Gradient Vaccine

Multi-task Optimization in Multilingual Translation

2022.04.08

Liang Chen

Mentor: Shuming Ma

Task Introduction and Problem

Task: Multilingual Machine Translation

TLDR: One Model doing Multi-direction Translation

Training_set = {(en-de),(en-fr),(fr-en),(de-en), … ...}

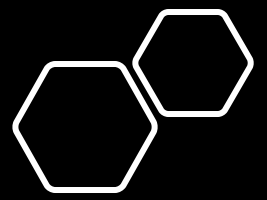
Test_set = {(en-de),(en-fr),(fr-en),(de-en), (fr-de), (de-fr)… ...}

Zero-shot

Problem:

1. Different translation directions might be conflicting to each other, in terms of gradient conflicting during training.

2. Different directions have imbalance data-size.



ICLR 2021 Spotlight

GRADIENT VACCINE: INVESTIGATING AND IMPROVING MULTI-TASK OPTIMIZATION IN MASSIVELY MULTILINGUAL MODELS

Zirui Wang[1,2]*, **Yulia Tsvetkov**[1], **Orhan Firat**[2], **Yuan Cao**[2]
[1]Carnegie Mellon University, [2]Google AI
{ziruiw,ytsvetko}@cs.cmu.edu, {orhanf,yuancao}@google.com

ABSTRACT

Massively multilingual models subsuming tens or even hundreds of languages pose great challenges to multi-task optimization. While it is a common practice to apply a language-agnostic procedure optimizing a joint multilingual task objective, how to properly characterize and take advantage of its underlying problem structure for improving optimization efficiency remains under-explored. In this paper, we attempt to peek into the black-box of multilingual optimization through the lens of loss function geometry. We find that gradient similarity measured along the optimization trajectory is an important signal, which correlates well with not only language proximity but also the overall model performance. Such observation helps us to identify a critical limitation of existing gradient-based multi-task learning methods, and thus we derive a simple and scalable optimization procedure, named Gradient Vaccine, which encourages more geometrically aligned parameter updates for close tasks. Empirically, our method obtains significant model performance gains on multilingual machine translation and XTREME benchmark tasks for multilingual language models. Our work reveals the importance of properly measuring and utilizing language proximity in multilingual optimization, and has broader implications for multi-task learning beyond multilingual modeling.

1 INTRODUCTION

Motivations

Naive joint training paradigm for Multilingual Translation is not optimal

- One possibility is that different task (language pair) may have conflicting loss geometries.
  - explore whether close language pairs share similar loss geometries while remote pairs may have conflicting gradients
  - identify what gradient/params are similar/conflicting, how to alleviate the conflicts to improve knowledge transfer and performance

Explore Loss Geometries in Multilingual Translation

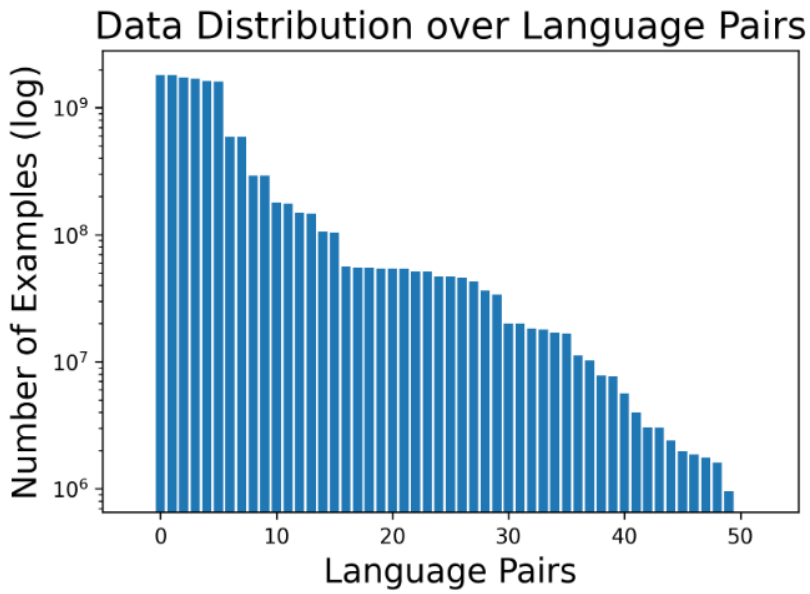
Part I

Experiment Setup

Goal: Study the gradient geometry of multilingual MT

Data: 25 languages (50 language pairs, en-xx & xx-en). 8B data.

