Efficient Model Editing for Safe Information Localization and Stitching

Han Zhao

02/26/2025

NSF Workshop on Safe AI

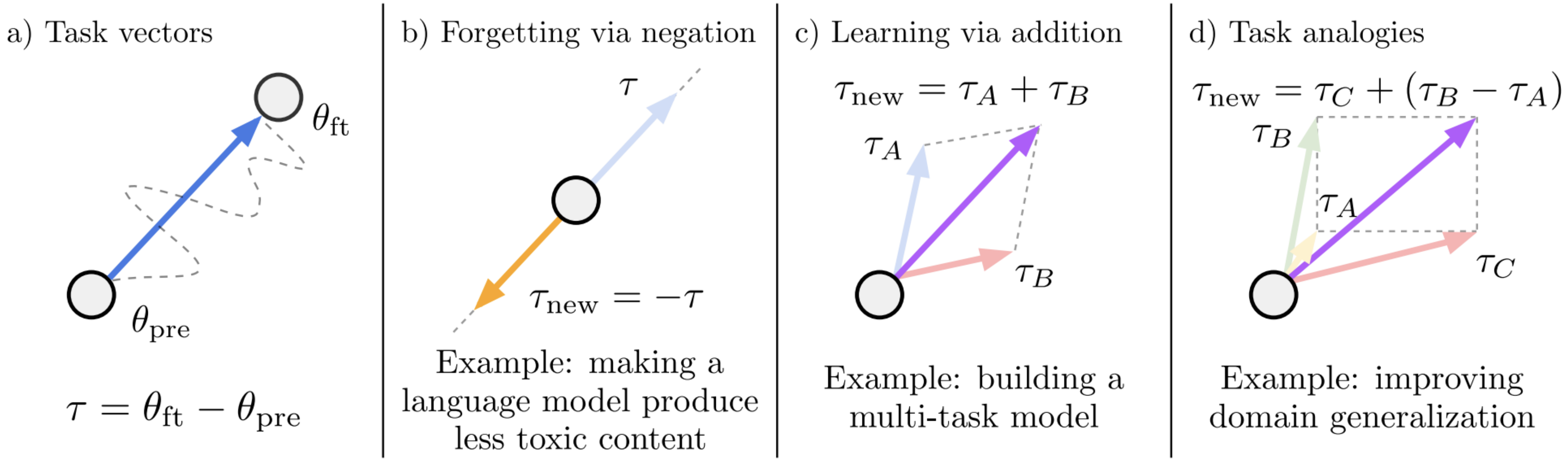
Joint work with Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang and Tong Zhang

hanzhao@illinois.edu
Department of Computer Science
University of Illinois Urbana-Champaign

ILLINOIS
Computer Science
GRAINGER COLLEGE OF ENGINEERING

Task Vectors for Large Models

Task vectors and model merging:



a) Task vectors
θ_{ft}
θ_{pre}
$\tau = \theta_{\text{ft}} - \theta_{\text{pre}}$
b) Forgetting via negation
τ
$\tau_{\text{new}} = -\tau$
Example: making a language model produce less toxic content
c) Learning via addition
$\tau_{\text{new}} = \tau_A + \tau_B$
τ_A
τ_B
Example: building a multi-task model
d) Task analogies
$\tau_{\text{new}} = \tau_C + (\tau_B - \tau_A)$
τ_B
τ_A
τ_C
Example: improving domain generalization

- Fine-tune k different tasks from a pre-trained model θ_{pre} to obtain fine-tuned model parameters $\theta_i, \forall i \in [k]$
- Task vector $\tau_i := \theta_i - \theta_{\text{pre}}$
- Arithmetic operations on task vectors for multi-tasking capability:

