OPT2022: 14th Annual Workshop on Optimization for Machine Learning

The Importance of Temperature in Multi-Task Optimization

David Mueller DAM@JHU.EDU
Mark Dredze MDREDZE@CS.JHU.EDU
Nicholas Andrews NOA@CS.JHU.EDU
Department of Computer Science, Johns Hopkins University
Human Language Technology Center of Excellence, Johns Hopkins University

Abstract

The promise of multi-task learning is that optimizing a single model on multiple related tasks will lead to a better solution for all tasks than independently trained models. In practice, optimization difficulties, such as conflicting gradients, can result in *negative* transfer, where multi-task models which perform *worse* than single-task models. In this work, we identify the optimization temperature—the ratio of learning rate to batch size—as a key factor in negative transfer. Temperature controls the level of noise in each optimization step, which prior work has shown to have a strong correlation with generalization. We demonstrate that, in some multi-task settings, negative transfer may arise due to poorly set optimization temperature, rather than inherently high task conflict. The implication of this finding is that in some settings, SGD with a carefully controlled temperature achieves comparable, and in some cases superior, performance to that of specialized optimization procedures such as PCGrad, MGDA, and GradNorm. In particular, our results suggest that the significant additional computational burden of these specialized methods may not always be necessary. Finally, we observe a conflict between the optimal temperatures of different tasks in a multi-task objective, with different levels of noise promoting better generalization for different tasks. Our work suggests the need for novel multi-task optimization methods which consider individual task noise-levels, and their impact on generalization.

1. Introduction

Multi-task learning (MTL)—the simultaneous optimization of multiple related tasks—has a long history in machine learning [1]. Modern deep learning has enabled multi-task learning in new applications and settings, such as estimating a shared policy across multiple Atari games [8], or segmenting and classifying regions of images to improve medical diagnoses [24] and autonomous driving [6]. Biasing a learner towards solutions that address multiple tasks can yield individual task predictors that generalize better to unseen data, and potentially mitigate underspecification [5] when considering a diverse but sufficiently related set of tasks.

Conceptually, multi-task learning benefits from considering multiple objectives jointly, but in practice these objectives may be at odds with one another, leading to multi-task models which generalize *worse* than single-task models, a phenomenon known as **negative transfer** [25]. To address negative transfer, prior work in multi-task learning has focused on mitigating **task conflict**: significant differences between the gradients of individual tasks [3, 26, 32, *inter alia*]. Although task conflict is a defining characteristic of multi-task learning relative to single-task learning, large amounts of conflict between task gradients can negatively impact optimization, and may signify an incompatibility between the tasks being learnt. This relationship remains poorly understood; how

much task conflict is too much, and even how to meaningfully measure task conflict, remain under explored questions.

Despite the open questions about the role of conflict in MTL, there are some cases where it is possible to identify when conflict will hurt generalization, such as when conflict between tasks yields a sub-optimal model from the perspective of a task's training objective. For example, high directional task conflict can cause SGD to get stuck in a poor local optima, preventing further minimization of the training loss [32]. In such a setting, negative transfer may arise due to tasks being under-fit, an artifact that is not specific to multi-task learning but is instead a result of *poor optimization*. Prior work in multi-task learning has focused on producing methods that attempt to mitigate negative transfer by preventing such artifacts of optimization. PCGrad [32] aims to prevent optimization from getting stuck in poor local minima, allowing training to continue minimizing the training loss of each task. Other methods like GradNorm and MGDA are motivated by the need to balance task losses, preventing any one task from dominated the learning trajectory, preventing other tasks from being under-fit. These methods share the perspective that negative transfer can be mitigated by preventing optimization from *under-fitting* or *over-fitting* the training objective.

In this work, we explore the role of gradient noise on negative transfer in multi-task learning. Specifically, we aim to understand when negative transfer may arise due to *poor temperature*, rather than significant task conflict. Recent studies on the dynamics of *single-task* stochastic gradient descent in neural networks have identified a relationship between generalization and the ratio of the learning rate and batch-size, termed the **temperature of SGD** [12, 23, 28, 29]. The temperature may be understood as regulating the level of noise in the gradient step taken during each iteration of SGD. Empirically, neural networks with small batches or large learning rates, i.e. *high temperatures*, have been shown to generalize better than low temperature models, despite minimizing the training loss equally well [7]. In the multi-task setting, we explore to what extent *negative transfer* may be attributed to poorly selected temperatures, rather than high amounts of task conflict. Our contributions are as follows: (1) We find that the *Uniform Average* objective, a common objective in multi-task learning, can have a significant effect on the noise of the multi-task gradient, and demonstrate that accounting for this when setting the temperature mitigates negative transfer in some common multi-task benchmarks; (2) We observe that, for a single multi-task objective, different temperatures may favor generalization on different tasks. This finding suggests a novel from of multi-task conflict, at the *noise level* of the gradient. Overall, our work highlights the importance of the connection between temperature and negative transfer when optimizing multi-task models.

