# Research Statement

I work on machine learning (ML) and algorithms, with my main goal being to **establish learning from *multiple* tasks, datasets, and computations on firm theoretical and practical foundations**, allowing scientists and engineers to confidently use the resulting algorithms to power new innovations. Theoretically, my research advances our understanding of training on multi-distribution data, which underlies everything from the foundation models powering the latest breakthrough AI systems to the go-to methods in distributed learning. My work also guides the design and analysis of learning-augmented algorithms, a leading paradigm of studying computation beyond worst-case instances by incorporating predictions from previously-seen instances. In practical applications, I use insights from my work combined with domain-specific knowledge and collaborations to design algorithms for distributed and efficient ML, incorporate learning into releasing statistics and scientific computing, and automate the application of ML to understudied modalities.

My work involves studying how to learn algorithms across multiple tasks—e.g. datasets, computational instances, or training runs—and translating the insights into practice. It can mainly be split into three thrusts:

1. **Multi-task & meta-learning:** In many applications, we want to learn a good learning algorithm—e.g. find a good initialization for gradient descent—from a large collection of heterogeneous datasets or tasks; for example, large language models (LLMs) are pretrained on a big, multi-distribution corpus before being fine-tuned on target data. Understanding such settings requires going beyond the single-distribution paradigm of classical ML. I have shown some of the first provable guarantees for gradient-based meta-learning, a major approach in this area with applications such as federated learning (FL) and vision. My theory describes task-similarity conditions under which learning from multiple tasks is useful and prescribes algorithms that can exploit this similarity. It has been directly built upon theoretically by scientists in diverse areas such as algorithmic game theory and reinforcement learning. As one example of its practical implications, it inspired my method for tuning hyperparameters in FL which was found to consistently improve regular tuners in both my own and in recent independent evaluations.
2. **Learning-augmented algorithms:** Also known as *algorithms with predictions*, this rapidly growing subfield of theoretical CS designs algorithms whose performance can be improved by learned predictions of their outputs. It is a leading way of analyzing algorithms beyond worst-case instances and has had a significant practical impact in areas such as databases and energy systems. My work provides the first systematic understanding of the critical *learning* aspect of learning-augmented algorithms, introducing a unified way to determine learnability and in doing so dramatically improving several existing theoretical results while proving many new ones. I have since worked on expanding the scope of learning-augmented algorithms beyond online and graph algorithms, including to privacy-preserving statistics as a Google intern and to scientific computing in collaboration with a domain expert at Georgia Tech.
3. **Efficient & automated ML:** My third area of focus is on extending the achievements of modern ML to low-resource settings and understudied tasks. A significant portion of my work involved developing our theoretical and practical understanding of optimization and generalization in model compression and neural architecture search, leading to better methods for these highly relevant areas. In the last few years, I have also led a push to make automated ML and foundation models work for diverse domains beyond vision, text, and audio. In addition to introducing several state-of-the-art methods via novel architecture search space design and transfer learning approaches, I also oversaw the creation of the main benchmark and co-organized the first major competition in this important applied area.

These lines of work combine tools from ML, algorithm design, and optimization to design methods and translate them into practically useful implementations. In the future, I aim to build upon these results in several main directions, connecting my aims of deploying principled and effective multi-task methods at scale and of democratizing ML to diverse modalities and resource levels. A major goal is to develop **an empirical theory of multi-distribution learning**, combining controlled experimentation with mathematical analysis to transform our understanding of modern learning beyond what classical theory can provide. This direction will build heavily on my theoretical experience in meta and representation learning alongside practical experience in core application areas such as language and the natural sciences. Another key focus is on **developing and deploying learning-augmented algorithms**, expanding my collaborations with theoretical CS while also focusing more deeply on areas such as scientific computing, applied statistics, and efficient large-scale ML. Lastly, building upon my pioneering work on automated ML for diverse tasks, I plan to create **new algorithms and workflows for bringing large-scale pretraining to bear on understudied applications**.

