MemMAP: Compact and Generalizable Meta-LSTM Models for Memory Access Prediction

Ajitesh Srivastava[1], *Pengmiao Zhang*[1], *Ta-Yang Wang*[1], *César De Rose*[2], *Rajgopal Kannan*[3], *Viktor K. Prasanna*[1]

[1]University of Southern California, USA
[2]Pontifical Catholic University of Rio Grande do Sul, Brazil
[3]Army Research Lab – West, USA

ML-based Prefetcher: Roadblocks

- ML based methods have achieved high accuracy in many domains including sequence prediction, but …
  - Large number of parameters => too much computation => slow prediction
  - Large number of parameters => storage requirement
  - Evolving patterns => need to quickly retrain models

- Specifically, for LSTM based access prediction [ICML 2018], existing work requires order of 10^6 parameters

- Practical ML prefetcher would require
  - training a **small model** with large traces that is **highly accurate** and **fast predictions**
  - **retraining** the model online **on-demand** to learn application specific models - require fast learning with small amount of data

Prior Work

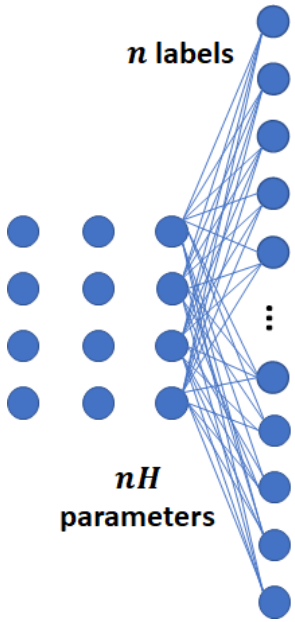
Srivastava, Ajitesh; Lazaris, Angelos; Brooks, Benjamin; Kannan, Rajgopal; Prasanna, Viktor K., **Predicting memory accesses: the road to compact ML-driven prefetcher**, Proceedings of the International Symposium on Memory Systems - MEMSYS '19, pp. 461-470, 2019.

- Revisiting the first question: How to address the ML roadblocks – accurate, fast, memory-efficient models

- Compact LSTM-based memory access prediction
  - *Accurate*: Predict the next access with high accuracy
  - *Highly compressed*: compression technique that results in a compact design with number of parameters reduced by a theoretical factor of $\frac{n}{\log n}$

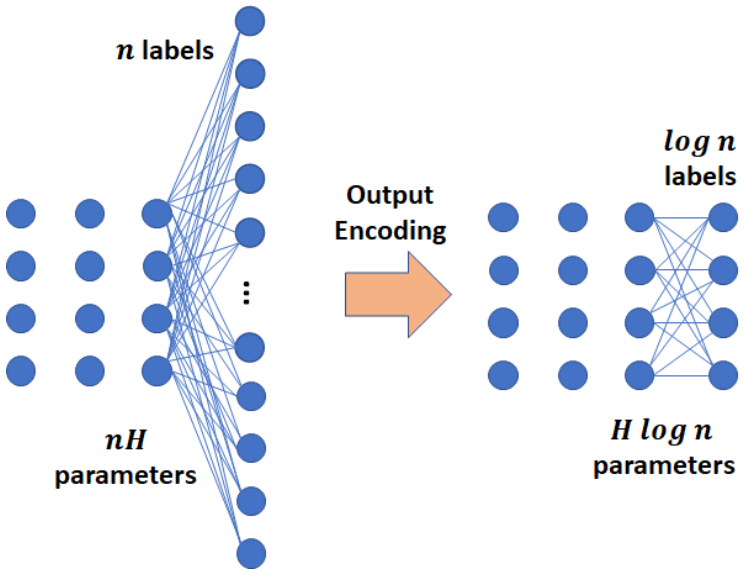
LSTM Size Challenge

- Number of labels (deltas) very large
- Traditional sequence prediction models have a large output layer: 1 for each of the n labels ($\sim 65K$ deltas)
- Number of computations dominated by output layer
- Hidden layer is much smaller ($H < 100$)



Approach: Compression Technique

- Label Encoding: Instead of delta, predict binary representation of delta
- Output layer reduced from n to $\log n$: Theoretical compression factor $O\left(\frac{n}{\log n}\right)$



Demonstrated >100x compression
(24K parameters)
Negligible loss in accuracy

- Challenge: All bits must be correct for correct prediction of delta

Scalability Issue!

