# Eliciting and Understanding Cross-Task Skills with Task-Level Mixture-of-Experts

Findings of EMNLP 2022



Qinyuan Ye    Juan Zha    Xiang Ren

Department of Computer Science & Information Sciences Institute
University of Southern California
http://inklab.usc.edu

# Background: Massive Multi-task Learning

**training a model on a multi-task mixture**

Train

**helpful for tasks seen in the mixture**

Test

**Muppet**: Aghajanyan et al., 2021
**ExT5**: Aribandi et al., 2021

**helpful for generalizing to unseen tasks**

Test

**CrossFit**: Ye et al., 2021
**Natural Instructions**: Mishra et al., 2021
**Meta-Tuning**: Zhong et al., 2021
**FLAN**: Wei et al., 2021
**T0**: Sanh et al., 2021

**Cross-task Generalization**

# A potential limitation…

**training a model on a multi-task mixture**

**Train** 

**the *same* set of weights**

**for tasks**
**in *different* domains**
**with *different* complexity**
**requiring *different* skills**

**humans**

**decompose and recompose skills**

**Can we train a model that explicitly emulate this?**

# Background: Task-level Mixture-of-Experts

**Mixture-of-Experts**
Jacobs et al., 1991

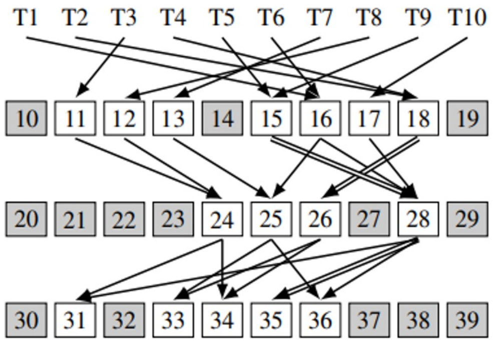
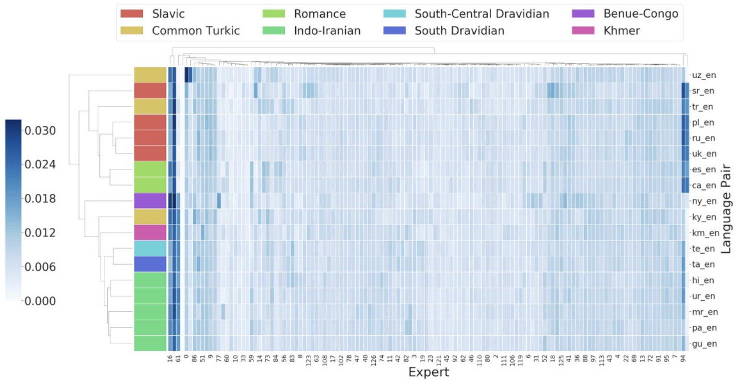
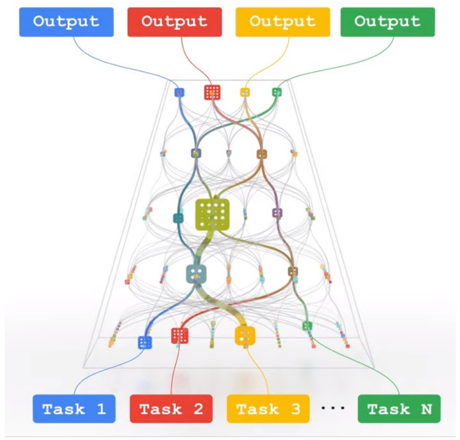



Figure 11: An actual routing map for MNIST-MTL.

**Routing Networks**
Rosenbaum et al., 2018



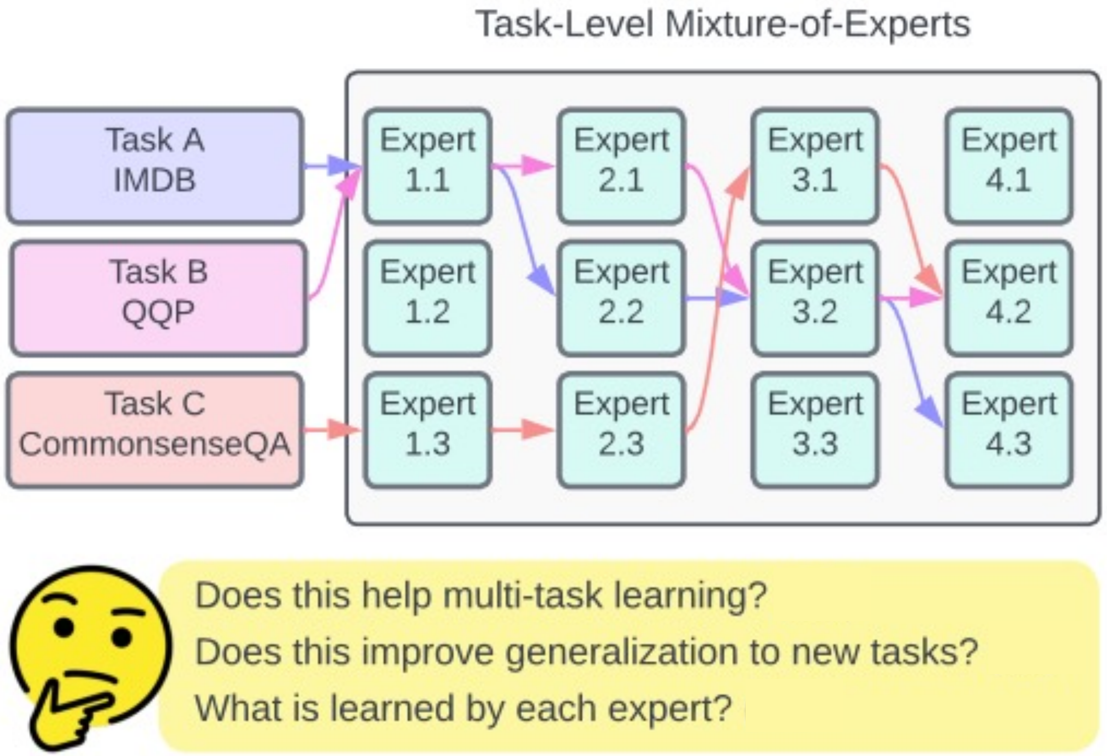

**Task-level MoE for Machine Translation**
Kudugunta et al., 2021




**Pathways**
Google AI Blog, 2021
Barham et al., 2022

# In this work

- **We train task-level mixture-of-expert models to multi-task on diverse NLP tasks**
  - Explicit, Flexible, Interpretable