## Multi-task and meta-learning

Alleviating the lack of data in individual tasks by incorporating data from related datasets is a long-standing approach in ML [7] that has enjoyed continued relevance in the age of fine-tuned pre-trained models. One setting is *meta-learning*, where a "meta-dataset" of tasks is used to learn a learning algorithm for subsequent tasks. For example, a dataset of mobile device data can be used to meta-learn an initialization of stochastic gradient descent (SGD) that yields a good personalized language model when fine-tuned on the data of a new user, a canonical problem in federated learning (FL) [35]. Meta-learning has also found important applications in areas such as few-shot learning [45], reinforcement learning [12], and fine-tuning LLMs [9].

**Contributions:** Showing meta-learning guarantees for iterative learning algorithms like SGD is challenging because of the complex relationship between the parameters of an algorithm and its performance. A key insight of my work is that we do not need to work with the exact performance metric and can instead use a good approximation to achieve meaningful results [18]. The closest analogy is that in (single-task) supervised classification we rarely optimize the (non-convex) classification loss and instead use *surrogate loss functions*. Similarly, the performance of learning algorithms can also often be approximated by a simple function of (a) task data and (b) parameters such as the initialization and step-size. For example, SGD performs provably well on a task if the distance from the initialization to the optimum is small. I developed the idea of optimizing such simple functions—i.e. using them as *surrogate algorithmic losses*—into a framework called "ARUBA" for designing meta-learning algorithms whose performance is provably better than single-task learning if the tasks are similar in a natural, algorithm-specific way [17]. For example, gradient descent with a meta-learned initialization performs well if the tasks' optima are close in average Euclidean distance. On the other hand, in the multi-armed bandit setting, we showed that the task-averaged regret will have a logarithmic dependence on the number of arms—unlike the square-root dependence that is minimax-optimal in the single-task setting—so long as a constant number of unknown arms is ever optimal on any task [23].

**Impact:** ARUBA has found significant use for meta-learning algorithm design, as it is both prescriptive and highly customizable: each algorithm-specific surrogate loss induces its own task-similarity measure and meta-learning method. For example, in algorithmic game theory, we showed how to speed up equilibrium-learning across games with nearby equilibria or similar potential functions [13]. We have also applied ARUBA to private [30], federated [26], and discontinuous [5] meta-learning, and it has been used by independent scientists to design meta-learning methods for multi-agent learning [32] and reinforcement learning [15], demonstrating its versatility and ease-of-use. In the FL application I introduced *FedEx*, a method for speeding up federated hyperparameter search with underlying provable guarantees [26]. FedEx's ability to consistently improve FL methods has been independently verified on the latest hyperparameter optimization benchmarks in FL [49], where its use improved the performance of eleven out of the twelve standard tuners assessed.

**Future work:** While my work has provided an initial understanding of how to fruitfully learn across multiple tasks, many questions remain unanswered, especially when dealing with modern models. I aim to build out a better understanding of learning across multiple distributions, focusing on a theory founded upon empirical observations, e.g. via small-scale training or by investigating public information like model checkpoints. As with my previous work on ARUBA, I will seek to develop a systematic and prescriptive theory, one that will help other researchers apply it to diverse domains such as those I study in my third research thrust.

An important first goal is understanding multi-distribution optimization of large-scale models trained on massive datasets derived from many sources. Drawing upon my background in multi-distribution optimization [17, 42, 11] and transfer learning [2, 24, 3, 44], I plan to start by understanding two key questions: (1) how optimizing over multiple distributions affects training dynamics and (2) an investigation of the role played by different specific pretraining datasets. My eventual goal is to design methods that produce better or cheaper results by using the resulting insights to improve training algorithms and data selection.

## Learning-augmented algorithms

While standard analysis of algorithms characterizes performance in the worst or average case, in domains ranging from database systems [27] to energy management [34] we can realize substantial gains by augmenting methods with *learned predictions* about their instances. This has inspired a large body of theoretical work on *algorithms with predictions* [36], focusing on quantifying improvement via prediction-dependent performance guarantees and designing methods that are *robust* to poor predictions. Such results can have a direct impact on important applications such as caching protocols, energy systems, and job scheduling.

