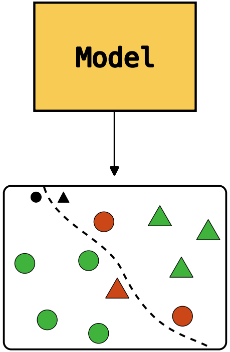
Model Update Regression

Raphael Schumann
Natural Language Processing PhD Student
Heidelberg University, Germany

February 14, 2023

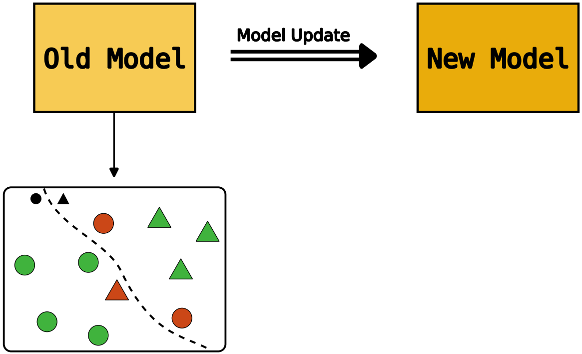
What is Model Update Regression?



Simple classification model as example:

- Train model
- Model makes correct and incorrect predictions

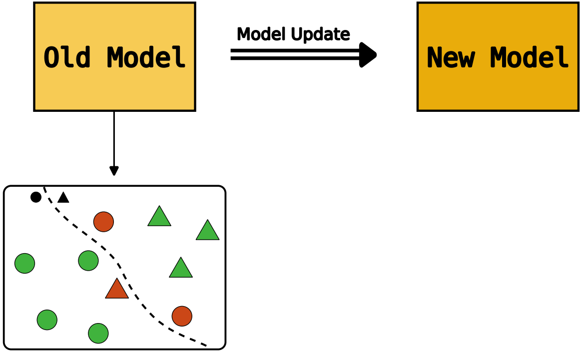
What is Model Update Regression?



Types of Model Updates includes:

- Architecture change
- Retrain with more data
- Retrain with different hyperparameter

What is Model Update Regression?



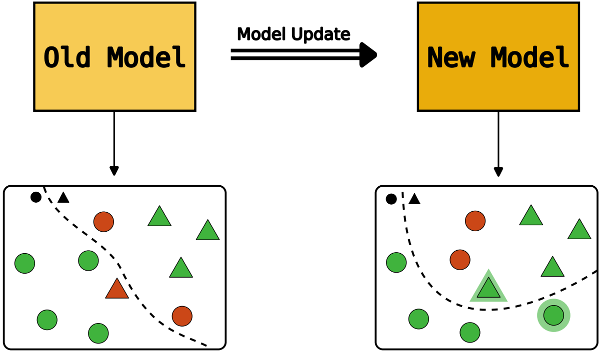
Types of Model Updates includes:

- Architecture change
- Retrain with more data
- Retrain with different hyperparameter

Motivation for Model Update:

- **Better accuracy**
- New features
- Smaller footprint

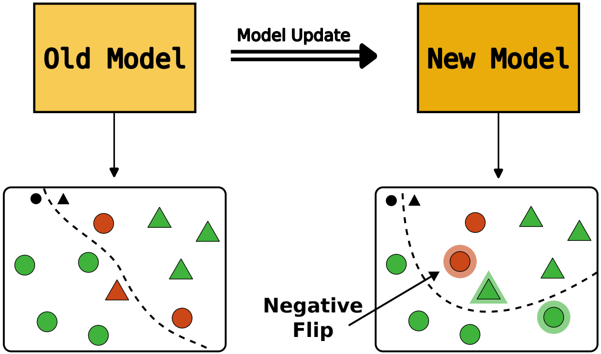
What is Model Update Regression?



New Model:

- Makes more correct predictions
- Incorrect predictions are flipped to correct ones

What is Model Update Regression?