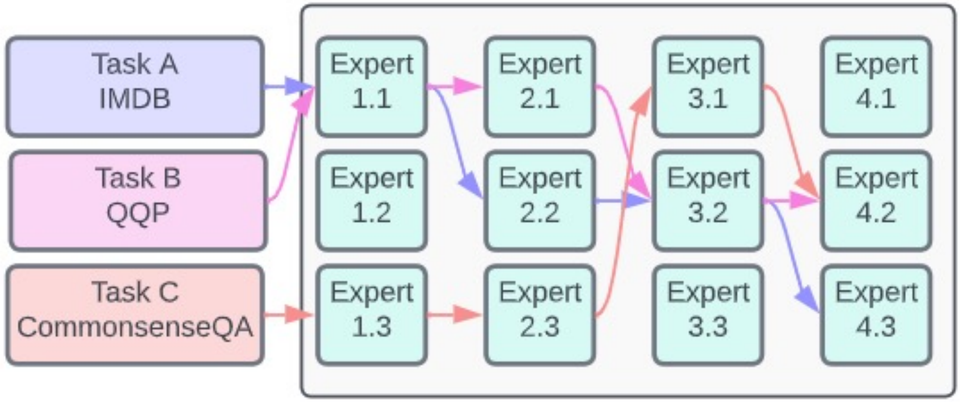
# Experiment Setup: Data

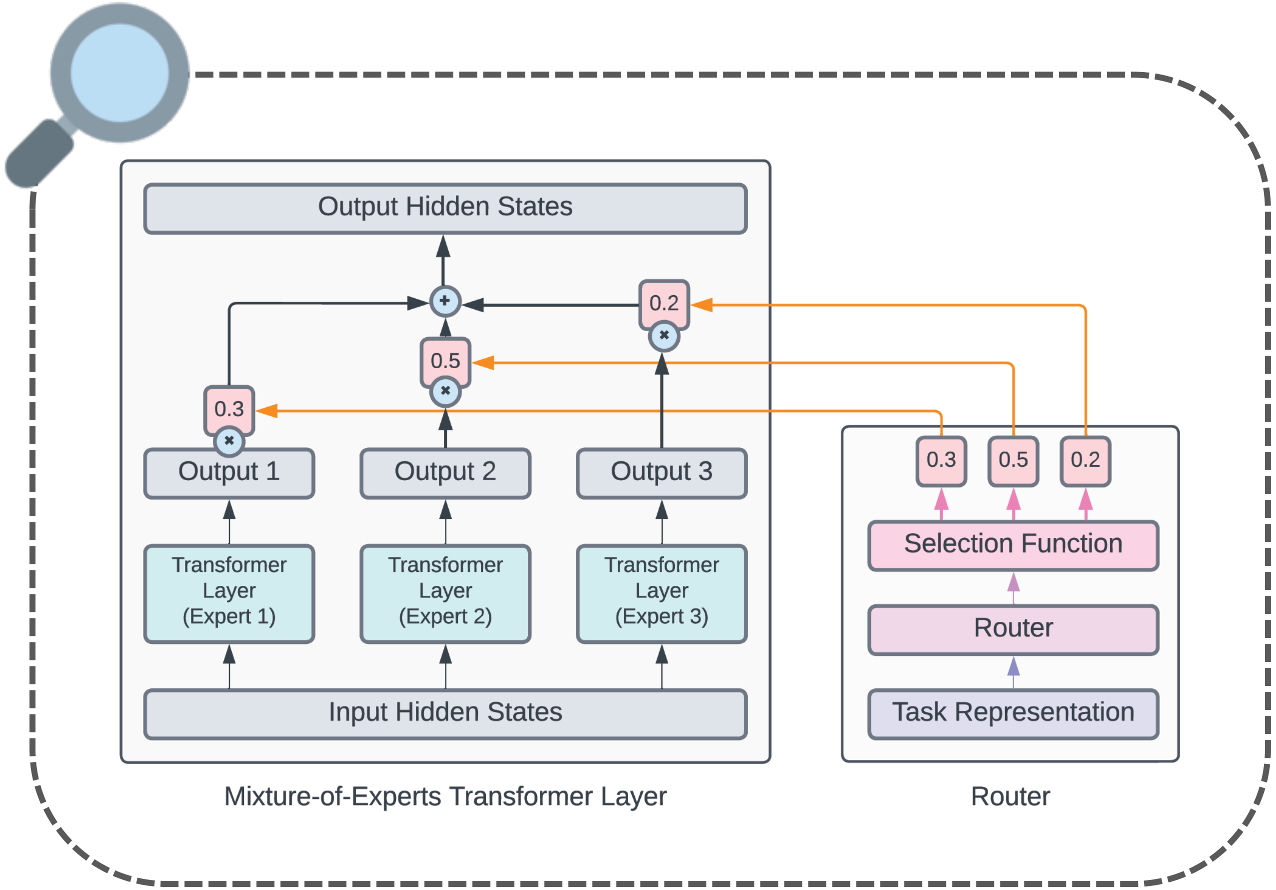
## CrossFit
## (Ye et al., 2021)

## Random Partition:
120 seen tasks for upstream multitask learning
18 unseen tasks for testing cross-task generalization


**Also tested on P3 dataset (Sanh et al., 2021)**

# Experiment Setup: Model



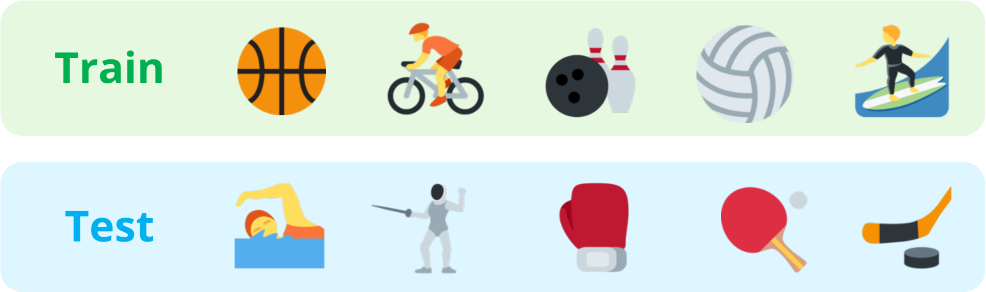
# Experiment Setup

## Data

### Random Partition in CrossFit

120 Train Tasks

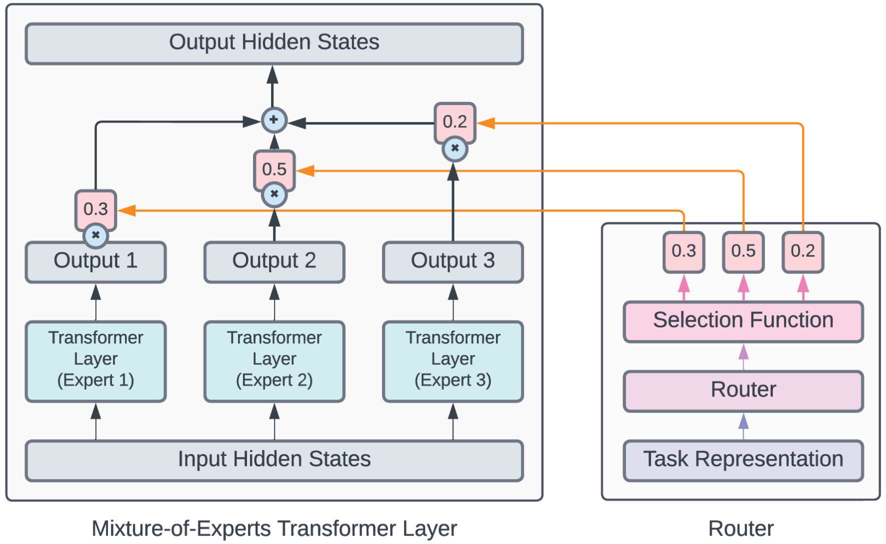
18 Test Tasks




## Model

### MoE-version of Transformer

Initialized from BART-Base (Lewis et al., 2020)



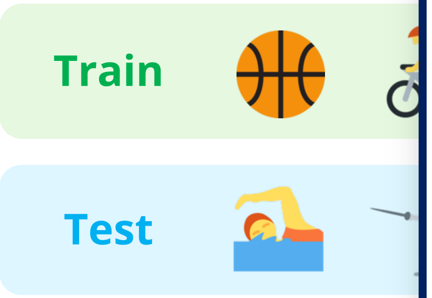
# Experiment Setup

## Data

### Random Partition in CrossFit

120 Trai

18 Test




**Expectation** **Reality**

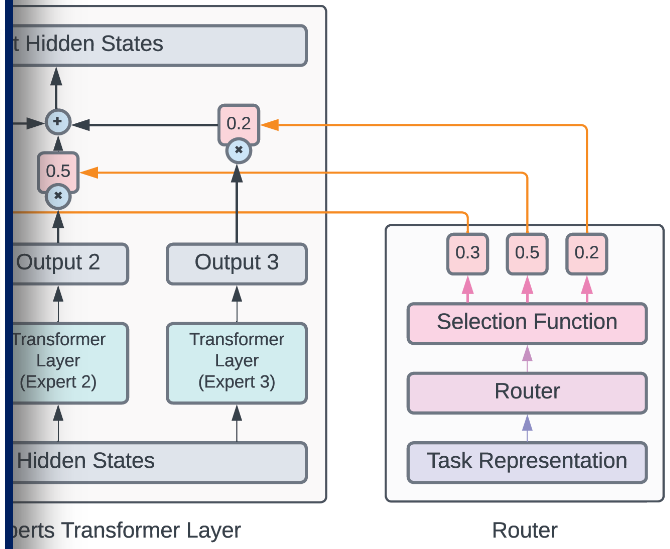
**@imjosef**

https://javascript.plainenglish.io/15-things-all-programmers-can-relate-to-7db1ce811b8