- Having one model per application is not a scalable approach
- Can m models predict A applications ($m \ll A$)?
- Different applications have different patterns – Retraining is necessary
- How to predict memory accesses for many applications with few models
  - Accurate: High accuracy
  - Adaptable: Quickly adapt to changing behavior
  - Generalizable: Generalize to applications not seen before

Meta-Learning Approach (MAML)

- Learn a meta-model that can adapt to specific application trace with small number of batches for retraining

Algorithm 1 Doubly Compressed LSTM with MAML

function MAML-DCLSTM(S)
    S: A set of applications
    Initialize θ and initial parameters α, β
    for $k \leftarrow 1$ to N_{epoch} **do**
        Sample batch of applications $A_i \sim S$
        for all A_i **do**
            Sample a batch D of m accesses from A_i
            Evaluate $\nabla_\theta L_{A_i}(f_\theta)$ using D, where L_{A_i} is the binary cross-entropy loss
            Compute the adapted parameters: $\theta_i' = \theta - \alpha \nabla_\theta L_{A_i}(f_\theta)$
            Sample accesses D_i' from A_i for the meta-update
        Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{A_i \sim S} L_{A_i}(f_{\theta_i'})$ using each D_i' and L_{A_i}
        return θ

Sample a trace

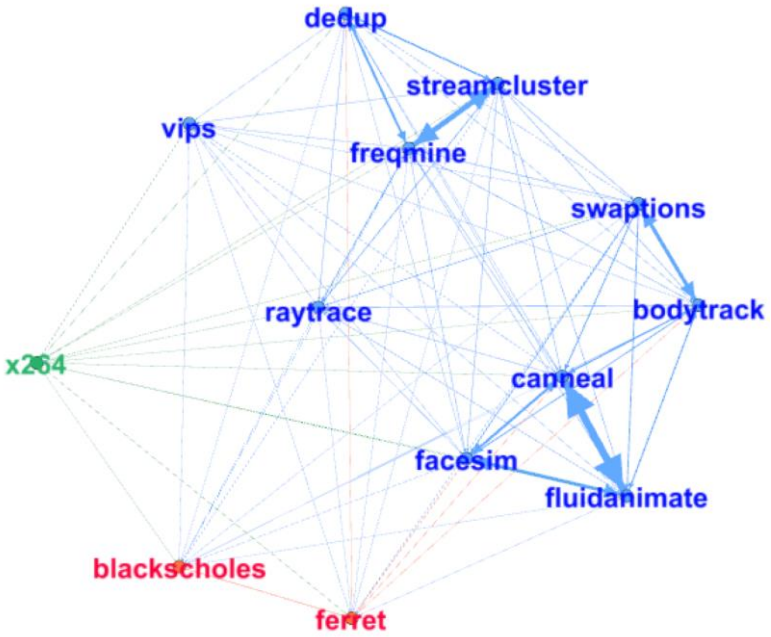
Sample a batch

Compute adapted parameters

Compute meta-model parameters

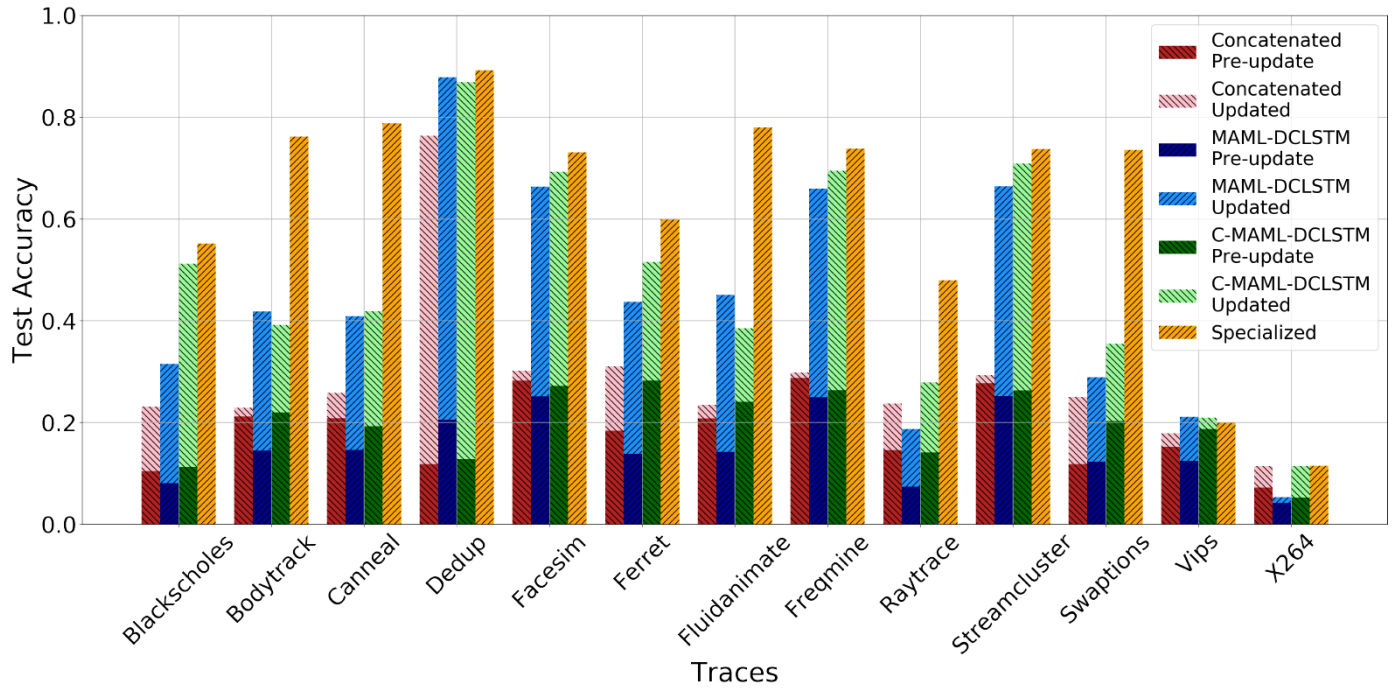
Clustered-Meta-Learning Approach (C-MAML)

- Highly dissimilar applications may result in slow adaptability
- We train one MAML for each “cluster”



