushing the limits of such models are hindered by inadequate access to resources and infrastructure.

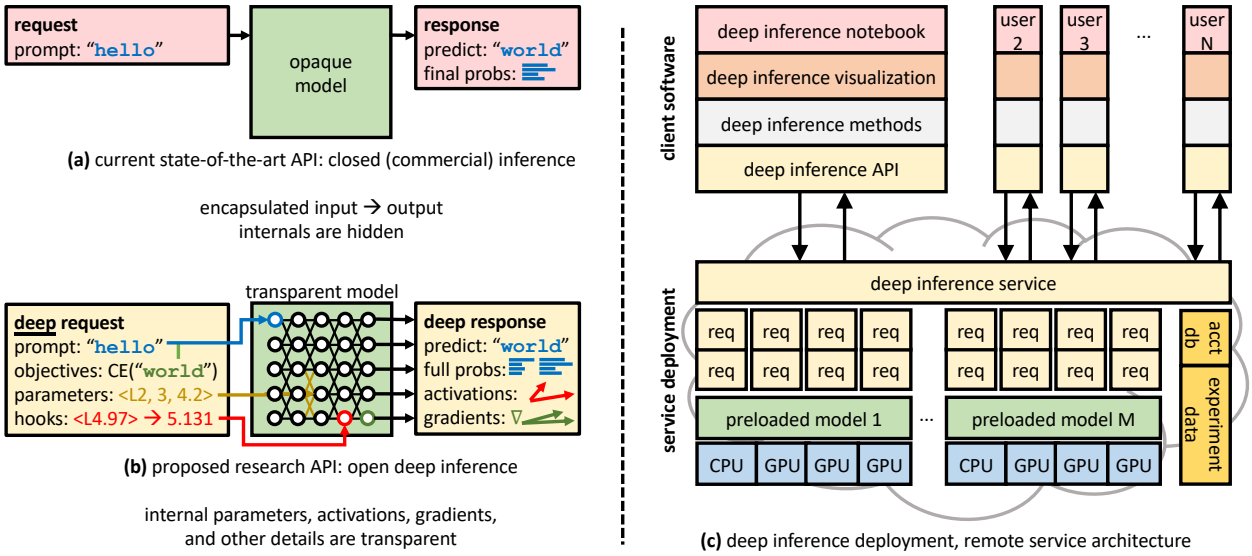

Figure 1: (a) Current services hosting large language models provide very limited interaction functionality (Top). One can send input text in a request, and is then provided an output string (and scores associated with the final predictions). (b) We propose developing infrastructure to provide deep access *to hosted language model instances (bottom), which will permit critical research without necessitating researchers hosting such models themselves. (c) The infrastructure consists of new software libraries and a deployed distributed service to be shared by researchers nationwide.*

We propose to develop infrastructure—open-source tools that can be replicated at other universities and a hosted instance that makes these available to the broader academic research community across a set of state-of-the-art models—to support advanced research on the modern, massive language models which are now central to AI. Figure 1 illustrates our proposed infrastructure agenda. The overarching aim is to develop tools and services that will support academic research on very large language models by permitting *deep inference*. Current state-of-the-art models are so large that even *running them*—nevermind training, which is substantially more computationally expensive—is not feasible for most academic researchers, owing to the cost and complexity of hardware necessary to support such inference.

As a result, many academics now use commercial inference servers to run experiments. Such services make large neural networks available only as opaque black-boxes that produce outputs directly from a set of inputs (Figure 1a), which greatly limits the experiments one can perform, and so constrains science. By contrast, deep inference (Figure 1b) would provide access to internal model activations, parameters, and gradients; this transparency is *necessary* for research investigating model inner-workings. To spread benefits and amortize the overhead of deep inference research, we propose deploying a service architecture that can support researchers at multiple institutions (Figure 1c), and developing software that can be replicated at other universities.

Leadership Team This project brings together an interdisciplinary group with deep expertise in machine learning (ML)/natural language processing (NLP), programming languages, and large-scale computing. PI Bau (Assistant Professor in Khoury College of Computer Sciences at Northeastern) is a leading researcher in interpretability of large neural networks [3–6] and editing of large models [7–10]. Bau also brings over 20 of software engineering experience in developing and deploying major software platforms for Google, Microsoft, and open-source nonprofits. Co-PI Wallace (Sy and Laurie Sternberg Interdisciplinary Associate Professor in Khoury) has extensive research expertise in NLP and interpretability of such models [11–18]. Co-PI Guha (Associate Professor in Khoury) brings deep experience in programming languages, including language-based security [19–22], GPU-accelerated domain specific languages [23, 24], and pre-trained models for code generation [25]. Co-PI Bell (Assistant Professor in Khoury) is an expert in software engineering and systems, including architectural design [26], testing and continuous integration [27–30], and analysis [31–33]. Team-member Schröter (Director of Research Computing at Northeastern University) organizes strategic planning for research computing resources at Northeastern, including on-premise hybrid cloud infrastructure, and he works with university researchers across all disciplines, to achieve research goals using shared high performance computing infrastructure.

2 Scientific Justification

2.1 The opportunities represented by large pretrained models

Large pre-trained models have shown impressive versatility across many difficult tasks and have shown apparent in-context generalization and meta-learning capabilities. These complex phenomena have emerged despite the simplicity of the self-supervised language modeling training regime [34]. Typical language model pre-training objectives are straightforward: for example, one commonly trains models to predict the next word in a sequence, given the preceding words. Despite this simplicity, when trained on huge datasets of text, large models such as GPT-3 [1] are capable—to varying degrees—of answering factual questions about the world [35], translating between natural languages [1], performing mathematical reasoning [36], writing computer code based on English specifications [37], obeying descriptive requests to perform a variety of tasks [1], and extrapolating performance of a new task given a small set of input examples [1].