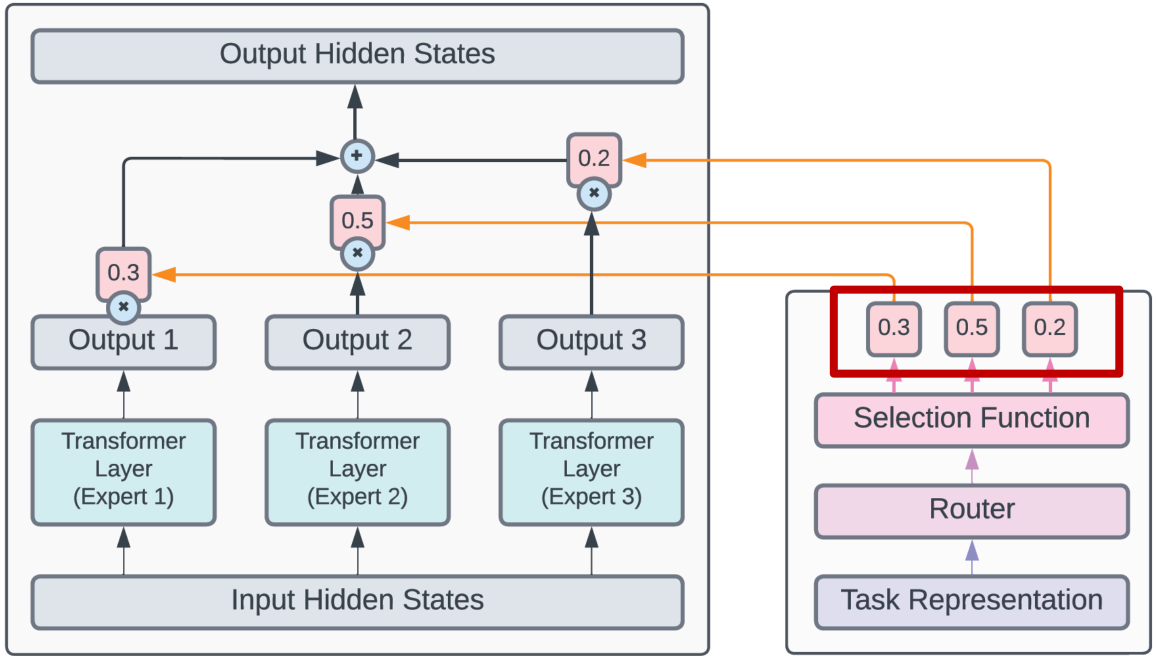
## Model

### MoE-version of Transformer

from BART-Base (Lewis et al., 2020)

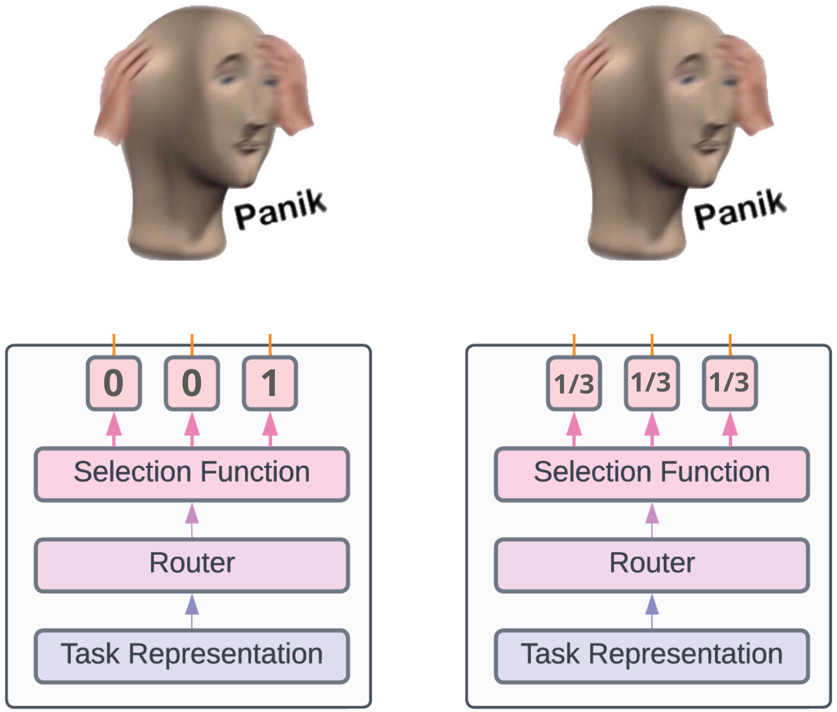


# How to make it work?




Mixture-of-Experts Transformer Layer

Router




Router

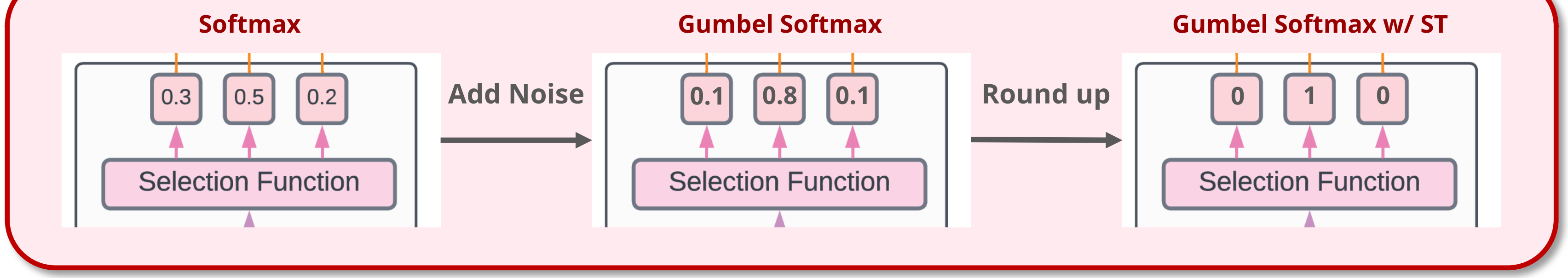
Router

**Degenerate to non-MoE transformer**

# How to make it work?

## Important Factors

| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| Softmax | Heterogenous | Yes | Yes |
| Gumbel Softmax | Homogeneous | No | No |
| Gumbel Softmax w/ Straight Through | | | |

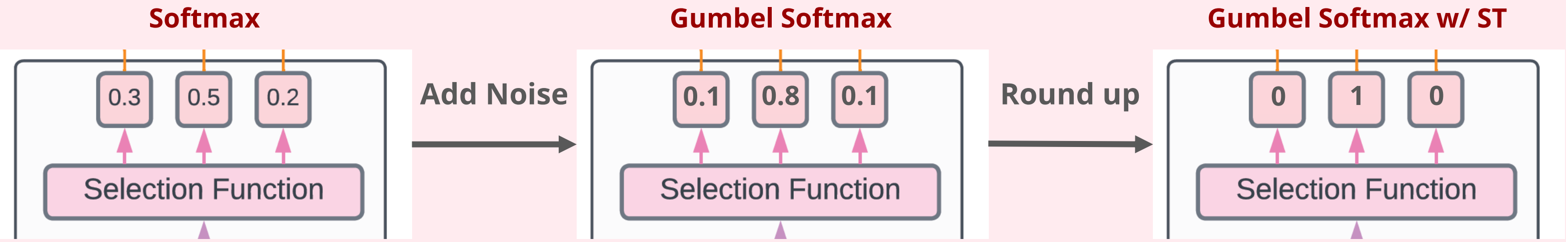


# How to make it work?

## Important Factors

| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| Softmax | Heterogenous | Yes | Yes |
| Gumbel Softmax | Homogeneous | No | No |
| Gumbel Softmax w/ Straight Through | | | |


**Sensitive to temperature and its annealing schedule**



# How to make it work?

## Important Factors

| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| Softmax | Heterogenous | Yes | Yes |
| Gumbel Softmax | Homogeneous | No | No |
| Gumbel Softmax w/ Straight Through | | | |

