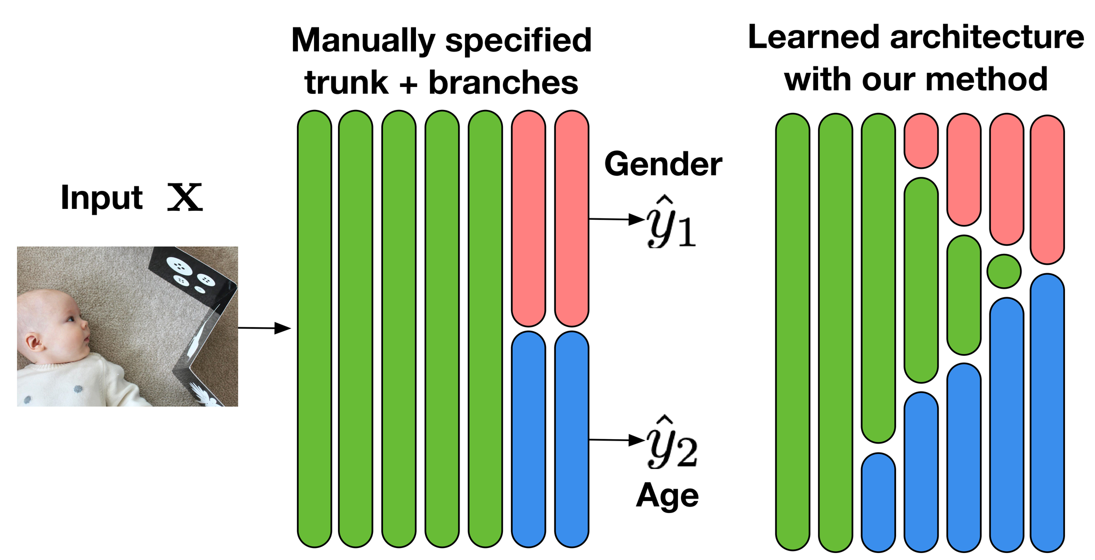
Stochastic Filter Groups for Multi-Task CNNs
Learning Specialist and Generalist Convolution Kernels

Felix Bragman*, Ryutaro Tanno*, Sébastien Ourselin, Daniel C. Alexander, M. Jorge Cardoso

