ICLR

LLM Augmented LLMs:

Expanding Capabilities Through Composition

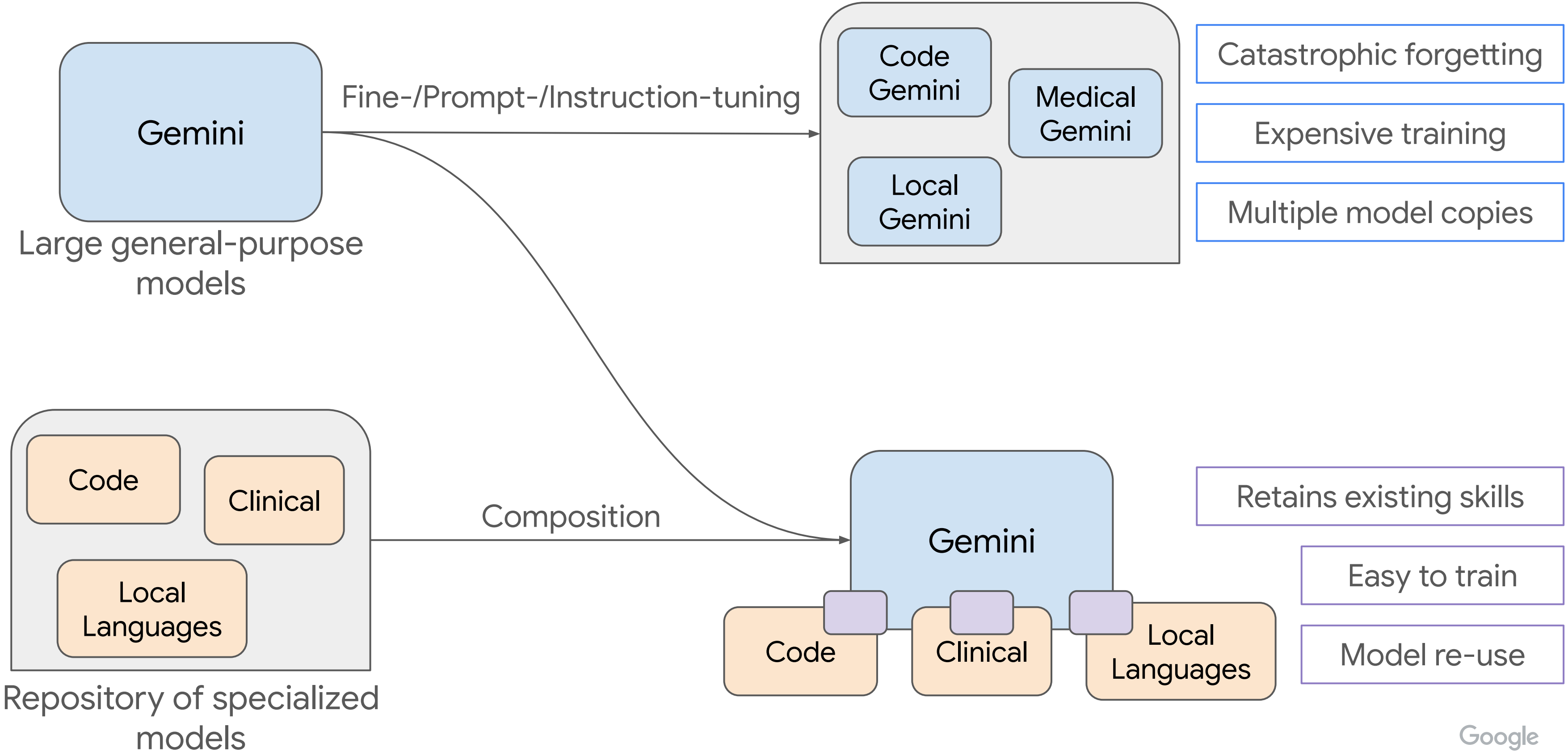
Rachit Bansal, Bidisha Samanta

Joint work with:
Sid Dalmia*, Nitish Gupta, Partha Talukdar, Prateek Jain, Abhishek Bapna

Google Research India
*Google DeepMind

Google Research
Google DeepMind

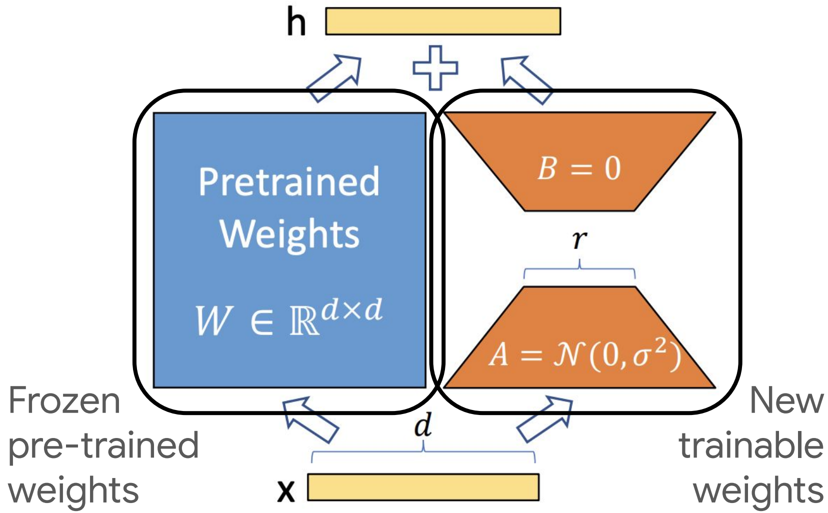
Motivation



Google

Related Work

Parameter Efficient Fine-Tuning



From "LoRA: Low-Rank Adaptation of Large Language Models"
Hu et al., 2021

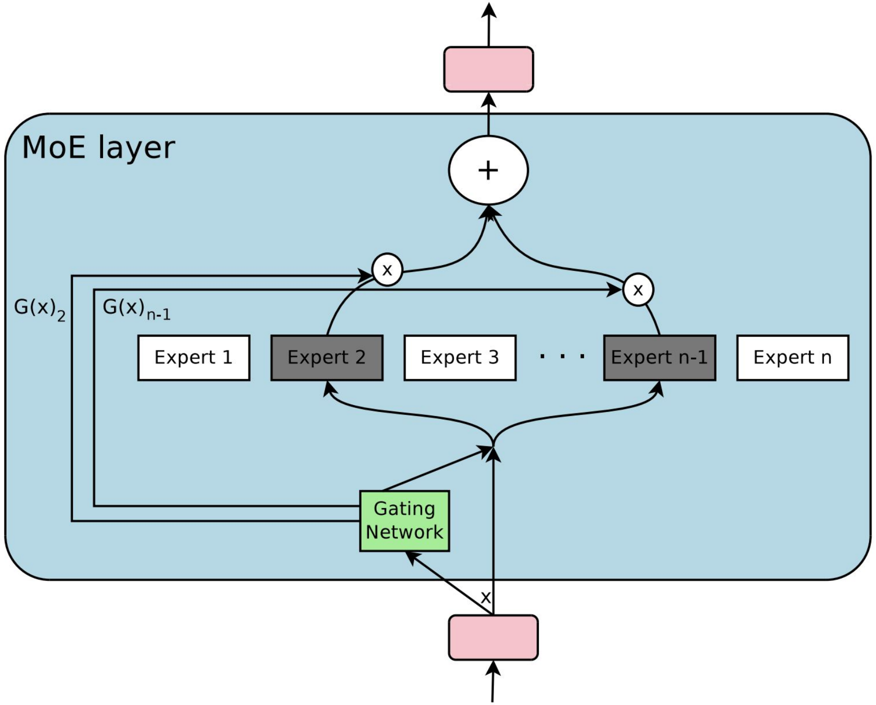
Introduces a small set of parameters to learn task-specific knowledge, keeping the base pre-trained model unchanged.

Reduced efficacy for large and varying domain knowledge, for instance when input tokenization needs to be differentiated.

New set of added parameters and training for each base model—hindering effective collaborative development of models.