Model: 6-6 Transformer-base



Evaluate: 3k semantically identical sentences are given in 25 languages

How to Measure Gradient Similarity?

- Train a Multilingual Model (50 directions)
- Compute Language-pair-wise Average Gradient Similiarity (en-fr,en-de,en-zh ...)
- Compute Gradient Similarity from different granularities (encoder|decoder|layer|training steps)

Main Observations

1. Gradient similarities reflect language proximities
2. Gradient similarities correlate positively with model quality
3. Gradient similarities evolve across layers and training steps.

1. Gradient similarities reflect language proximities

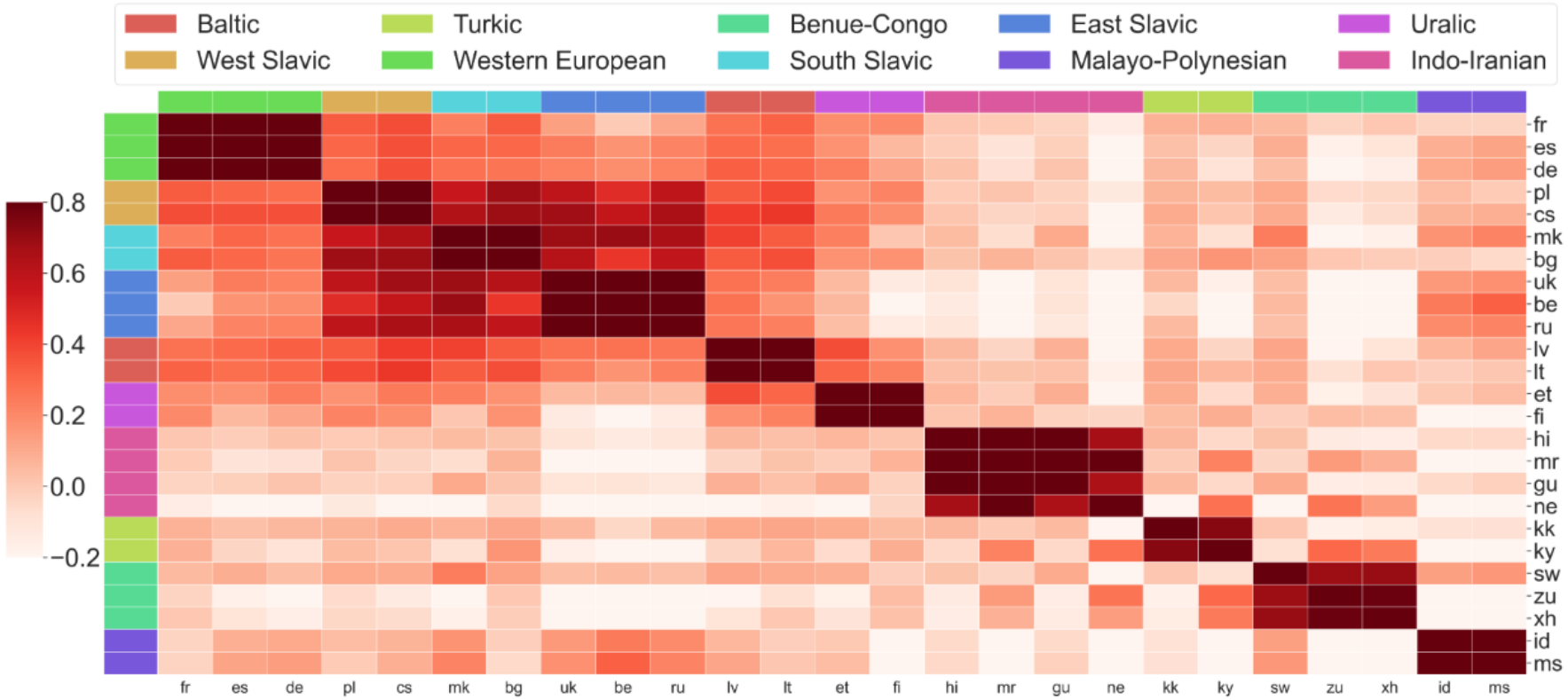
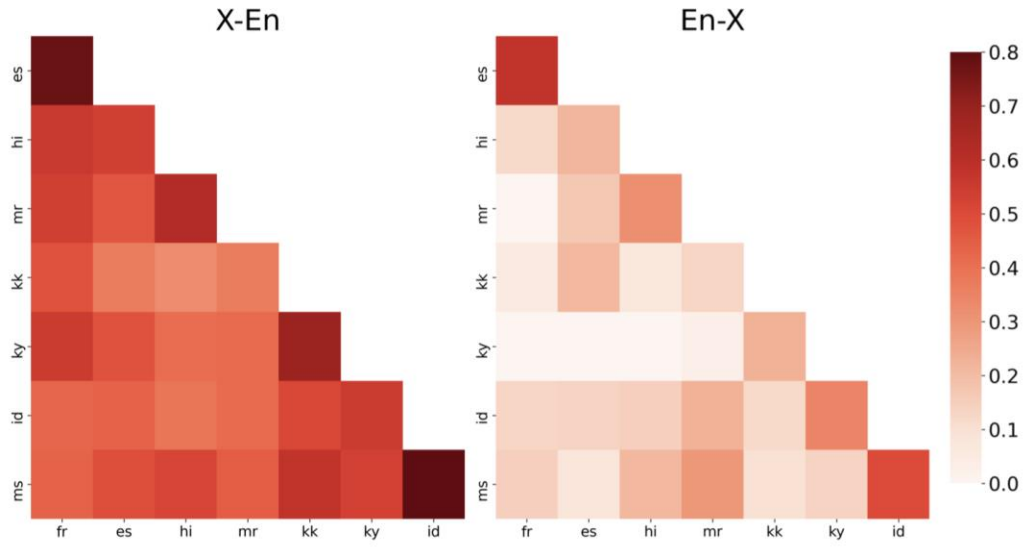