2. Background & Related Work

2.1. Negative Transfer & Multi-Task Optimizers

To properly address negative transfer, prior work in multi-task learning has largely focused on producing *specialized multi-task optimizers* [SMTOs, 18] which aim to mitigate task conflict during training. Although these methods are largely motivated by *generalization* performance, i.e. negative transfer, they often directly target ways to improve optimization of the *training objective*. For example, MGDA [26] aims to find a pareto-stationary solution in the training landscape, such that no task loss can be reduced without increasing another. GradNorm [3] aims to learn task weights such that tasks are learned at similar rates, so no task can dominate the optimization objective. Yu et al. [32] show that, under certain conditions, multi-task optimization can converge to poor local minima, and propose PCGrad as a method to prevent such conditions. The hypothesis of the SMTOs

listed above is that the training objective of the tasks being considered are tied to negative transfer; better minimizing each task's training loss will reduce negative transfer.

However, directly mitigating conflict between the direction and magnitude of task gradients may not be necessary to mitigate negative transfer. Lin et al. [21] showed that randomly weighting tasks throughout training could achieve impressively strong multi-task models, which outperform several conflict mitigation methods. Recently, Kurin et al. [18] found that many SMTOs may behave as regularizers; replacing conflict mitigation with L2 regularization can often achieve comparable results. In this work we examine the extent to which negative transfer may arise to due an *improper amount of noise* in the gradient step, rather than significant conflict in gradient directions or magnitudes. Importantly, the connection between noise and generalization is not observable from the perspective of the training objective, suggesting the need for a new perspective in multi-task optimization.

2.2. The Temperature of SGD & The Linear Scaling Rule

It has long been thought that a small batch-size is key to neural network generalization [20]. Keskar et al. [16] identified that large-batch models converge to "sharper" minima than small-batch models, which had previously been tied to worse generalization [9]. However, Goyal et al. [7], Keskar et al. [16] demonstrated the *linear scaling rule*: by scaling the learning rate linearly with respect to the batch-size, one could achieve good generalization with large batch-sizes. Smith and Le [28] investigated SGD dynamics as a stochastic differential equation and found that the underlying variable influencing the convergence to sharp or flat minima was not batch-size, but rather the noise-scale of SGD, also known as the *temperature of SGD*, characterized as $T = \frac{\epsilon}{B}$, where ϵ is the learning rate and B is the batch-size. They argued that successful generalization depends on finding an optimal T for the given problem, which has been corroborated in other studies [13, 23, 27]. While some works have justified this connection theoretically, demonstrating the preference of a noisy optimizer towards flat minima [11, 27, 31], other work has empirically shown that common measurements of flatness are not necessarily correlated with temperature *or* generalization [14]. Despite our inconsistent understanding of this phenomena, it is clear that the connection between T and generalization plays an important role in deep learning.

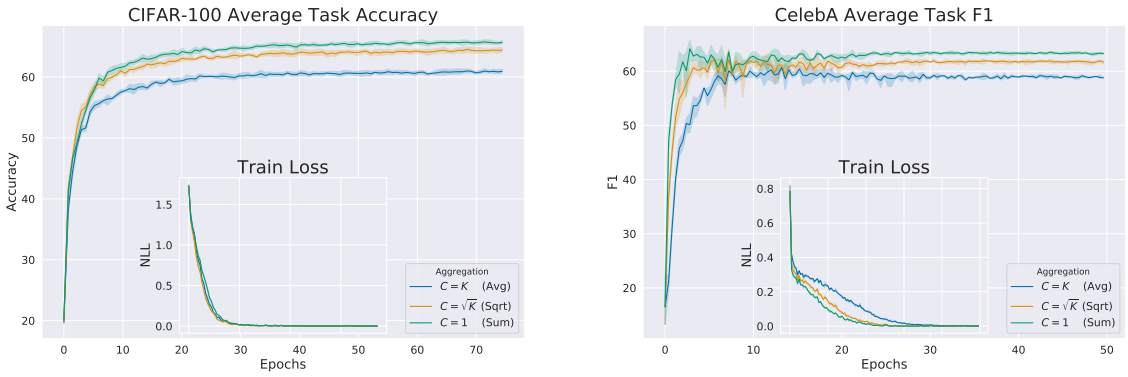
3. The Optimization Temperature of Multi-Task Problems

3.1. The Noise of the Multi-Task Objective

In multi-task learning we assume K tasks each consisting of labeled examples $\{(\mathbf{x}_i^{(k)}, y_i^{(k)})\}_{i=1}^{N_k}$ for $k = 1, 2, \ldots, K$, where N_k is the number of examples for the kth task. We assume that the input space $\mathcal{X}$ is the same for all tasks, and each task has a *task-specific* output space $\mathcal{Y}_k$. Tasks are modeled using a neural network $f_\theta : \mathcal{X} \to \mathbb{R}^D$ where the parameters θ are shared across all K tasks. To obtain task-specific predictions, we introduce projections h_{ϕ_k} such that $h_{\phi_k} \circ f_\theta : \mathcal{X} \to \mathcal{Y}_k$. We write the loss function of the kth task as ℓ_k and let $\Theta = (\theta, \phi_1, \ldots, \phi_K)$. Then the *uniform multi-task loss* (UMTL) is defined as:

$$\mathcal{L}(\Theta) = \frac{1}{CB} \sum_{k=1}^{K} \sum_{i=1}^{B} \ell_k((h_{\phi_k} \circ f_\theta)(x_i^{(k)}), y_i^{(k)}) \tag{1}$$

where $x_i^{(k)}$ are sampled uniformly at random with replacement from $\mathcal{D}_k$ with a mini-batch size of B. C is a hyper-parameter which determines the *scaling* of the UTML. The role of C is of primary