$$\theta_m := \theta_{\text{pre}} + \sum_{i \in [k]} \alpha_i \tau_i$$

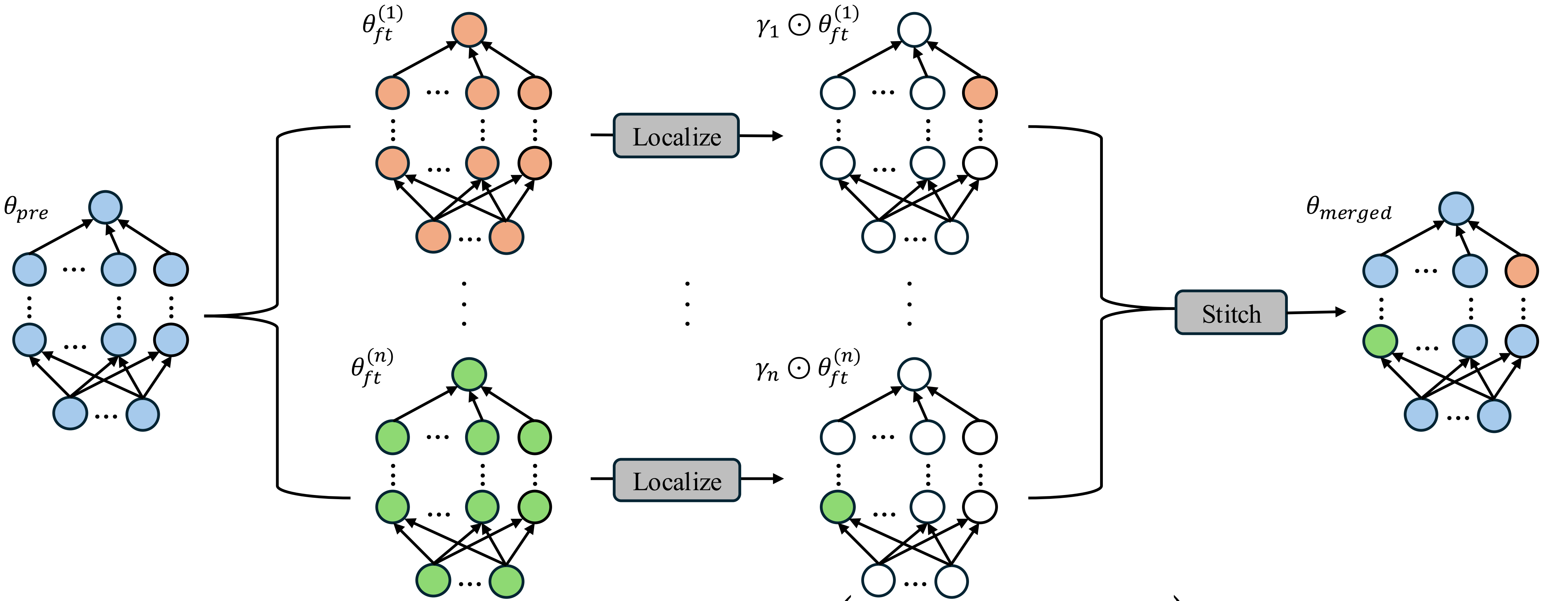
for some $\sum_{i \in [k]} \alpha_i = 1, \alpha_i \geq 0, \forall i \in [k]$

Figure credit to Ilharco et al., “Editing models with task arithmetic”

Sparsity Localization

Sparse task vectors for model merging?

- Computationally more efficient
- Less task interference among task vectors, hence suboptimal multi-task performance of the merged model
- Mechanistic interpretability: more transparent and explainable (?)