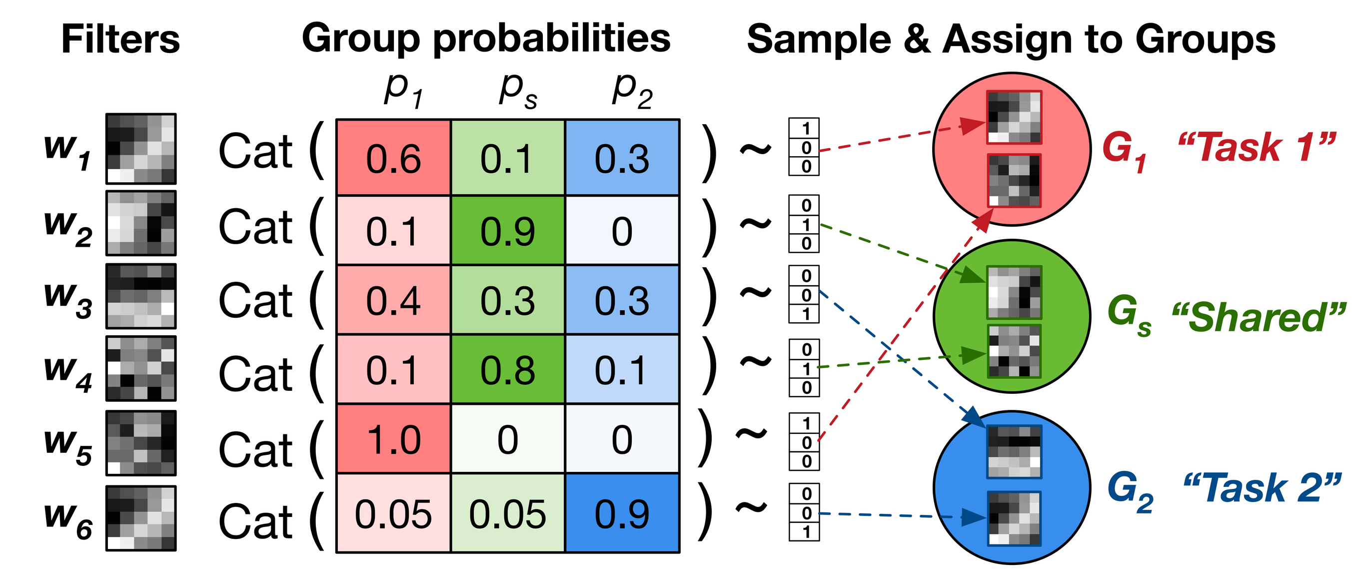
1 Summary

- Benefits of **multi-task learning (MTL)** depend on the nature of feature sharing and the network architecture
- **Problem:** these architectures are **manually pre-specified** which can be **suboptimal**
- **Solution:** we propose Stochastic Filter Groups (SFG); a principled mechanism to learn the amount of feature sharing and separation between tasks
- We show the benefits of SFGs in two multi-task problems



2 What are Stochastic Filter Groups?

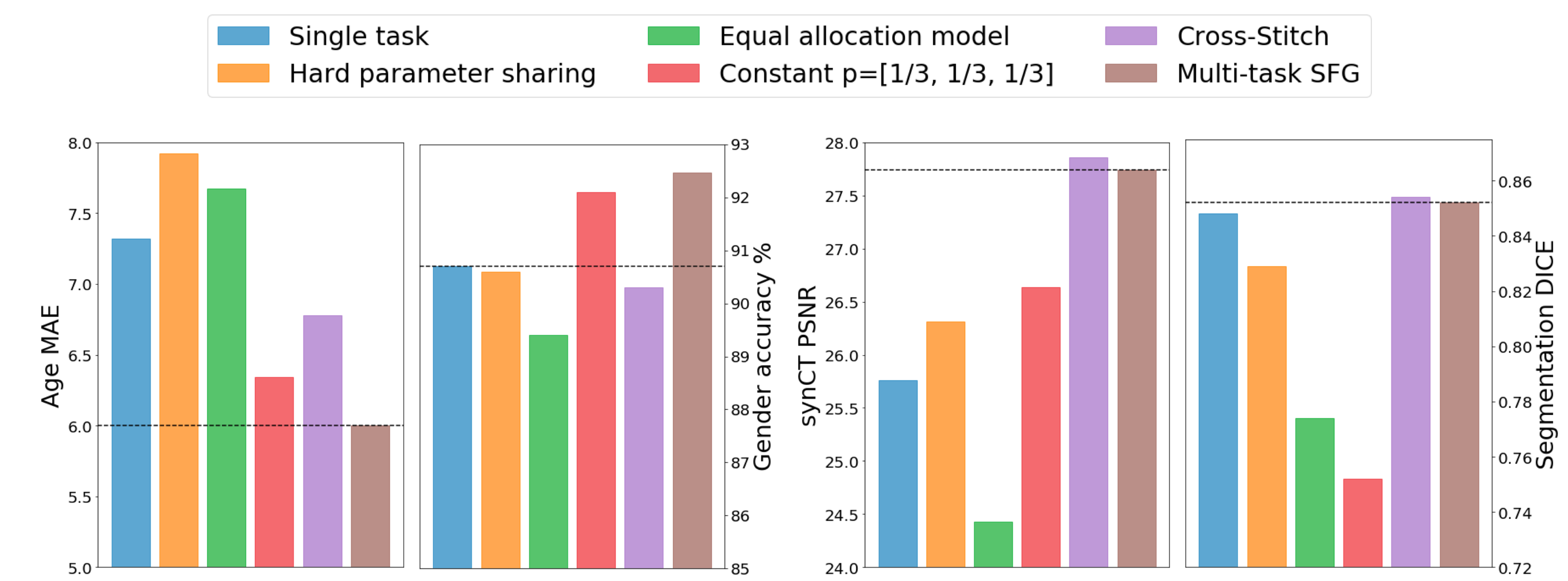
- **Core idea:** cluster convolution kernels into task specific and shared groups in each layer of a CNN
- We define **task-specific groups** as the set of filters that are only updated to minimise corresponding task losses , while the **shared group** follows the same logic but is leaned to optimise **all** task
- Jointly learn the grouping and convolution kernel weights