(*a*) (*b*)

Figure 1: Different settings of C can yield solutions which equally minimize the training objective, but generalize in significantly different ways. Here we see multi-task CIFAR-100 and CelebA models trained with the Uniform Multi-Task Objective. A setting of $C = K$ (averaging task-gradients) incurs significantly higher negative transfer than a setting of $C = 1$ (summing task gradients), despite either method minimizing the training objective.

interest in this work, as it has a significant effect on the *covariance* of the multi-task gradient. Let $G_{k,\theta}$ be a random variable representing the gradient of a single sample $\nabla_\theta \ell_k(x) : x \sim \mathcal{D}_k$ given model parameters θ – then the noise of task k is described by $\Sigma(G_{k,\theta})$. Let $G_\theta = \frac{1}{C}\sum_{k=1}^{K} G_{k,\theta}$ represent the *multi-task gradient* as a sum of random variables. Then, assuming $x \sim \mathcal{D}_k$ is sampled independently from all other k, the multi-task noise can be written as

$$\Sigma(G_\theta) = \frac{1}{C^2}\Sigma(G_{1,\theta}) + \cdots + \frac{1}{C^2}\Sigma(G_{K,\theta}) = \frac{1}{C^2}\sum_{k=1}^{K}\Sigma(G_{k,\theta}) \tag{2}$$

We therefore have the following relationship: the multi-task gradient covariance scales with the *number of tasks* and scales inverse-quadratically with C. When $C = K$, yielding the *Uniform Average* multi-task objective, the noise of the gradient is the sum of individual task variances divided by K^2. A more principled choice may be $C = \sqrt{K}$, which would result in the multi-task noise being the *average single-task noise*. Nearly all multi-task work considers some variant of the UMTL as a baseline, often with $C = K$ or $C = 1$, although few provide justification for their choice.

3.2. The Uniform Multi-Task Objective

We now empirically examine the effects of C on negative transfer in multi-task models. The Uniform Average Multi-Task objective ($C = K$) is a common objective in multi-task literature [15, 21, 26, *inter alia*], often used to motivate the need for more sophisticated methods which mitigate conflict between tasks. However, $C = K$ reduces the gradient noise to a fraction of the average task covariance; here we ask if this can be responsible for significant negative transfer. In Figure 1 we plot the training loss and validation performance of multi-task ResNet-18 models on CIFAR-100 and CelebA for $C = 1, \sqrt{K}$, and K, with the optimal *single-task* learning rate and batch-size.[1]

1. For details regarding the models and hyper-parameters for each setting, please see App. A.

From the perspective of the training objective, C has little impact on optimization, as all models equivalently minimize the training loss. However, the uniform *average* multi-task objective exhibits significantly higher "negative transfer" than the uniform *sum* multi-task objective ($C = 1$). Interestingly, this suggests a *linear scaling rule* for multi-task learning, e.g. the gradient should scale linearly with the number of tasks. Because we use the optimal single-task temperature, our results suggest that setting $C = 1$ significantly reduces the need for broad hyper-parameter sweeps when the best average single-task optimization temperature is already known.

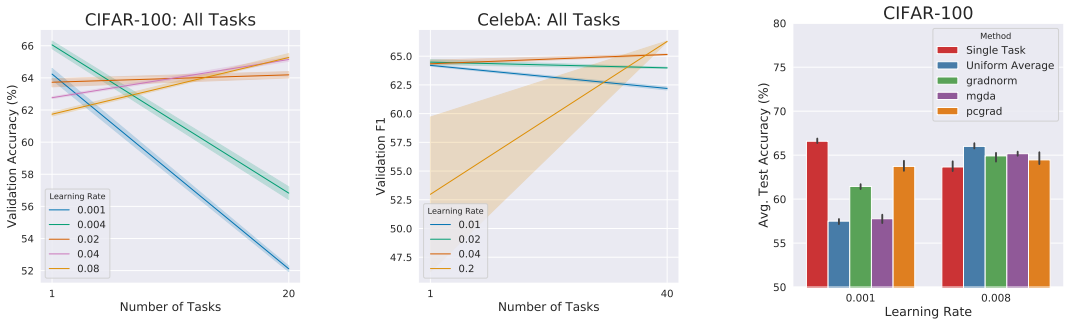
In Figure 2, we examine negative transfer in CIFAR-100 (a) and CelebA (b) ResNet18 models using the uniform *average* objective ($C = K$). We plot average task test performance in single- and multi-task models for a range of optimization temperatures (in App. C we provide similar plots for individual tasks). We find that temperatures which are optimal for single-task models exhibit strong negative transfer in multi-task settings. However, scaling the temperature by the number of tasks, which has the same effect as setting $C = 1$, again largely mitigates negative transfer. Indeed, in Figure 2 (c) we compare single-task models to multi-task models trained with the Uniform Average objective, as well as multi-task models trained with the PCGrad [32], MGDA [26], and GradNorm [3] SMTOs, on CIFAR-100. We find that, for the optimal single-task temperature (a low learning rate), the uniform average objective exhibits strong negative transfer, and SMTOs are highly beneficial. However, when the temperature is appropriately scaled, the uniform average objective is more competitive, generalizing comparably to the SMTOs. We provide additional discussion and results for CelebA and MNISTS datasets in App. B.