Related Work

Mixture of Experts



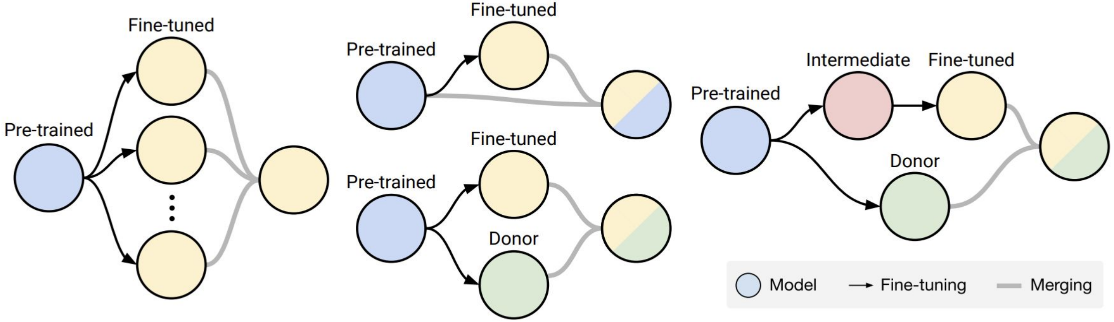
From "Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer"; Shazeer et al., 2017

Routing a given input to certain sparsified model parameters at each layer for efficient distributed training and inference.

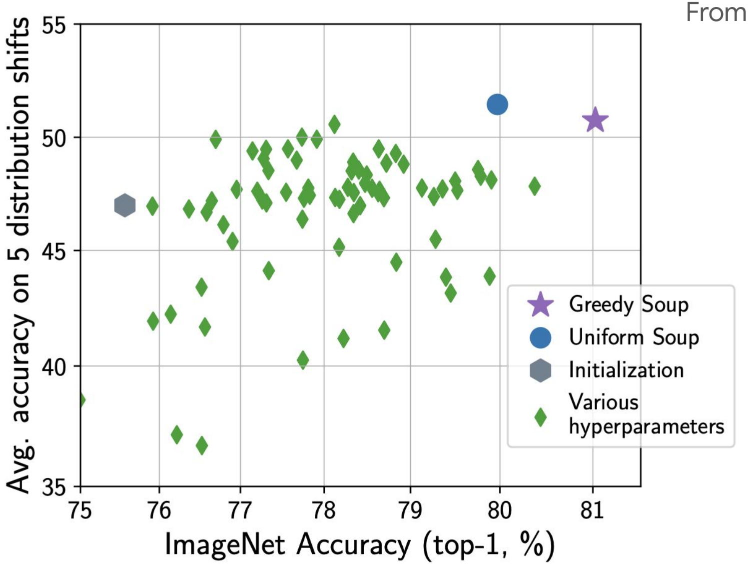
Parameters are divided into "experts" pre-hoc, i.e., specialized models cannot be trained independently and introduced post-training.

Distinction of capabilities across "experts" is unclear—all parameters need to be available in memory at test time.

Related Work
Model Merging



From “Merging Models with Fisher-Weighted Averaging”; Matenda et al., 2022



From “Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time”; Wortsman et al., 2022

Merging parameters from different models to obtain an enhanced new model.

Forgetting of existing capabilities is likely since existing models are not restored.

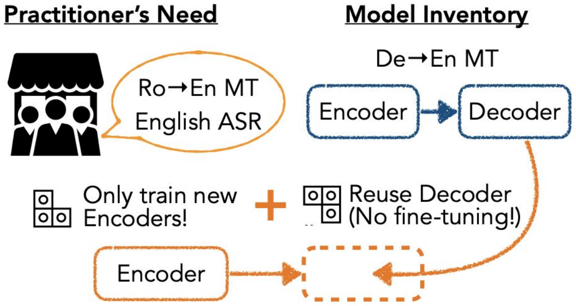
Unlikely to result into a model that enables new tasks as a composition of original models.

Google

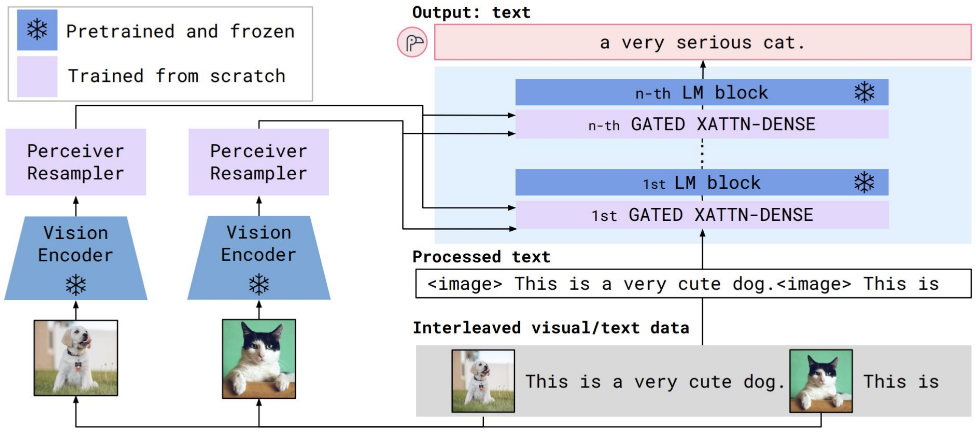
Related Work

Multi-modal Compositionality

Based on a similar motivation as our work, re-uses existing encoder and decoder models for new tasks and capabilities.



From "LegoNN: Building Modular Encoder-Decoder Models"; Dalmia et al., 2022



From "Flamingo: a Visual Language Model for Few-Shot Learning"; Alayrac et al., 2022

Composes independent encoder and decoder models together—not established for composing decoder-only language models.

Inputs are divided and distributed across different models, hence assumes a clear distinction of capabilities at the input level.

Prior Work

Axes of Interest

	Fine-tuning	LoRA	MoE	Tool-use and Routing	**CALM**
Modularity		✅	✅		✅
Training Efficiency *easy and cheap to train*		✅			✅
Model re-use *Independently trained pre-existing model reuse*				✅	✅
Avoids Forgetting *retain existing skills*				✅	✅

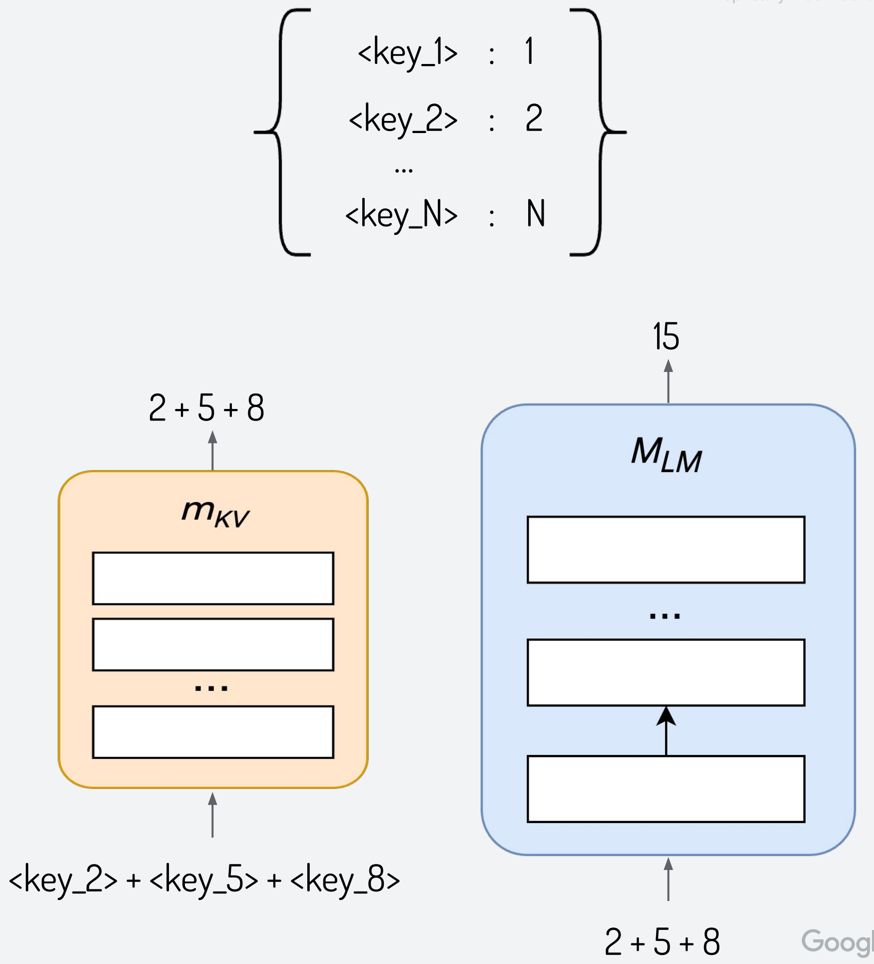
A Synthetic Example

Created a set of key-value pairs, mapping strings to integers.