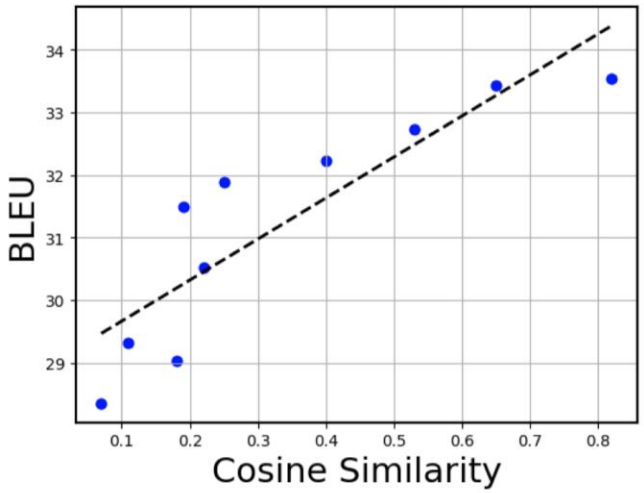
Figure 1: Cosine similarities of encoder gradients between *xx-en* language pairs averaged across all training steps. Darker cell indicates pair-wise gradients are more similar. Best viewed in color.[4]

2. Gradient similarities correlate positively with model quality

a. gradients in X-EN is more similar then that in EN-X



(a)