What has enabled the emergence of such apparently sophisticated behaviors? Smaller models

trained with exactly the same objectives do not exhibit the same range of capabilities. Therefore, it is widely believed that the main enabler has been an increase in model size by several orders of magnitude. The varied complex behaviors engendered by this model scaling have motivated a nascent and growing subfield in which researchers aim to characterize the capabilities of models and probe *how* they work. This line of research was galvanized by BERT [38], a precursor of modern (much larger) models. Despite its modest size by current standards, BERT demonstrated capabilities sufficiently interesting to motivate a body of work clarifying the structure and internal representations of that model [39]. These efforts were possible in part because BERT-scale networks are sufficiently small that academic researchers can run (and probe) them on academic-scale hardware, i.e., single mid-tier GPU equipped machines.

2.2 Huge models have created a crisis of transparency

While the emergence of very large models, such as GPT-3, has energized the NLP and broader ML research communities, at the same time the dominant success of such models presents the research community with a new crisis of transparency that is very different from the previous generation of "large-scale" AI. When AlexNet shocked the computer vision community in 2012 by winning the ImageNet Visual Recognition Challenge [40], that model encapsulated its complexity in 62 million learned parameters. The size was large for the period, but still sufficiently small for academic labs to be able to reproduce, validate, modify, retrain, and study the model. Similarly, when the first successful pre-trained models for NLP—e.g., ELMO [41] and BERT [38]—emerged, these were large by the standards of the time, but still small enough for academic researchers to run, interrogate, and tinker with locally, enabling important research into the capabilities and limitations of the "first generation" of pre-trained neural NLP models [39]. That accessibility led to an explosion of creativity and innovation, with a doubling of AI papers published annually from 2011 to 2021, and a 30-fold increase in the annual number of AI-related patents filed [42].

However, the current advance represented by GPT-3 scale models is qualitatively different. The 175-billion parameter GPT-3 model is private. Alternative, open-access large language models (such as OPT [43], Bloom [44], and NEO-X [45]) are technically available to researchers, but are often *de facto* inaccessible due to their size. Academic researchers do not in general have sufficient resources to run such models, and so they are unable to probe them in depth. Most academic work on analyzing large language models therefore relies on the paid Application Programming Interfaces (API) made available by OpenAI or other commercial vendors.

The primary advantage to using inference API services is that they obviate the need for one to run (very large) models locally to interact with them. However, this approach comes with a critical trade-off: Commercial inference APIs provide only limited access to model outputs, which ensures that model weights remain proprietary. However, this precludes researchers from probing the internals of models and characterizing the internal mechanisms that models have learned from data. These limits threaten to slow the pace of innovation, shielding new developments behind the cloak of private ownership where advances in AI cannot be subject to the kind of competitive scrutiny that is provided by independent academics. Next we make the case for the need for deep inference among academic researchers and characterize specific research lines that are only possible to pursue with such access.

2.3 Research Enabled by Deep Inference

When asked online, [1] over 400 members of the research community say that the proposed National Deep Inference service would support their research. Several researchers point out the strong

[1] People on a Twitter thread in December 2022 were asked to respond if they had research that would be enabled by a transparent national inference service for large models.

need given the practical difficulties of investigating models whose parameters do not fit into the memory of a typical research computing node. Professor Boaz Barak (Harvard) observes, "Any model that doesn't fit on one GPU starts to be complicated for researchers to use even if they do have enough GPUs to fit... A central engineering resource that all academics can share would be a game changer." Professor Tom Dietterich (Oregon State) says, "I strongly support a public National Deep Inference service.... We will want to support many different things: fine tuning, access to the training data, access to external resources." Professor Ana Marasović (University of Utah) notes, "Having academic access ... would enable not only machine learning academics, but also academics without expertise in training models, to study large language models."

The need for a large-model inference service is widely recognized because the community has discovered that such models exhibit qualitatively different capabilities than small models. A recent survey of established benchmarks [34] catalogued over 175 different capabilities that emerge in large-scale models but that do not appear in smaller models. These include the ability to perform multidigit arithmetic, unscramble words, and correctly select truthful answers when baited by commonly-stated misconceptions.

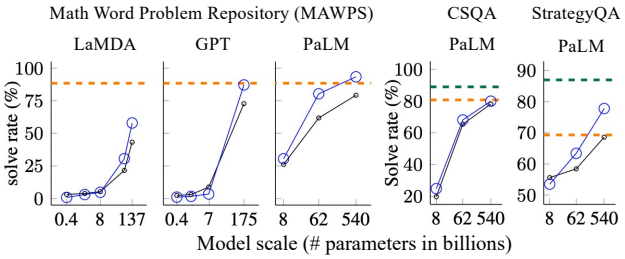

Figure 2: Capabilities that emerge in large models (data from [46]). Mathematical reasoning, complex sequential question-answering, and multistep strategic reasoning do not appear in small language models, but they emerge in the largest language models when chain-of-thought prompts are used. The mechanisms giving rise to these capabilities are not well-understood. A core goal of this infrastructure proposal is to enable academic researchers to develop a better understanding of how and why such large-language-model capabilities emerge.