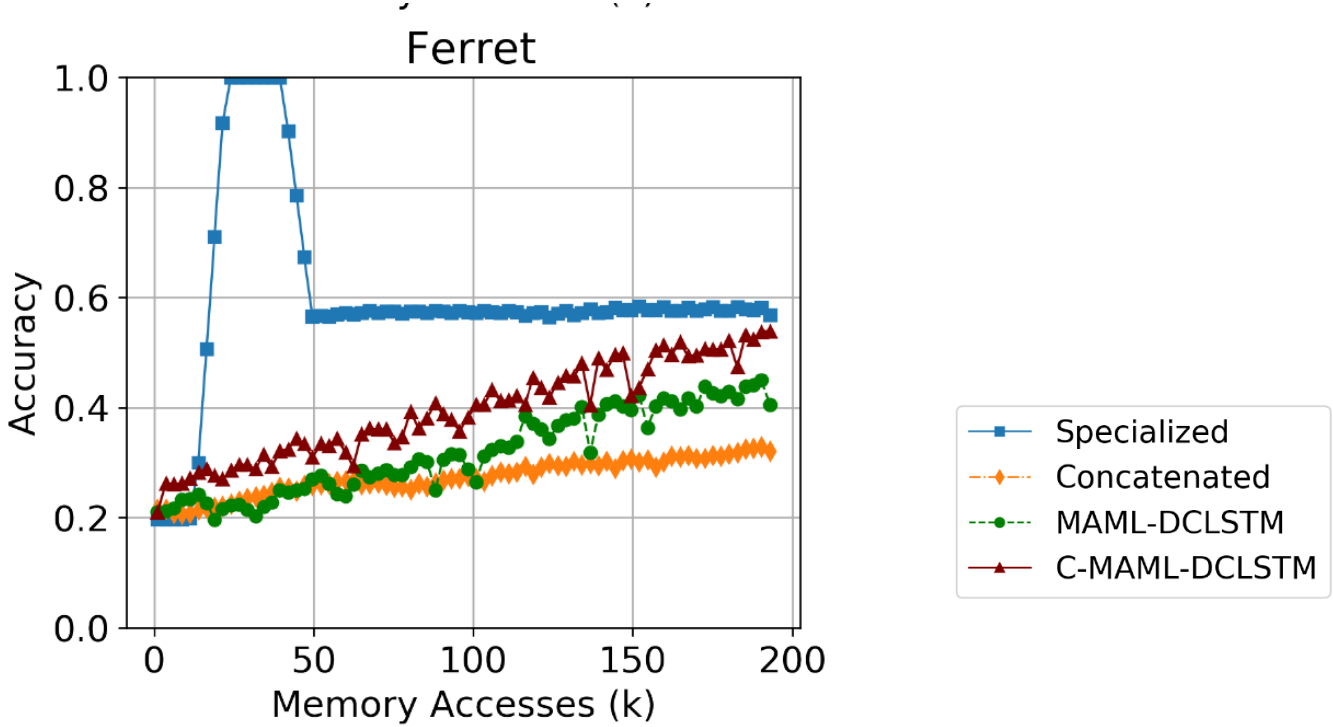
Accuracy

- In majority of the traces, C-MAML comes close to the accuracy of specialized model (from MEMSYS 2019)