Can we borrow and compose relevant capabilities from ***two models*** to perform arithmetic over these keys?

A small PaLM that has memorized the key-value pairs

A larger PaLM that has arithmetic reasoning built into it



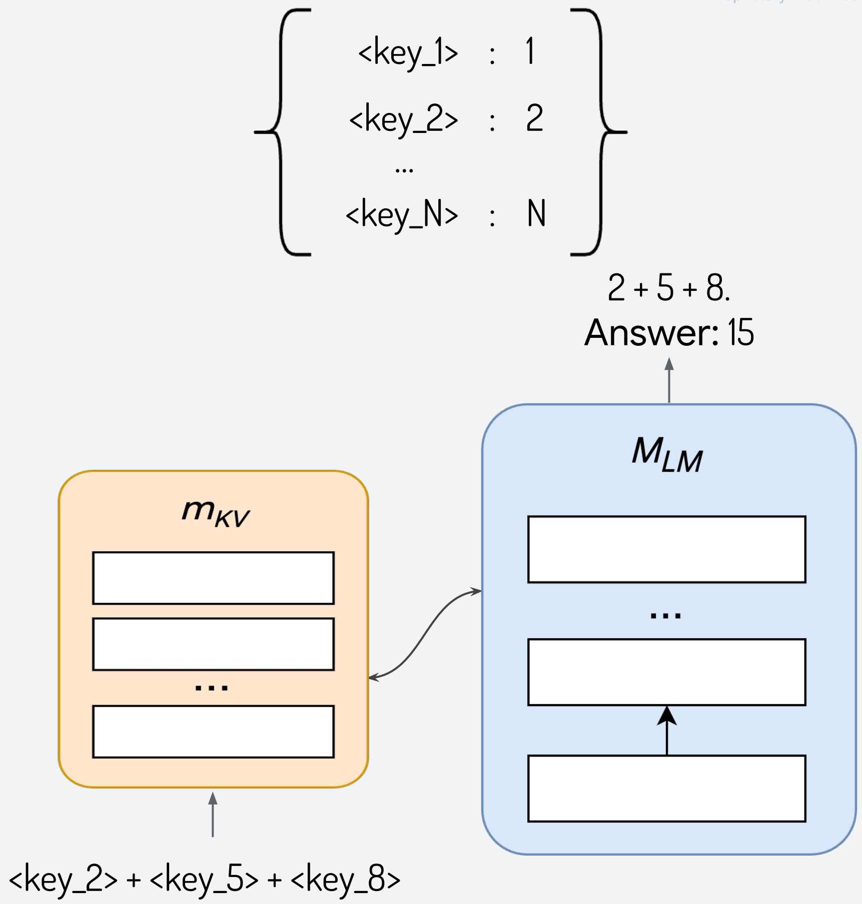
A Synthetic Example

Created a set of key-value pairs, mapping strings to integers.

Can we borrow and compose relevant capabilities from ***two models*** to perform arithmetic over these keys?

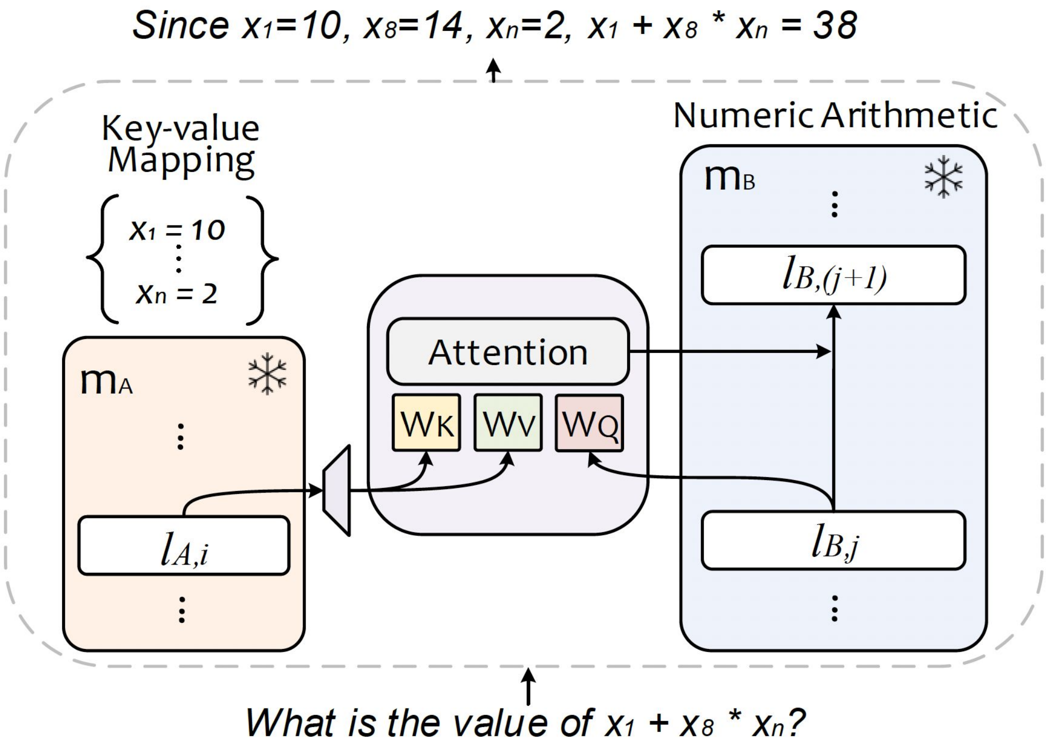
A small PaLM that has memorized the key-value pairs

A larger PaLM that has arithmetic reasoning built into it



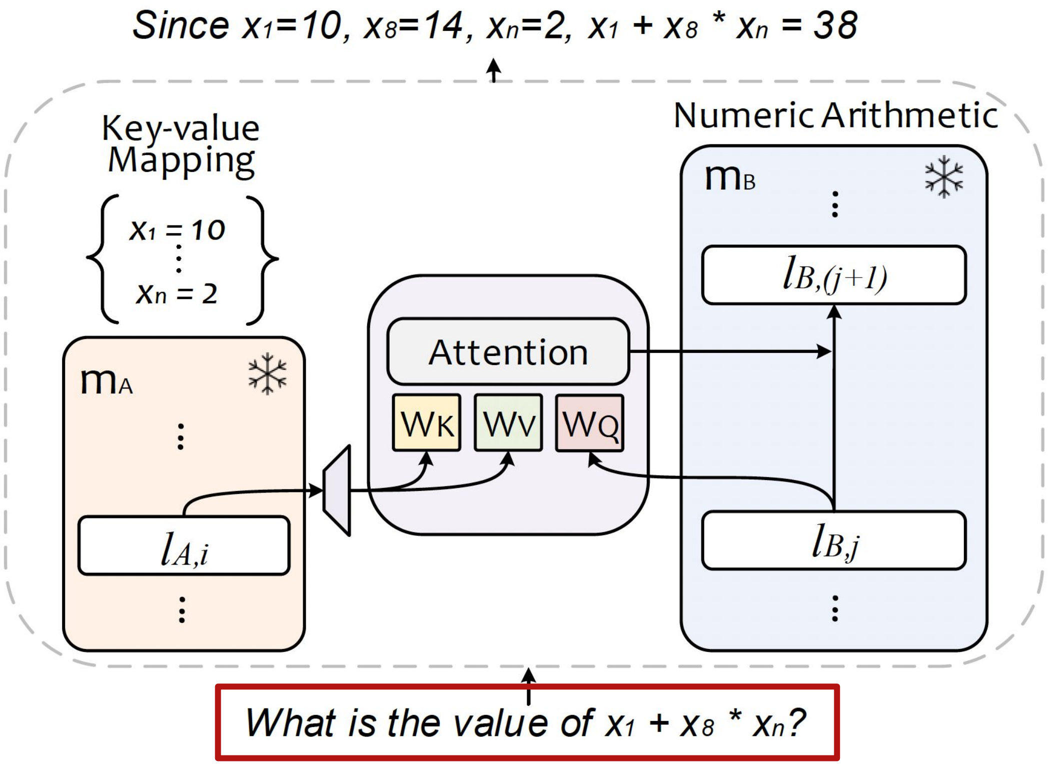
Methodology

Learn to ***align*** and ***cross-attend*** representations from the two models.