Several emergent behaviors are among the most interesting that have been observed in any machine-learned model. For example, strategic multi-step reasoning [47] and handling complex question sequences in a conversation [48] are hallmarks of high-level reasoning. Large models can handle these tasks when prompted to use "chain-of-thought" generation [46], but this ability does not emerge in small models[2] (Figure 2). Another striking characteristic of very large models is that they have been observed to perform well under domain shift [50]. Robustness of model behavior under such shifts has been a major concern in machine learning, and it is an unexpected development to see very large models that become more robust to distribution shift, without special training methods, as the parameter count grows beyond 100 billion.

2.4 Understanding Learned Algorithms

The emergent capabilities of large models pose a fundamental question: *How do they work*? When a large model makes a surprising decision, what information does the model use to inform its decision, and what rules does the model apply to make its choice? Understanding such mechanisms is important when working to distinguish profound computational capabilities from the mere appearance of capabilities.

Representation probing. One major line of inquiry investigates internal mechanisms by asking: what information does the network contain? For example, it has been found that even when a language model is conditioned to output falsehoods, it may contain a hidden state that represents the true answer internally [51], suggesting that large models may exhibit deceptive or pandering behavior. Such a gap between external failure modes and internal state can only be identified

[2]Instruction fine tuning [49] can induce CoT in large models at about 62 billion parameters.

by probing model internals. Representation probing has been used to characterize the behaviors of smaller models [39, 52–56], but applying these methods to understand large models requires transparent inference that provides access to internal state.

Attention mapping. A model can also be understood by asking: What parts of the input is it attending to? Analyzing attention in small models has revealed how simple dependencies are processed [57–59], including the discovery of very explicit copying circuits in transformer models [60]. Analyzing per-token model probabilities can reveal model self-knowledge [61] and differences between human and AI-generated text [62]. Extending these lines of inquiry to large models will require transparent access to model internals.

Causal mediation analysis. Another way emergent learned algorithms can be understood is through measuring the impact of modifying individual computational steps within a model. Such causal analysis has been applied to identify attention heads that mediate gender bias in language models [63]; indirect object identification in sentences that name multiple subjects [64]; and the recall of world knowledge within large language models, such as knowledge of the relationships, associations and properties of real-world entities [9, 10]. Applying these methods to large models will require direct access to internal state.

2.5 Lightweight Fine-Tuning

One of the most compelling properties of large language models is their ability to perform *few-shot* learning, i.e., learn to perform new tasks from a handful of examples. In the era of massive pre-trained base models, methods to efficiently fine-tune them for specific tasks of interest has become an important topic. Yet this important direction for research requires access to model gradients, not provided by existing commercial offerings.

Continuous prompt tuning. One strategy for few-shot learning entails *prompting* [65] models with in-context examples such as sets of fill-in-the-blank prompts. A downside to this paradigm is that it requires specifying a discrete prompt (i.e., template) upfront suitable to the task under consideration. Recent work has sought to instead effectively *learn* prompts from a few labeled examples, for example by synthesizing *continuous*-valued soft prompts via fine-tuning [66–68]. However, even this lightweight training strategy requires access to gradients in the underlying language model, which is not available under with commercial inference services.

Training adapter layers. Another approach to lightweight fine-tuning involves fitting *adapter layers* [69], which are free parameters inserted into the network and then fine-tuned for a specific task (while other network parameters remain fixed). Adapter layer fine-tuning has been shown to permit comparatively efficient adaptation of large language models to new tasks [70]. But, like continuous prompt fine-tuning, updating adapter parameters requires gradients, and therefore evaluating and improving adapter methods requires an interface to models that supports deep inference. NDIF will provide this capability.

3 The Need for New Deep Inference Infrastructure

3.1 The three computational challenges facing large language model research

In the broader ML research ecosystem, there are three main computational challenges that confront researchers studying large language models, detailed below. This proposal is focused on the third of these challenges, since deep inference for research is not well-served by existing efforts.

1. **Training** a model at the 175-billion parameter plus scale is prohibitive for any individual academic lab. With 3,000× more parameters than AlexNet, the increase in scale has far outstripped the growth of hardware density given by Moore's law. At this size, training such a model can only be justified at national scale: for example, Google has estimated that the Microsoft

