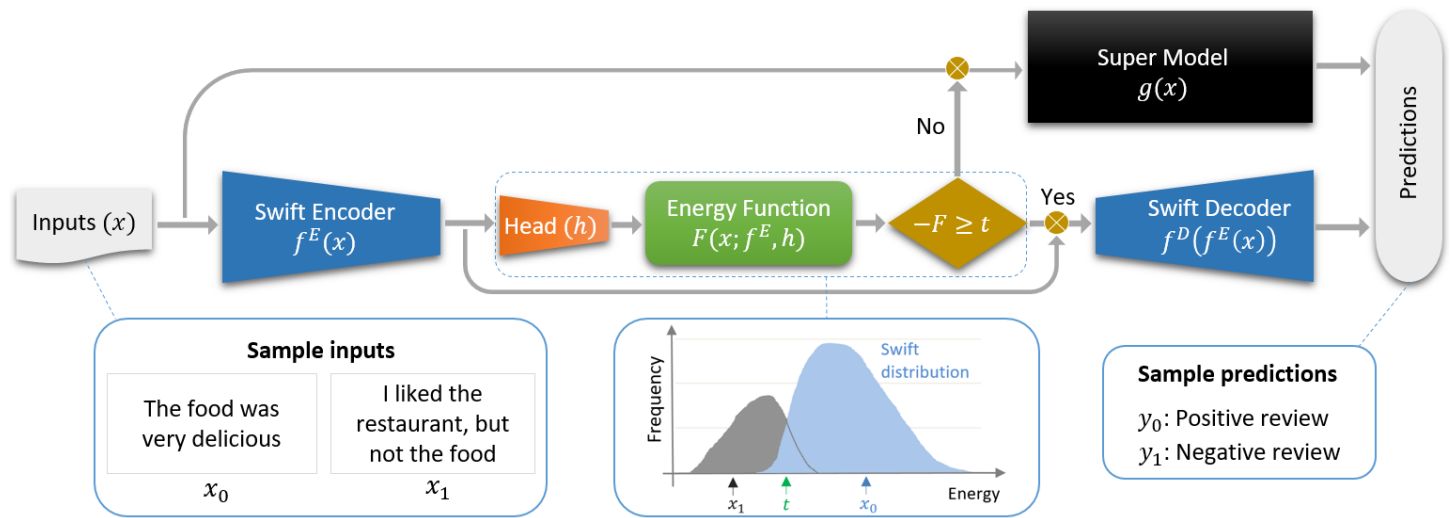


E-LANG: Energy-based Joint Inferencing of Super and Swift Language Models



Package Requirements

- Anaconda (version 2020.07)
- All the other requirements are listed in **environment.yml** file
- After installing Anaconda, use the following command to create an conda environment with the required packages:

```
conda env create -f environment.yml
```

- You can activate the environment using the following command:

```
conda activate elang
```

Commands for training and evaluation

Fine-tuning of T5 for downstream tasks

```
python main.py --train
                --model_type large
                --benchmark glue
                --task sst2
                --train_batch_size 8
                --steps 20000
                --save_steps 1000
```

- NOTE: before finetuning, the pre-trained T5 models need be downloaded from [the T5 repository](#). The datasets (e.g., GLUE) are also required to be downloaded from [Tensorflow-datasets](#).

Evaluation of T5 for downstream tasks

```
python main.py --eval
                --model_type large
```

```
--benchmark glue
--task sst2
--eval_batch_size 1
```

Training energy head of task-specific Swift

```
python main.py --train_head
                --model_type large
                --benchmark glue
                --task sst2
                --train_batch_size 8
                --steps 10000
                --save_steps 1000
```

Task-specific energy-based joint inference

```
python main.py --eval
                --model_type large
                --teacher_model_type 11b
                --benchmark glue
                --task sst2
                --eval_batch_size 1
                --head
                --router=energy
                --thresholds_list=1.0,1.5,2.0,2.5
```

Distillation-based fine-tuning of T5 for downstream tasks

```
python main.py --train
                --distill
                --model_type large
                --teacher_model_type 11b
                --benchmark glue
                --task sst2
                --train_batch_size 8
                --steps 20000
                --save_steps 1000
```

