	Small	Medium	Large
Steps btw. validations	100	100	1000
Attention	N	N	Y
Classifier dropout rate	0.4	0.2	0.2
Classifier hidden dim.	128	256	512
Max pool projection dim.	128	256	512

Table 5: Hyperparameter settings for target-task models and target-task training for ELMo-style models. Small-data tasks are RTE and WNLI; medium-data tasks are CoLA, SST, and MRPC; large-data tasks are STS, QQP, MNLI, and QNLI. STS has a relatively small training set, but consistently patterns with the larger tasks in its behavior.

A Additional Pretraining Task Details

DisSent To extract discourse model examples from the WikiText-103 corpus (Merity et al., 2017), we follow the procedure described in Nie et al. (2019) by extracting clause-pairs that follow specific dependency relationships within the corpus (see Figure 4 in Nie et al., 2019). We use the Stanford Parser (Chen and Manning, 2014) distributed in Stanford CoreNLP version 3.9.1 to identify the relevant dependency arcs.

Cross-Sentence Attention For MNLI, QQP, QNLI, and STS with ELMo-style models, we use an attention mechanism between all pairs of words representations, followed by a $512D \times 2$ BiLSTM with max-pooling over time, following the mechanism used in BiDAF (Seo et al., 2017).

Alternative Tasks Any large-scale comparison like the one attempted in this paper is inevitably incomplete. Among the thousands of publicly available NLP datasets, we also performed initial trial experiments on several datasets for which we were not able to reach development-set performance above that of the random encoder baseline in the pretraining or as an intermediate task with ELMo. These include image-caption matching with MSCOCO (Lin et al., 2014), following Kiela et al. (2018); the small-to-medium-data text-understanding tasks collected in NLI format by Poliak et al. (2018); ordinal common sense inference (Zhang et al., 2017); POS tagging on the Penn Treebank (Marcus et al., 1993); supertagging on CCGBank (Hockenmaier and Steedman, 2007); and a variant objective on our Reddit data, inspired by Yang et al. (2018), where the model is trained to select which of two candidate replies to a given comment is correct.

B Hyperparameters and Optimization Details

See Section 5 for general comments on hyperparameter tuning.

Validation We evaluate on the validation set for the current training task or tasks every 1,000 steps, except where noted otherwise for small-data target tasks. During multitask learning, we multiply this interval by the number of tasks, evaluating every 9,000 steps during GLUE multitask training, for example.

Optimizer For BERT, we use the same optimizer and learning rate schedule as Devlin et al. (2019), with an initial learning rate of 1e-5 and training for a maximum of three epochs at each stage (or earlier if we trigger a different early stopping criterion). For all other experiments, we use AMSGrad (Reddi et al., 2018). During pretraining, we use a learning rate of 1e-4 for classification and regression tasks, and 1e-3 for text generation tasks. During target-task training, we use a learning rate of 3e-4 for all tasks.

Learning Rate Decay We multiply the learning rate by 0.5 whenever validation performance fails to improve for more than 4 validation checks. We stop training if the learning rate falls below 1e-6.

Early Stopping We maintain a saved checkpoint reflecting the best validation result seen so far. We stop training if we see no improvement after more than 20 validation checks. After training, we use the last saved checkpoint.

Regularization For BERT models, we follow the original work. For non-BERT models, we apply dropout with a drop rate of 0.2 after the character CNN in pretraining experiments or after ELMo, after each LSTM layer, and after each MLP layer in the task-specific classifier or regressor. For small-data target tasks, we increase MLP dropout to 0.4 during target-task training.

Preprocessing For BERT, we follow Devlin et al. (2019) and use the WordPiece (Wu et al., 2016) tokenizer. For all other experiments, we use the Moses tokenizer for encoder inputs, and set a maximum sequence length of 40 tokens. There is no input vocabulary, as we use ELMo's character-based input layer.