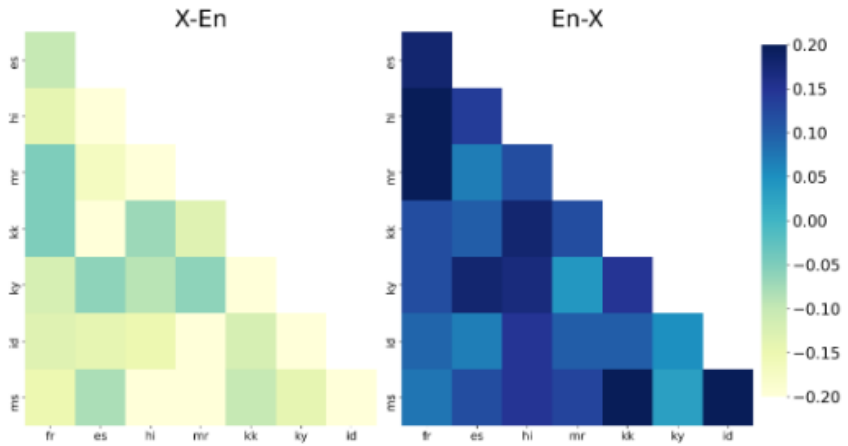


(b)

b. Pair EN-FR data with different extra task (en-de,en-hi …)

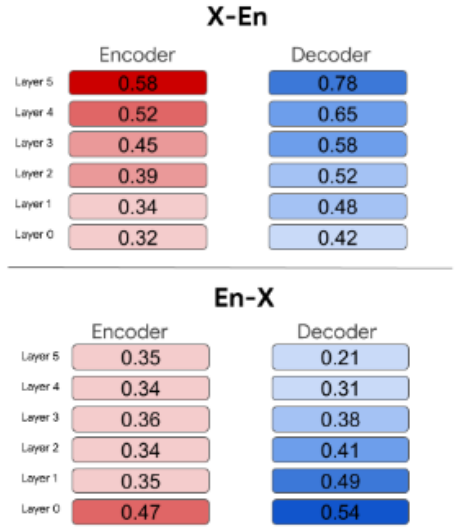
- Plot EN-FR BLEU with gradient similarity

3. Gradient similarities evolve across layers and training steps

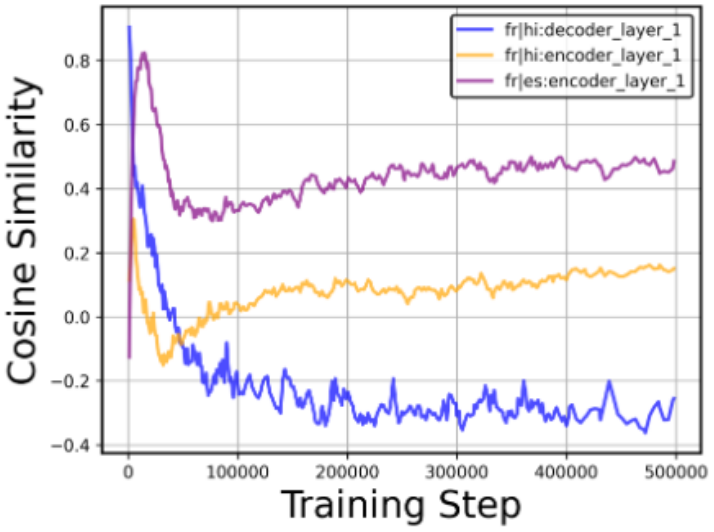


(a)

Similarity difference in encoder and decoder.



(b)



(c)

Similarity evolves across layers and training steps

Gradients should be more consistent when the encoder/decoder only needs to handle one single language