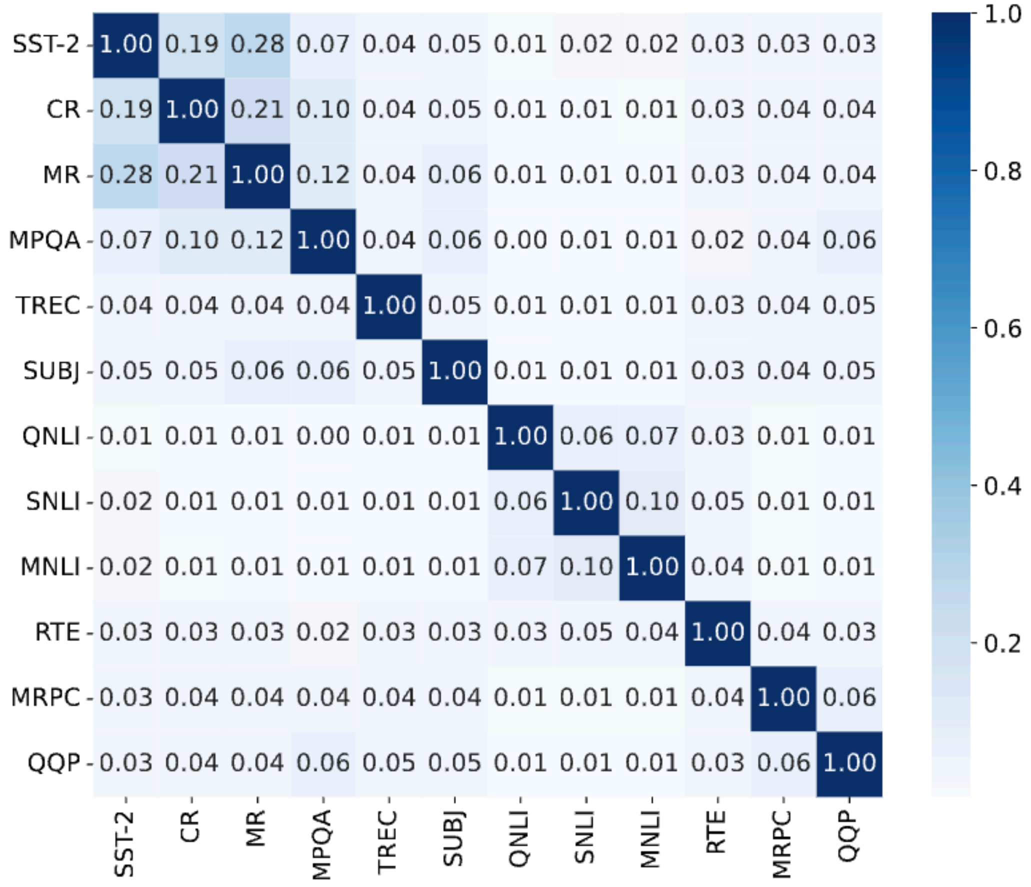
$$\gamma_i := \arg\min_{\gamma \in \{0,1\}^d} \mathscr{L}_i \left(\theta_{\text{pre}} + \gamma \odot \tau_i \right)$$

“Localize-and-Stitch: Efficient Model Merging via Sparse Task Arithmetic”, He et al., TMLR’ 24