Methodology

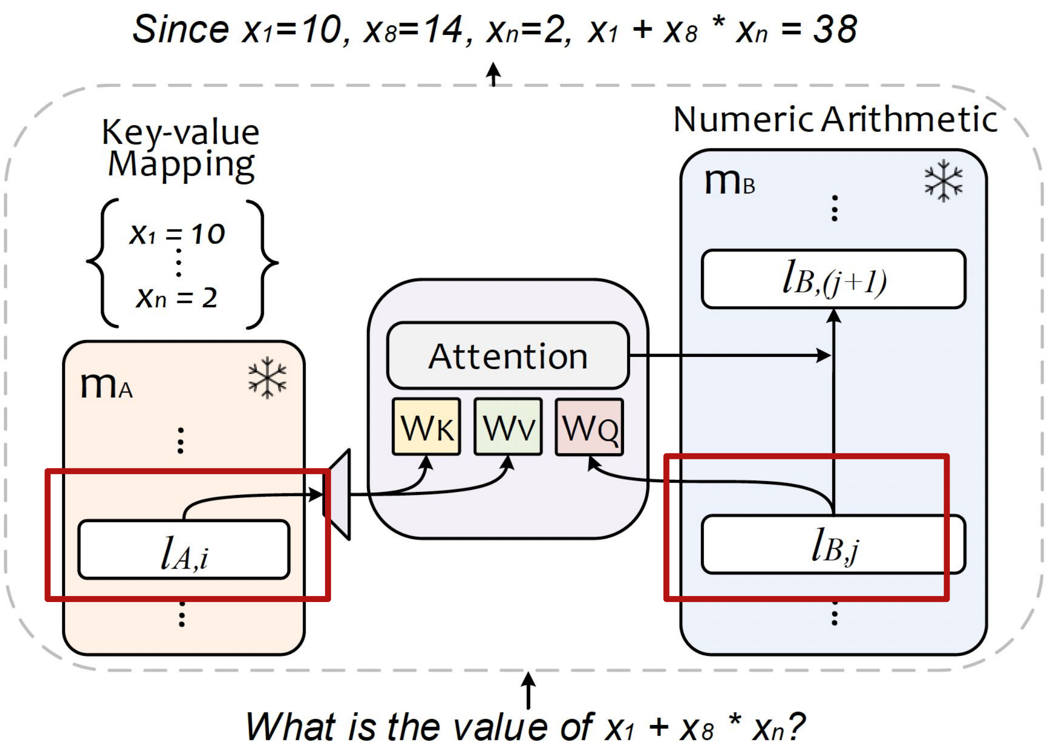
Inputs are passed to both models.



Methodology

Inputs are passed to both models.

Obtain layer representations from selected layers in m_A and m_B.



Methodology

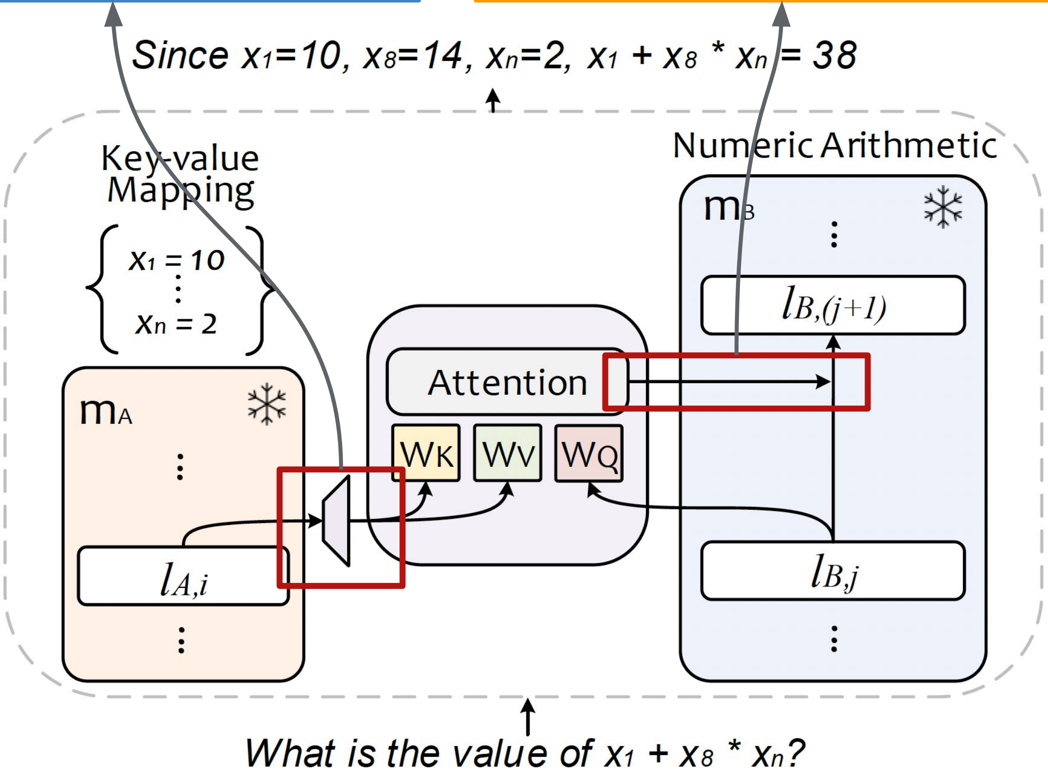
A linear projection mapping layer representations from mA to mB.

Cross-attended representations added as residual connections.

Inputs are passed to both models.

Obtain layer representations from selected layers in m_A and m_B.

Learn to ***align*** and ***cross-attend*** representations from the two models.



Methodology

Inputs are passed to both models.

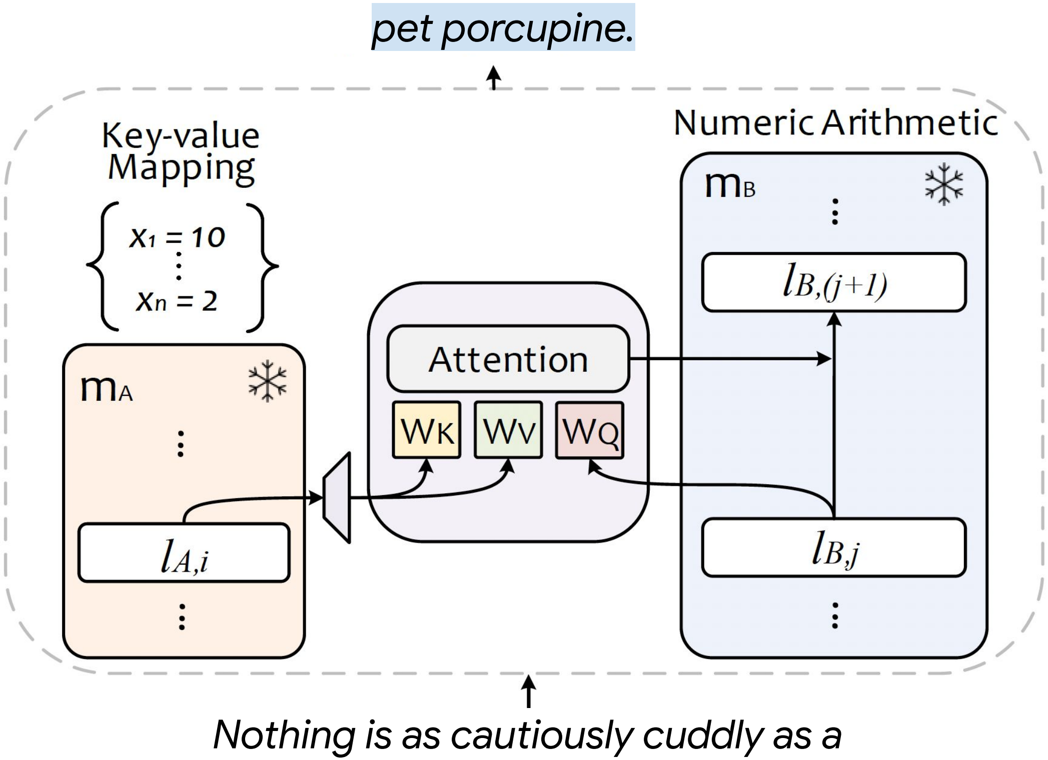
Obtain layer representations from selected layers in m_A and m_B.