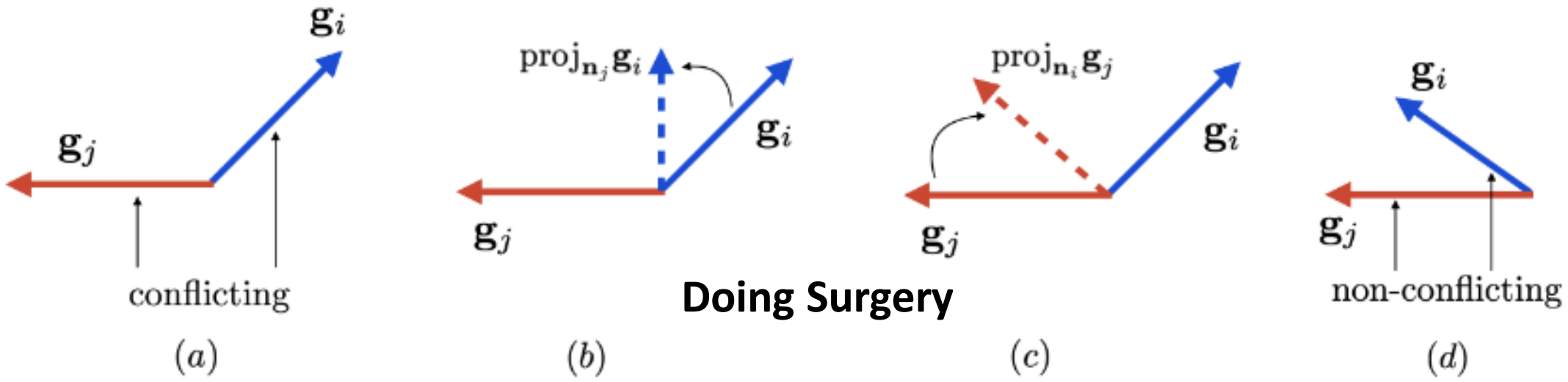
Alleviation of Gradient Conflict via Gradient Vaccine

Part II

Background: Gradient Surgery

PCGrad: Project Conflicting Gradients

1. Identify Gradient Conflict: cos_sim < 0
2. Solve Conflict: project to normal plane

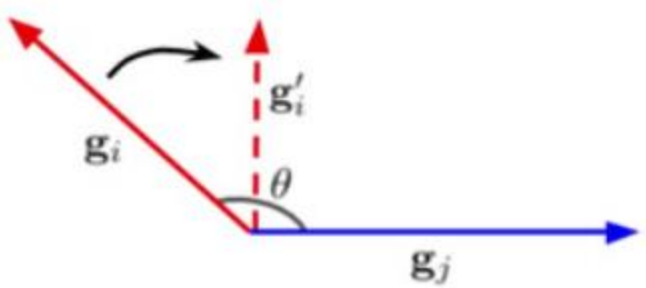


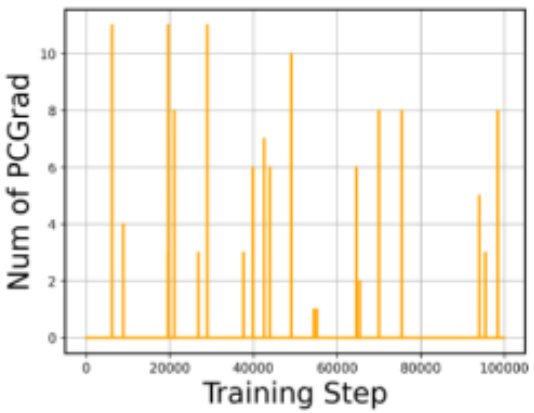
Doing Surgery

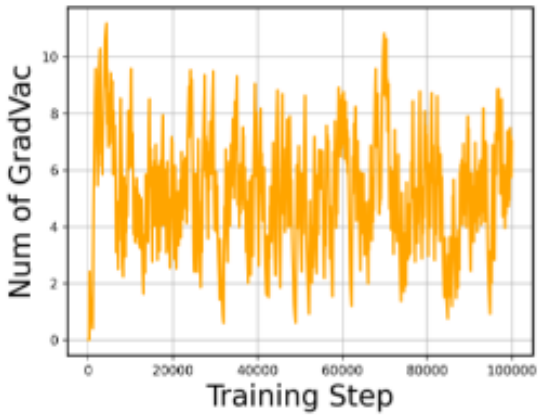
$$\mathbf{g}_i = \mathbf{g}_i - \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_j\|^2}\mathbf{g}_j.$$

Gradient Surgery

- Previous method, normal surgery (PCGrad)
  - Problem: surgery can be very sparse during training







- Motivation is simple: if gradients from two tasks are **conflicting**, remove their conflicts.