Our results **do not** suggest that conflict is not a problem in multi-task learning, nor that SMTOs are unnecessary—indeed we see that they still improve performance on CelebA—but rather suggest that the *role* of conflict in negative transfer has been overestimated due to a lack of attention given to other artifacts of optimization such as temperature. Such a finding is inline with the results of Kurin et al. [18], who similarly show that SMTOs may provide benefits largely as a form of regularization, a benefit which can be replicated through stronger L2 regularization using a vanilla SGD optimizer.

4. Different Tasks Have Different Optimal Optimization Temperatures

In Section 3 we explore how optimization temperature may cause negative transfer from the perspective of the *average task* performance. However, a more fine-grained view of task generalization across different temperatures (App. C) reveals a surprising phenomenon: different temperatures may generalize better or worse for different tasks. This is surprising because all models share the exact same objective (the uniform average loss) and minimize the objective equally well. We corroborate this finding on the Cityscapes dataset, a 3-task dataset with significant noise disparities, using pre-trained in Figure 3. Here we see that each task individually benefits the most from a unique temperature, despite the objective remaining constant; depth estimation generalizes best with a learning rate of 0.002, yet instance segmentation generalizes best with a learning rate of 0.005. We emphasize again that each model minimizes the training objective equivalently. However, the *noise* of the gradient step clearly biases the model towards *generalization* on certain tasks over others.

This result also suggests that there exists a *conflict* between the preferred temperature of optimization of tasks in a multi-task model. In principle, the trade-off between task generalization is not desirable; it would be preferable to find a single solution that exhibits equally optimal generalization properties for all tasks simultaneously. To the best of our knowledge, we are the first to note on this particular type of *noise conflict* between tasks in a multi-task setting. Our results suggest the



(*a*) (*b*) (*c*)

Figure 2: **(a) & (b):** On CIFAR-100 and CelebA, negative transfer can be associated with a poorly set optimization temperature, rather than an inadequate training objective. Average task generalization can attain comparable or even better performance than single-task performance when the temperature is scaled by an appropriate factor. On CelebA, the highest temperature is so sub-optimal for single-task models that they often perform near random chance. **(c):** At optimal single-task temperatures, SMTOs often outperform the uniform average baseline. However, at higher temperatures, the uniform average baseline can attain competitive performance to most SMTOS.

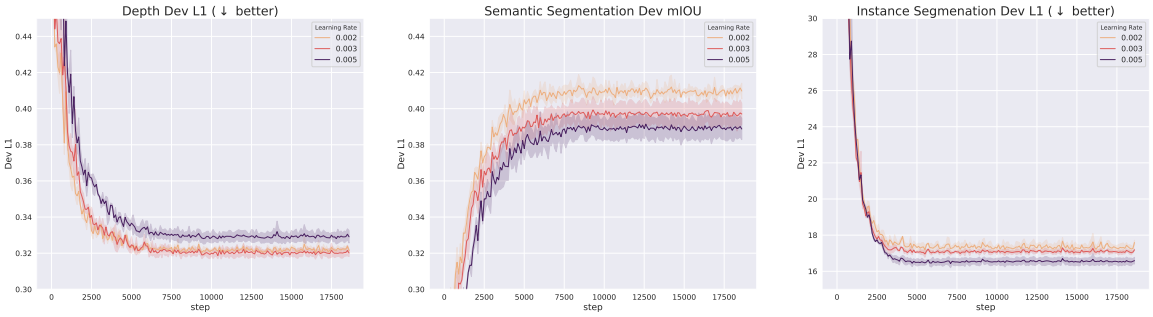


Figure 3: Different temperatures promote better generalization for different tasks in the Cityscapes dataset. A learning rate of 0.002 finds the best performance for semantic segmentation, yet a learning rate of 0.005 finds the best generalization from the perspective of instance segmentation. We note that all models achieve equal loss from the perspective of the training objective; thus different noise levels are biased towards *generalization* on different tasks.

need for novel multi-task optimizers which can address disparities in the optimal temperatures for different tasks in a single model.