Definitions:
G_1 := filters updated w.r.t task-1 loss
G_S := filters updated w.r.t task-1 loss and task-2 loss
G_2 := filters updated w.r.t task-2 loss

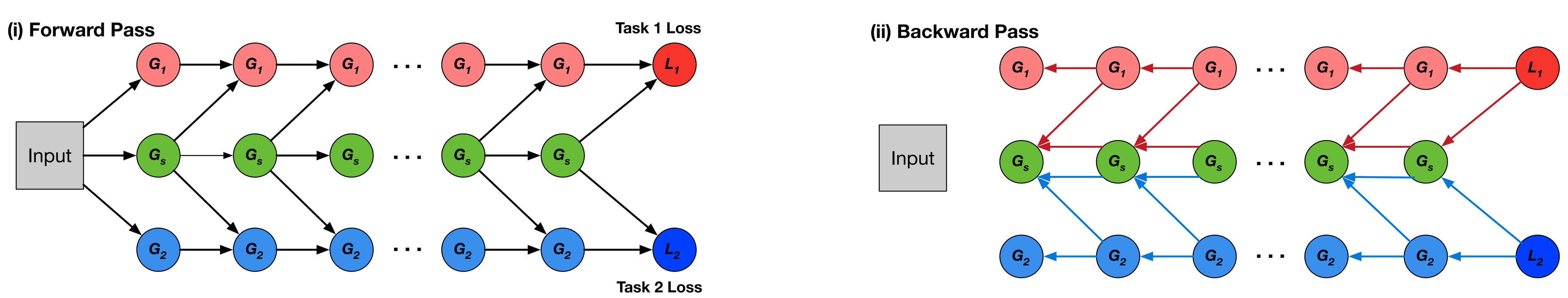
4 SFGs improve MTL performance

- Learning the allocation of kernels in MTL improves task performance
- We compared SFG-MTL architectures against: a) single-task networks, hard-parameter sharing networks, MTL networks with no learned allocation and Cross-Stitch networks [Misra et al, 2016]
- Dataset 1: UTKFace - age and gender prediction
- Dataset 2: Prostate MRI – CT synthesis and segmentation



3 Optimisation of network weights and kernel grouping

① **Sparse routing of features for desired gradient flow**



② **Filter assignment as T+1 group drop-out**

- Cast learning of grouping probabilities and filter weights as variational inference [Gal et al. 2018]
- Extended binary dropout to categorical distributions
  - Minimise the following variational objective:

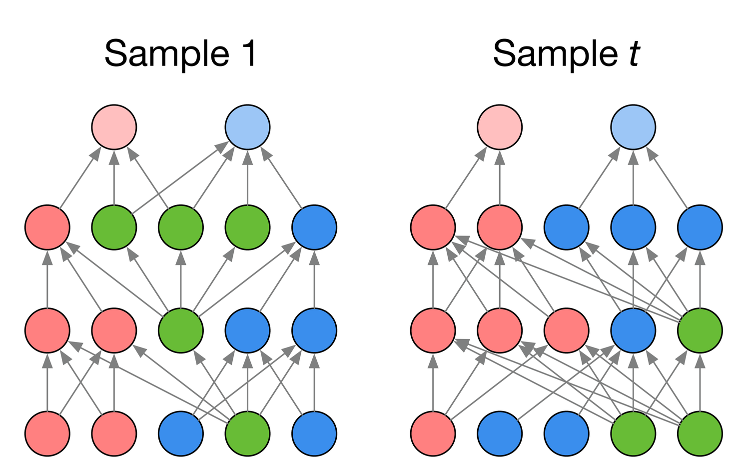
$$\mathcal{L}_{\mathrm{MC}}(\phi)=-\frac{N}{M}\sum_{i=1}^{M}\Big[\log p\Big(y_i^{(1)}|\mathbf{x}_i,\mathcal{W}_i\Big)+\log p\Big(y_i^{(2)}|\mathbf{x}_i,\mathcal{W}_i\Big)\Big]$$