**Contributions:** My work in this area makes two fundamental contributions: (1) addressing the crucial question of *learning* in learning-augmented algorithms and (2) extending the field's scope beyond its origins in online and graph problems. The first direction is important because, while the field had produced useful algorithms with predictions, the question of where the predictions themselves came from was not systematically addressed. As the name suggests, predictions often come from ML applied to algorithmic data, and so the question becomes whether and how they can be efficiently learned. My work makes the key observation that, just like for learning tasks, existing performance guarantees for learning-augmented algorithms can be *also* converted into surrogate losses [19]. Optimizing these approximations led to several improvements upon the theoretical state-of-the-art, including (1) improved bounds on the number of samples required for bipartite matching and several other graph algorithms with predictions, (2) the first sample complexity guarantees for learning-augmented page migration along with more general results for online ski-rental, and (3) the first *regret* bounds whatsoever for all of these problems and others, ensuring performance as good as that of the best fixed predictor on sequences of adversarially chosen instances. The latter is an important setting in many applications such as job-scheduling, where the instance distribution is non-stationary.

**Outlook:** Distilling my approach above into two steps—(1) proving an optimizable prediction-dependent performance bound and (2) applying online learning to minimize it across instances—yields a **powerful tool for showing end-to-end guarantees for algorithms with predictions**, i.e. results that address both how to use predictions and how to learn them. Because it focuses on surrogate loss functions amenable to optimization, the framework also leads to efficient and practical prediction-learning methods. Thus it has already been adopted by others in the algorithms with predictions community, e.g. for showing sample complexity and regret bounds for problems in discrete convex analysis [40, 41, 37], and has been called a "standard tool in the literature" for addressing when predictions can be efficiently learned [48]. The idea of learning performance bounds has also been an important design principle in my own work on **expanding the scope of learning-augmented algorithms** beyond online and graph algorithms. In particular, we have recently used algorithms with predictions to rigorously study the use of multi-dataset information in differential privacy (DP), where statistics are released about sensitive datasets while protecting individuals by injecting noise. For tasks such as private covariance estimation and multiple quantile release, we introduced learning-augmented algorithms that can be provably learned by optimizing well-chosen surrogate losses on external datasets [1]. We also made novel insights even in the single-task setting, such as the first method for DP quantile release that does not require knowing an interval containing the data. Our results yielded substantial reductions in the error of privately released statistics, especially at high privacy levels [16].

**Future work:** The algorithms with predictions literature has developed a powerful set of design principles for data-driven methods, including robustness to poor predictions and—from my own work—learnability of performance guarantees. I intend to continue to apply these in areas of theoretical CS—e.g. metrical task systems [10], graph algorithms [8], and mechanism design [4]—where learning-augmented algorithms have already shown great promise, and also to develop them in other important subfields of CS and data science.

The first area is **scientific computing**, where I have experience developing both classical [20] and data-driven [39, 43] tools and which is awash in multi-instance data. For example, physics simulations can require solving thousands of related linear systems, so learning can both directly accelerate standard solvers and indirectly speed up ML approaches that use them to generate training data [31]. I have already begun work on the theory of data-driven linear system solvers [21] and aim to expand this line of work to (1) incorporate more expressive models for preconditioning and smoothing and (2) use learning to help solve other important numerical tasks, e.g. matrix decomposition and approximation, eigenvalue problems, and nonlinear systems.

My work on private statistics [16, 1] opens another area of application: improving **statistical algorithms across multiple datasets**. Statistical procedures are often run on multiple related instances, which might be used to mutually improve the results; for example, inverse solvers use expensive sampling schemes on numerous related distributions. As another example, evaluating the fairness of ML systems involves hard estimation tasks because intersectional subgroups can have few samples. However, a model provider might have clients willing to share statistics to improve the estimation quality of other clients. Using both my past work and the existing statistical literature, I aim to design robust that benefit from such multi-dataset information.

A last direction is **learning-augmented inference** to accelerate generative AI. Inference on LLMs and other systems is both expensive and queried many times, and one can significantly reduce inference time [46] and possibly even the number of queries by exploiting their relatedness. I aim to develop learning-augmented approaches to reduce the cost of these processes on average while ensuring robust performance for all users.