5. Conclusion

In this work we identify temperature as a key factor in negative transfer in several common multi-task benchmarks. We show that, when the optimization temperature is carefully considered, negative transfer can be mitigated through the relationship between optimization temperature and generalization in deep learning. Our results highlight the importance of considering different factors of optimization when evaluating the causes of negative transfer; in particular, we show that vanilla

SGD with the uniform average objective is a stronger baseline that previously believed, a finding corroborated by Kurin et al. [18]. Finally, we observe a conflict between the optimal temperatures for different tasks in a multi-task model, which yields ways to bias multi-task optimization towards certain tasks without changing the objective. More importantly, however, it highlights a novel direction for future work in multi-task optimization, demonstrating the need for methods which can mitigate discrepancies between the optimal noise-levels of each task in a single model.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful feedback and comments on this work. We also thank Steven Reich for his input and contributions to earlier versions of this work. This work was supported, in part, by the Human and Language Technology Center of Excellence at Johns Hopkins University.

References

[1] R. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *ICML*, 1993.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[3] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. volume 80 of *Proceedings of Machine Learning Research*, pages 794–803, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/chen18a.html.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yian Ma, Cory McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification presents challenges for credibility in modern machine learning, 2020.

[6] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2020. doi: 10.1109/TITS.2020.2972974.

[7] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

[8] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. Multi-task deep reinforcement learning with popart. Technical report, DeepMind, 2019. URL https://www.aaai.org/ojs/index.php/AAAI/article/view/4266.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, January 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

[10] Po-Chen Hsieh and Chia-Ping Chen. Multi-task learning on MNIST image datasets, 2018. URL https://openreview.net/forum?id=S1PWi_lC-.

[11] W. Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, J. K. Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations, 2020.

[12] Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

[13] Stanisław Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd, 2018.

[14] Simran Kaur, Jeremy Cohen, and Zachary C. Lipton. On the maximum hessian eigenvalue and generalization, 2022. URL https://arxiv.org/abs/2206.10654.

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima, 2017.

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[18] Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and M Pawan Kumar. In defense of the unitary scalarization for deep multi-task learning. *arXiv preprint arXiv:2201.04122*, 2022.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] Yann LeCun, Leon Bottou, Genevieve Orr, and Klaus-Robert Müller. Efficient backprop. 08 2000.

[21] Baijiong Lin, Feiyang Ye, Yu Zhang, and Ivor W. Tsang. Reasonable effectiveness of random weighting: A litmus test for multi-task learning, 2021. URL https://arxiv.org/abs/2111.10603.

[22] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[23] Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[24] Sachin Mehta, Ezgi Mercan, Jamen Bartlett, Donald Weaver, Joann Elmore, and Linda Shapiro. Y-Net: Joint Segmentation and Classification for Diagnosis of Breast Biopsy Images. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2018.

[25] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

[26] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 527–538. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7334-multi-task-learning-as-multi-objective-optimization.pdf.

[27] Samuel Smith, Erich Elsen, and Soham De. On the generalization benefit of noise in stochastic gradient descent. In *International Conference on Machine Learning*, pages 9058–9067. PMLR, 2020.

[28] Samuel L. Smith and Quoc V. Le. A bayesian perspective on generalization and stochastic gradient descent. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJij4yg0Z.

[29] Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

[30] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[31] Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wXgk_iCiYGo.

[32] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.

Appendix A. Experiment & Dataset Details

A.1. CelebA

Data: The CelebA dataset [22] is a dataset consisting of over 200,000 images of celebrity faces. Each image is annotated with 40 distinct, binary characteristics, ranging from objective facial attributes, such as nose size and hair color, to highly subjective attributes, such as whether or not the celebrity is attractive or chubby. In a multi-task setting, each attribute is treated as a separate binary classification task. We use CelebA's standard training, validation and test splits, which consist of $162,770$ training images, $19,867$ validation images, and $19,962$ test images. We additionally resize all images to RBG images of size 64×64. CelebA severely suffers from class imbalance – several classes are largely positive or negative. As such, we often opt to report F1, rather than accuracy, as it more accurately represents the models true generalization capabilities. **Model:** For our CelebA experiments we use a ResNet-18 architecture. In the multi-task setting, all but the ultimate layer parameters are shared across all tasks. For task-specific classification we take the output of the penultimate layer and feed it to task-specific linear binary classifier. In the single-task setting, each task is learnt with an independent ResNet18 model. The model is trained with a batch-size of 256 for 25 epochs. We found the optimal temperature in the single-task setting to be a batch-size of 256 and a learning rate of 0.02, and using SGD with a momentum coefficient of 0.9.

A.2. CIFAR-100

Data: The CIFAR-100 dataset [17] consists of 60,000 32×32 RGB images, each of which belongs to a *coarse* and *fine-grained* class. In the multi-task setting, each *coarse-grained* class is treated as a separate task, which is itself a multi-class classification problem among the *fine-grained* classes that make up that coarse-grained task. In this sense, CIFAR-100 is a *multi-source* multi-task dataset, in that each task has its own domain of images associated with it, and no image is labeled for more than a single task. Each coarse-grained class is associated with exactly 5 distinct fine-grained classes, and thus each task is a 5-class classification problem. The dataset is split such that each task (coarse-grained class) has 2,500 training images and 500 test images. This corresponds to 500 training images and 100 test images per *fine-grained* class. We take a random sample of 500 images per task from the training set to construct a validation split. This results in 40,000 training images, 10,000 validation images, and 10,000 test images in the full multi-task setting. **Model:** For our CIFAR-100 experiments we use a ResNet-18 model. In the multi-task setting we take the output of the penultimate layer and feed it to task-specific linear softmax classifiers, sharing all layers below across all tasks. In the single-task setting each task is learnt with an full, independent ResNet18 model. All models are trained with a batch-size of 16 for 75 epochs. We found the optimal temperature in the single-task setting to be a batch-size of 16 with a learning rate of 0.004, and a learning rate decay of 0.99 using SGD with a momentum coefficient of 0.9.