Localize-and-Stitch

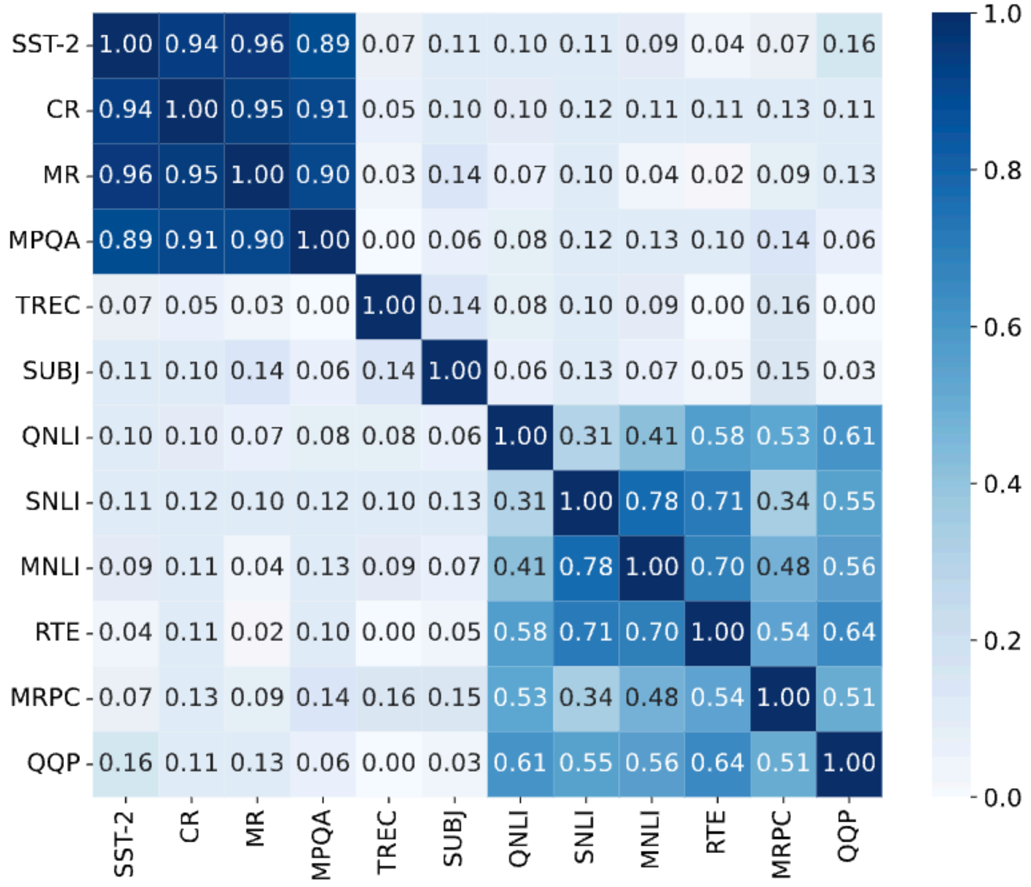
Experiments

- Base models: RoBERTa-base
- Tasks: 12 GLUE
- Sparsity: ~1% - 10%



	SST-2	CR	MR	MPQA	TREC	SUBJ	QNLI	SNLI	MNLI	RTE	MRPC	QQP
SST-2	1.00	0.19	0.28	0.07	0.04	0.05	0.01	0.02	0.02	0.03	0.03	0.03
CR	0.19	1.00	0.21	0.10	0.04	0.05	0.01	0.01	0.01	0.03	0.04	0.04
MR	0.28	0.21	1.00	0.12	0.04	0.06	0.01	0.01	0.01	0.03	0.04	0.04
MPQA	0.07	0.10	0.12	1.00	0.04	0.06	0.00	0.01	0.01	0.02	0.04	0.06
TREC	0.04	0.04	0.04	0.04	1.00	0.05	0.01	0.01	0.01	0.03	0.04	0.05
SUBJ	0.05	0.05	0.06	0.06	0.05	1.00	0.01	0.01	0.01	0.03	0.04	0.05
QNLI	0.01	0.01	0.01	0.00	0.01	0.01	1.00	0.06	0.07	0.03	0.01	0.01
SNLI	0.02	0.01	0.01	0.01	0.01	0.01	0.06	1.00	0.10	0.05	0.01	0.01
MNLI	0.02	0.01	0.01	0.01	0.01	0.01	0.07	0.10	1.00	0.04	0.01	0.01
RTE	0.03	0.03	0.03	0.02	0.03	0.03	0.03	0.05	0.04	1.00	0.04	0.03
MRPC	0.03	0.04	0.04	0.04	0.04	0.04	0.01	0.01	0.01	0.04	1.00	0.06
QQP	0.03	0.04	0.04	0.06	0.05	0.05	0.01	0.01	0.01	0.03	0.06	1.00

(a) Jaccard similarity of pairwise task masks.