## Efficient and automated machine learning

While large-scale neural networks have achieved incredible success in recent years, progress is distributed very unevenly. Methodological development has focused on a set of well-studied domains—vision, text, and audio—and data and compute demands have made it difficult for academic and some industry researchers to apply state-of-the-art ML. This has led to important research directions aimed at making such models more efficient and widely applicable, such as neural architecture search (NAS) and more broadly automated machine learning (AutoML). These are also fundamentally about learning from computations: how do we efficiently use expensive training runs to figure out the best architectures and hyperparameters for our task?

**Contributions:** I have a broad set of contributions in this area, but the main line of work is **AutoML for diverse tasks**, a broad agenda of allowing any scientist with data from any domain to quickly obtain good performance using ML. I have led this push together with several collaborators, starting with the ambitious goal of discovering the "right" operation for a novel domain, in the same way that convolutions are the "right" operation for vision data. To do so, we came up with XD, a search space over linear operations constrained—like convolutions—to run in time at most linearithmic in the input size [39]. This constraint on the expressivity allowed us to find XD operations that outperformed comparably sized neural partial differential equation (PDE) solvers and protein folders, which we used as examples of relevant but understudied domains. To address issues with scaling this initial approach, we introduced the more efficient DASH method [43], which restricted the operation search space even further to just the kernel sizes and dilations of basic convolutions. DASH obtained better performance than expert, hand-crafted architectures on seven out of ten tasks in NAS-Bench-360 [47], a benchmark of diverse tasks we built to address the lack of consistent evaluation in this area.

Beyond diverse tasks, I have worked in many other areas of efficient and automated ML. For efficient ML, I showed that with the right initialization and regularization, low-rank neural networks can compete with sparsity or tensor-based approaches in terms of accuracy per parameter, while being more GPU-friendly [25]; low-rank methods are now a key component of fine-tuning foundation models [14]. In AutoML, I developed the foundational methods in the challenging federated setting, introducing the aforementioned FedEx approach [26] and taking part in an industry collaboration that demonstrated the power of hyperparameter transfer in FL [28]. Lastly, in NAS, I made the first progress on understanding the breakthrough technique of *weight-sharing* [38, 33], in which the performance of multiple neural networks is (noisily) evaluated by training one large "supernet." My work advanced both the optimization [29] and statistical [22] understanding of this approach, which underlies much of NAS today, including my own work on diverse tasks.

**Impact:** Spurred by our work, AutoML for diverse tasks has become an important direction for applied research. Twenty-four teams participated in our AutoML Decathlon competition at NeurIPS 2022, which evaluated automated methods on a diverse set of ten tasks; a similar competition was incorporated into the AutoML Cup at the AutoML 2023 conference. Our method DASH turned out to be a hard baseline to beat in the Decathlon and was even incorporated by the second-place team's submission; a multi-objective variant has since been developed by a leading AutoML lab [6]. Our NAS-Bench-360 benchmark [47] has also been crucial for advancing the state-of-the-art; in my own work, we used it to show that many cheap NAS heuristics do not generalize beyond vision tasks [50] and it was the main evidence of success for ORCA [44], a breakthrough method that showed that state-of-the-art AutoML on diverse tasks could be achieved with transfer learning from readily available pretrained models like RoBERTa and Swin Transformer.

**Future work:** While there has been strong progress in building out methods, benchmarks, and a community dedicated to AutoML for diverse tasks, there remains a great deal of effort before ML is truly accessible to most fields of science and engineering. To push this forward, I plan to work with experts in diverse domains and design methods that both (1) optimize their metrics while satisfying their constraints and (2) can be generalized to large classes of learning problems. To do so, I aim to incorporate ideas from multi-objective optimization and transfer learning, with the goal of developing as simple a workflow as possible for general-purpose ML. At the same time, I also plan to work on a better understanding of **AutoML in the age of large-scale pretrained models**. While most automated approaches often assume a single dataset, in practice transfer learning is used because of its strong performance benefits. In fact, in work with my collaborators we found that tuning hyperparameters on out-of-distribution data first can often be helpful [28] and that pretrained models do not even need to be in the same modality to be useful for attaining state-of-the-art on downstream tasks [44]. My goal here is to understand how to best integrate AutoML approaches into ML workflows dominated by fine-tuning or even in-context learning, taking advantage of outside data or even prior computation while still incorporating useful techniques developed by NAS and other subfields.