New Model:

- Makes more correct predictions
- Incorrect predictions are flipped to correct ones

But:

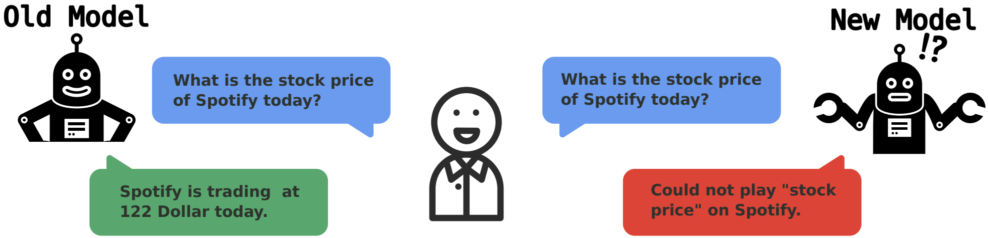
- Also flips correct predictions to incorrect
- Negative flips cause regression in classification models

Why is Model Update Regression Bad?

User of Virtual Assistant:

- Learns what kind of commands work
- Consistently uses the same commands

Why is Model Update Regression Bad?



Updated system with New Model:

- Negative flips have direct negative impact on user experience
- Overall better performance only perceived over time

→ **User gets the impression that the system has degraded**

Why is Model Update Regression Bad?

Other consequences of lacking backward compatibility in ML models:

- Humans lose trust in Human-AI collaboration teams
  - [Bansal et al. 2019]
- Downstream systems can break if they are not robust to novel errors
  - [Srivastava et al. 2020]
- Fluctuating categorization of images in a user's gallery
  - [Shen et al. 2020b]
- Inconsistent content moderation
- Inconsistent fraud detection

How to Measure Regression?

$$\text{NFR} = \frac{1}{|\mathcal{D}_{reg}|} \sum_{x,y \in \mathcal{D}_{reg}} \mathbb{1}[f_{\theta_{old}}(x) = y \wedge f_{\theta_{new}}(x) \neq y]$$

[Yan et al. 2021]

Negative Flip: Instance (x, y) that is classified correctly by the old model $(f_{\theta_{old}})$ and incorrectly by the new model $(f_{\theta_{new}})$.

Negative Flip Rate (NFR): Ratio of negative flips to all instances in the regression set $(\mathcal{D}_{reg})$ e.g. development or test set.

Causation and Mitigation of Negative Flips

Negative Flips are caused by:

- Stochasticity in optimization [Srivastava et al. 2020]
  - Changing random seed introduces negative flips [Somepalli et al. 2022]
- Amplified by extent of model change [Yan et al. 2021]

Let's look at concrete settings and strategies to mitigate negative flips!

Update Model Architecture

Update Model Architecture

ImageNet Classification (ILSVRC12)

Model Name	Method	ACC↑	NFR↓
ResNet-18 (Old Model)		69.8	0.0
➔ ResNet-50 (New Model)	No Treatment	74.2	4.9

Paraphrase Classification (MRPC)

Model Name	Method	ACC↑	NFR↓
$\text{BERT}_{\text{BASE}}$ (Old Model)		86.0	0.0
➔ $\text{BERT}_{\text{LARGE}}$ (New Model)	No Treatment	87.8	5.9

Update Model Architecture

Train with additional distillation loss:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \sum_{i}^{|\mathcal{D}_{train}|} \beta \; D_{KL}[p_{\theta_{new}}(x_i), p_{\theta_{old}}(x_i)]$$

$\mathcal{L}_{CE}$ is the cross entropy loss
D_{KL} is the KL divergence of old and new probabilities over training instances
$\beta = 1$ is regular knowledge distillation

Focal Distillation:
Focus the distillation loss on specific instances

$$\beta = \mathbb{1}[f_{\theta_{old}}(x_i) = y_i]$$

- Only instances that were correct by the old model
- Static throughout training

[Yan et al. 2021]

$$\beta = \mathbb{1}[p_{\theta_{old}}(y_i|x_i) > p_{\theta_{new}}(y_i|x_i)]$$

- Old model has higher probability for correct class than new model
- Dynamic selection during training

[Xie et al. 2021]

Update Model Architecture

ImageNet Classification (ILSVRC12)

Model Name		**Method**	**ACC↑**	**NFR↓**
ResNet-18	(Old Model)		69.8	0.0
➔ ResNet-50 (New Model)		No Treatment	**74.2**	4.9
		Focal Distillation	73.7	**2.9**
		Dynamic FD	-	-