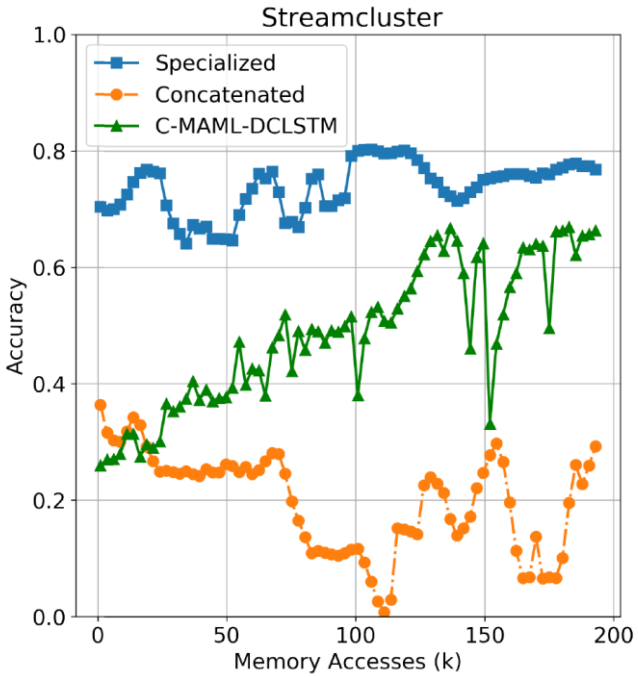
Adaptability

- C-MAML has a faster increase in accuracy as more accesses are seen



Generalizability

- Predicting for an unseen application with the meta-model



Conclusions

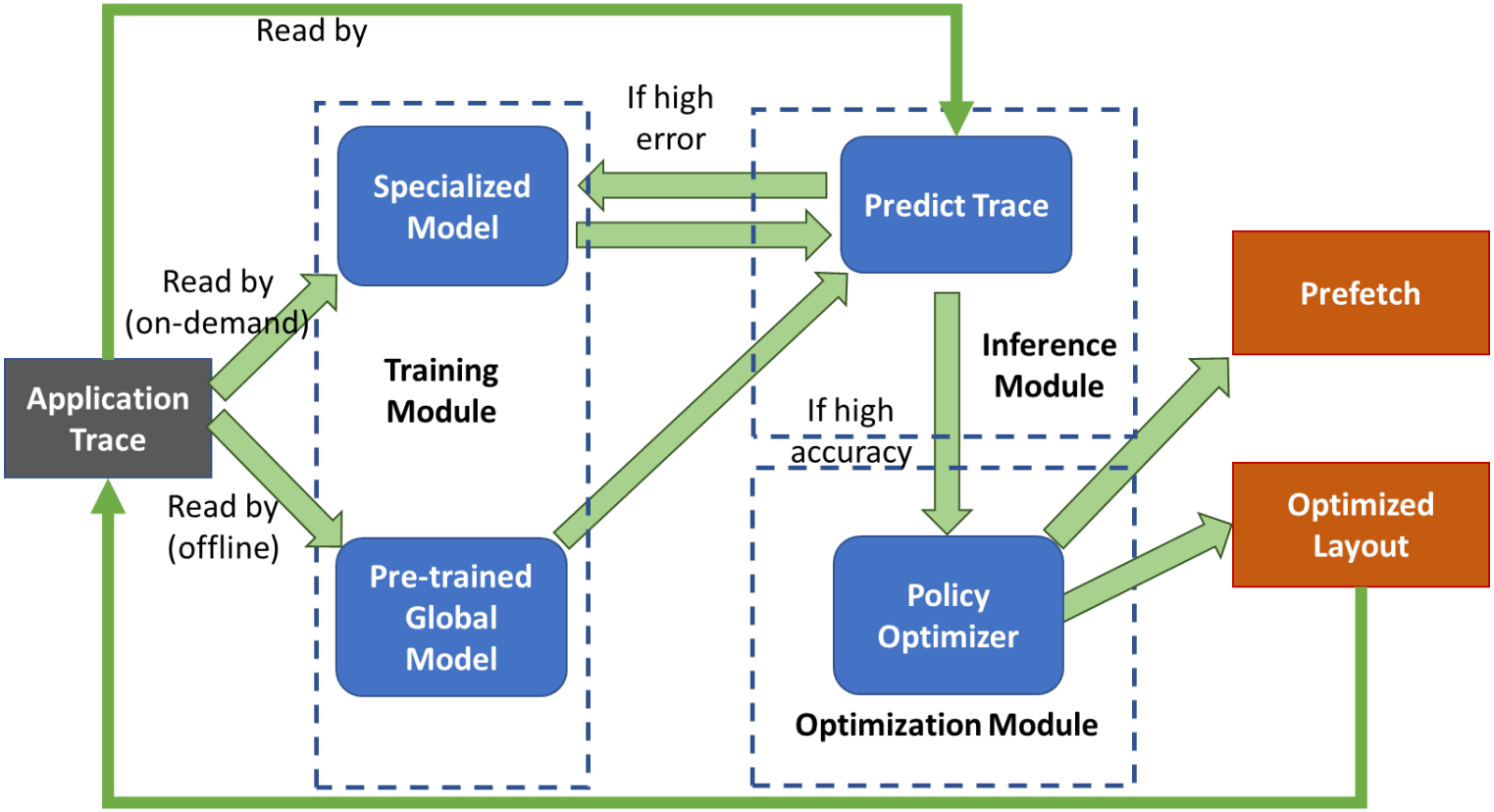
- ML models to support prefetching must be made accurate and compact
- We proposed a compact C-MAML-LSTM
  - > 100x compression in model size
  - 3 models for 13 traces
  - Can adapt to unseen patterns quickly using meta-learning
  - Can generalize to unseen applications
- Opens major opportunities to improve prefetching

The End

Back Up Slides

Our Approach: The Big Picture

- Using pre-trained model and fast pipelined retraining for prefetching policy and memory layout optimization



Target Architecture