For English text generation tasks, we use the Moses tokenizer to tokenize our data, but use a

word-level output vocabulary of 20,000 types for tasks that require text generation. For translation tasks, we use BPE tokenization with a vocabulary of 20,000 types. For all sequence-to-sequence tasks we train word embeddings on the decoder side.

Target-Task-Specific Parameters For non-BERT models, to ensure that baseline performance for each target task is competitive, we find it necessary to use slightly different models and training regimes for larger and smaller target tasks. We used partially-heuristic tuning to separate GLUE tasks into big-, medium- and small-data groups, giving each group its own heuristically chosen task-specific model specifications. Exact values are shown in Table 5.

Sequence-to-Sequence Models We found bilinear attention to be helpful for the SkipThought and Reddit pretraining tasks but not for machine translation, and report results for these configurations. For ELMo-style models, we use the max-pooled output of the encoder to initialize the hidden state of the decoder, and the size of this hidden state is equal to the size of the output of our shared encoder. For BERT, we use the representation corresponding to the `[CLS]` token to initialize the hidden state of the decoder. We reduce the dimension of the output of the decoder by half via a learned linear projection before the output softmax layer.

C Multitask Learning Methods

Our multitask learning experiments have three somewhat distinctive properties: (i) We mix tasks with very different amounts of training data—at the extreme, under 1,000 examples for WNLI, and over 1,000,000,000 examples from LM BWB. (ii) Our goal is to optimize the quality of the shared encoder, not the performance of any one of the tasks in the multitask mix. (iii) We mix a relatively large number of tasks, up to eighteen at once in some conditions. These conditions make it challenging but important to avoid overfitting or underfitting any of our tasks.

Relatively little work has been done on this problem, so we conduct a small experiment here. All our experiments use the basic paradigm of randomly sampling a new task to train on at each step, and we experiment with two hyperparameters that can be used to control over- and underfitting: The probability with which we sample each task and the weight with which we scale the loss for each task. Our experiments follow the setup in Appendix B, and do not use the ELMo BiLSTM. For validation metrics like perplexity that decrease from high starting values during training, we include the transformed metric $1 - \frac{metric}{250}$ in our average, where the constant 250 was tuned in early experiments.

	Pretraining Tasks			
Sampling	**GLUE**	**S1**	**S2**	**S3**
Uniform	69.1	53.7	82.1	31.7
Proportional	**69.8**	52.0	83.1	36.6
Log Proportional	68.8	54.3	82.9	31.2
Power 0.75	69.3	51.1	82.7	**37.9**
Power 0.66	69.0	53.4	82.8	35.5
Power 0.5	69.1	**55.6**	**83.3**	35.9

Table 6: Comparison of sampling methods on four subsets of GLUE using uniform loss scaling. The reported scores are averages of the development set results achieved for each task after early stopping. Results in **bold** are the best within each set.

	Loss Scaling		
Sampling	**Uniform**	**Proportional**	**Power 0.75**
Uniform	69.1	**69.7**	**69.8**
Proportional	**69.8**	69.4	69.6
Log Proportional	68.8	68.9	68.9
Power 0.75	69.3	69.1	69.0

Table 7: Combinations of sampling and loss scaling methods on GLUE tasks. Results in **bold** are tied for best overall GLUE score.

Task Sampling We consider several approaches to determine the probability with which to sample a task during training, generally making this probability a function of the amount of data available for the task. For task i with training set size N_i, the probability is $p_i = f(N_i)/\sum_j f(N_j)$, where $f(N_i) = 1$ (Uniform), N_i (Proportional), $log(N_i)$ (Log Proportional), or N_i^a (Power a) where a is a constant.

Loss Scaling At each update, we scale the loss of a task with weight $w_i = f(N_i)/max_j f(N_j)$, where $f(N_i) = 1$ (Uniform), N_j (Proportional), or N_j^a (Power a).