Paraphrase Classification (MRPC)

Model Name		**Method**	**ACC↑**	**NFR↓**
$\text{BERT}_{\text{BASE}}$	(Old Model)		86.0	0.0
➔ $\text{BERT}_{\text{LARGE}}$ (New Model)		No Treatment	87.8	5.9
		Focal Distillation	88.5	4.9
		Dynamic FD	**88.7**	**2.5**

Update Model Architecture

ImageNet Classification (ILSVRC12)

Model Name	**Method**	**ACC↑**	**NFR↓**
ResNet-18 (Old Model)		69.8	0.0
→ ResNet-50 (New Model)	No Treatment	74.2	4.9
	Focal Distillation	73.7	2.9
	Ensemble (16x)	**77.8**	**1.6**

Ensembling new models reduces negative flips, but is often infeasible in practice.

Strategies to avoid the **inference cost penalty**:

- Choose most centric model from the ensemble [Xie et al. 2021]
- Distill from the ensemble [Yan et al. 2021]

Update Model Architecture

Specialized Methods

Backward Compatible Reranking [Cai et al. 2022]

- For structured prediction tasks
- Use old model to rerank top beams of new model

Backward-Compatible Representation Learning [Shen et al. 2020a]

- Avoid backfilling embeddings after model update
- Add old classifier loss when training new embeddings

Update Training Data

Update Training Data

Context:

Work done during my 2022 internship at Amazon AWS Lex (chatbot service)

➔ focus on intent classification task [Schumann et al. 2023]

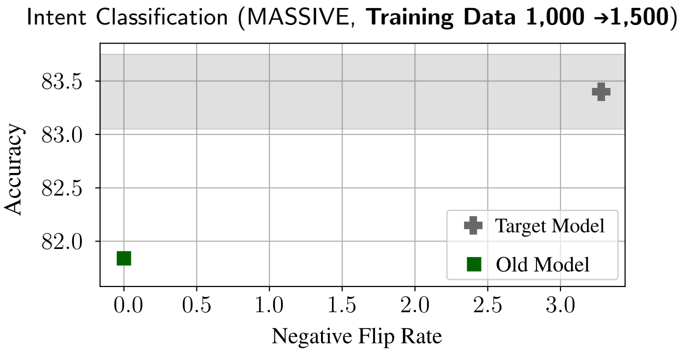
Motivation:

Data updates are more frequent than architecture updates

Prerequisites:

- We do assume full access to the old data when training the new model
  - $\mathcal{D}_{updated} = \mathcal{D}_{old} + \mathcal{D}_{new}$
- Data update consists of i.i.d training data
- We use **RoBERTa$_{\textbf{BASE}}$** as pretrained model for all experiments
  - add classification layer
- MASSIVE dataset is intent classification with 60 classes [FitzGerald et al. 2022]

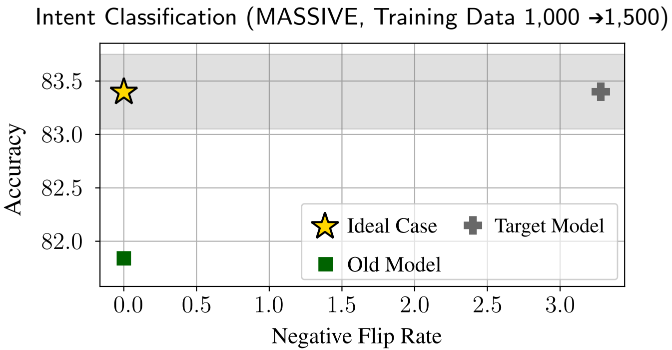
Update Training Data



Model Name	Weights	Initialization	Data	ACC↑	NFR↓
Old Model	θ_{old}	θ_{pre}	$\mathcal{D}_{old}$	81.8	0.0
Target Model	θ_{target}	θ_{pre}	$\mathcal{D}_{updated}$	83.4	3.3

Gray area is the accuracy confidence interval of the target model. It dictates the level of accuracy a new model should reach.