From Surgery to Vaccine(prevent before happening)

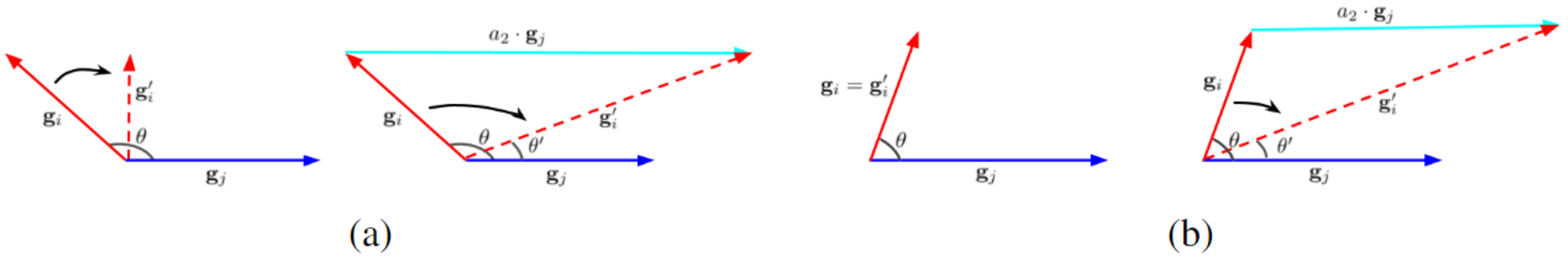


Figure 5: Comparing PCGrad (left) with GradVac (right) in two cases. **(a):** For negative similarity, both methods are effective but GradVac can utilize adaptive objectives between different tasks. **(b):** For positive similarity, only GradVac is active while PCGrad stays "idle".

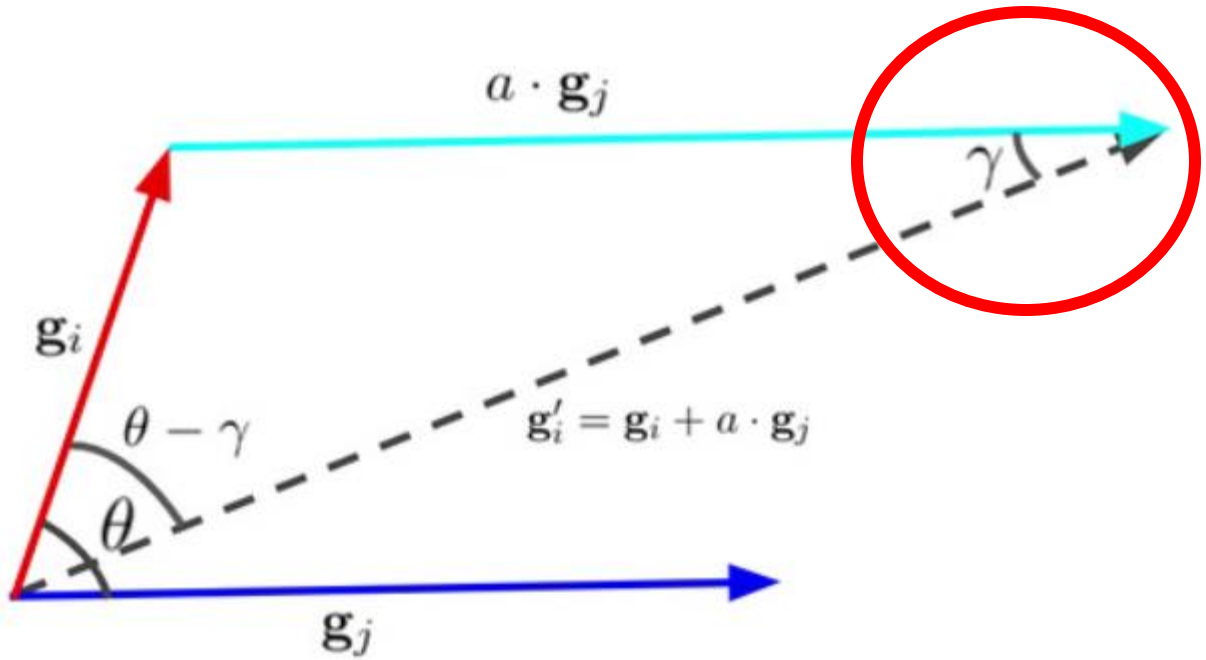
$$\mathbf{g}_i' = \mathbf{g}_i + \frac{\|\mathbf{g}_i\|(\phi_{ij}^T\sqrt{1-\phi_{ij}^2} - \phi_{ij}\sqrt{1-(\phi_{ij}^T)^2})}{\|\mathbf{g}_j\|\sqrt{1-(\phi_{ij}^T)^2}} \cdot \mathbf{g}_j.$$