Experiments For task sampling, we run experiments with multitask learning on the full set of nine GLUE tasks, as well as three subsets: single sentence tasks (S1: SST, CoLA), similarity and paraphrase tasks (S2: MRPC, STS, QQP), and

inference tasks (S3: WNLI, QNLI, MNLI, RTE). The results are shown in Table 6.

We also experiment with several *combinations* of task sampling and loss scaling methods, using only the full set of GLUE tasks. The results are shown in Table 7.

While no combination of methods consistently offers dramatically better performance than any other, we observe that it is generally better to apply only one of non-uniform sampling and non-uniform loss scaling at a time rather than apply both simultaneously, as they provide roughly the same effect. Following encouraging results from earlier pilot experiments, we use power 0.75 task sampling and uniform loss scaling in the multitask learning experiments shown in Table 2.

D Additional Target Task Correlations

Tables 8, 9, and 10 respectively show the full target task correlations computed over pretraining, intermediate ELMo, and intermediate BERT experiments.

See Section 7 for a discussion about correlations for the pretraining experiments. The general trends in correlation vary significantly between the three experimental settings, which we take to roughly indicate the different types of knowledge encoded in ELMo and BERT. The exception is that WNLI is consistently negatively correlated with the other target tasks and often the overall GLUE score.

For intermediate ELMo experiments, correlations are generally low, with the exception of MNLI with other tasks. CoLA is negatively correlated with most other tasks, while QQP and SST are positively correlated with most tasks.

For intermediate BERT experiments, correlations with the GLUE score are quite high, as we found that intermediate training often negatively impacted GLUE score. QQP is highly negatively correlated with most other tasks, while the smaller tasks like MRPC and RTE are most highly correlated with overall GLUE score.

E Additional Learning Curves

Figure 3 shows learning curves reflecting the amount of target-task data required to train a model on each GLUE task, starting from three selected encoders. See Section 7 for discussion.

F Diagnostic Set Results

Tables 11 and 12 show results on the four coarse-grained categories of the GLUE diagnostic set for all our pretraining experiments. This set consists of about 1000 expert-constructed examples in NLI format meant to isolate a range of relevant phenomena. Results use the target task classifier trained on the MNLI training set.

No model achieves performance anywhere close to human-level performance, suggesting that *either* none of our pretrained models extract features that are suitable for robust reasoning over text, or that the MNLI training set and the MNLI target-task model are not able to exploit any such features that exist. See Section 7 for further discussion.

While no model achieves near-human performance, the use of ELMo and appears to be helpful on examples that highlight world knowledge and lexical-semantic knowledge, and less so on examples that highlight complex logical reasoning patterns or alternations in sentence structure. This relative weakness on sentence structure is somewhat surprising given the finding in Zhang and Bowman (2018) that language model pretraining is helpful for tasks involving sentence structure.

Using BERT helps significantly with understanding sentence structure, lexical knowledge, and logical reasoning, but does not seem to help on world knowledge over using ELMo. Encouragingly, we find that intermediate training of BERT on all of our pretraining tasks outperforms intermediate training on one or no tasks in two of the four categories.