A.3. MNISTS

Data: The MNISTS dataset [10] consists of 3 MNIST-like datasets, each consisting of 50,000 28×28 greyscale training images, and 10,000 validation and test images. Each dataset is a multi-class classification dataset. MNIST [19] is the canonical handwritten digit recognition dataset, and consists of classifying images of individual digits as $0 - 9$. FashionMNIST [30] is similarly a 10-class classification problem, consisting of greyscale images which must be classified as one

of T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot. Finally, NotMNIST (https://www.kaggle.com/lubaroli/notmnist) is a dataset consisting of images of individual letters, pulled from a large variety of fonts. The dataset is a 26-class classification dataset where the task is to classify each image's letter, a-z. **Model:** We use the LetNet architecture for our MNIST experiments. This model shares 2 convolutions layers, followed by a shared feed forward layer. The representations from this model are fed into 2 task-specific feedforward layers, the first with a relu activation, and the second with a softmax output as the classifiers prediction. Our model uses a dropout rate of 0.5 in all layers. We found the optimal temperature in the single-task setting to be a batch-size of 32 and a learning rate of 0.001, using SGD with a momentum coefficient of 0.9.

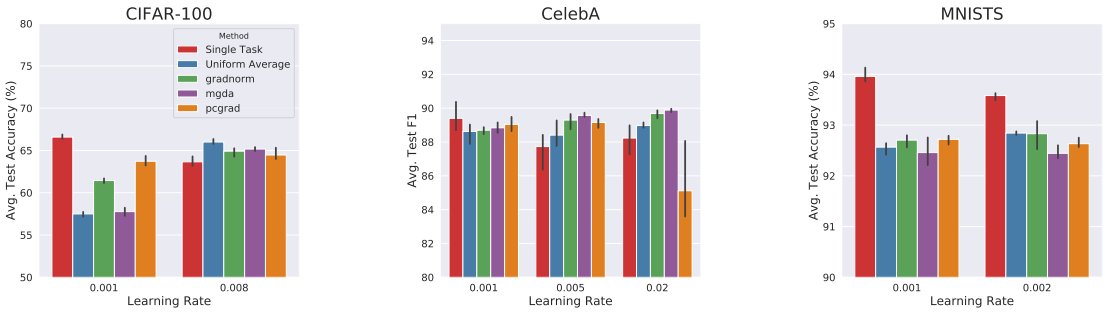
A.4. Cityscapes

Data: The Cityscapes dataset [4] comprises 5,000 images of urban streets. Each image contains pixel-level annotations for semantic segmentation, instance segmentation of people and cars, and depth ('disparity') labels; these are considered to be three separate tasks. We train on the training set of 2,975 images and report each task metric on the validation set of 500 images after each epoch. All images are downsized to 128×256. For the instance segmentation task we follow the setup of Sener and Koltun [26] and train and evaluate on the proxy task of estimating the center of mass of each pixel. **Model:** Our experiments on Cityscapes uses a DeepLabV3 model Chen et al. [2], which consists of a pre-trained dilated ResNet50 model, followed by task-specific Atrous Spatial Pyramid Pooling layers. We use a batch-size of 16, and we optimize with Adam with default hyperparameters.

Appendix B. SMTOs on CelebA and MNISTS

Here we provide additional results and discussion on comparing SMTOs to the uniform average objective across a small set of temperatures in Figure 4. On CIFAR-100, we see in the low-temperature regime that the uniform average multi-task objective incurs heavy negative transfer. Additionally, in this regime conflict mitigation methods are very impactful, and can mitigate negative transfer significantly. However, in the high temperature regime, the uniform average objective suffers very little negative transfer. In this regime, SMTOs provide very little benefit over the uniform average objective. On the MNISTS dataset, we see a similar story; SMTOs are very beneficial over the uniform average objective in low temperature regimes, but their benefit drops in high temperature regimes.

In the CelebA dataset we see a different story – here, SMTOs retain large benefits over the uniform average objective even in the high temperature regime. Kurin et al. [18] found that typical models for CelebA over-fit, and that SMTOs could act as an explicit form of regularization (i.e. they prevent minimization of the training loss). Regardless, CelebA provides a concrete setting where SMTOs can improve performance over the uniform average objective even when the temperature is treated appropriately; This suggests that conflict mitigation is not entirely unnecessary in multi-task learning. However, taken all together, this section suggests that we should be careful when employing expensive conflict mitigation methods before making sure that the baseline (UMTL) is appropriately optimized.