Arguments:

- **train**: if included, train/finetune the model
- **distill**: if included, train/finetune the model along with distillation from teacher
- **train_head**: if included, train the extra head
- **steps**: number of training steps
- **save_steps**: frequency of saving the checkpoints
- **eval**: evaluate the model
- **head**: consider using the extra head
- **router**: type of the routing mechanism: e.g., 'energy', 'softmax', 'entropy', or 'random'
- **thresholds_list**: list of threshold values for the routing mechanism (separated by comma)
- **model_type**: the model type for the Swift (student) T5 model
- **teacher_model_type**: the model type for the Super (teacher) T5 model
- **benchmark**: select the benchmark: e.g., 'glue' or 'super_glue'
- **task**: select the downstream task in the benchmark: e.g., 'cola', 'sst2', etc.

Experimental Results

		GLUE						SuperGLUE						WMT	
		MNLI	QNLI	SST2	RTE	MRPC	COLA	RTE	BoolQ	MRC	COPA	CB	WIC	WSC	EnRo
Swift (Large)	Time (ms)	216	283	57	263	160	56	287	303	201	96	223	185	133	1609
	Accuracy (%)	89.7	93.9	95.5	90.3	90.9	62.7	88.5	84.3	80.7	81.0	92.0	72.7	86.5	28.6
Super (11B)	Time (ms)	821	980	281	964	433	213	818	3205	1731	268	844	671	2211	3041
	Accuracy (%)	91.7	95.9	96.6	92.4	91.7	69.1	93.1	89.4	84.9	93.0	93.1	77.4	89.4	28.9
E-LANG	Accuracy (%)	91.7	96.0	96.6	92.4	92.2	69.5	93.2	88.7	84.9	90.0	93.1	78.1	89.4	28.9
	FLOPs ($\times 10^{11}$)	47.8	25.7	29.5	50.4	11.5	39.9	42.0	50.8	46.9	52.6	13.4	40.3	20.6	63.4
	Time (ms)	582	495	132	716	190	147	671	1978	1022	222	302	447	545	2800
	Swift Ratio (%)	49	75	70	46	91	58	56	45	50	43	89	57	81	30
	Speed-up (FLOPs)	1.8X	3.4X	2.9X	1.7X	7.6X	2.2X	2.1X	1.7X	1.9X	1.7X	6.5X	2.2X	4.2X	1.4X
	Speed-up (time)	1.4X	2.0X	2.1X	1.4X	2.3X	1.5X	1.2X	1.6X	1.7X	1.2X	2.8X	1.5X	4.1X	1.1X

Table 1: Joint inference results with T5 architecture on GLUE and SuperGLUE development sets, and WMT’s English-to-Romanian translation. The FLOPs for Super and Swift are respectively 87×10^{11} and 4.25×10^{11} .

	MNLI	QNLI	SST2	RTE	MRPC	COLA	Average
Super (11B)	87.0/91.7	87.0/95.9	87.0/96.6	87.0/92.4	87.0/91.7	87.0/69.1	87.0/89.5
Random (Encoder)	78.5/91.5	61.9/95.3	58.7/96.3	60.2/91.2	47.5/91.9	61.6/67.2	61.4/88.9
Softmax (Encoder)	57.7/91.6	36.5/95.9	34.6/96.5	52.0/92.3	13.8/92.1	45.7/69.3	40.1/89.6
Entropy (Encoder)	55.7/91.6	27.1/96.0	40.2/96.5	50.7/92.0	23.0/92.2	48.1/69.3	40.8/89.6
Energy (Swift _{small})	71.3/91.0	58.8/95.6	47.0/96.6	71.2/88.5	55.0/91.4	75.3/68.3	63.1/88.5
Energy (Swift _{base})	54.5/91.5	50.5/95.8	35.9/96.6	55.8/90.6	44.0/91.9	50.6/68.4	48.5/89.1
Energy (Decoder)	57.9/90.6	68.1/95.5	75.8/96.3	60.5/91.5	20.2/90.9	45.1/69.3	54.6/89.0
Energy (Encoder)	47.8/91.7	25.7/96.0	32.0/96.6	50.4/92.4	11.5/92.2	39.9/69.5	34.5/89.7

Table 2: Ablation study on different T5-based scenarios. Each cell shows FLOPs/Accuracy.