Learn to ***align*** and ***cross-attend*** representations from the two models.

Training over:

PaLM Training Sub-set

t_A



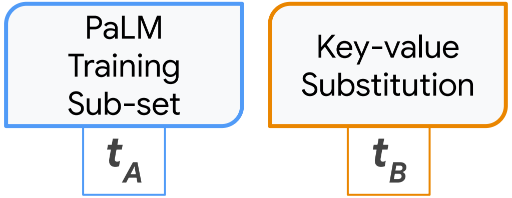
Methodology

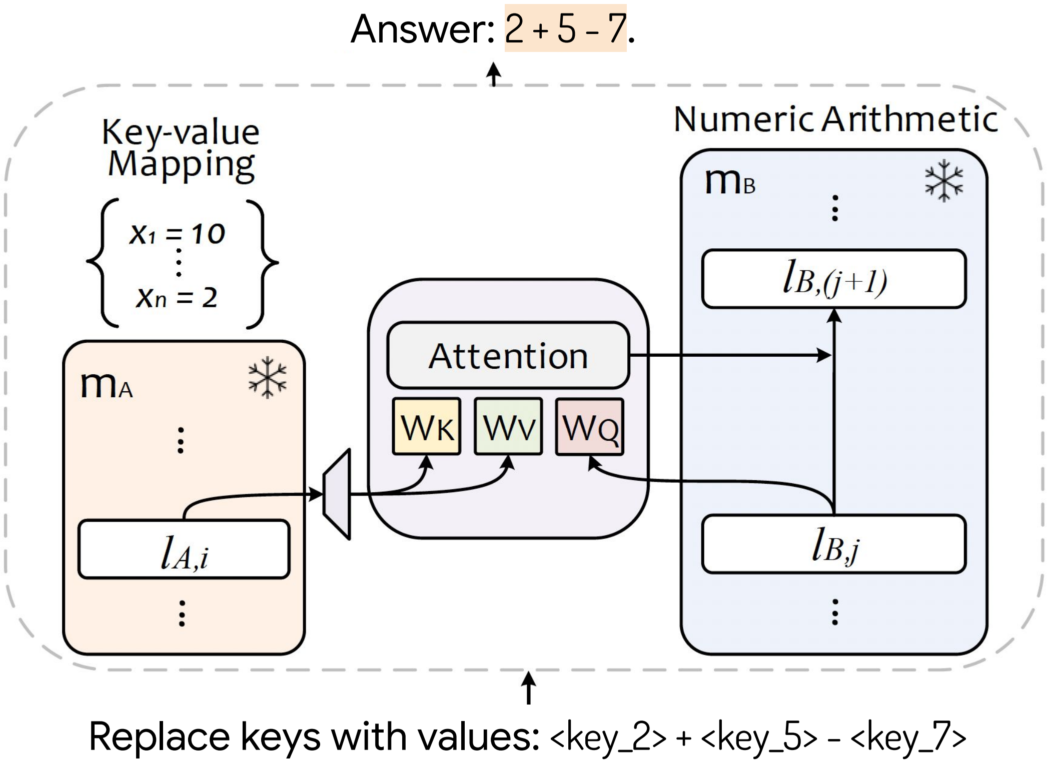
Inputs are passed to both models.

Obtain layer representations from selected layers in m_A and m_B.

Learn to ***align*** and ***cross-attend*** representations from the two models.

Training over:





Methodology

Inputs are passed to both models.

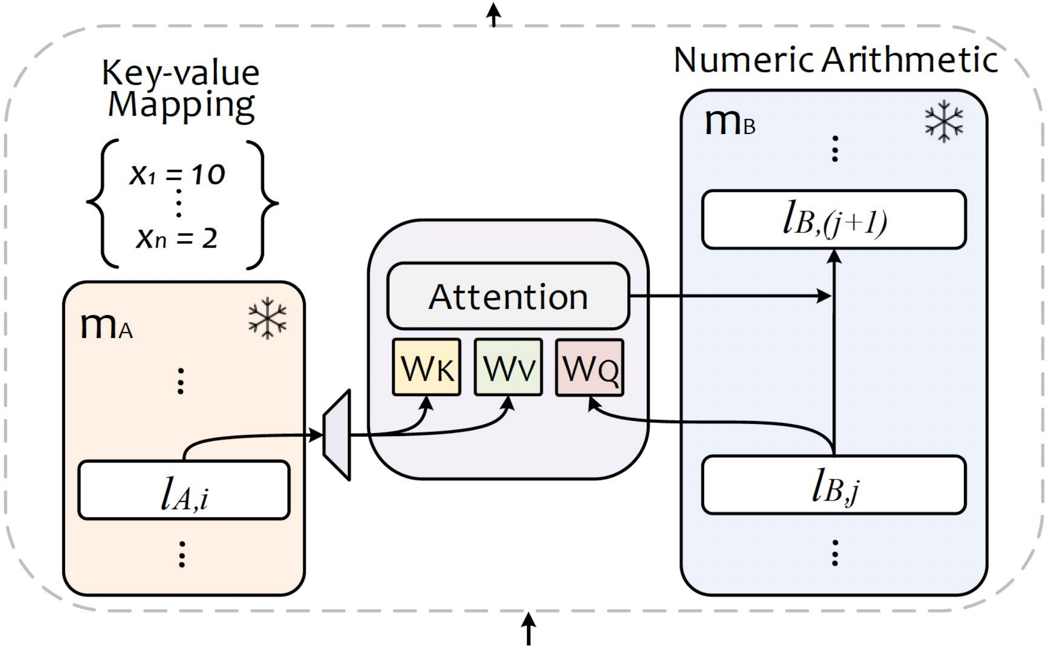
Obtain layer representations from selected layers in m_A and m_B.

Learn to ***align*** and ***cross-attend*** representations from the two models.

Training over:

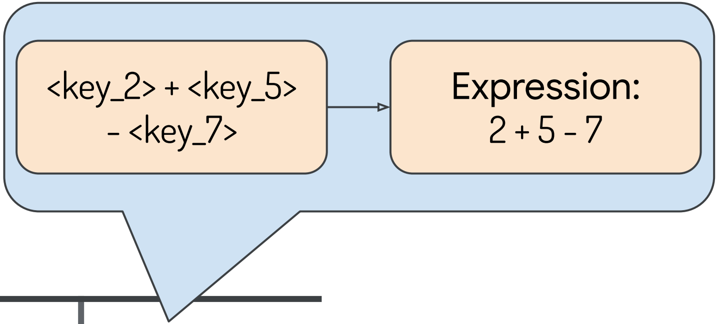
PaLM Training Sub-set	Key-value Substitution	Held-in Key-value arithmetic
t_A	t_B	$t_{A \cap B}$

CoT: <key_1> - <key_4> + <key_8> = 1 - 4 + 8.
Solving: 1 - 4 = -3; -3 + 8 = 5.
Answer: 5



What is <key_1> - <key_4> added to <key_8>?