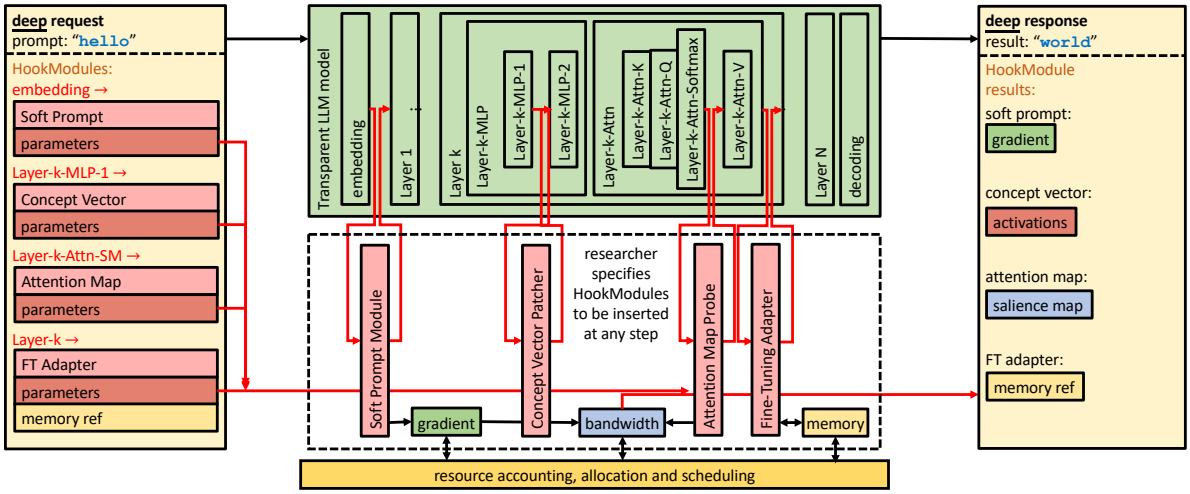

Figure 3: Details of a deep inference request. Unlike commercial inference that provides no transparency, with the NDIF, researchers can execute flexible experiments by inserting computations in the internals of the deep network inference process. To maintain safe and efficient co-tenancy, experiment computations are packaged as `HookModules` *that enable resource accounting and scheduling.*

cluster used by OpenAI to train GPT-3 consumed about 1.2Gwh of electricity to train that model once—that is the power of about 120 average US homes for a year.

2. **Simple inference (predictions)**: even after model training is complete, using a trained model for inference (i.e., to make predictions) can be prohibitive. 175 billion parameters would require 700Gb of RAM at single-precision, or 350Gb of RAM at half-precision. A machine capable of loading the parameters of modern neural networks in GPUs requires 5-10 high-end parallel computing devices, totalling between $100,000-$200,000; the cost of buying or renting time on such a machine puts it out of reach of most academic groups. To meet these needs, several commercial inference services (such as the OpenAI API) are available that amortize the cost of running large models across many users. However, all existing services are geared toward commercial applications, providing high speed and high reliability but providing only superficial access to outputs of models (Figure 1a).
3. **Deep inference to enable research**: to understand the mechanisms within large models, researchers use techniques such as representation probing, causal mediation analysis, salience mapping, direct model editing, soft prompting, and lightweight fine-tuning. However, these methods require access to model internals that go beyond the needs of commercial applications and that are not provided by commercial inference services. Furthermore, since providing researchers with full access to parameters or gradients would allow a user to inspect or even duplicate a model, providing open research access could put private investment in proprietary models at risk. Therefore, despite the demand and need from academic researchers, we should not expect commercial inference services to provide deep inference capabilities for large proprietary models.

3.2 Existing efforts to *train* large language models with open parameters

The need to **train** open, non-commercial 175-billion plus parameter models has been widely recognized, and several efforts are already in progress toward this end. Our proposal will make such models available as research subjects for academic researchers. Those models include:

- Meta OPT, a set of commercial language models trained by Meta, with parameters that are made available to academic researchers. OPT includes a 175-billion parameter model.

- BigScience Bloom, a 176-billion parameter multilingual model trained by BigScience, a collaboration of European agencies, the Huggingface company, and many others.
- Eleuther AI GPT-NeoX and GPT-J, 20-billion and 6-billion parameter language models trained by a research collaborative with support from Stability.AI, CoreWeave, and Google. The Eleuther team plans to continue with training a 150-200-billion parameter model.
- Tsinghua GLM, a 130-billion-parameter Chinese-English model supported by Zhipu.AI.
- Yandex YaLM, a 100-billion-parameter Russian-English model from Yandex.

Several ongoing efforts are also training large models to incorporate human feedback to explicitly follow instructions, similar to OpenAI's InstructGPT and ChatGPT: BigScience BloomZ fine-tunes Bloom to add human feedback; CarperAI will fine-tune EluetherAI models.

Despite the availability of such models with more than 100 billion parameters, academic research using these large models remains prohibitive because of the high cost of hardware and the complexity of software. Our proposal addresses this problem.

3.3 Existing commercial services supporting simple inference

Commercial inference services address the high overhead of running large pretrained language models by offering shared *inference API* services. These preload a few important models on shared severs, allowing many users to amortize the cost of running models. Our current proposal adopts a similar model, but unlike commercial inference services, **the goal of our proposed service is to support fundamental research**, which imposes requirements that go beyond commercial inference services. Commercially available inference APIs include: The OpenAI inference API providing access to OpenAI's GPT-3 and other large models; The Azure inference API, which offers several Microsoft-proprietary models; the Huggingface inference API; and the Cohere inference API.

Unfortunately, existing commercial solutions treat large models as a opaque black boxes. Commercial inference APIs provide natural-language responses to natural-language prompts, and they can also provide numeric scores for alternative predicted responses. However, commercial APIs don't allow reading or overriding specific activations, parameters, or gradients within the model while inference occurs. Such internal access is key for researchers who investigate large models.

4 Implementation

Large models pose several engineering challenges: loading can take minutes or hours, so a model should be used by as many requests and users as possible before being unloaded. Furthermore, a model spanning multiple GPU devices incurs orders of magnitude higher communication costs compared to a single-GPU model. The challenge is amplified by research needs: Unlike production applications of deep networks that have a simple, fixed input-output pattern, scientists investigating model internals will extract and alter hidden state data in novel ways, and experiment probes can drive bandwidth demands. A final challenge is to enable research flexibility while protecting users from each other: errors by a researcher must not degrade the functioning of the service.

We will address these engineering challenges though a modular software development architecture to enable flexible research experimental design, while maintaining efficient and safe co-tenancy of the service. Capabilities of the service will be enabled in phases, with modular functionality deployed incrementally to gather community feedback, participation, and testing.

4.1 Modular Software Architecture

All common experimental designs for deep model inference can be expressed by inserting additional computations between standard steps of deep network inference. To empower researchers to run complex experiments, the service API will enable such inserted computations using established modular idioms in the `PyTorch` [71] deep learning framework.

