1 Question answering task on the SQUADv2 dataset

	SQUADv2 (Exact Match)	SQUADv2 (F1)
Adam	48.41 ± 0.57	49.99 ± 0.54
M-FAC	49.80 ± 0.43	52.18 ± 0.20

Table 1: Comparing M-FAC optimizer (without weight decay) against HuggingFace's Adam baseline on the **bert-tiny** model.

	SQUADv2 (Exact Match)	SQUADv2 (F1)
Adam	54.80 ± 0.47	58.13 ± 0.31
M-FAC	58.02 ± 0.39	61.35 ± 0.24

Table 2: Comparing M-FAC optimizer (without weight decay) against HuggingFace's Adam baseline on the **bert-mini** model.

2 Text classification on a subset of GLUE tasks

	SST-2 (Acc.)	MRPC (F1)	MRPC (Acc.)	STS-B (Pearson)	STS-B (Spearman)
Adam	80.11 ± 0.65	81.68 ± 0.33	69.90 ± 0.32	64.39 ± 5.02	66.52 ± 5.67
M-FAC	81.86 ± 0.76	82.77 ± 0.22	72.94 ± 0.37	80.15 ± 0.52	80.62 ± 0.43

	QQP (F1)	QQP (Acc.)	MNLI-m (Acc.)	MNLI-mm (Acc.)	QNLI (Acc.)
Adam	81.09 ± 0.15	77.58 ± 0.08	65.36 ± 0.13	66.78 ± 0.15	77.85 ± 0.15
M-FAC	84.29 ± 0.08	79.71 ± 0.13	68.28 ± 3.29	68.98 ± 3.05	81.17 ± 0.43

Table 3: Comparing M-FAC optimizer (without weight decay) against HuggingFace's Adam baselines on the **bert-tiny** model.

	SST-2 (Acc.)	MRPC (F1)	MRPC (Acc.)	STS-B (Pearson)	STS-B (Spearman)
Adam	85.46 ± 0.58	84.57 ± 0.36	76.57 ± 0.80	82.09 ± 0.54	82.64 ± 0.71
M-FAC	84.20 ± 0.58	85.06 ± 1.63	78.87 ± 2.33	84.66 ± 0.30	84.65 ± 0.30

	QQP (F1)	QQP (Acc.)	MNLI-m (Acc.)	MNLI-mm (Acc.)	QNLI (Acc.)
Adam	86.45 ± 0.12	82.43 ± 0.10	73.30 ± 0.20	74.85 ± 0.09	83.85 ± 0.10
M-FAC	86.75 ± 0.20	82.67 ± 0.23	74.59 ± 0.41	75.95 ± 0.14	83.70 ± 0.13

Table 4: Comparing M-FAC optimizer (without weight decay) against HuggingFace's Adam baselines on the **bert-mini** model.

3 Text classification on a subset of GLUE tasks (evaluation on the official test sets)

	SST-2 (Acc.)	MRPC (F1)	MRPC (Acc.)	STS-B (Pearson)	STS-B (Spearman)
AdamW	83.2	81.1	71.1	74.3	73.6
M-FAC	83.4*	81.9*	72.7*	75.3*	73.2*

	QQP (F1)	QQP (Acc.)	MNLI-m (Acc.)	MNLI-mm (Acc.)	QNLI (Acc.)
AdamW	62.2	83.4	70.2	70.3	81.5
M-FAC	62.8	83.9	71.0	70.5	81.7

Table 5: Comparing M-FAC optimizer (without weight decay) against authors' (https://github.com/google-research/bert) **tuned bert-tiny** competitive baselines on a subset of GLUE benchmark test sets. * Modest tuning of learning rate and dampening because of an extremely low number of samples (*i.e.* gradients) in the dataset.