Results



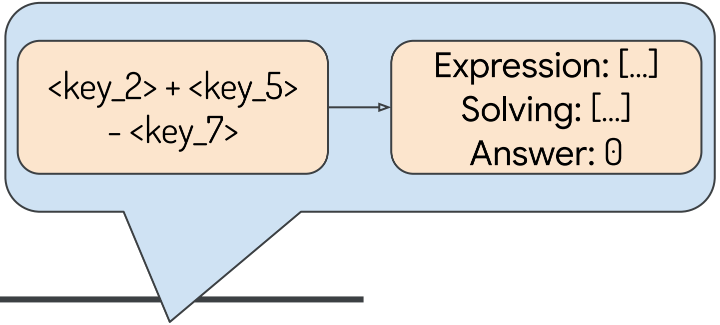
	Key-Value
Model-1	98.10

Results

2 + 5 - 7 → Solving: 2 + 5 = 7; 7 - 7 = 0. Answer: 0

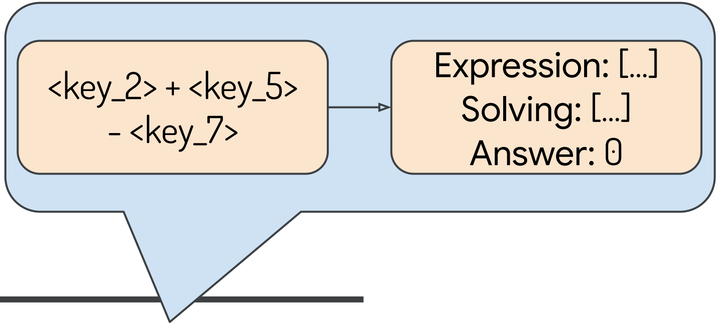
	Key-Value	Numeric Arithmetic
Model-1	98.10	4.20
Model-2	–	73.70

Results



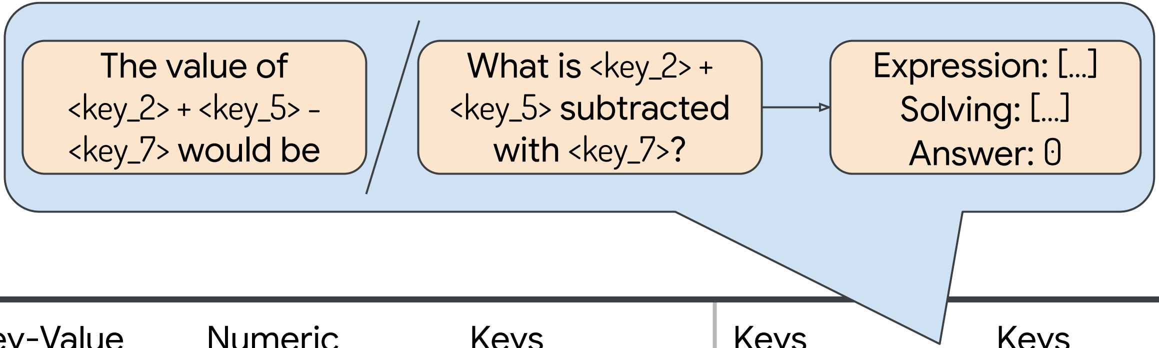
	Key-Value	Numeric Arithmetic	Keys Arithmetic
Model-1	98.10	4.20	–
Model-2	–	73.70	–
Composed	92.90	72.00	84.30

Results



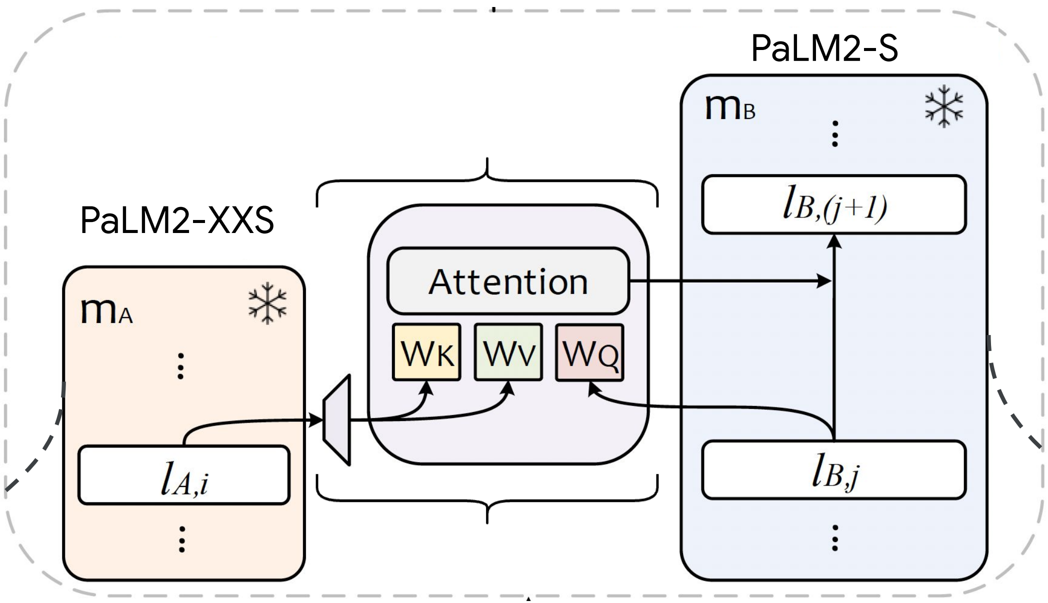
	Key-Value	Numeric Arithmetic	Keys Arithmetic
Model-1	98.10	4.20	–
Model-2	–	73.70	–
Composed	92.90	72.00	84.30
Composed w/ Random M1	25.60	70.90	20.25
Cascade* (M1 ⟶ M2)	98.10	73.70	72.30

Results



	Key-Value	Numeric Arithmetic	Keys Arithmetic	Keys Arithmetic (OOD)	Keys Arithmetic (OOD-Hard)
Model-1	98.10	4.20	–	–	–
Model-2	–	73.70	–	–	–
Composed	92.90	72.00	84.30	66.40	30.30
Composed w/ Random M1	25.60	70.90	20.25	14.95	7.75
Cascade* (M1 ⟶ M2)	98.10	73.70	72.30	51.00	27.55

Key Results: Multi-linguality



PaLM2-NTL

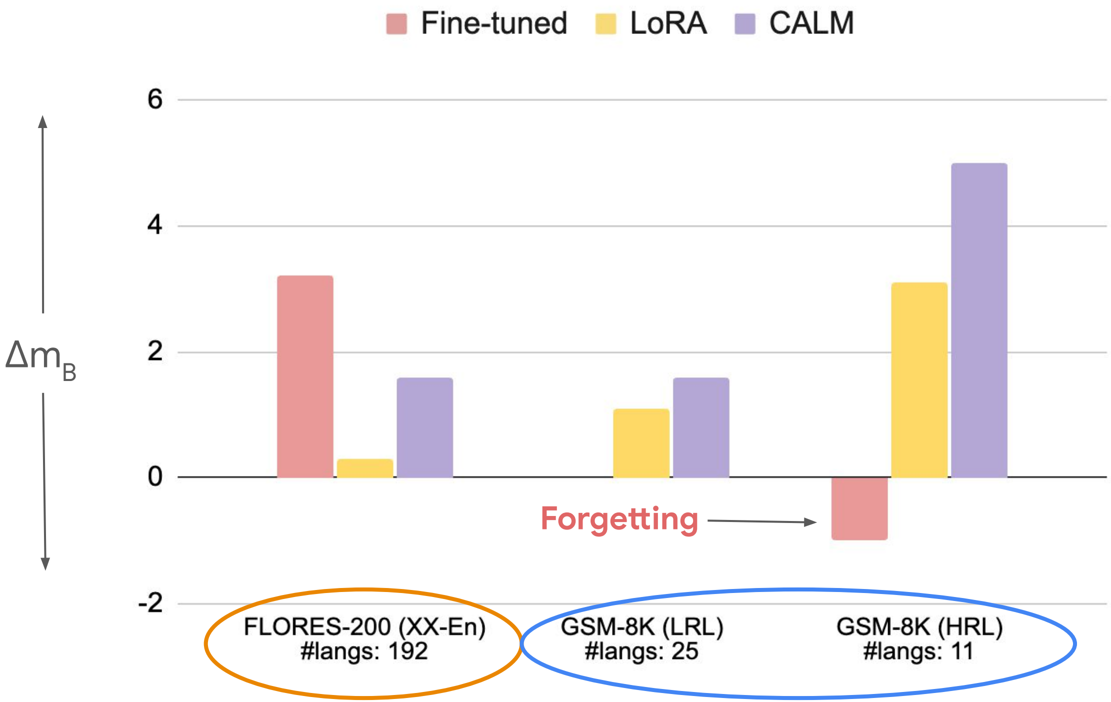
Re-using a model trained for the ***Next Thousand Languages (NTL)*** effort for tail- and torso- languages.

trained on a mixture of the NTL corpus and the PaLM2 training corpora

PaLM2-S

A general purpose language model **primarily trained on English data.** Encomposes reasoning and logic.