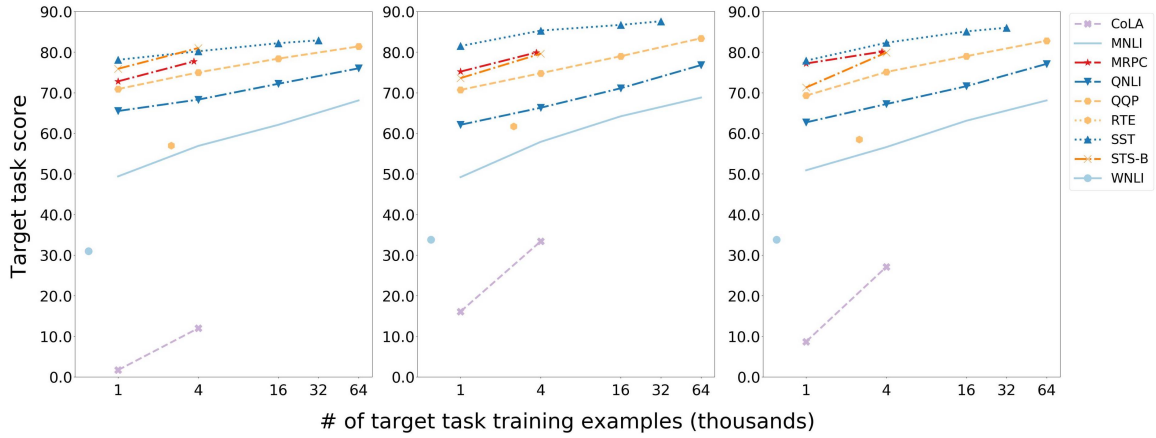

Figure 3: Target-task training learning curves for each GLUE task with three encoders: the random encoder without ELMo (left), random with ELMo (center), and MTL Non-GLUE pretraining (right).

Task	Avg	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI
CoLA	0.86	1.00								
SST	0.60	0.25	1.00							
MRPC	0.39	0.21	0.34	1.00						
STS	-0.36	-0.60	0.01	0.29	1.00					
QQP	0.61	0.61	0.27	-0.17	-0.58	1.00				
MNLI	0.54	0.16	0.66	0.56	0.40	0.08	1.00			
QNLI	0.43	0.13	0.26	0.32	0.04	0.27	0.56	1.00		
RTE	0.34	0.08	0.16	-0.09	-0.10	0.04	0.14	0.32	1.00	
WNLI	-0.21	-0.21	-0.37	0.31	0.31	-0.37	-0.07	-0.26	0.12	1.00

Table 8: Pearson correlations between performances on different target tasks, measured over all pretraining runs reported in Table 2.

Task	Avg	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI
CoLA	0.07	1.00								
SST	0.32	-0.48	1.00							
MRPC	0.42	-0.20	0.29	1.00						
STS	0.41	-0.40	0.26	0.21	1.00					
QQP	0.02	0.08	0.26	0.18	0.15	1.00				
MNLI	0.60	-0.21	0.33	0.38	0.72	0.21	1.00			
QNLI	0.50	0.10	0.03	0.12	0.63	-0.01	0.72	1.00		
RTE	0.39	-0.13	-0.15	0.21	0.27	-0.04	0.60	0.59	1.00	
WNLI	-0.14	0.02	0.23	-0.29	-0.02	0.15	0.02	-0.25	-0.22	1.00

Table 9: Pearson correlations between performances on different target tasks, measured over all ELMo runs reported in Table 3. Negative correlations are underlined.

Task	Avg	CoLA	SST	MRPC	STS	QQP	MNLI	QNLI	RTE	WNLI
CoLA	0.71	1.00								
SST	0.41	0.32	1.00							
MRPC	0.83	0.67	0.62	1.00						
STS	0.82	0.34	0.21	0.60	1.00					
QQP	-0.41	0.01	0.04	-0.05	-0.64	1.00				
MNLI	0.73	0.31	0.10	0.42	0.69	-0.68	1.00			
QNLI	0.73	0.38	0.29	0.56	0.43	-0.11	0.62	1.00		
RTE	0.88	0.47	0.22	0.56	0.87	-0.70	0.68	0.55	1.00	
WNLI	0.45	-0.10	-0.03	0.20	0.79	-0.89	0.65	0.11	0.69	1.00

Table 10: Pearson correlations between performances on different target tasks, measured over all BERT runs reported in Table 3. Negative correlations are underlined.