# How to make it work?

## Important Factors

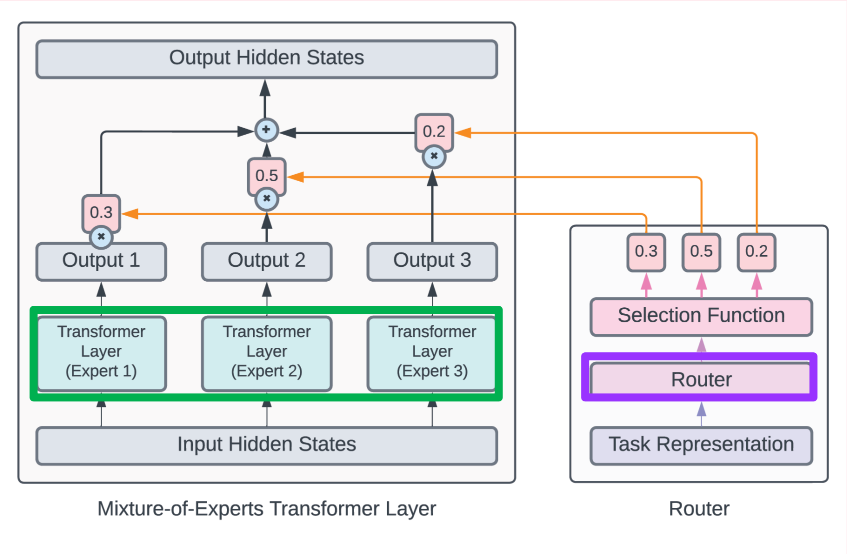
| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| Softmax | Heterogenous | Yes | Yes |
| Gumbel Softmax | Homogeneous | No | No |
| Gumbel Softmax w/ Straight Through | | | |

See also Aghajanyan et al., 2021

# How to make it work?

## Important Factors

| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| | | Yes | Yes |
| | | No | No |

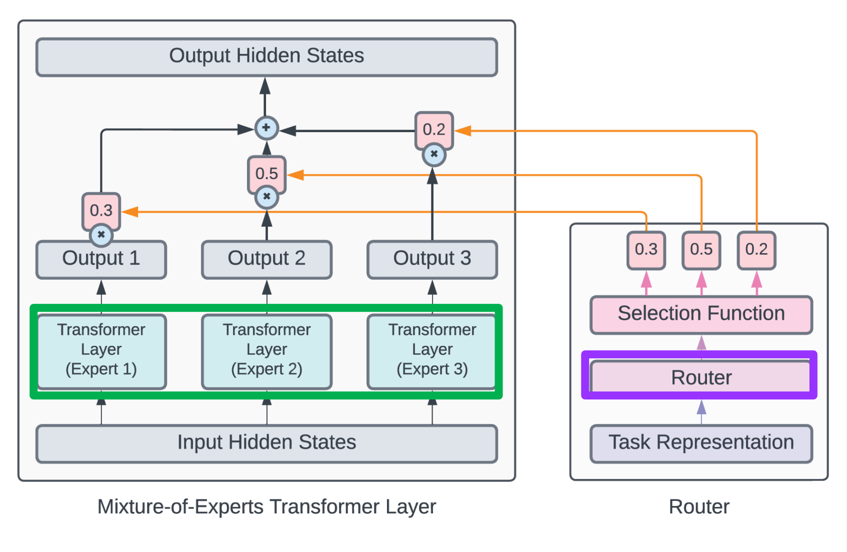



**Smaller LR** **Larger LR**

# How to make it work?

## Important Factors

| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| | | Yes | Yes |
| | | No | No |




Smaller LR

Larger LR

Also pay attention to how you clip the gradients!

See also Ponti et al., 2022

# How to make it work?

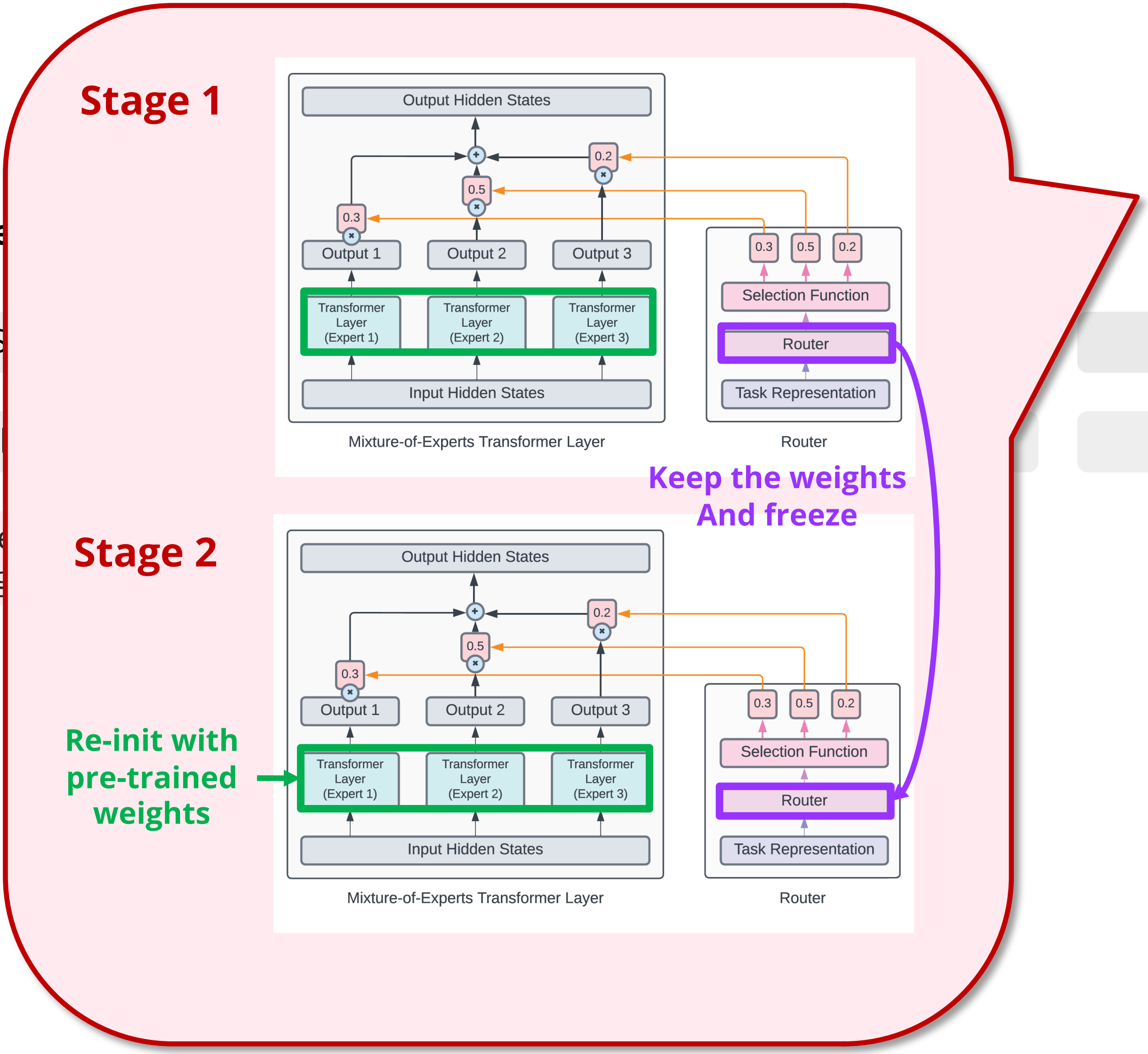
## Important Factors

Se
S
Gum
Gumbe
Straig

Two Stage Training

Yes

No



# How to make it work?

## Important Factors

| Selection | Batching | Two-speed LR | Two Stage Training |
|---|---|---|---|
| Softmax | Heterogenous | Yes | Yes |
| Gumbel Softmax | Homogeneous | No | No |
| Gumbel Softmax w/ Straight Through | | | |

# How to make it work?

## Less Important Factors

| Router | Task Representation | Freeze Task Repre. |
| --- | --- | --- |
| MLP | Random | Yes |
| LSTM | Text Embedding | No |
| Transformer | Fisher Information Task Embedding (Vu et al., 2020) | |