Figure 3 illustrates the service design. In addition to the standard input, an experiment request specifies a set of `HookModule` computations that execute bounded computations that can be assembled into an experiment. Each `HookModule` is a `PyTorch` `torch.nn.Module` that encapsulates, bounds and accounts for computing resources consumed prior to the processing of the request. The accounting allows a scheduler to plan and allocate requests in an order that prevents any one user from exhausting GPU memory or otherwise interfering with the shared service functionality.

4.2 Phased Hardware Deployment

The service will be backed by a 400-GPU cluster that will be procured and installed in phases. Sets of GPU devices will be hosted on high-memory nodes (with 8 or more GPUs per node), configured to maximize throughput for our application, for example, sized to enable a full inference pass for a large model to be able to be executed on a single node where possible. Suitable hardware design will reduce communication overhead and reduce the need for exotic interconnects between nodes.

While the software stack is under development, we will deploy the cluster in phases to capitalize on hardware advances during service deployment. Each year we plan to expand the cluster by 80 GPUs, with full capacity deployed in the fifth year.

4.3 Phased Software Development and Service Deployment

To ensure broad but efficient use of large model inference capabilities, our implementation will provide several software features that will be deployed in five phases.

Phase 1. Develop and deploy single-pass inference, resource accounting, and low-latency low-bandwidth and interfaces to a selected set of preloaded large models. Basic transparent state access will be enabled, and this will allow testing by a limited set of users.

Phase 2. Enable high-latency high-throughput batch access to enable larger-scale experiments, and remote data caching to reduce overhead for iterated algorithms such as multi-token generation, soft-prompt tuning, adapter training, and causal tracing.

Phase 3. Support remote data processing and backpropagation to allow scientists to fine-tune models and perform other nontrivial computations before results are communicated off-device.

Phase 4. Support on-premise jobs to support experiments that go beyond the scope of our ordinary services such as access to external data or other complex interactions.

Phase 5. Support self-hosted operation, with custom deployments in other clusters and clouds.

4.4 Outreach and Training

The goal of our project is to enable a broad range of impactful research into large language models. Therefore, after the initial API is deployed, we will conduct outreach and training to enable researchers to use the facility, and to gather feedback to improve it.

In phases 3 and 4, we will begin to train PhD students to use the service, and we will hold a pair of workshops aimed at enabling researchers. One research workshop will be geared towards the machine learning community, and a second research workshop will be interdisciplinary, including social science and humanities research applications. Then in phase 5 we will prepare instructional materials and hold an educational workshop for teachers and students. To facilitate this work, we plan development of tutorials and code releases aimed at researchers and teachers.

5 Evaluation and Oversight

Our project is driven by four measurable goals:

1. **Advance scientific understanding** of large language models.
2. Provide **broad access** to researchers and students for inference not served elsewhere.

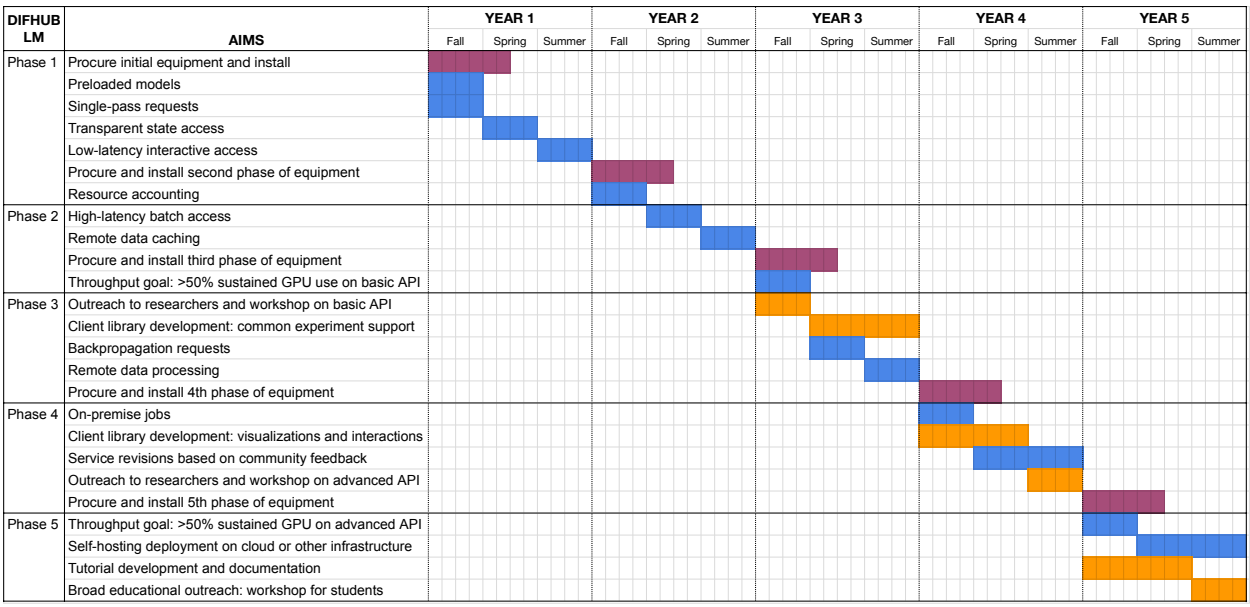

Figure 4: Five-phase deployment timeline.

3. Enable **efficient use** of scarce computational resources.
4. **Train students** on large models, to build the next generation of AI engineers and researchers.