(*a*) (*b*) (*c*)

Figure 4: Average task test results on CIFAR-100 (a), CelebA (b), and MNISTS (c) datasets for single-task models, uniform average multi-task models, and models trained with GradNorm, MGDA, and PCGrad, over a small set of temperatures. Note that, due to the high computational cost of running SMTOs on the full CelebA dataset, we consider a random subset of tasks here. We see that in low temperature regimes SMTOs largely outperform the uniform average baseline. However, when the temperature is set appropriately high, the benefits of SMTOs on MNISTS and CIFAR-100 largely disappears.

Appendix C. Full Task Scaling Plots for CIFAR-100 & CelebA

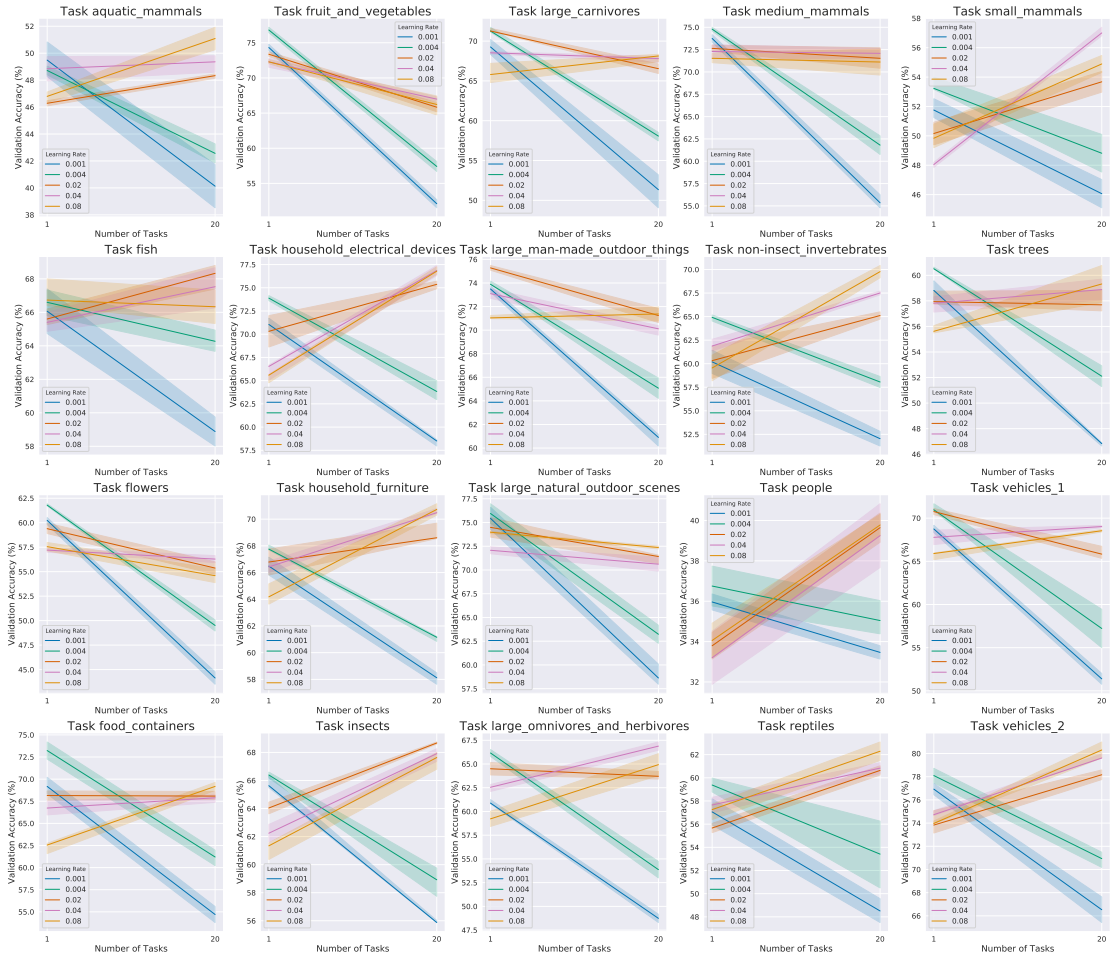


Figure 5: Full results from every task considered in CIFAR-100, in single- and multi-task settings. We see that the single-task optimal temperature is significantly sub-optimal in the multi-task setting, compared to temperatures scaled by a factor of 10 to 20.