# References


[1] Kareem Amin, Travis Dick, **Mikhail Khodak**, and Sergei Vassilvitskii. Private algorithms with private predictions. arXiv, 2023.

[2] Sanjeev Arora, **Mikhail Khodak**, Nikunj Saunshi, and Kiran Vodrahalli. A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and LSTMs. In *ICLR*, 2018.

[3] Sanjeev Arora, Hrishikesh Khandeparkar, **Mikhail Khodak**, Nikunj Saunshi, and Orestis Plevrakis. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, 2019.

[4] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *NeurIPS*, 2016.

[5] Maria-Florina Balcan, **Mikhail Khodak**, Dravyansh Sharma, and Ameet Talwalkar. Learning-to-learn non-convex piecewise-Lipschitz functions. In *NeurIPS*, 2021.

[6] Thomas Boot, Nicolas Cazin, Willem Sanberg, and Joaquin Vanschoren. Efficient-DASH: Automated radar neural network design across tasks and datasets. In *IEEE Intelligent Vehicles Symposium*, 2023.

[7] Rich Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[8] Justin Y. Chen, Sandeep Silwal, Ali Vakilian, and Fred Zhang. Faster fundamental graph algorithms via learned predictions. In *ICML*, 2022.

[9] Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. Meta-learning via language model in-context tuning. In *ACL*, 2022.

[10] Nicholas Christianson, Junxuan Shen, and Adam Wierman. Optimal robustness-consistency tradeoffs for learning-augmented metrical task systems. In *AISTATS*, 2023.

[11] Lucio M. Dery, Paul Michel, **Mikhail Khodak**, Graham Neubig, and Ameet Talwalkar. AANG: Automating auxiliary learning. In *ICLR*, 2023.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[13] Keegan Harris*, Ioannis Anagnostides*, Gabriele Farina, **Mikhail Khodak**, Tuomas Sandholm, and Zhiwei Steven Wu. Meta-learning in games. In *ICLR*, 2023.

[14] Edward 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 *ICLR*, 2022.

[15] Vanshaj Khattar, Yuhao Ding, Bilgehan Sel, Javad Lavaei, and Ming Jin. A CMDP-within-online framework for meta-safe reinforcement learning. In *ICLR*, 2023.

[16] **Mikhail Khodak**, Kareem Amin, Travis Dick, and Sergei Vassilvitskii. Learning-augmented private algorithms for multiple quantile release. In *ICML*, 2023.

[17] **Mikhail Khodak**, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. In *NeurIPS*, 2019.

[18] **Mikhail Khodak**, Maria-Florina Balcan, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *ICML*, 2019.

[19] **Mikhail Khodak**, Maria-Florina Balcan, Ameet Talwalkar, and Sergei Vassilvitskii. Learning predictions for algorithms with predictions. In *NeurIPS*, 2022.

[20] **Mikhail Khodak**, Richard L. Berger, Thomas Chapman, and Jeffrey A. F. Hittinger. Development and application of a multi-fluid simulation code for modeling interpenetrating plasmas. In *APS Division of Plasma Physics Meeting*, 2015.

[21] **Mikhail Khodak**, Edmond Chow, Maria-Florina Balcan, and Ameet Talwalkar. Learning to relax: Setting solver parameters across a sequence of linear system instances. arXiv, 2023.

[22] **Mikhail Khodak**, Liam Li, Nicholas Roberts, Maria-Florina Balcan, and Ameet Talwalkar. A simple setting for understanding neural architecture search with weight-sharing. AutoML Workshop, 2020.

[23] **Mikhail Khodak***, Ilya Osadchiy*, Keegan Harris, Maria-Florina Balcan, Kfir Y. Levy, Ron Meir, and Zhiwei Steven Wu. Meta-learning adversarial bandit algorithms. In *NeurIPS*, 2023.

[24] **Mikhail Khodak***, Nikunj Saunshi*, Yingyu Liang, Tengyu Ma, Brandon Stewart, and Sanjeev Arora. A la carte embedding: Cheap but effective induction of semantic feature vectors. In *ACL*, 2018.