Gradient Vaccine (treat before too late)

Suppose we a “target” cosine similarity between two gradients:

$$\cos(\theta) = \phi_{ij} = \frac{\mathbf{g}_i \cdot \mathbf{g}_j}{\|\mathbf{g}_i\|\|\mathbf{g}_j\|}$$

$$\cos(\gamma) = \phi_{ij}^T = \frac{\mathbf{g}_i' \cdot \mathbf{g}_j}{\|\mathbf{g}_i'\|\|\mathbf{g}_j\|}$$



(In the plane spanned by the two gradient vectors)

Then, by Law of Sines:

$$\frac{\|\mathbf{g}_i\|}{\sin(\gamma)} = \frac{a\|\mathbf{g}_j\|}{\sin(\theta - \gamma)}$$

$$\Rightarrow \frac{\|\mathbf{g}_i\|}{\sin(\gamma)} = \frac{a\|\mathbf{g}_j\|}{\sin(\theta)\cos(\gamma) - \cos(\theta)\sin(\gamma)}$$

$$\Rightarrow \frac{\|\mathbf{g}_i\|}{\sqrt{1-(\phi_{ij}^T)^2}} = \frac{a\|\mathbf{g}_j\|}{\phi_{ij}^T\sqrt{1-\phi_{ij}^2} - \phi_{ij}\sqrt{1-(\phi_{ij}^T)^2}}$$

$$\Rightarrow a = \frac{\|\mathbf{g}_i\|(\phi_{ij}^T\sqrt{1-\phi_{ij}^2} - \phi_{ij}\sqrt{1-(\phi_{ij}^T)^2})}{\|\mathbf{g}_j\|\sqrt{1-(\phi_{ij}^T)^2}}$$

How to implement during training ?

interactions change drastically across tasks, layers and training steps. To incorporate these three factors, we exploit an exponential moving average (EMA) variable for tasks i, j and parameter group k (e.g. the k-th layer) as:

$$\hat{\phi}_{ijk}^{(t)} = (1-\beta)\hat{\phi}_{ijk}^{(t-1)} + \beta\phi_{ijk}^{(t)}, \tag{3}$$

where $\phi_{ijk}^{(t)}$ is the computed gradient similarity at training step t, β is a hyper-parameter, and $\hat{\phi}_{ijk}^{(0)} = 0$. The full method is outlined in Algorithm 1 (Appendix E). Notice that gradient surgery is a special case of our proposed method such that $\phi_{ij}^{T} = 0$. As shown in the right of Figure 5(a) and 5(b),

- Empirically, we find performing GradVac by layers and on low-resource languages to work generally the best