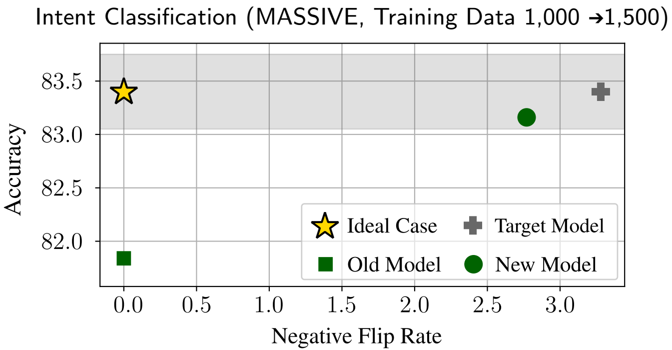
Update Training Data



Model Name	**Weights**	**Initialization**	**Data**	**ACC↑**	**NFR↓**
Old Model	θ_{old}	θ_{pre}	$\mathcal{D}_{old}$	81.8	0.0
Target Model	θ_{target}	θ_{pre}	$\mathcal{D}_{updated}$	83.4	3.3

The ideal case is a model that maintains target accuracy but has no negative flips.

Update Training Data



Model Name	**Weights**	**Initialization**	**Data**	**ACC↑**	**NFR↓**
Old Model	θ_{old}	θ_{pre}	$\mathcal{D}_{old}$	81.8	0.0
Target Model	θ_{target}	θ_{pre}	$\mathcal{D}_{updated}$	83.4	3.3
New Model	θ_{new}	θ_{old}	$\mathcal{D}_{updated}$	83.2	**2.8**

The *New Model* is initialized by the *Old Model* and thus achieves lower NFR than the *Target Model* which is initialized by the pretrained model.

Update Training Data

Proposed Method: Backward Compatible Weight Interpolation (BCWI)

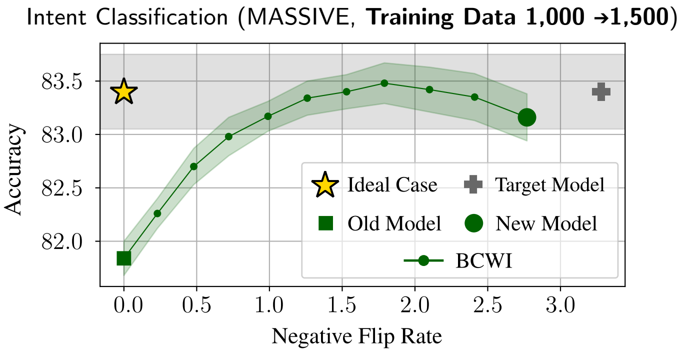
BCWI is the linear interpolation between the weights of the old model and new model:

$$\boxed{\theta_{\mathsf{BCWI}} = \alpha\theta_{old} + (1-\alpha)\theta_{new}}$$

θ_{old} are the weights of the old model
θ_{new} are the weights of the new model
α is the interpolation parameter

More about weight interpolation later...

Update Training Data



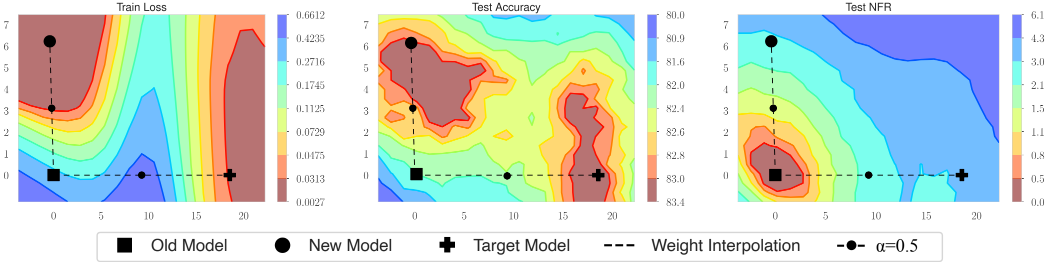
Model Name	**Weights**	**Initialization**	**Data**	**ACC↑**	**NFR↓**
Old Model	θ_{old}	θ_{pre}	$\mathcal{D}_{old}$	81.8	0.0
Target Model	θ_{target}	θ_{pre}	$\mathcal{D}_{updated}$	83.4	3.3
New Model	θ_{new}	θ_{old}	$\mathcal{D}_{updated}$	83.2	2.8
BCWI α=0.4	$\alpha\theta_{old} + (1-\alpha)\theta_{new}$			83.4	1.4
BCWI α=0.6				83.1	**0.8**