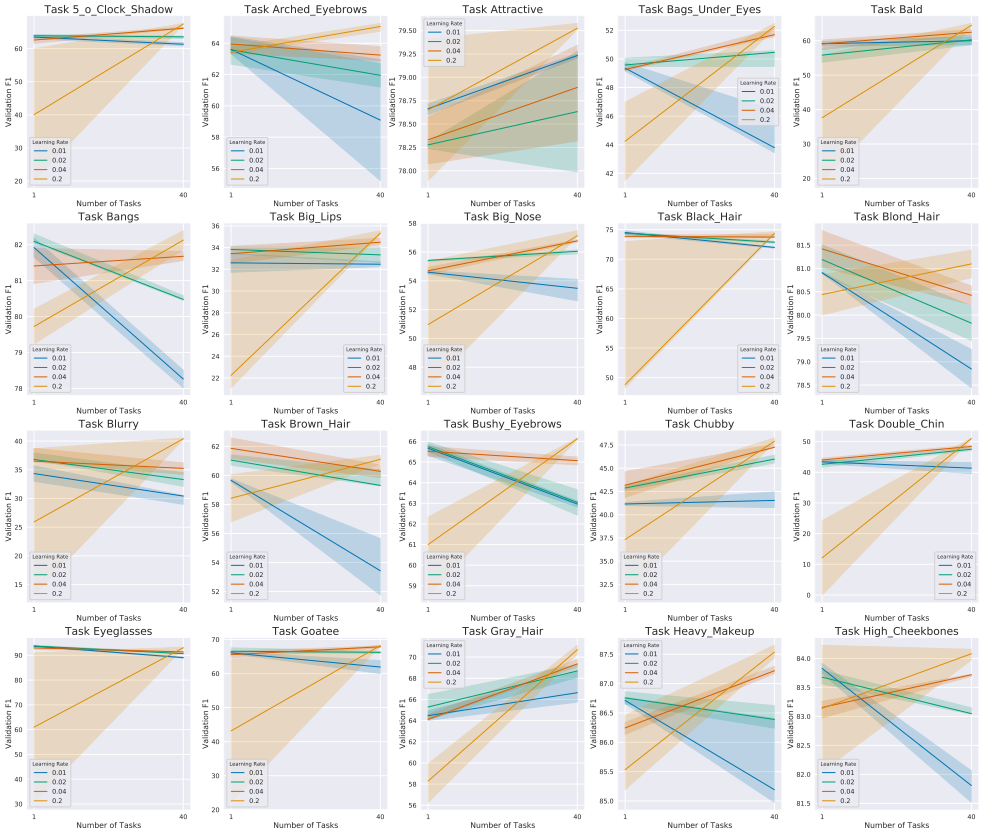


Figure 6: Full results from every task considered in CelebA (tasks 1-20), in the single- and fully multi-task settings. Similar to CIFAR-100 and MNISTS, we see a significant amount of homogeneity in these graphs, suggesting that independent tasks have similar levels of gradient noise, and that averaging tasks together uniformly lowers gradient noise of SGD.

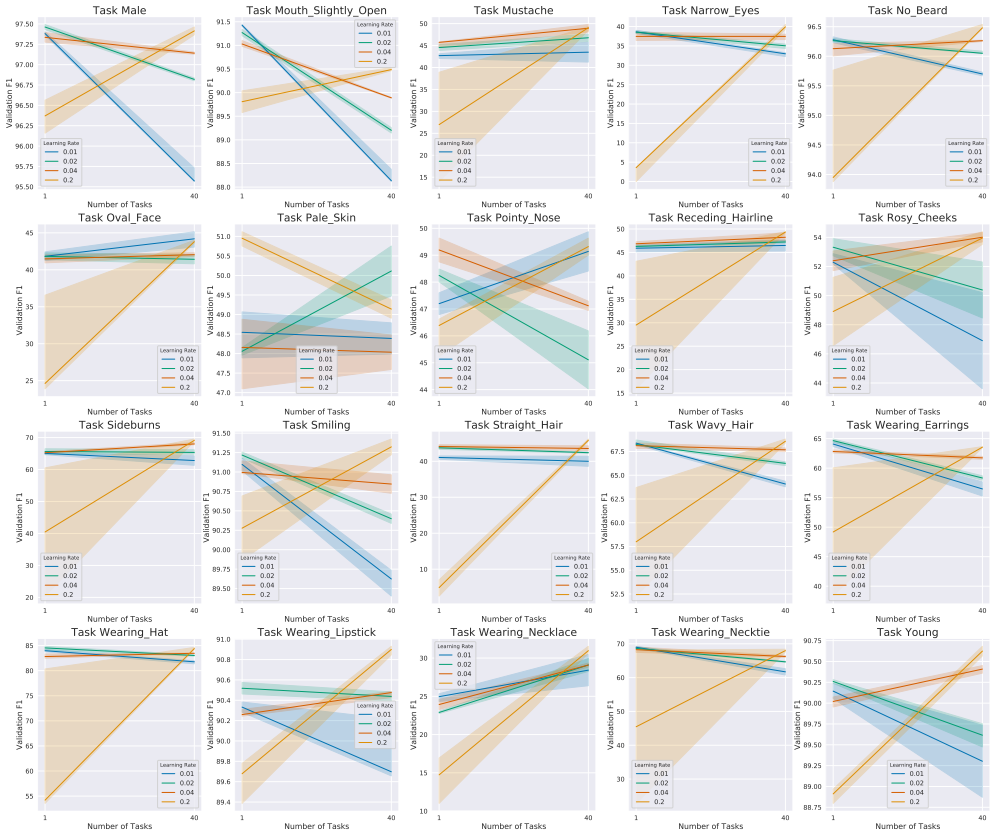


Figure 7: Full results from every task considered in CelebA (tasks 21-40), in the single- and fully multi-task settings. Similar to CIFAR-100 and MNISTS, we see a significant amount of homogeneity in these graphs, suggesting that independent tasks have similar levels of gradient noise, and that averaging tasks together uniformly lowers gradient noise of SGD.