Algorithm 1 GradVac Update Rule

Require: EMA decay β, Model Components $\mathcal{M} = \{\theta_k\}$, Tasks for GradVac $\mathcal{G} = \{\mathcal{T}_i\}$
Initialize model parameters
Initialize EMA variables $\hat{\phi}_{ijk}^{(0)} = 0, \forall i, j, k$
Initialize time step $t = 0$
while not converged **do**
  Sample minibatch of tasks $\mathcal{B} = \{\mathcal{T}_i\}$
  for $\theta_k \in \mathcal{M}$ **do**
    Compute gradients $\mathbf{g}_{ik} \leftarrow \nabla_{\theta_k} \mathcal{L}_{\mathcal{T}_i}, \forall \mathcal{T}_i \in \mathcal{B}$
    Set $\mathbf{g}'_{ik} \leftarrow \mathbf{g}_{ik}$
    for $\mathcal{T}_i \in \mathcal{G} \cap \mathcal{B}$ **do**
      for $\mathcal{T}_j \in \mathcal{B} \setminus \mathcal{T}_i$ in random order **do**
        Compute $\phi_{ijk}^{(t)} \leftarrow \frac{\mathbf{g}'_{ik} \cdot \mathbf{g}_{jk}}{\|\mathbf{g}'_{ik}\| \|\mathbf{g}_{jk}\|}$ 计算两个任务之间的梯度相似度
        if $\phi_{ijk}^{(t)} < \hat{\phi}_{ijk}^{(t)}$ **then**
          Set $\mathbf{g}'_{ik} = \mathbf{g}'_{ik} + \frac{\|\mathbf{g}'_{ik}\|(\hat{\phi}_{ijk}^{(t)}\sqrt{1-(\phi_{ijk}^{(t)})^2} - \phi_{ijk}^{(t)}\sqrt{1-(\hat{\phi}_{ijk}^{(t)})^2})}{\|\mathbf{g}_{jk}\|\sqrt{1-(\hat{\phi}_{ijk}^{(t)})^2}} \cdot \mathbf{g}_{jk}$
        end if
        Update $\hat{\phi}_{ijk}^{(t+1)} = (1-\beta)\hat{\phi}_{ijk}^{(t)} + \beta\phi_{ijk}^{(t)}$
      end for
    end for
    Update θ_k with gradient $\sum \mathbf{g}'_{ik}$
  end for
  Update $t \leftarrow t + 1$
end while

Main Result & Ablation

	En→Any					*Any→En*				
	en-fr	en-cs	en-hi	en-tr	avg	fr-en	cs-en	hi-en	tr-en	avg
Monolithic Training										
(1) Bilingual Model	41.80	24.76	5.77	9.77	20.53	36.38	29.17	8.68	13.87	22.03
(2) Multilingual Model	37.24	20.22	13.69	18.77	22.48	34.29	27.66	18.48	22.01	25.61
Multi-task Training										
(3) GradNorm (Chen et al., 2018b)	37.02	18.78	11.57	15.44	20.70	34.58	27.85	18.03	22.37	25.71
(4) MGDA (Sener & Koltun, 2018)	38.22	17.54	12.02	13.69	20.37	35.05	26.87	18.28	22.41	25.65
(5) PCGrad (Yu et al., 2020)	37.72	20.88	13.77	18.23	22.65	34.37	27.82	18.78	22.20	25.79
(6) PCGrad w. all_layer	38.01	21.04	13.95	18.46	22.87	34.57	27.84	18.84	22.48	25.93
Our Approach										
(7) GradVac w. fixed_obj	38.41	21.12	13.75	18.68	22.99	34.55	27.97	18.72	22.14	25.85
(8) GradVac w. whole_model	38.76	21.32	14.22	18.89	23.30	34.84	28.01	18.85	22.24	25.99
(9) GradVac w. all_layer	**39.27***	**21.67***	**14.88***	**19.73***	**23.89**	**35.28***	**28.42***	**19.07***	**22.58***	**26.34**

Table 1: BLEU scores on the WMT dataset. The best result for multilingual model is **bolded** while underline signifies the overall best, and * means the gains over baseline multilingual models are statistically significant with $p < 0.05$.

GradVaccine alleviates data imbalance

Any→En	High	Med	Low	All
T=1	28.56	28.51	19.57	24.95
T=5	28.16	28.42	24.32	26.71
GradVac	**28.99**	**28.94**	**24.58**	**27.21**

En→Any	High	Med	Low	All
T=1	22.62	21.53	12.41	18.18
T=5	22.04	21.43	13.07	18.25
GradVac	**24.20**	**21.83**	**13.30**	**19.08**

Temperature Sampling

$$P_D(i) = \frac{q_i^{1/\tau}}{\sum_{k=1}^{n} q_k^{1/\tau}} \text{ where } q_i = \frac{|D_{\text{train}}^i|}{\sum_{k=1}^{n} |D_{\text{train}}^k|}. \quad (4)$$

Summary

- Gradients similairity is highly related to language similarity
- In Multilingual NMT, Encoder and decode have different gradients and loss geometries considering different directions. (X-EN , EN-X)
- Propose a more aggressive method to alleviate the conflicting gradients by vaccine task-by-task, layer-by-layer