These goals correspond to metrics that we will track. To measure our progress in realizing **impact** by providing **broad access** and **efficiency**, we will track and aim to increase:

- **Number of monthly academic users** of the deployed service, a core measure of reach.
- **Sustained GPU utilization** in the deployed service, a core measure of efficiency.
- **Number of peer-reviewed research works** that use our service or software in experiments.
- **The number of deployments** of our software stack on clusters beyond the initial service.

In addition, to ensure that our project remains impactful, efficient and fair, we will recruit an external advisory council to meet and review the project's progress and goals two times per year.

6 Operations and Maintenance

After the deep inference facility is deployed and operational, it will transition to regular operations and maintenance. Operations are not part of this RI-1 proposal, and funding for operations will be raised in future grant proposals. Ongoing operations will include three kinds of activities:

1. Oversight of research operations, including the provisioning of service to allow prioritization of important science objectives; and continuing outreach to the research community. This will be done by forming an academic research operations team.
2. Inference service operations, consisting of operation, monitoring, maintenance, updating, and security of the hardware cluster and the software service. These duties will be taken on by the Northeastern research computing operations team.
3. Open source code maintenance, consisting of coordinating, testing, and updating the open-source code base produced for NDIF. Coordination of the open source code base will be done by a software engineering team.

Continuing operations will be funded under future grants to further advance scientific goals.

7 Broader Impacts

7.1 Benefits to the wider U.S. research community

Academic NLP researchers in the U.S. are currently limited in their ability to run and analyze the very large language models which now dominate the field. Such groups will in general be unable to procure hardware sufficient to run these models locally. But current hosted offerings (e.g., as provided by OpenAI) are severely limited in terms of what they will provide to users.

The proposed infrastructure research will define new approaches to providing granular access to very large language models, and instantiate such access in an academically oriented end-point. This will confer substantial benefit to the U.S. AI research community, members of which must currently rely on private enterprises such as OpenAI to access such large-scale models. Such access is, however, restricted to such a degree that it precludes pursuing important, emerging technical questions related to interpreting and adapting massive language models. Providing such granular access to multiple large models via an intuitive API will greatly expedite critical NLP research.

7.2 Opportunities for student training

Integrating research into undergraduate curricula. We are committed to ensuring that undergraduates at Northeastern and beyond benefit from the proposed infrastructure. To this end we will develop materials—lectures, exercises, and homeworks—that cover analysis of large language models. We will pilot and refine these materials in relevant courses at Northeastern (e.g., Machine Learning I and II, Natural Language Processing, and Practical Neural Networks). Bau and Wallace regularly lead these offerings. Further, Wallace, as Director of the Bachelors in Data Science program (and serves on the undergraduate curriculum committee), is well-positioned to ensure that the developed materials are incorporated into the curricula of these courses.

Importantly, once developed, we will make these materials—which will use the hosted API developed and instantiated under this project—available to faculty at other institutions, scaling the impact by enabling undergraduates in CS across the U.S. to gain hands-on experience analyzing and working with the internals of massive language models, which increasingly dominating the AI landscape. Note that such exercises are not currently possible at the vast majority of institutions given the resources required to run such models. And even if students are willing and able to pay for access via a commercial API, they would not be able to access model internals in a granular way, in turn severely limiting the kinds of analysis possible.

Supporting undergraduate and graduate research. To train the next generation of researchers in AI and NLP, we must provide them the capabilities to run, analyze, and improve state-of-the-art systems—i.e., very large language models. The proposed infrastructure will provide that, allowing students to engage in cutting edge NLP research in a way not currently possible at most institutions.

Institutional commitment to Diversity and Inclusion (DEI). Khoury College of Computer Sciences is a leader in increasing representation in Computer Science (CS). Khoury is home to the Center for Inclusive Computing (CIC [72]), which aims to increase the representation of women majoring in CS across the U.S. Also, with NSF support, Khoury has developed the Align program [73], which provides a pathway to an MS in CS for students without CS backgrounds. This unique program attracts a notably diverse student body: For example, in 2018 half of the incoming class was female, and 15% was Hispanic, Latino, African-American, Native American, or Pacific Islander [74]. Khoury also has a verified College-wide Broadening Participation in Computing plan [75].

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[2] John Schulman, Barret Zoph, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Luke Metz, Michael Pokorny, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright, Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, and Nick Ryder. *ChatGPT: Optimizing Language Models for Dialogue*. `https://openai.com/blog/chatgpt/`. 2022.

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. "Understanding the role of individual units in a deep neural network". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30071–30078.

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. "Network dissection: Quantifying interpretability of deep visual representations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6541–6549.

[5] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. "Explaining explanations: An overview of interpretability of machine learning". In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE. 2018, pp. 80–89.

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks". In: *International Conference on Learning Representations*. 2018.

[7] David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. "Rewriting a deep generative model". In: *European conference on computer vision*. Springer. 2020, pp. 351–369.

[8] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. "Rewriting geometric rules of a gan". In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), pp. 1–16.

[9] Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. "Locating and editing factual associations in gpt". In: *Advances in Neural Information Processing Systems*. 2022.

[10] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. "Mass-editing memory in a transformer". In: *arXiv preprint arXiv:2210.07229* (2022).

[11] Pouya Pezeshkpour, Sarthak Jain, Byron Wallace, and Sameer Singh. "An Empirical Comparison of Instance Attribution Methods for NLP". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Online: Association for Computational Linguistics, June 2021, pp. 967–975. DOI: `10.18653/v1/2021.naacl-main.75`. URL: `https://aclanthology.org/2021.naacl-main.75`.