Pretr.	Knowledge	Lexical Semantics	Logic	Predicate/Argument Str.
		Baselines		
Random	17.6	19.6	12.5	26.9
		GLUE Tasks as Pretraining Tasks		
CoLA	15.3	24.2	14.9	**31.7**
SST	16.1	24.8	16.5	28.7
MRPC	16.0	**25.2**	12.6	26.4
QQP	12.8	22.5	12.9	30.8
STS	16.5	20.2	13.0	27.1
MNLI	16.4	20.4	**17.7**	29.9
QNLI	13.6	21.3	12.2	28.0
RTE	16.3	23.1	14.5	28.8
WNLI	**18.8**	19.5	13.9	29.1
		Non-GLUE Pretraining Tasks		
DisSent WP	18.5	24.2	15.4	27.8
LM WP	14.9	16.6	9.4	23.0
LM BWB	15.8	19.4	9.1	23.9
MT En-De	13.4	24.6	14.8	30.1
MT En-Ru	13.4	24.6	14.8	30.1
Reddit	13.9	20.4	14.1	26.0
SkipThought	15.1	22.0	13.7	27.9
		Multitask Pretraining		
MTL All	16.3	21.4	11.2	28.0
MTL GLUE	12.5	21.4	15.0	30.1
MTL Outside	14.5	19.7	13.1	26.2

Table 11: GLUE diagnostic set results, reported as R_3 correlation coefficients ($\times 100$), which standardizes the score of random guessing by an uninformed model at roughly 0. Human performance on the overall diagnostic set is roughly 80. Results in **bold** are the best overall.

Pretr.	Knowledge	Lexical Semantics	Logic	Predicate/Argument Str.
ELMo with Intermediate Task Training				
Random^E	19.2	22.9	9.8	25.5
CoLA^E	17.2	21.6	9.2	27.3
SST^E	19.4	20.5	9.7	28.5
MRPC^E	11.8	20.5	12.1	27.4
QQP^E	17.5	16.0	9.9	**30.5**
STS^E	18.0	18.4	9.1	25.5
MNLI^E	17.0	23.2	14.4	23.9
QNLI^E	17.4	**24.1**	10.7	30.2
RTE^E	18.0	20.2	8.7	28.0
WNLI^E	16.5	19.8	7.3	25.2
DisSent WP^E	16.3	23.0	11.6	26.5
MT En-De^E	19.2	21.0	13.5	29.7
MT En-Ru^E	20.0	20.1	11.9	21.4
Reddit^E	14.7	22.3	**15.0**	29.0
SkipThought^E	20.5	18.5	10.4	26.8
MTL GLUE^E	**20.6**	22.1	14.7	25.3
MTL Non-GLUE^E	15.7	23.7	12.6	29.0
MTL All^E	13.8	18.4	10.8	26.7
BERT with Intermediate Task Training				
Single-Task^B	20.3	36.3	21.7	40.4
CoLA^B	18.5	34.0	23.5	40.1
SST^B	19.8	36.0	23.2	39.1
MRPC^B	20.6	33.3	20.9	37.8
QQP^B	17.4	35.7	23.8	40.5
STS^B	**21.3**	34.7	24.0	40.7
MNLI^B	19.1	34.0	23.3	**41.7**
QNLI^B	20.3	38.4	24.4	41.5
RTE^B	15.4	32.6	20.2	38.5
WNLI^B	20.8	35.8	23.1	39.3
DisSent WP^B	17.9	34.0	23.7	39.1
MT En-De^B	18.6	33.8	20.7	37.4
MT En-Ru^B	14.2	30.2	20.3	36.5
Reddit^B	16.5	29.9	22.7	37.1
SkipThought^B	15.8	35.0	20.9	38.3
MTL GLUE^B	17.0	35.2	24.3	39.6
MTL Non-GLUE^B	18.7	37.0	21.8	40.6
MTL All^B	17.8	**40.3**	**27.5**	41.0

Table 12: GLUE diagnostic set results, reported as R_3 correlation coefficients ($\times 100$), which standardizes the score of random guessing by an uninformed model at roughly 0. Human performance on the overall diagnostic set is roughly 80. Results in **bold** are the best by section.