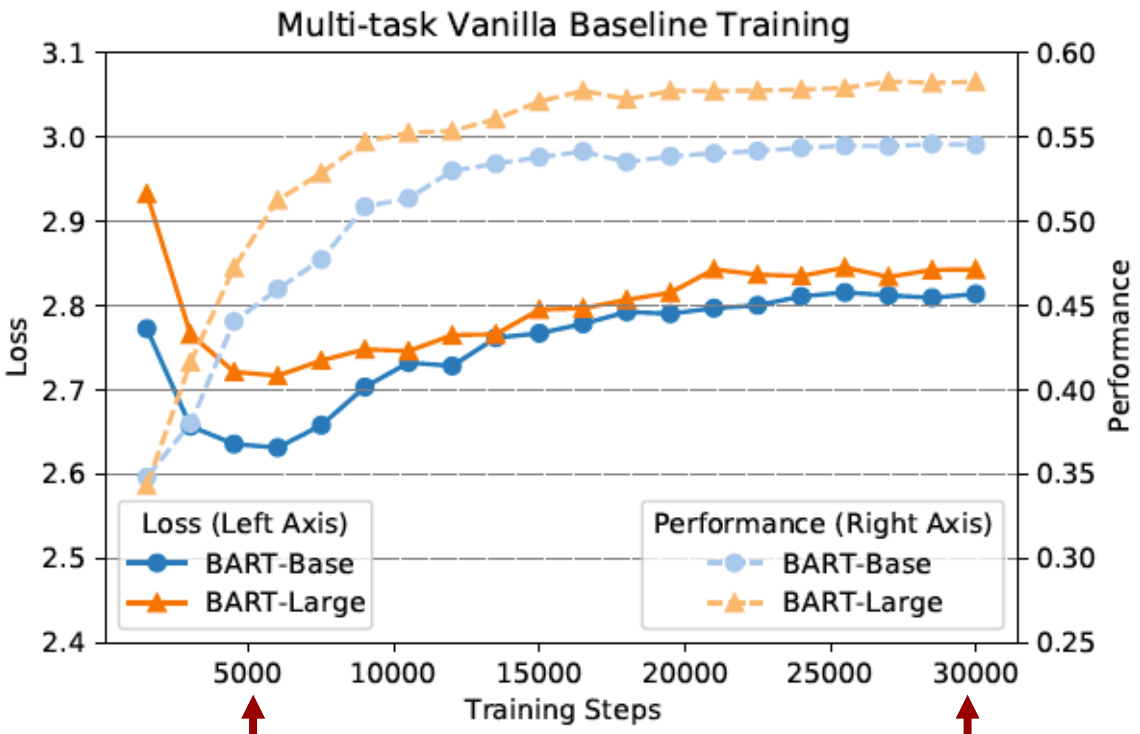
# How to make it work?

## Less Important Factors

| Router | Task Representation | Freeze Task Repre. |
| --- | --- | --- |
| MLP | Random | Yes |
| LSTM | Text Embedding | No |
| Transformer | Fisher Information Task Embedding (Vu et al., 2020) | |

# How to make it work?

## Discrepancy Between Loss and Performance




**Select model based on dev loss**

**Select model based on dev performance**

# How well does the model perform?

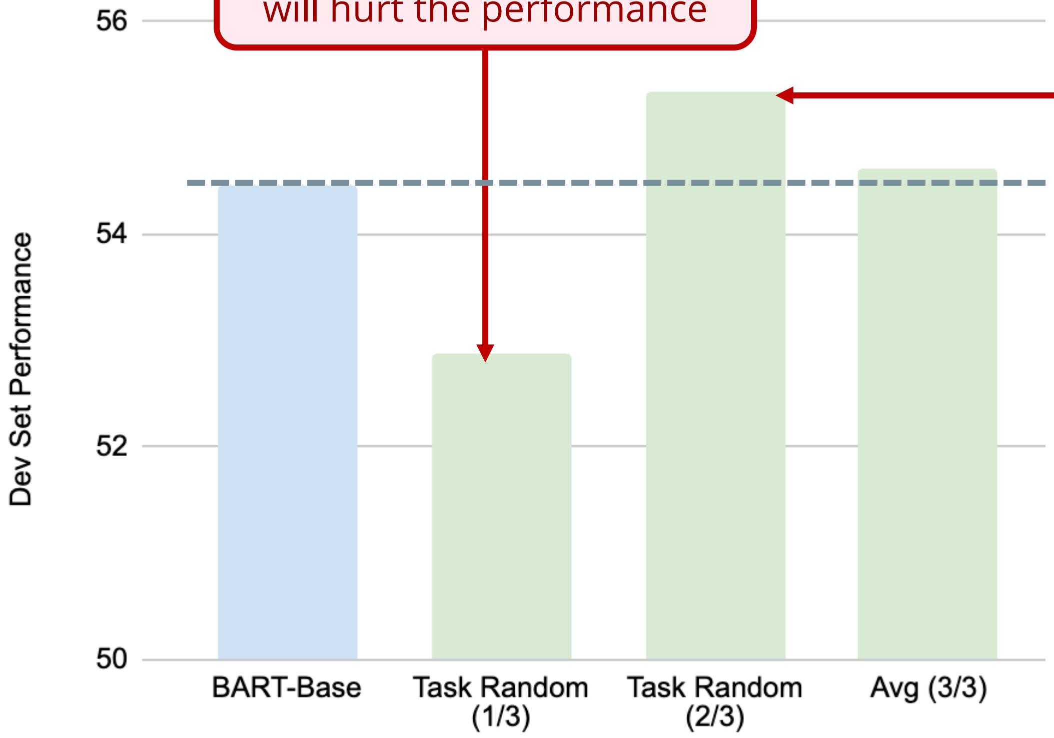
Vanilla Multi-tasking
Random/Avg Task-level Routing
Learned Task-level Routing

# How well does the model perform?

- Multi-task learning on the train set of 120 tasks
  - Report the average of performance on dev set

Vanilla Multi-tasking
Random/Avg Task-level Routing
Learned Task-level Routing

Introducing sparsity naively will hurt the performance




Surprisingly, randomly choosing 2 out of 3 experts improves performance

Sharing all weights may limit performance

# How well does the model perform?

- Multi-task learning on the train set of 120 tasks
  - Report the average of performance on dev set

**Vanilla Multi-tasking**
**Random/Avg Task-level Routing**
**Learned Task-level Routing**

Task-level MoE can match the performance of the best baseline

# How well does the model perform?

- Multi-task learning on the train set of 120 tasks
  - Report the average of performance on dev set