[12] Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. "Explaining Black Box Predictions and Unveiling Data Artifacts through Influence Functions". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. Online: Association for Computational Linguistics, July 2020, pp. 5553–5563. DOI: 10.18653/v1/2020.acl-main.492. URL: https://aclanthology.org/2020.acl-main.492.

[13] Jered McInerney, Geoffrey Young, Jan-Willem van de Meent, and Byron C. Wallace. "That's the Wrong Lung! Evaluating and Improving the Interpretability of Unsupervised Multimodal Encoders for Medical Data". In: *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*. 2022.

[14] Sarthak Jain, Varun Manjunatha, Byron C. Wallace, and Ani Nenkova. "Influence Functions for Sequence Tagging Models". In: *Proceedings of the Findings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*. 2022.

[15] Xiongyi Zhang, Jan-Willem van de Meent, and Byron Wallace. "Disentangling Representations of Text by Masking Transformers". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 778–791. URL: https://aclanthology.org/2021.emnlp-main.60.

[16] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. "ERASER: A Benchmark to Evaluate Rationalized NLP Models". In: *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*. 2020, pp. 4443–4458.

[17] Sarthak Jain and Byron C. Wallace. "Attention is not Explanation". In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2019, pp. 3543–3556.

[18] Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. "Learning to Faithfully Rationalize by Construction". In: *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*. 2020, pp. 4459–4473.

[19] Arjun Guha, Mark Reitblatt, and Nate Foster. "Machine Verified Network Controllers". In: *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. 2013.

[20] Arjun Guha, Matthew Fredrikson, Benjamin Livshits, and Nikhil Swamy. "Verified Security for Browser Extensions". In: *IEEE Security and Privacy (Oakland)*. 2011.

[21] Arjun Guha, Shriram Krishnamurthi, and Trevor Jim. "Using Static Analysis for Ajax Intrusion Detection". In: *World Wide Web Conference (WWW)*. 2009.

[22] Rian Shambaugh, Aaron Weiss, and Arjun Guha. "Rehearsal: A Configuration Verification Tool for Puppet". In: *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*. 2016.

[23] Abhinav Jangda, Sandeep Polisetty, Arjun Guha, and Marco Serafini. "Accelerating Graph Sampling for Graph Machine Learning Using GPUs". In: *European Conference on Computer Systems (EuroSys)*. 2021.

[24] Abhinav Jangda and Arjun Guha. "Model-Based Warp-Level Tiling for Image Processing Programs on GPUs". In: *International Conference on Parallel Architectures and Compilation Techniques (PACT)*. 2020.

[25] Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. *MultiPL-E: A Scalable and Extensible Approach to Benchmarking Neural Code Generation*. 2022. DOI: 10.48550/ARXIV.2208.08227.

[26] Nicolas Viennot, Mathias Lécuyer, Jonathan Bell, Roxana Geambasu, and Jason Nieh. "Synapse: A Microservices Architecture for Heterogeneous-Database Web Applications". In: *Proceedings of the Tenth European Conference on Computer Systems*. EuroSys '15. Bordeaux, France: Association for Computing Machinery, 2015. ISBN: 9781450332385. DOI: 10.1145/2741948.2741975. URL: https://doi.org/10.1145/2741948.2741975.

[27] Jonathan Bell, Owolabi Legunsen, Michael Hilton, Lamyaa Eloussi, Tifany Yung, and Darko Marinov. "DeFlaker: Automatically Detecting Flaky Tests". In: *Proceedings of the 2018 International Conference on Software Engineering*. ICSE 2018. 2018. URL: http://jonbell.net/publications/deflaker.

[28] Abdulrahman Alshammari, Christopher Morris, Michael Hilton, and Jonathan Bell. "FlakeFlagger: Predicting Flakiness Without Rerunning Tests". In: *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. 2021, pp. 1572–1584. DOI: 10.1109/ICSE43902.2021.00140.

[29] Jonathan Bell and Gail Kaiser. "Unit Test Virtualization with VMVM". In: *ICSE*. 2014.

[30] Jonathan Bell, Eric Melski, Gail Kaiser, and Mohan Dattatreya. "Accelerating Maven by Delaying Test Dependencies". In: *3rd International Workshop on Release Engineering*. RELENG '15. Florence, Italy: IEEE Press, 2015, p. 28. URL: http://dl.acm.org/citation.cfm?id=2820690.2820703.

[31] Jonathan Bell and Gail Kaiser. "Phosphor: Illuminating Dynamic Data Flow in Commodity JVMs". In: *ACM International Conference on Object Oriented Programming Systems Languages & Applications*. OOPSLA '14. Portland, Oregon, USA: ACM, 2014, pp. 83–101. ISBN: 978-1-4503-2585-1. DOI: 10.1145/2660193.2660212. URL: http://doi.acm.org/10.1145/2660193.2660212.

[32] Jonathan Bell and Luís Pina. "CROCHET: Checkpoint and Rollback via Lightweight Heap Traversal on Stock JVMs". In: *Proceedings of the 2018 European Conference on Object-Oriented Programming*. ECOOP 2018. 2018.

[33] Katherine Hough and Jonathan Bell. "A Practical Approach for Dynamic Taint Tracking with Control-Flow Relationships". In: *ACM Trans. Softw. Eng. Methodol.* 31.2 (2021). ISSN: 1049-331X. DOI: 10.1145/3485464. URL: https://doi.org/10.1145/3485464.