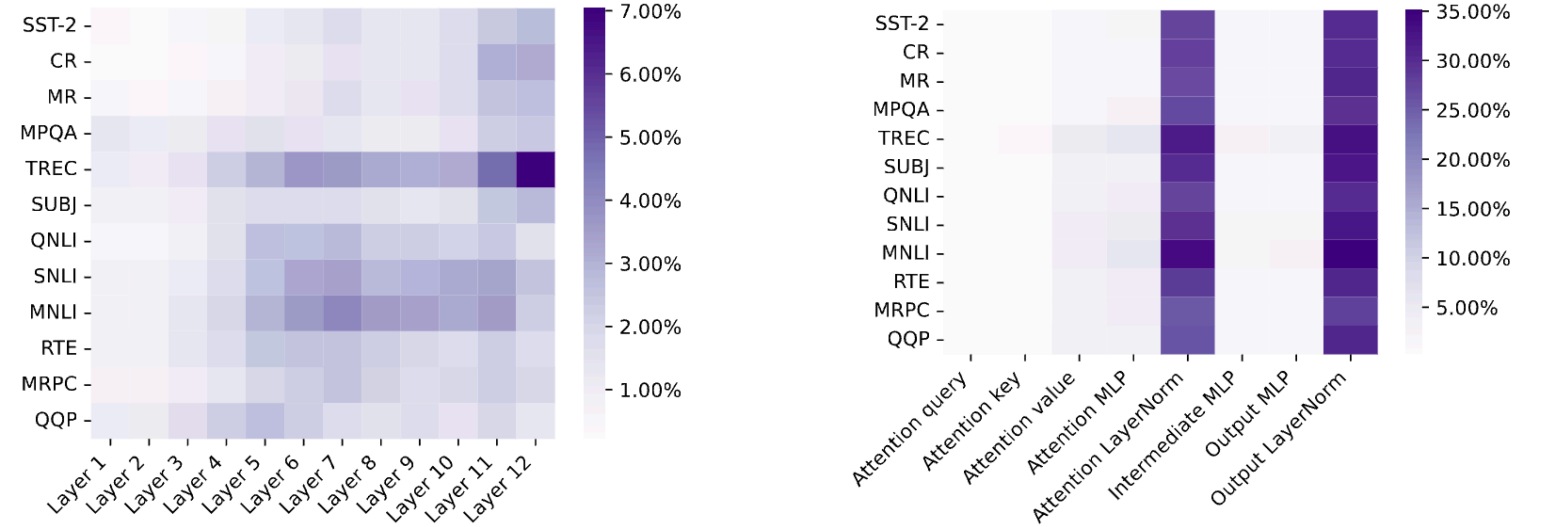
	SST-2	CR	MR	MPQA	TREC	SUBJ	QNLI	SNLI	MNLI	RTE	MRPC	QQP
SST-2	1.00	0.94	0.96	0.89	0.07	0.11	0.10	0.11	0.09	0.04	0.07	0.16
CR	0.94	1.00	0.95	0.91	0.05	0.10	0.10	0.12	0.11	0.11	0.13	0.11
MR	0.96	0.95	1.00	0.90	0.03	0.14	0.07	0.10	0.04	0.02	0.09	0.13
MPQA	0.89	0.91	0.90	1.00	0.00	0.06	0.08	0.12	0.13	0.10	0.14	0.06
TREC	0.07	0.05	0.03	0.00	1.00	0.14	0.08	0.10	0.09	0.00	0.16	0.00
SUBJ	0.11	0.10	0.14	0.06	0.14	1.00	0.06	0.13	0.07	0.05	0.15	0.03
QNLI	0.10	0.10	0.07	0.08	0.08	0.06	1.00	0.31	0.41	0.58	0.53	0.61
SNLI	0.11	0.12	0.10	0.12	0.10	0.13	0.31	1.00	0.78	0.71	0.34	0.55
MNLI	0.09	0.11	0.04	0.13	0.09	0.07	0.41	0.78	1.00	0.70	0.48	0.56
RTE	0.04	0.11	0.02	0.10	0.00	0.05	0.58	0.71	0.70	1.00	0.54	0.64
MRPC	0.07	0.13	0.09	0.14	0.16	0.15	0.53	0.34	0.48	0.54	1.00	0.51
QQP	0.16	0.11	0.13	0.06	0.00	0.03	0.61	0.55	0.56	0.64	0.51	1.00

(b) Cosine similarity of masked task vectors.

"Localize-and-Stitch: Efficient Model Merging via Sparse Task Arithmetic", He et al., TMLR' 24

Localize-and-Stitch

Localized modules:



(a) Distribution of localized regions in different network layers in the RoBERTa-base model.

(b) Distribution of localized regions in different network components in the RoBERTa-base model.