**Vanilla Multi-tasking**
**Random/Avg Task-level Routing**
**Learned Task-level Routing**

Task-level MoE can match the performance of the best baseline

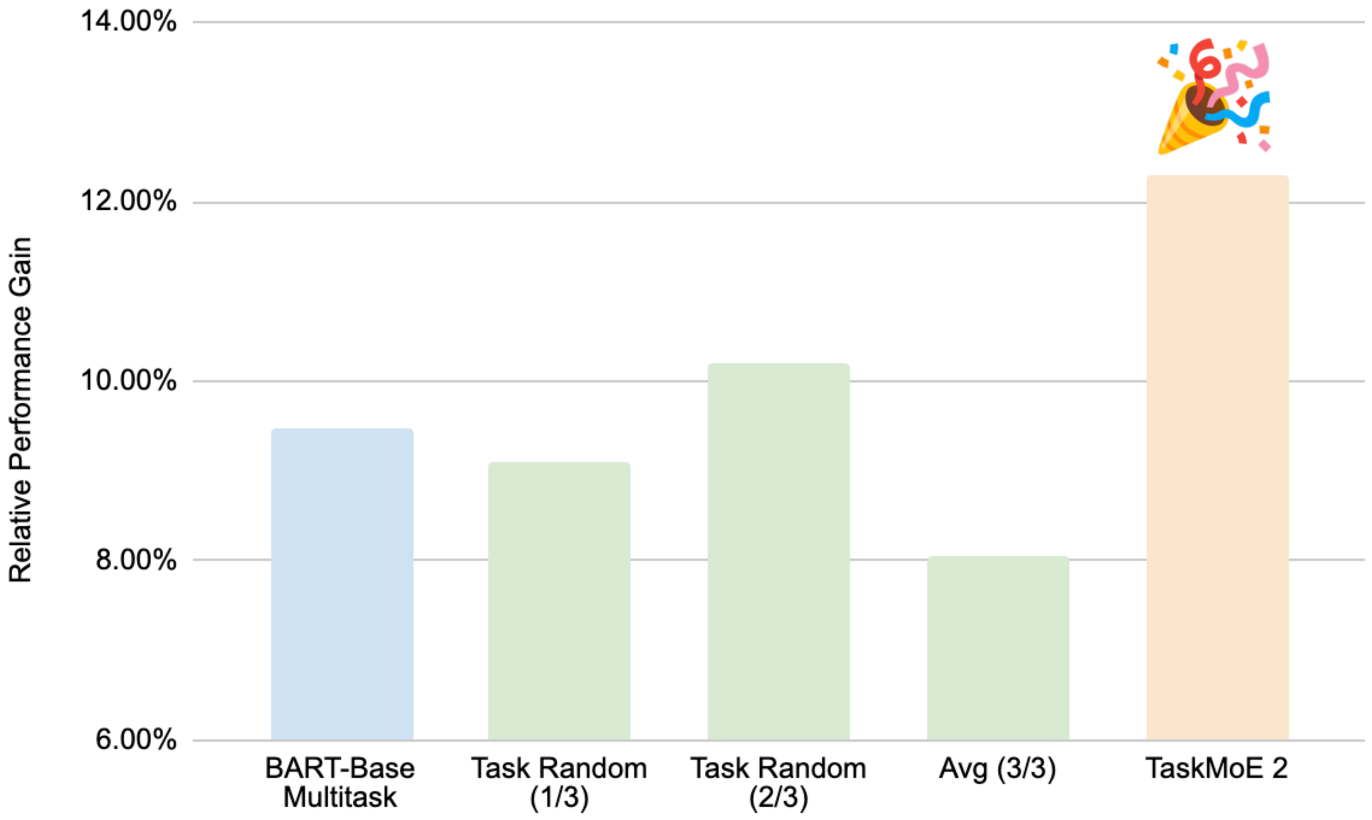

It uses less computation during inference

| Computation | 1x | 1x | 2x | 3x | 1x | 1x |
|---|---|---|---|---|---|---|

# How well does the model perform?

- Cross-task generalization to 18 unseen tasks
    - Report the average of relative performance gain on unseen tasks

**Vanilla Multi-tasking**
**Random/Avg Task-level Routing**
**Learned Task-level Routing**

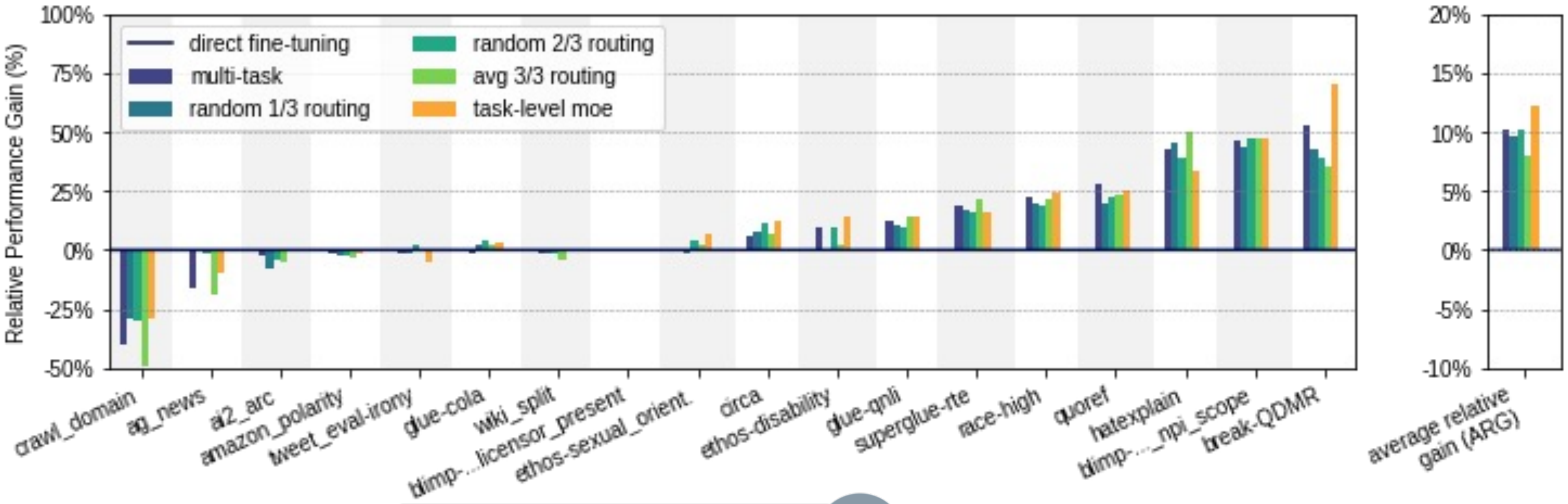


# How well does the model perform?

- Cross-task generalization to 18 unseen tasks
  - Report the average of relative performance gain on unseen tasks