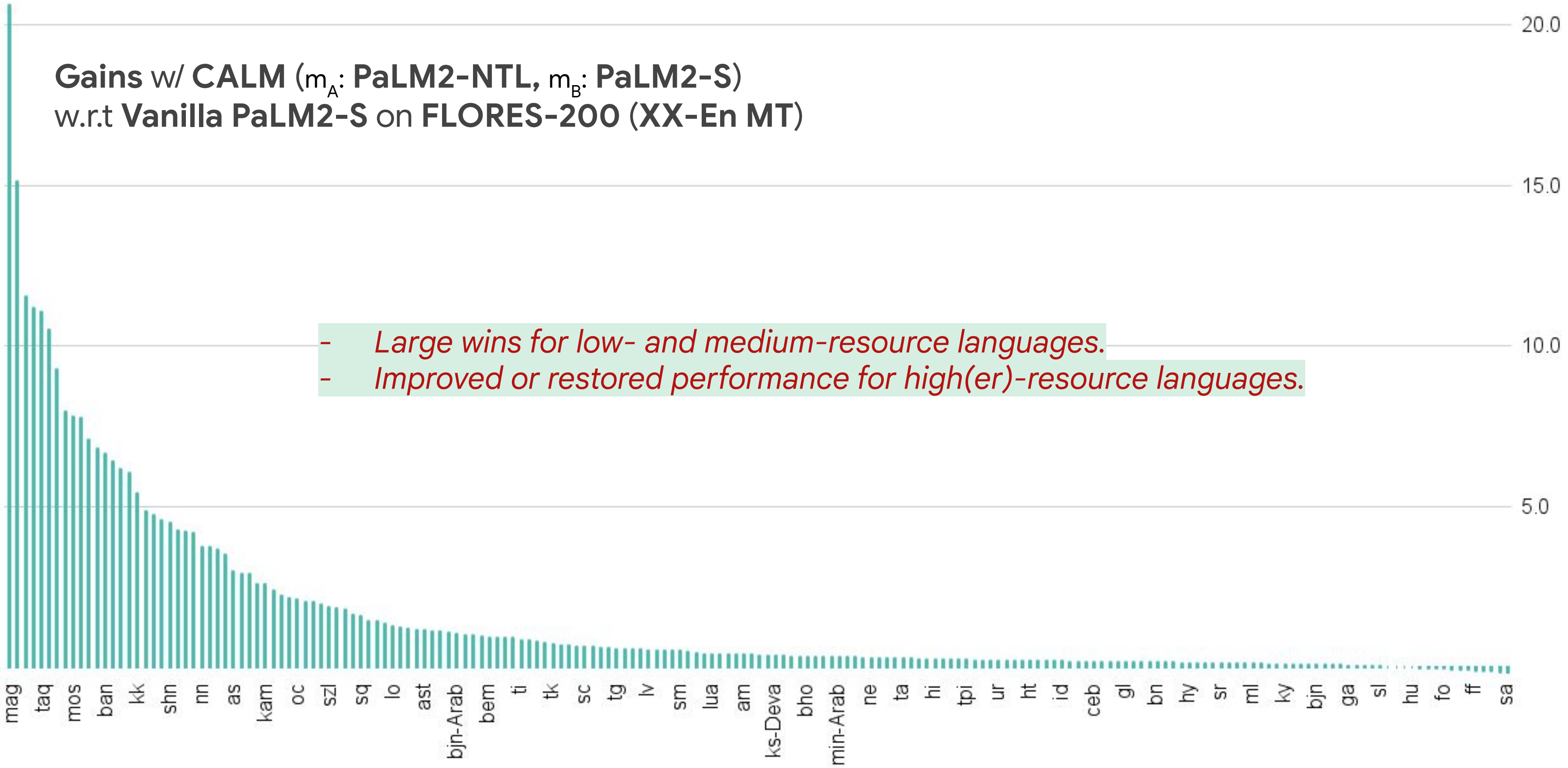
Key Results: Multi-linguality



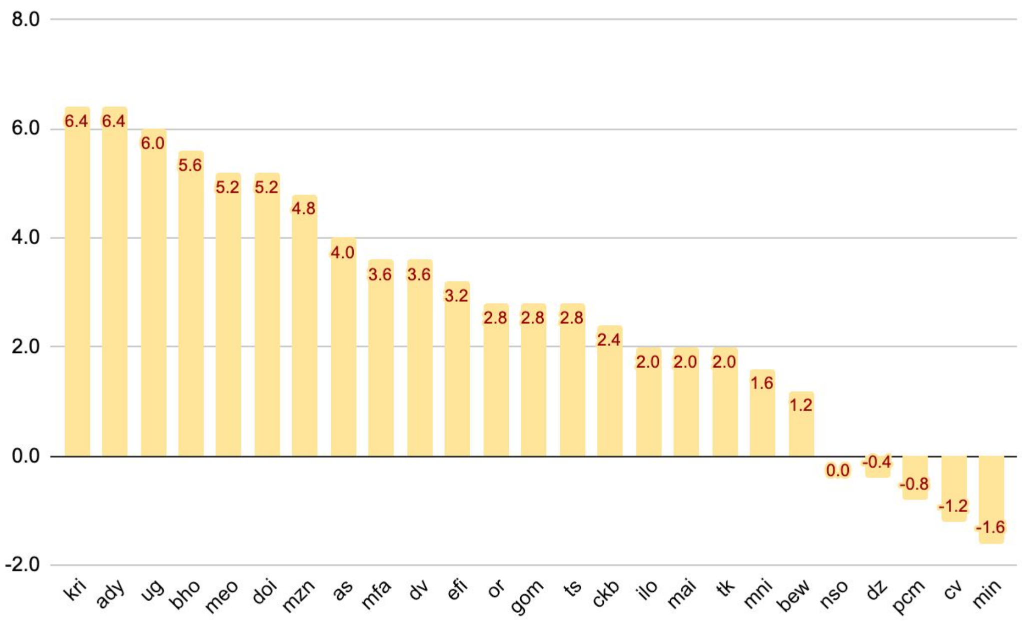
Evaluating for translation and grade-school numeric arithmetic

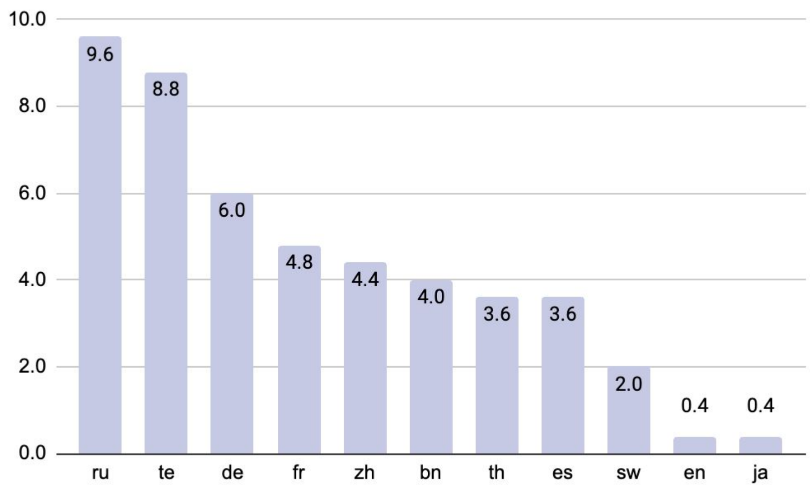
CALM leads to improvement and prevents catastrophic forgetting unlike fine-tuned baselines