Properties of BCWI and Baselines

	Additional Memory	Training Time	Tune Trade-Off	Inference Cost
EWC	$\|F\| + \|\theta_{old}\|$	$(F+)$ 1.9x	retrain	1x
Prior WD	$\|\theta_{old}\|$	1.1x	retrain	1x
Mixout	$\|\theta_{old}\|$	1.6x	retrain	1x
Distillation	$\|\mathcal{D}_{train}\|$	1.3x	retrain	1x
BCWI	-	1x	post training	1x

BCWI is faster to train, uses less GPU memory and the NFR/Accuracy trade-off can be tuned without retraining the model.

BCWI Loss and Error Landscape



Further finetuning the old model places the new model in **the same local minimum** and thus makes weight interpolation effective. Low Negative Flip Rate is centered around the old model.

Potential to use "Git Re-Basin" [Ainsworth, Hayase, and Srinivasa 2022] and leverage permutation symmetries to move target model into the same basin in order to make it "averageable" with the old model.

Visualization Technique by [Izmailov et al. 2018]

Parenthesis: Weight Interpolation/Averaging

Weights along the same training trajectory

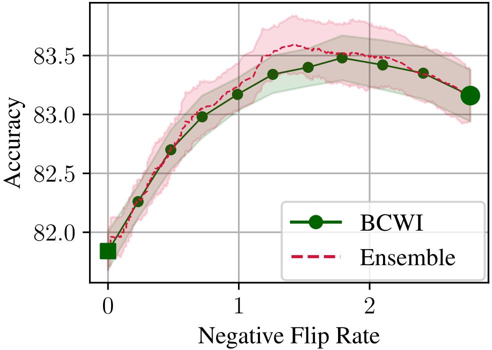
- Checkpoint Averaging
- Stochastic Weight Averaging [Izmailov et al. 2018]

Weights optimized independently but same initialization

Because of the many nonlinearities, it is not clear that linear interpolation of model weights can result in high accuracy solutions. [Ilharco et al. 2022]

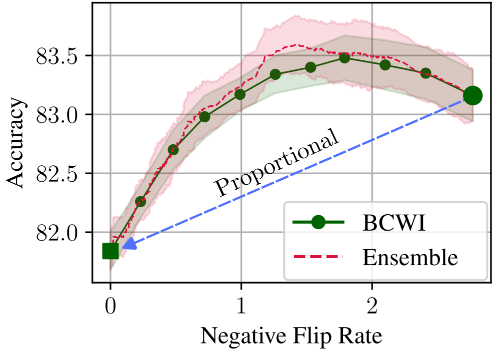
- **Model Soup**: Average weights finetuned from same pretrained model [Wortsman et al. 2022] ➔ SoupBCWI in our paper
- Bias-Variance-Covariance-*Locality* decomposition [Rame et al. 2022]
  - Locality term: Squared Euclidean Distance.
- **Fisher Merging**: Use Fisher information matrix as importance weighting when averaging model weights [Matena and Raffel 2021] ➔ FisherBCWI in our paper

Weight Interpolation vs. Probability Ensemble



Weight interpolation produces similar results as a weighted ensemble of output probabilities, but without the inference cost.

Weight Interpolation vs. Probability Ensemble



Accuracy Gain	1.4
Negative Flip Rate	2.8
Positive Flip Rate	4.2

- Blue line is the trajectory when negative and positive flips are flipped back proportionally

Why are negative flips get flipped back disproportional when interpolating towards the old model?

Conclusion BCWI Paper

Backward Compatible Weight Interpolation (BCWI) effectively reduces regression during data updates. It is easy to implement and does not increase training or inference time.

Backward Compatibility During Data Updates by Weight Interpolation, 2023,
Raphael Schumann, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao and Yi Zhang

- Second data update scenario of adding more classes
- Experiments on more datasets
- SoupBCWI, FisherBCWI

Summary & What's Next

Regression in model updates is an **important and understudied problem** with real world implications. Architecture update regression is best mitigated by **distillation based methods** and data update regression with **weight interpolation** of new and old model.