$$+\lambda_1\cdot\sum_{l=1}^{L}\sum_{k=1}^{K_l}||\mathbf{M}^{(l),k}||^2-\lambda_2\cdot\sum_{l=1}^{L}\sum_{k=1}^{K_l}\mathcal{H}(\mathbf{p}^{(l),k})$$

③ **Continuous relaxation using Gumbel-Softmax [Jang et al. 2016]**

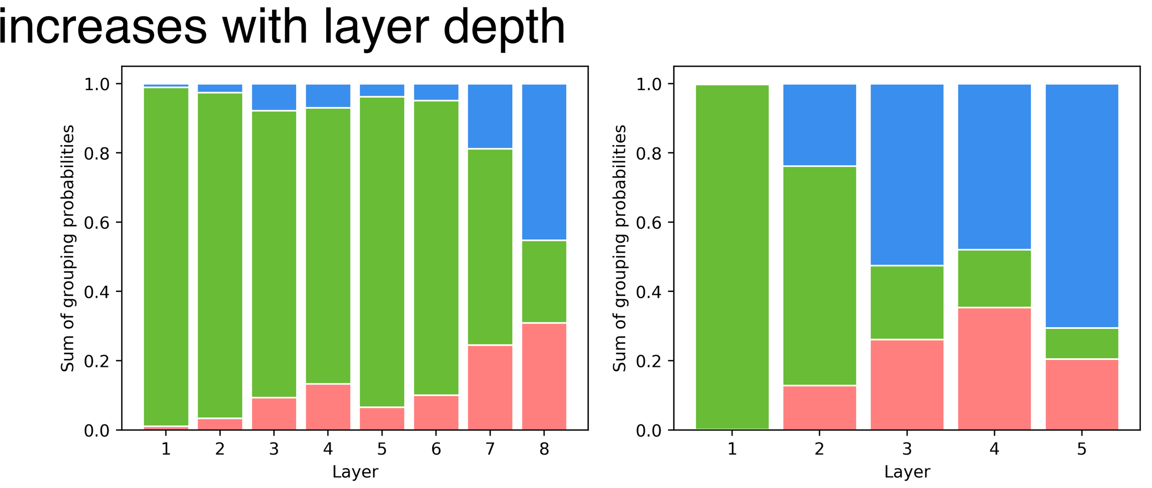
- First two terms have **zero** gradients w.r.t assignments probabilities **p**

$$\mathrm{Softmax}\left([g_i+\log p_i]/\tau\right) \quad g\sim\mathrm{Gumbel}(0,1)$$