**Vanilla Multi-tasking**
**Random/Avg Task-level Routing**
**Learned Task-level Routing**

**Breaking down to each task**



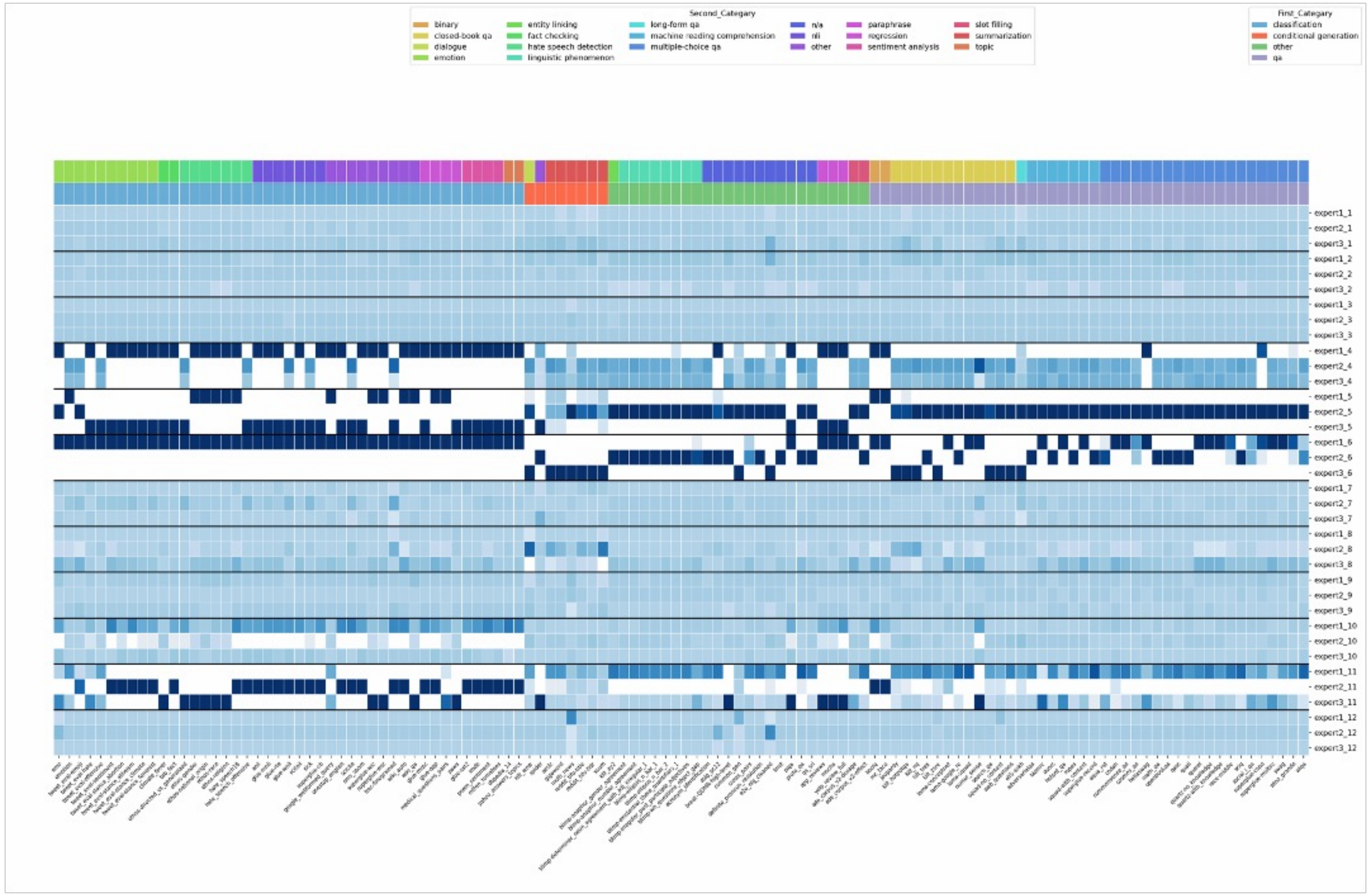

Avoid negative transfer

**Also tested on P3 dataset (Sanh et al., 2021)**

Improve average performance

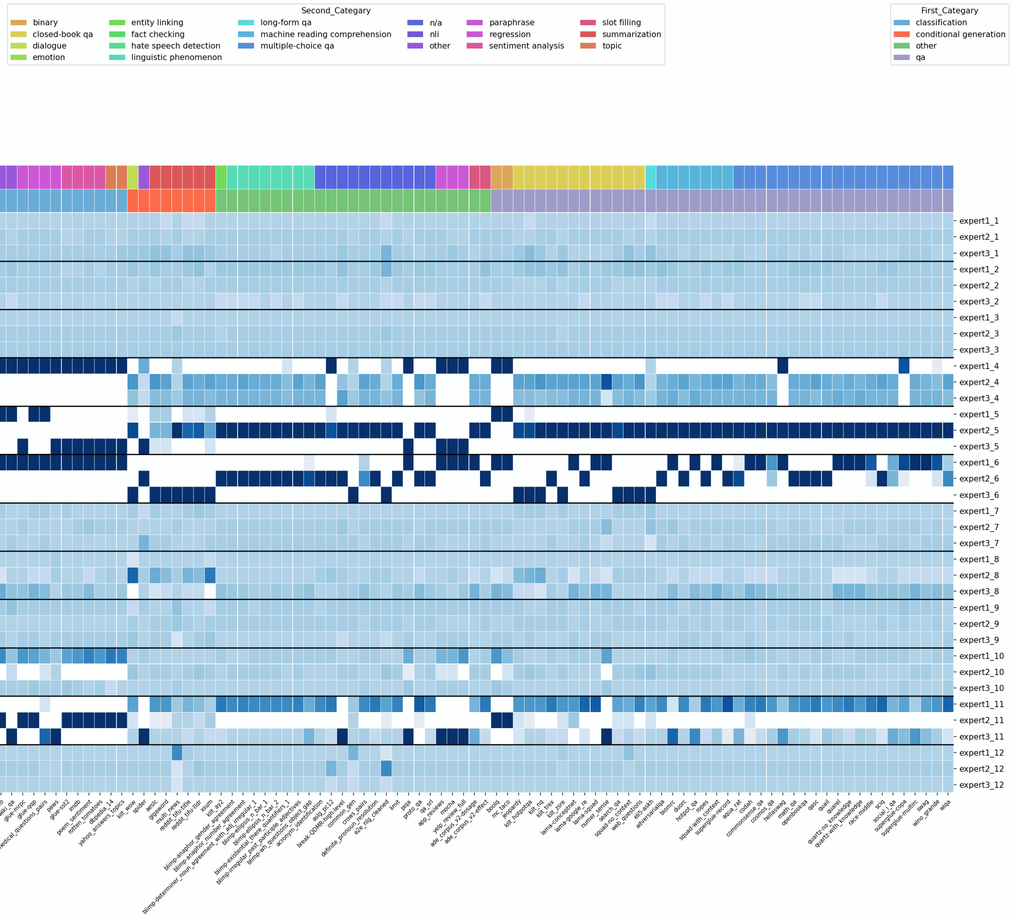
# What can we learn from the learned routes?