[34] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. "Emergent abilities of large language models". In: *arXiv preprint arXiv:2206.07682* (2022).

[35] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. "Language Models as Knowledge Bases?" In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 2463–2473. DOI: 10.18653/v1/D19-1250. URL: https://aclanthology.org/D19-1250.

[36] Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, et al. "A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level". In: *Proceedings of the National Academy of Sciences* 119.32 (2022), e2123433119.

[37] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. "Evaluating large language models trained on code". In: *arXiv preprint arXiv:2107.03374* (2021).

[38] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[39] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A primer in BERTology: What we know about how bert works". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 842–866.

[40] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[41] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. "Semi-supervised sequence tagging with bidirectional language models". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 1756–1765. DOI: 10.18653/v1/P17-1161. URL: https://aclanthology.org/P17-1161.

[42] Daniel Zhang, Erik Brynjolfsson, John Etchemendy, Terah Lyons, James Manyika, Juan Carlos Niebles, Michael Sellitto, Yoav Shoham, Nestor Maslej, Andre Barbe, Helen Ngo, Latisha Harry, Ellie Sakhaee, Benjamin Bronkema-Bekker, et al. "The AI index 2021 annual report". In: (2022). URL: https://aiindex.stanford.edu/wp-content/uploads/2022/03/2022-AI-Index-Report_Master.pdf.

[43] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. "Opt: Open pre-trained transformer language models". In: *arXiv preprint arXiv:2205.01068* (2022).

[44] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. "Bloom: A 176b-parameter open-access multilingual language model". In: *arXiv preprint arXiv:2211.05100* (2022).

[45] Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. "GPT-NeoX-20B: An Open-Source Autoregressive Language Model". In: *Proceedings of BigScience Episode\# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*. 2022, pp. 95–136.

[46] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. "Chain of thought prompting elicits reasoning in large language models". In: *Advances in Neural Information Processing Systems*. 2022.

[47] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. "Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies". In: *Transactions of the Association for Computational Linguistics* 9 (2021), pp. 346–361.

[48] Amrita Saha, Vardaan Pahuja, Mitesh Khapra, Karthik Sankaranarayanan, and Sarath Chandar. "Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[49] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[50] Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. "Prompting GPT-3 To Be Reliable". In: *arXiv preprint arXiv:2210.09150* (2022).

[51] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. "Discovering Latent Knowledge in Language Models Without Supervision". In: *arXiv preprint arXiv:2212.03827* (2022).

[52] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. "Learning to generate reviews and discovering sentiment". In: *arXiv preprint arXiv:1704.01444* (2017).

[53] Yonatan Belinkov and James Glass. "Analysis methods in neural language processing: A survey". In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 49–72.

[54] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. "Knowledge neurons in pretrained transformers". In: *arXiv preprint arXiv:2104.08696* (2021).

[55] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. "Transformer Feed-Forward Layers Are Key-Value Memories". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 5484–5495.

[56] Yonatan Belinkov. "Probing classifiers: Promises, shortcomings, and advances". In: *Computational Linguistics* 48.1 (2022), pp. 207–219.

[57] Jesse Vig. "A multiscale visualization of attention in the transformer model". In: *arXiv preprint arXiv:1906.05714* (2019).

[58] Jesse Vig and Yonatan Belinkov. "Analyzing the structure of attention in a transformer language model". In: *arXiv preprint arXiv:1906.04284* (2019).

[59] Samira Abnar and Willem H Zuidema. "Quantifying Attention Flow in Transformers". In: *ACL*. 2020.

[60] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. "In-context learning and induction heads". In: *arXiv preprint arXiv:2209.11895* (2022).

[61] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield Dodds, Nova DasSarma, Eli Tran-Johnson, et al. "Language models (mostly) know what they know". In: *arXiv preprint arXiv:2207.05221* (2022).

[62] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. "Gltr: Statistical detection and visualization of generated text". In: *arXiv preprint arXiv:1906.04043* (2019).

[63] Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M Shieber. "Investigating Gender Bias in Language Models Using Causal Mediation Analysis." In: *NeurIPS*. 2020.

[64] Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. "Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 small". In: *arXiv preprint arXiv:2211.00593* (2022).

[65] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. "Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing". In: *ACM Comput. Surv.* (2022). ISSN: 0360-0300. DOI: 10.1145/3560815. URL: https://doi.org/10.1145/3560815.

[66] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353.

[67] Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. "WARP: Word-level Adversarial ReProgramming". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 4921–4933.

[68] Guanghui Qin and Jason Eisner. "Learning How to Ask: Querying LMs with Mixtures of Soft Prompts". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 5203–5212.

[69] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. "Parameter-Efficient Transfer Learning for NLP". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 2790–2799. URL: https://proceedings.mlr.press/v97/houlsby19a.html.

[70] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. "On the Effectiveness of Adapter-based Tuning for Pretrained Language Model Adaptation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 2208–2222. DOI: 10.18653/v1/2021.acl-long.172. URL: https://aclanthology.org/2021.acl-long.172.

[71] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[72] *Center for Inclusive Computing at Northeastern University*. 2023. URL: https://cic.northeastern.edu/.

[73] *Align MS in Computer Science at Northeastern*. 2023. URL: https://www.khoury.northeastern.edu/programs/align-masters-of-science-in-computer-science/.

[74] Kamna Shastri. "Northeastern University creates new generation of tech leaders". In: *International Examiner* (2018).

[75] *Verified Departmental BPC Plans*. 2023. URL: https://bpcnet.org/verified-departmental-bpc-plans/.