[25] **Mikhail Khodak**, Neil Tenenholtz, Lester Mackey, and Nicolò Fusi. Initialization and regularization of factorized neural layers. In *ICLR*, 2021.

[26] **Mikhail Khodak**, Renbo Tu, Tian Li, Liam Li, Maria-Florina Balcan, Virginia Smith, and Ameet Talwalkar. Federated hyperparameter tuning: Challenges, baselines, and connections to weight-sharing.

In *NeurIPS*, 2021.

[27] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *SIGMOD*, 2018.

[28] Kevin Kuo, Pratiksha Thaker, **Mikhail Khodak**, John Nguyen, Daniel Jiang, Ameet Talwalkar, and Virginia Smith. On noisy evaluation in federated hyperparameter tuning. In *MLSys*, 2023.

[29] Liam Li*, **Mikhail Khodak***, Maria-Florina Balcan, and Ameet Talwalkar. Geometry-aware gradient algorithms for neural architecture search. In *ICLR*, 2021.

[30] Jeffrey Li, **Mikhail Khodak**, Sebastian Caldas, and Ameet Talwalkar. Differentially private meta-learning. In *ICLR*, 2020.

[31] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *ICLR*, 2021.

[32] Sen Lin, Mehmet Dedeoglu, and Junshan Zhang. Accelerating distributed online meta-learning via multi-agent collaboration under limited communication. In *MobiHoc*, 2021.

[33] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *ICLR*, 2019.

[34] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. Renewable and cooling aware workload management for sustainable data centers. In *SIGMETRICS*, 2012.

[35] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[36] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, Cambridge, UK, 2021.

[37] Taihei Oki and Shinsaku Sakaue. Faster discrete convex function minimization with predictions: The M-convex case. In *NeurIPS*, 2023.

[38] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *ICML*, 2018.

[39] Nicholas Roberts*, **Mikhail Khodak***, Tri Dao, Liam Li, Chris Ré, and Ameet Talwalkar. Rethinking neural operations for diverse tasks. In *NeurIPS*, 2021.

[40] Shinsaku Sakaue and Taihei Oki. Discrete-convex-analysis-based framework for warm-starting algorithms with predictions. In *NeurIPS*, 2022.

[41] Shinsaku Sakaue and Taihei Oki. Rethinking warm-starts with predictions: Learning predictions close to sets of optimal solutions for faster $L$-/$L^\natural$-convex function minimization. In *ICML*, 2023.

[42] Nikunj Saunshi, Yi Zhang, **Mikhail Khodak**, and Sanjeev Arora. A sample complexity separation between non-convex and convex meta-learning. In *ICML*, 2020.

[43] Junhong Shen*, **Mikhail Khodak***, and Ameet Talwalkar. Efficient architecture search for diverse tasks. In *NeurIPS*, 2022.

[44] Junhong Shen, Liam Li, Lucio Dery, Corey Staten, **Mikhail Khodak**, Graham Neubig, and Ameet Talwalkar. Cross-modal fine-tuning: Align then refine. In *ICML*, 2023.

[45] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.

[46] Benjamin Spector and Chris Ré. Accelerating LLM inference with staged speculative decoding. arXiv, 2023.

[47] Renbo Tu*, Nicholas Roberts*, **Mikhail Khodak**, Junhong Shen, Frederic Sala, and Ameet Talwalkar. NAS-Bench-360: Benchmarking diverse tasks for neural architecture search. In *NeurIPS: Datasets and Benchmarks Track*, 2022.

[48] Jan van den Brand, Sebastian Forster, Yasamin Nazari, and Adam Polak. On dynamic graph algorithms with predictions. In *SODA*, 2024.

[49] Zhen Wang, Weirui Kuang, Ce Zhang, Bolin Ding, and Yaliang Li. FedHPO-Bench: A benchmark suite for federated hyperparameter optimization. In *ICML*, 2023.

[50] Colin White, **Mikhail Khodak**, Renbo Tu, Shital Shah, Sébastien Bubeck, and Debadeepta Dey. A deeper look at zero-cost proxies for lightweight NAS. In *ICLR: Blog Track*, 2022.