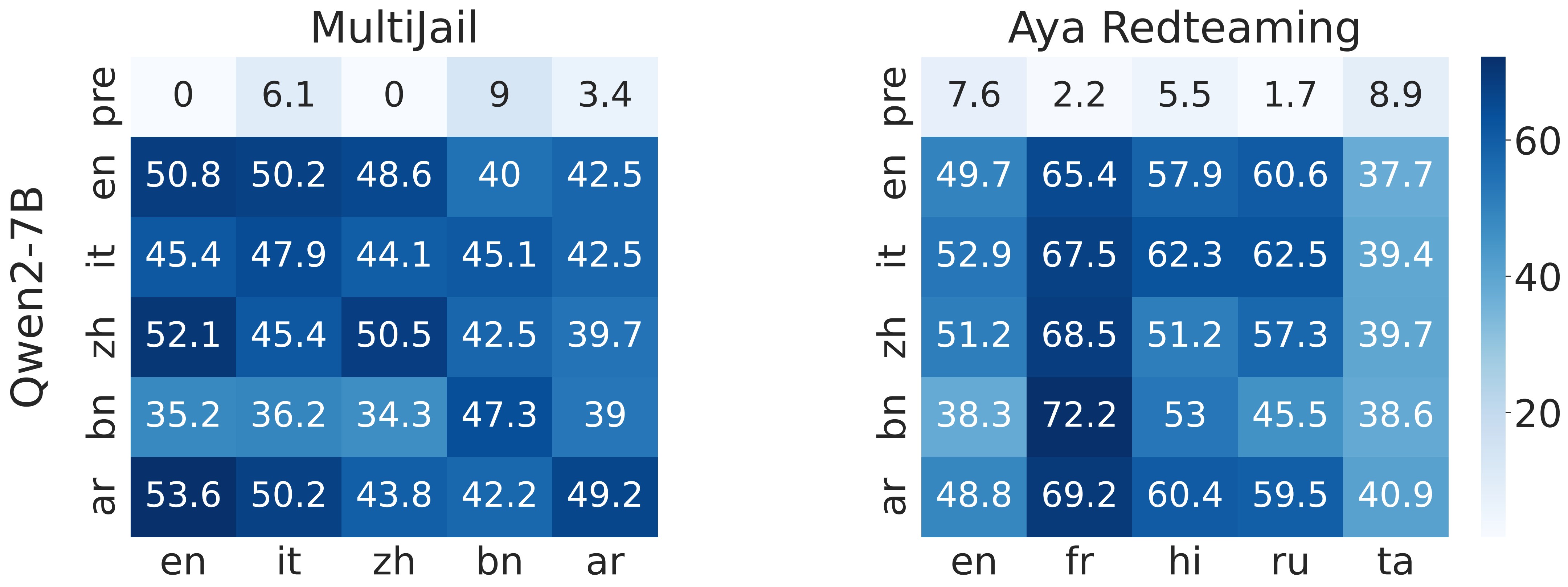
- The localized regions are predominantly found in the LayerNorm parameters
- Percentage = % of parameters in each module localized by the mask

"Localize-and-Stitch: Efficient Model Merging via Sparse Task Arithmetic", He et al., TMLR' 24

Safety Information Localization

Cross-lingual fine-tuning attacks:

- Fine-tuning on adversarial examples in one language breaks multilingual safety alignment



MultiJail

	en	it	zh	bn	ar
pre	0	6.1	0	9	3.4
en	50.8	50.2	48.6	40	42.5
it	45.4	47.9	44.1	45.1	42.5
zh	52.1	45.4	50.5	42.5	39.7
bn	35.2	36.2	34.3	47.3	39
ar	53.6	50.2	43.8	42.2	49.2