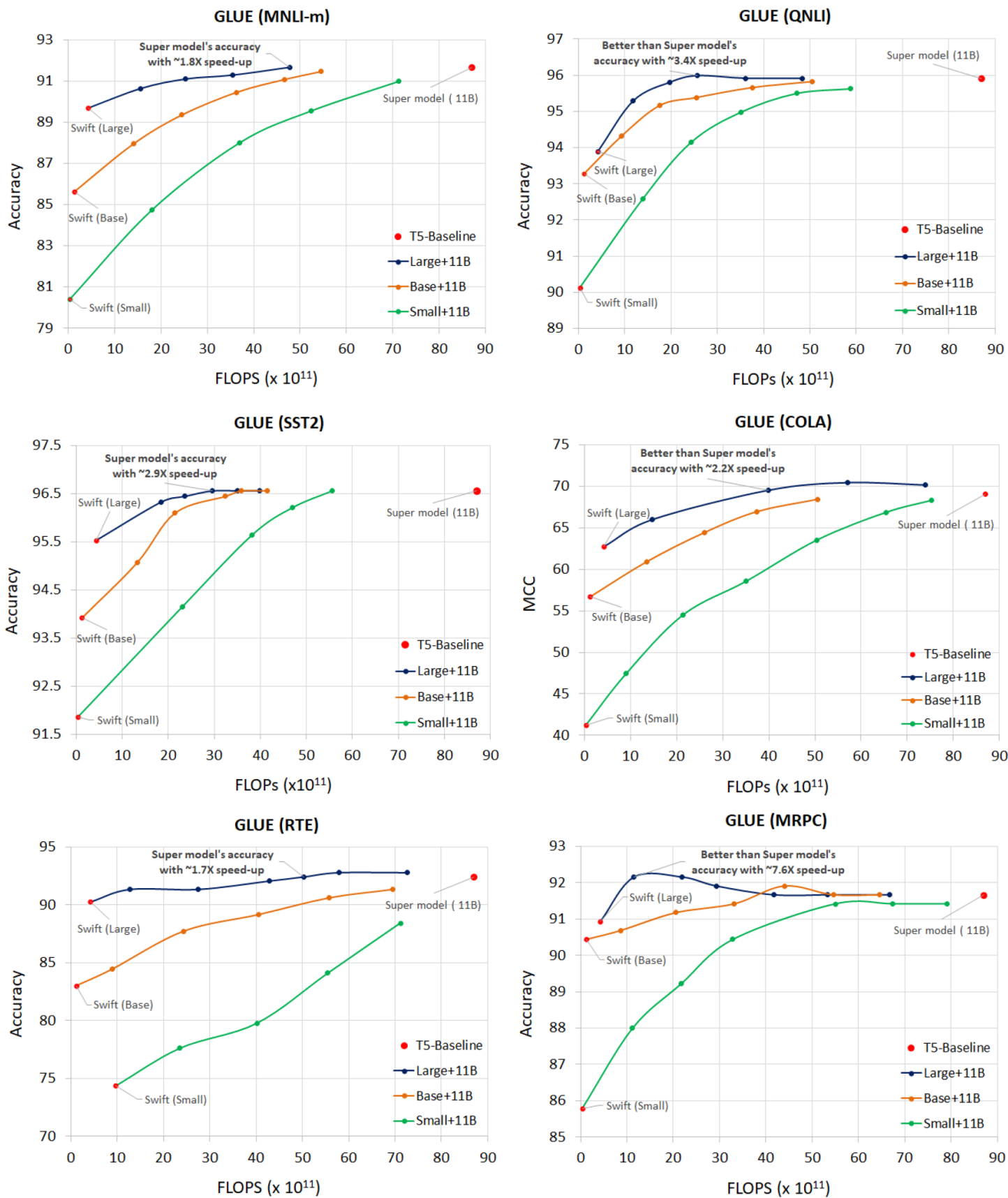


Figure 5: Trade-off curves with T5 backbone on GLUE tasks.