- Learning Dynamics



# What can we learn from the learned routes?

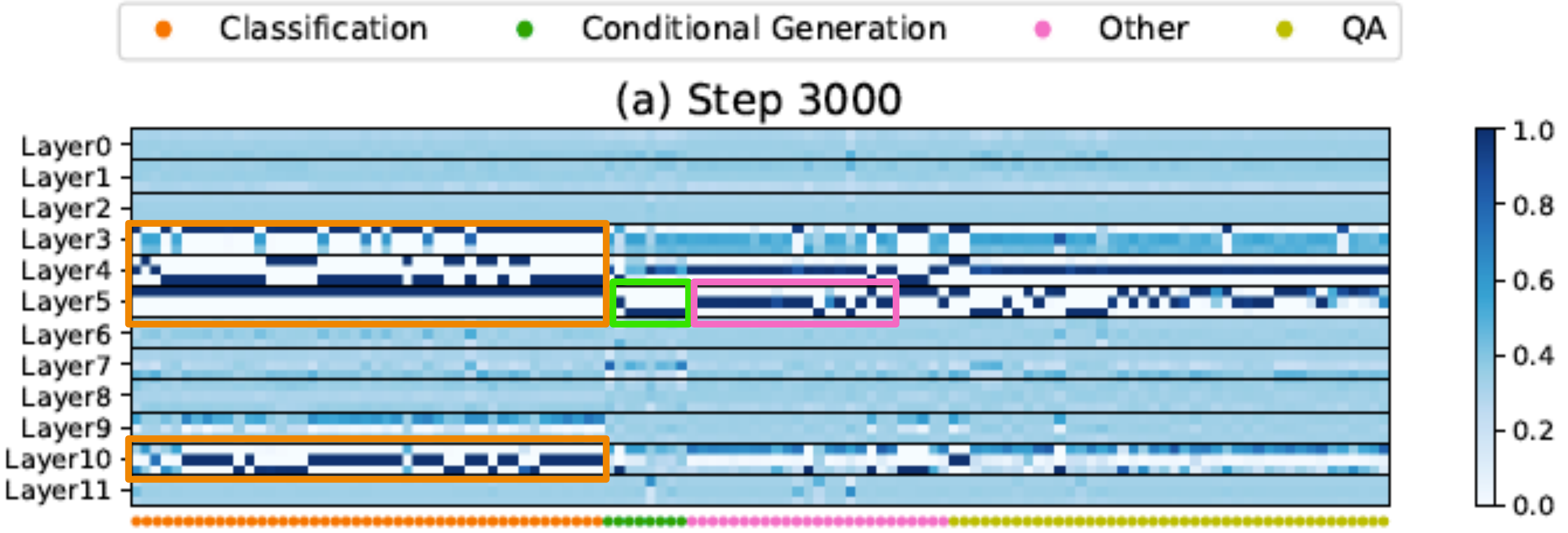
- Learning Dynamics



# What can we learn from the learned routes?

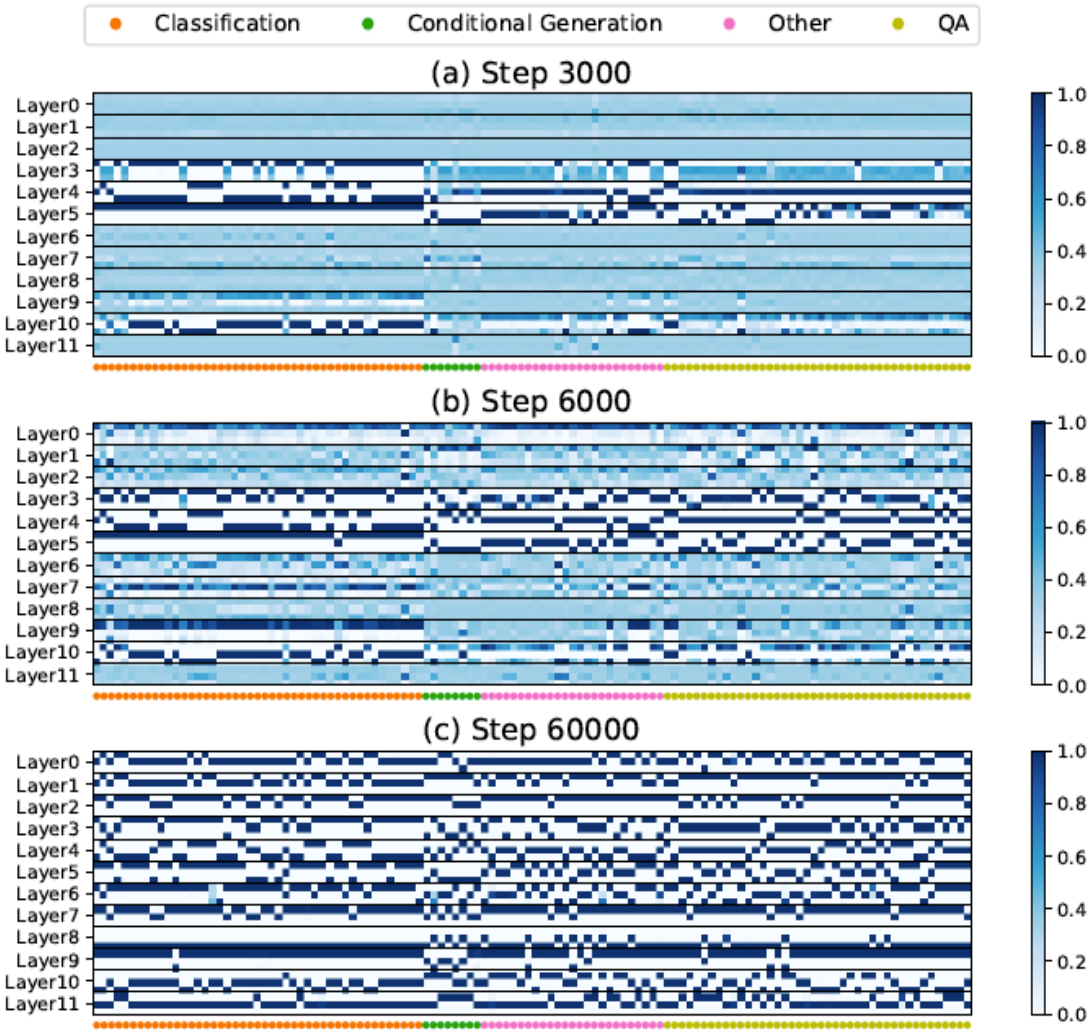
- Learning Dynamics

**Developing patterns early on**



# What can we learn from the learned routes?

- Learning Dynamics




More fine-grained

# What can we learn from the learned routes?

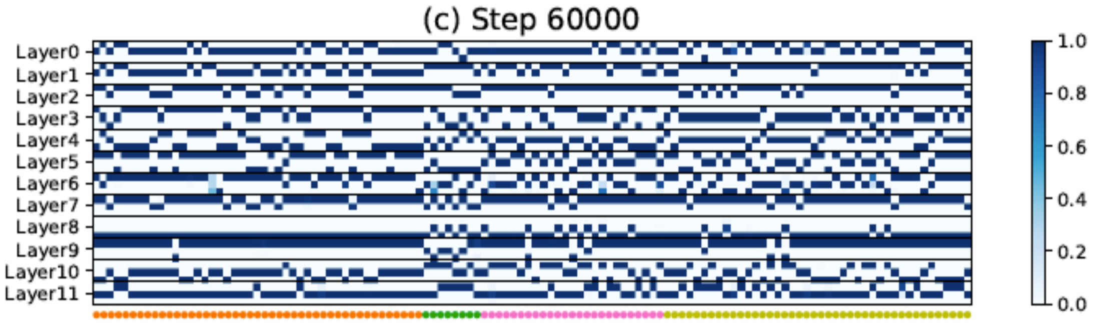
- How do we interpret these routes?

**Manually-defined Features**

| | T1 | T2 | T3 | … | … | … | Tn |
|---|---|---|---|---|---|---|---|
| **Extractive?** | Y | Y | N | | | | Y |
| **Use world knowledge?** | N | N | N | | | | Y |
| … | | | | | | | |
| **Has short input?** | N | Y | Y | | | | N |

$N_{features}$ x $N_{tasks}$

**Learned Routes**



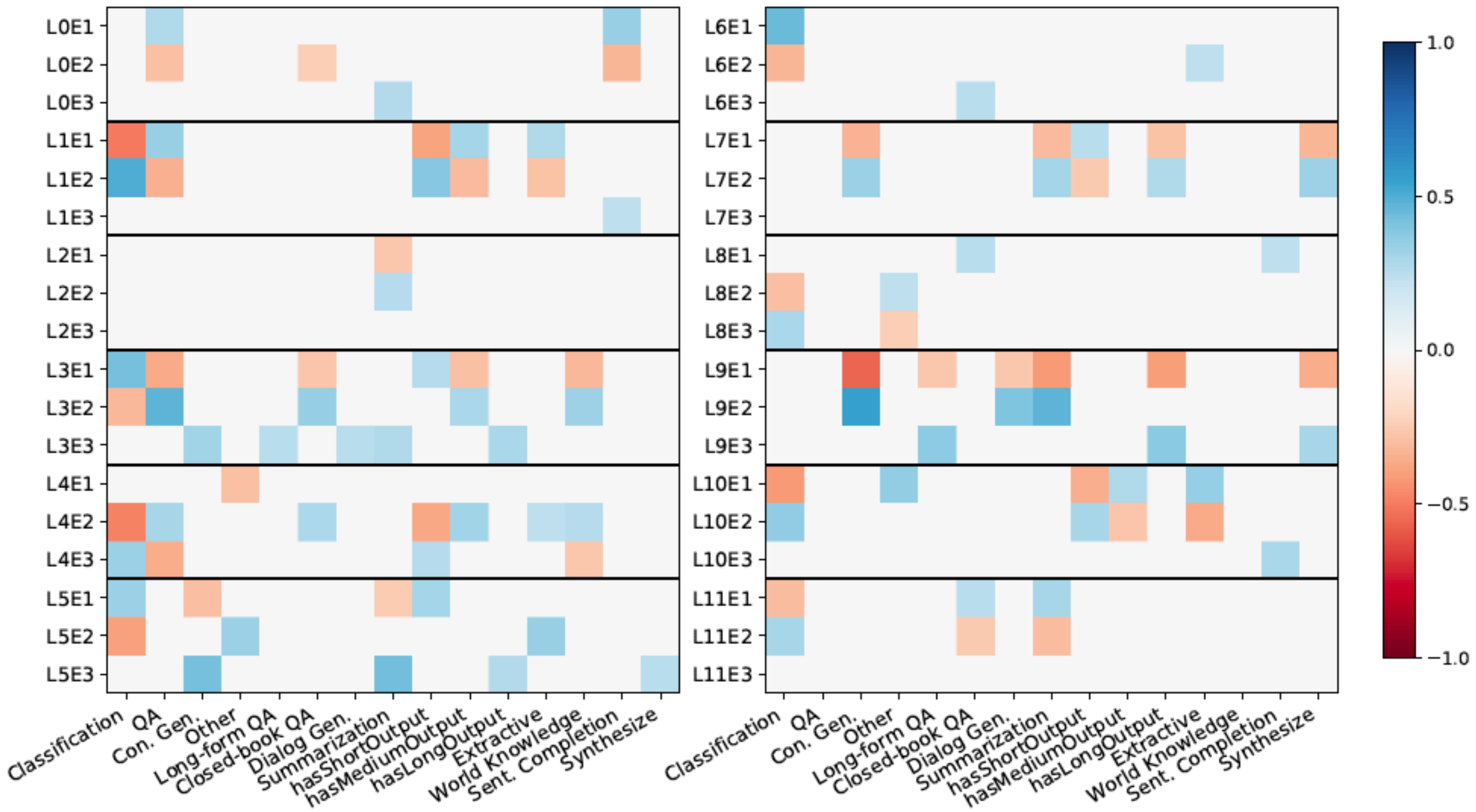

$N_{experts}$ x $N_{tasks}$

→ **Correlation between features and experts**

# What can we learn from the learned routes?

- How do we interpret these routes?

**Pearson correlation, p<0.01**




**Verified with expert disabling experiments**

# Conclusions

- We explored ...
  - Adapting transformer models to be task-level mixture-of-expert models
  - Training such models to multi-task on diverse NLP tasks
- We found that ...
  - Some design choices matter a lot
  - The resulting models are better at generalizing to unseen tasks
  - Learned routes and experts partly align with task characteristics defined by us

# Looking Forward

- Making few-shot learning more computationally efficient
  - Explore the area between in-context learning and fine-tuning
- Data Augmentation → Task Augmentation
  - From 160 tasks (CrossFit) / 36 tasks (P3) to more “tasks”