Aya Redteaming

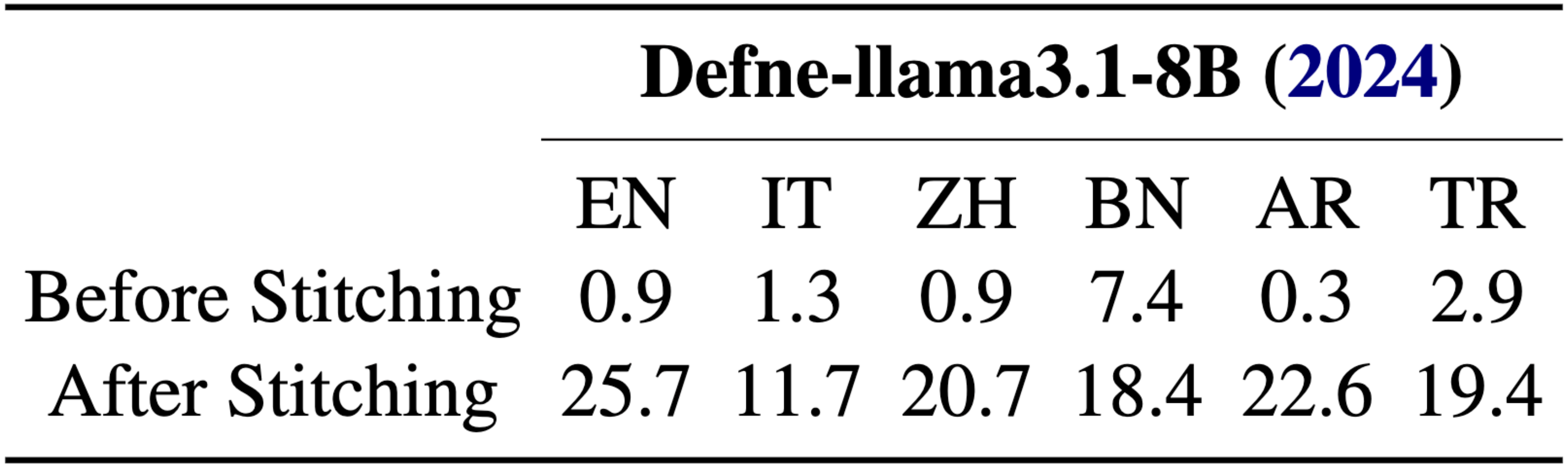
	en	fr	hi	ru	ta
pre	7.6	2.2	5.5	1.7	8.9
en	49.7	65.4	57.9	60.6	37.7
it	52.9	67.5	62.3	62.5	39.4
zh	51.2	68.5	51.2	57.3	39.7
bn	38.3	72.2	53	45.5	38.6
ar	48.8	69.2	60.4	59.5	40.9

“Towards Understanding the Fragility of Multilingual LLMs against Fine-Tuning Attacks”, Poppi et al., NAACL’ 25

Safety Information Localization

Cross-lingual fine-tuning attacks:

- Localization:
  - Overlapped localized regions of different tasks contain knowledge shared across them
  - Different languages share common safety regions in multilingual LLMs, explaining cross-lingual transfer
- Stitching attack:
  - Stitching localized jailbroken regions onto benign models breaks the safety alignment of unseen languages



	Defne-llama3.1-8B (2024)					
	EN	IT	ZH	BN	AR	TR
Before Stitching	0.9	1.3	0.9	7.4	0.3	2.9
After Stitching	25.7	11.7	20.7	18.4	22.6	19.4

"Towards Understanding the Fragility of Multilingual LLMs against Fine-Tuning Attacks", Poppi et al., NAACL' 25

Why do Task Vectors Work?

Properties of task vectors:

- Fine-tuning regime: small task vector norms
- Near task-orthogonality
- Local smoothness of the loss