Multi-linguality: Translation



Multi-linguality: Arithmetic Reasoning

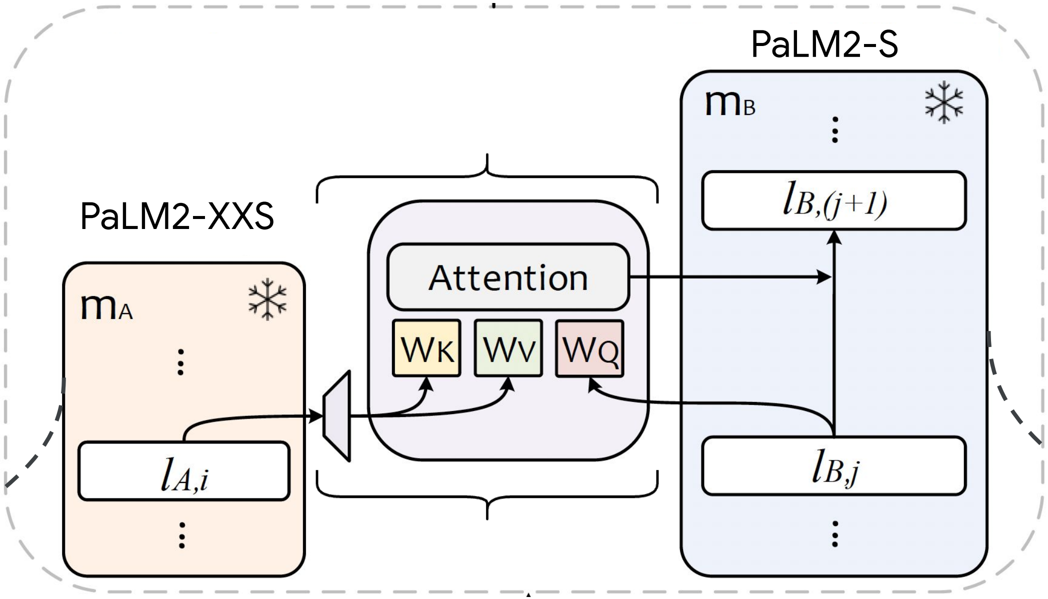




Gains w/ **CALM** (m_A: **PaLM2-NTL**, m_B: **PaLM2-XXS**)
w.r.t **Vanilla PaLM2-S** on **M-GSM** and **LR-GSM8K**.

- *Large performance improvements across most languages.*

Key Results: Code



PaLM2-GitHub

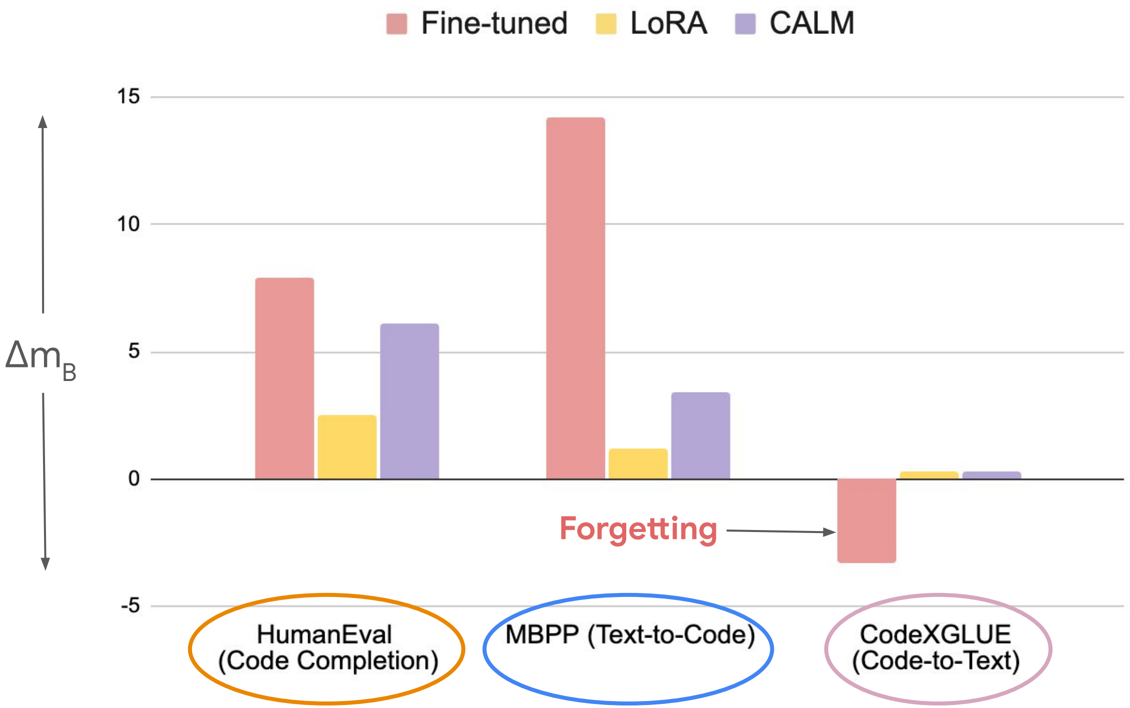
Re-using a model trained on **a snapshot of GitHub code heads** across numerous code languages.

trained on a mixture of GitHub code and the PaLM2 training corpora

PaLM2-S

A general purpose language model with **underrepresented code data.** Encomposes reasoning and logic.

Key Results: Code



Evaluating for code completion, code generation, and code explanation

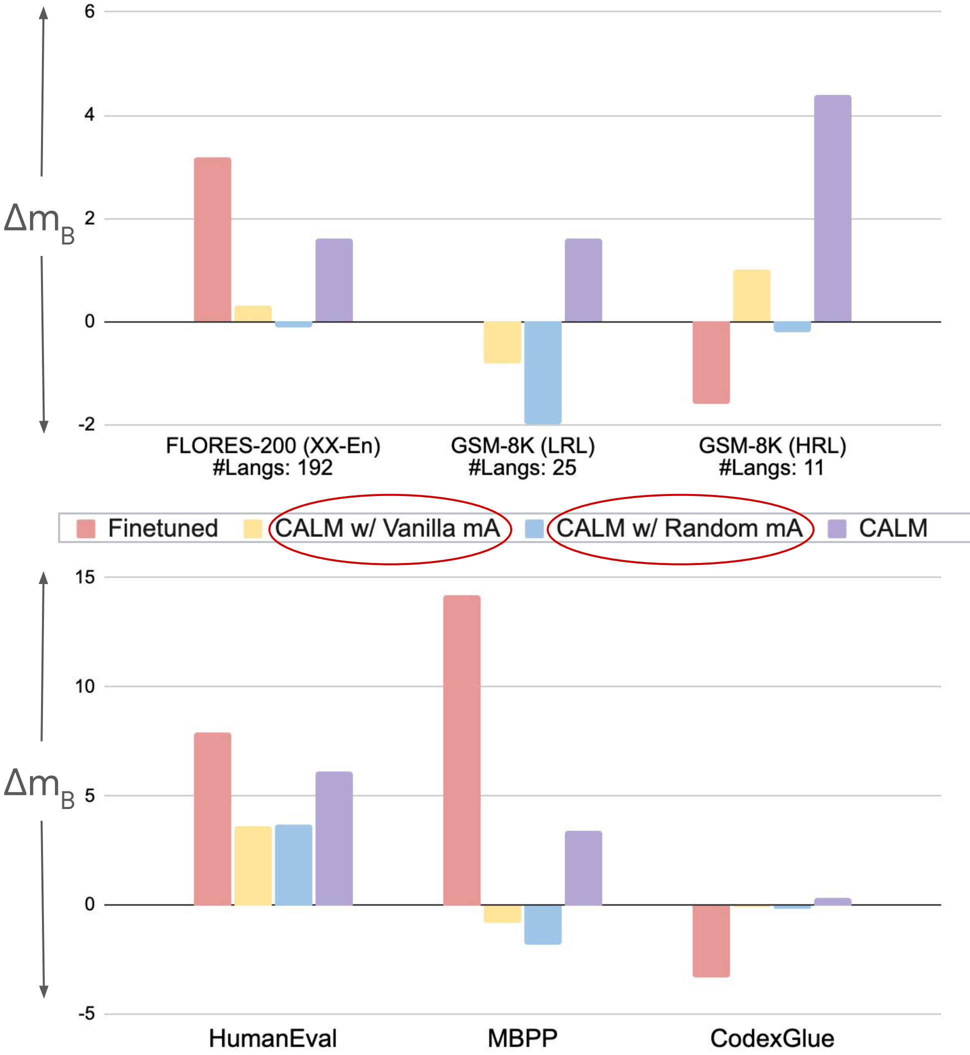
CALM leads to improvement and prevents catastrophic forgetting unlike fine-tuned baselines

Ablations

To investigate the source of improvements w/ CALM

Replacing mA w/ **an un-specialized vanilla variant**

Replacing mA w/ **a random variant**



Conclusion

This work enables a paradigm shift from expensive fine/instruction-tuning towards model composition for augmenting new knowledge in LLMs.

Through composition, we allow users and product teams to build new capabilities by composing their own small LMs with Google's LLMs.

Thank You!