What's Next?

- How do we measure regression in seq2seq tasks, e.g. summarization, translation?
  - Output changes but is still correct
  - Gradual badness scale of negative flips
- Can we reduce regression for in-context learning when moving between LLMs or when providing more examples?
- What about open ended text generation of LLMs?

Advertisement

Also check out my other work on https://schumann.pub which features **interactive demos** of the following:

Vision and Language Navigation

Navigation Instructions Generation

Thank You!

Thank You!

References I

Ainsworth, Samuel K., Jonathan Hayase, and Siddhartha S. Srinivasa (2022). "Git Re-Basin: Merging Models modulo Permutation Symmetries". In: *ArXiv* abs/2209.04836.

Bansal, Gagan, Besmira Nushi, Ece Kamar, Dan Weld, Walter Lasecki, and Eric Horvitz (Jan. 2019). "Updates in Human-AI Teams: Understanding and Addressing the Performance/Compatibility Tradeoff". In: *AAAI Conference on Artificial Intelligence*. AAAI.

Cai, Deng, Elman Mansimov, Yi-An Lai, Yixuan Su, Lei Shu, and Yi Zhang (2022). "Measuring and Reducing Model Update Regression in Structured Prediction for NLP". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho.

FitzGerald, Jack G. M. et al. (2022). "MASSIVE: A 1M-Example Multilingual Natural Language Understanding Dataset with 51 Typologically-Diverse Languages". In: *ArXiv* abs/2204.08582.

References II

Ilharco, Gabriel, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt (2022). "Patching open-vocabulary models by interpolating weights". In: *ArXiv* abs/2208.05592.

Izmailov, P, AG Wilson, D Podoprikhin, D Vetrov, and T Garipov (2018). "Averaging weights leads to wider optima and better generalization". In: *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pp. 876–885.

Matena, Michael and Colin Raffel (2021). "Merging Models with Fisher-Weighted Averaging". In: *ArXiv* abs/2111.09832.

Rame, Alexandre, Matthieu Kirchmeyer, Thibaud Rahier, Alain Rakotomamonjy, patrick gallinari, and Matthieu Cord (2022). "Diverse Weight Averaging for Out-of-Distribution Generalization". In: *Advances in Neural Information Processing Systems*. Ed. by Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho.

Schumann, Raphael, Elman Mansimov, Yi-An Lai, Nikolaos Pappas, Xibin Gao, and Yi Zhang (2023). "Backward Compatibility During Data Updates by Weight Interpolation". In: *arXiv preprint arXiv:2301.10546*.

References III

Shen, Yantao, Yuanjun Xiong, Wei Xia, and Stefano Soatto (2020a). "Towards Backward-Compatible Representation Learning". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6367–6376.

— (2020b). "Towards backward-compatible representation learning". In: *CVPR 2020*.

Somepalli, Gowthami, Liam Fowl, Arpit Bansal, Ping Yeh-Chiang, Yehuda Dar, Richard Baraniuk, Micah Goldblum, and Tom Goldstein (2022). "Can neural nets learn the same model twice? investigating reproducibility and double descent from the decision boundary perspective". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13699–13708.

Srivastava, Megha, Besmira Nushi, Ece Kamar, Shital Shah, and Eric Horvitz (2020). "An Empirical Analysis of Backward Compatibility in Machine Learning Systems". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*. KDD '20. Virtual Event, CA, USA: Association for Computing Machinery, pp. 3272–3280.

Wortsman, Mitchell et al. (2022). "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time". In: *ICML*. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 23965–23998.

References IV

Xie, Yuqing, Yi-An Lai, Yuanjun Xiong, Yi Zhang, and Stefano Soatto (Aug. 2021). "Regression Bugs Are In Your Model! Measuring, Reducing and Analyzing Regressions In NLP Model Updates". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, pp. 6589–6602.

Yan, Sijie, Yuanjun Xiong, Kaustav Kundu, Shuo Yang, Siqi (Tiffany) Deng, Meng Wang, Wei Xia, and Stefano Soatto (2021). "Positive-congruent training: Towards regression-free model updates". In: *CVPR 2021*.