Cross-task generalization of the merged model:

$$\theta_m := \theta_{\text{pre}} + \sum_{i \in [k]} \alpha_i \tau_i$$

For each task $i \in [k]$:

$$\mathscr{L}_i(\theta_m) - \mathscr{L}_i(\theta_i) \leq 2L_i C(1+\epsilon)$$

Loss of merged model

Near-orthogonality, local smoothness & fine-tuning regime

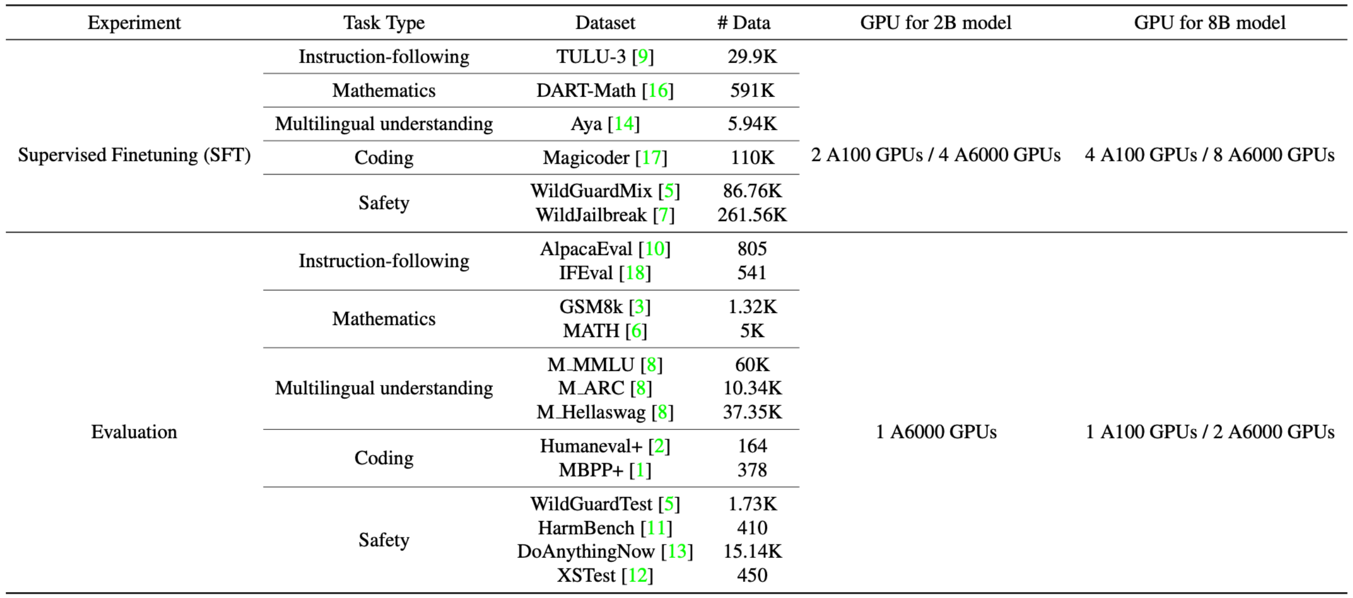
“Efficient Model Editing with Task Vector Bases: A Theoretical Framework and Scalable Approach”, Zeng et al., arXiv: 2502.01015

Thanks

Thanks for the generous support from the NSF SLES program!

Ongoing projects:

- MergeBench: a comprehensive benchmark for evaluation of different localization and merging methods



Experiment	Task Type	Dataset	# Data	GPU for 2B model	GPU for 8B model
Supervised Finetuning (SFT)	Instruction-following	TULU-3 [9]	29.9K	2 A100 GPUs / 4 A6000 GPUs	4 A100 GPUs / 8 A6000 GPUs
	Mathematics	DART-Math [16]	591K		
	Multilingual understanding	Aya [14]	5.94K		
	Coding	Magicoder [17]	110K		
	Safety	WildGuardMix [5] WildJailbreak [7]	86.76K 261.56K		
Evaluation	Instruction-following	AlpacaEval [10] IFEval [18]	805 541	1 A6000 GPUs	1 A100 GPUs / 2 A6000 GPUs
	Mathematics	GSM8k [3] MATH [6]	1.32K 5K		
	Multilingual understanding	M_MMLU [8] M_ARC [8] M_Hellaswag [8]	60K 10.34K 37.35K		
	Coding	Humaneval+ [2] MBPP+ [1]	164 378		
	Safety	WildGuardTest [5] HarmBench [11] DoAnythingNow [13] XSTest [12]	1.73K 410 15.14K 450