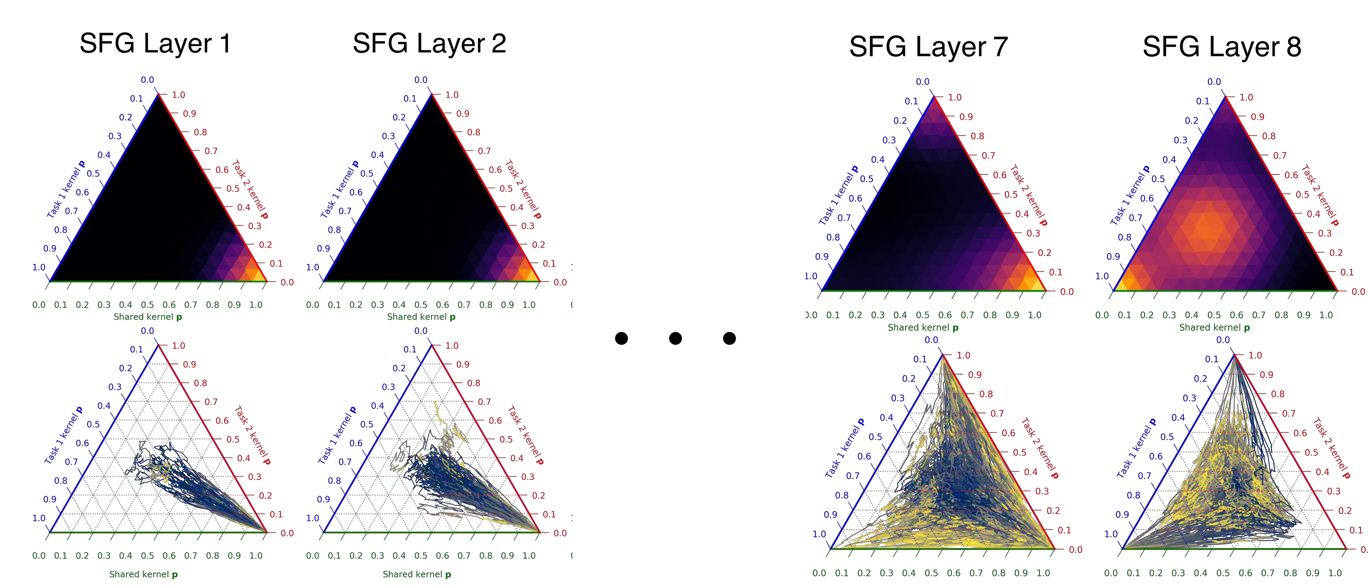


5 Qualitative Results

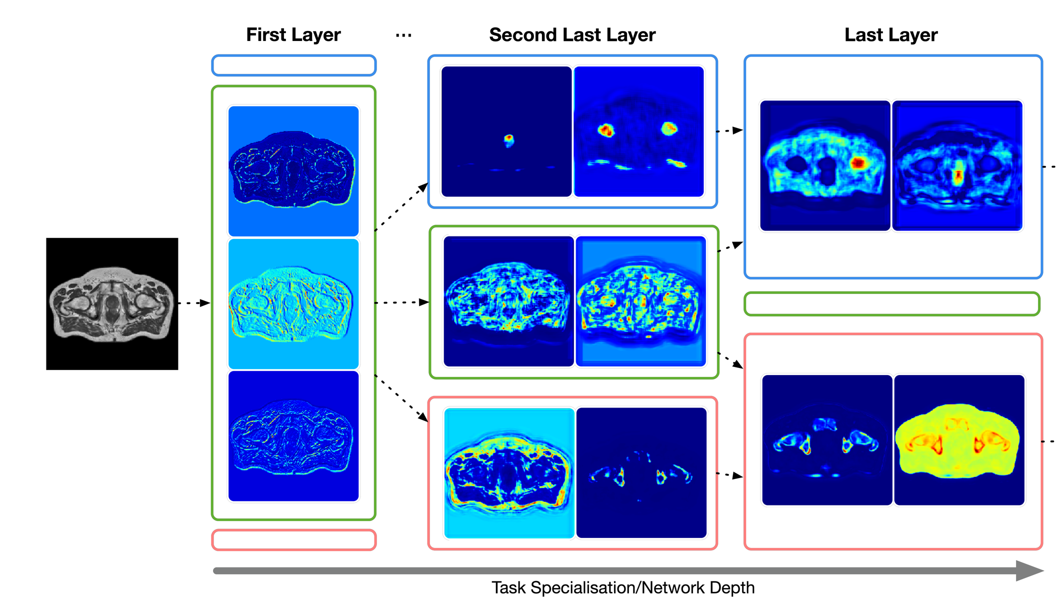
- We can visualise the learned allocation of kernels in a CNN with SFG modules
- Across both datasets, the ratio of task-specific groups increases with layer depth



- Density plots of probabilities **p** illustrate learned grouping
- Training trajectories reveal some kernels converge faster to corresponding groups



- Activation maps for kernels with low grouping entropy confirm increasing task specialisation



- Maps for kernels with high grouping entropy show uncertainty in feature utility for maximising task performance