A Hyperparameters

All models use a fusion layer size of 300 and dropout (Srivastava et al., 2014) of 0.5 is applied to the input embeddings, the joint premise-hypothesis layer and all subsequent layers. These layers use an embedding size of 512. Our models are trained with Adam (Kingma and Ba, 2015) with initial learning rate of 0.003 which is exponentially decayed by 0.5 after each 10K iterations. A batch size of 64 is used. Hyperparameters were lightly tuned to optimize performance on the SNLI+MultiNLI development sets and are used for all of our experiments.

We observed that performance on the NLI development set and downstream tasks are correlated, up to a point. Beyond this, we found it's possible to obtain stronger models on NLI that perform worse on downstream evaluations, perhaps due to overfitting towards NLI. For example, using larger fusion layer sizes can obtain better NLI development set performance but tends to hurt on downstream tasks.

Conneau et al. (2017) reported that training with SGD led to increased generalization on downstream evaluations compared to Adam. We did not observe this difference. We noticed models trained with SGD did not overfit on NLI as severely, while models trained with Adam required early stopping. We observed 50K iterations of training was sufficient for our models and beyond this overfitting became an issue.

B Image-sentence retrieval

We experimented with image-sentence retrieval evaluations on COCO (Lin et al., 2014), comparing our models to existing pre-trained sentence embedding models from Skip-thoughts (Kiros et al., 2015), InferSent (Conneau et al., 2017) and Multitask embeddings (Subramanian et al., 2018). Models are evaluated using Recall@K and median rank. All models use ResNet-152 pre-trained features that are held fixed during training. Results are reported in Table 4. Surprisingly, the best performing model on average for these tasks is (glove+news,1), which is a bag-of-words model. It matches or outperforms Multitask on all evaluations and outperforms InferSent on all but one metric (R@10).

C Evaluating binary representations

In order to determine the effectiveness of our semantic hashing layer, we also evaluated binary representations on all downstream evaluations. We trained 3 models on NLI that are based on (glove,3): 256,1024 and 4096-bit codes. Results are reported in Tables 5 and 6 and include the baseline (glove,3) 4096-dim embeddings as a reference baseline. Unsurprisingly, we observe a performance hit across all tasks, with some downstream tasks being affected more than others. Interestingly, both 1024 and 4096-bit codes perform comparable on most tasks. However, moving the dimensionality down to 256 results in a significant performance drop across almost all tasks. In future work we intend to explore InferLite binary codes for large-scale retrieval evaluations. The fact that our binary codes still perform effectively on most STS tasks, as well as MRPC, loosely indicates that they should also be effective for retrieval.

D Probing tasks

We also evaluate our models on the 10 probing tasks introduced by Conneau et al. (2018). The goal of these evaluations is to more effectively understand what properties are encoded in generic sentence embeddings. We refer the reader to Conneau et al. (2018) for full task descriptions. Some tasks, such as BShift and SOMO require context, consequently bag-of-word models obtain random performance. We compare our models to InferSent as well as the GatedConv model introduced in Conneau et al. (2018), which shares many similarities to our own models. Results for these tasks are in Table 7. Note that our models do not make use of positional embeddings.

E Ablation study

In order to better understand which components of our model contribute most to downstream performance, we perform an ablation study. Here, we consider 6 components and train InferLite with each of these components removed, fixing the rest of the model. Each modification we control for is described below:

Mean Pooling. We replace max pooling reduction with mean pooling:

$$s = \text{meanpool}\{F\}_T \tag{9}$$

In Conneau et al. (2017) it was shown that mean pool performs significantly worse than max pooling on downstream evaluations. This experiment is to verify the same result holds for InferLite.

	Image Annotation				Image Search			
Model	R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
ST (Kiros et al., 2015)	37.9	72.2	84.3	2	30.6	66.2	81.0	3
InferSent (Conneau et al., 2017)	42.6	75.3	87.3	2	33.9	69.7	83.8	3
Multitask (Subramanian et al., 2018)	43.0	76.0	87.0	2	33.8	70.1	83.6	2.8
glove,1	42.6	75.9	87.0	2	33.5	69.9	83.8	2.8
glove+news,1	**43.8**	76.6	87.1	2	**34.3**	**70.3**	**84.2**	2.8
glove+query,1	43.1	76.5	87.8	2	33.6	69.6	83.7	2.8
glove+news+query,1	43.0	76.6	87.7	2	33.9	69.6	83.8	2.8
glove,3	42.6	**76.8**	**88.0**	2	33.8	69.7	83.6	2.8
glove+news,3	42.7	76.5	87.5	2	34.0	69.7	83.6	2.8
glove+query,3	43.5	76.7	87.7	2	34.2	70.1	84.1	**2.6**
glove+news+query,3	42.9	76.1	87.0	2	33.9	70.0	84.0	**2.6**

Table 4: COCO test-set results for image-sentence retrieval experiments. All models use ResNet-152 pre-trained features. R@K is Recall@K (high is good). Med *r* is the median rank (low is good). Best results are bolded.

Model	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC
semhash,256	73.7	81.2	83.2	86.2	78.4	59.0	71.6/80.9
semhash,1024	76.3	83.2	87.8	88.4	81.3	74.0	74.4/82.2
semhash,4096	77.7	83.7	89.6	89.1	82.3	78.6	74.9/82.4
glove,3	**80.9**	**84.1**	**92.4**	**89.6**	**85.8**	**90.0**	**76.5/83.4**

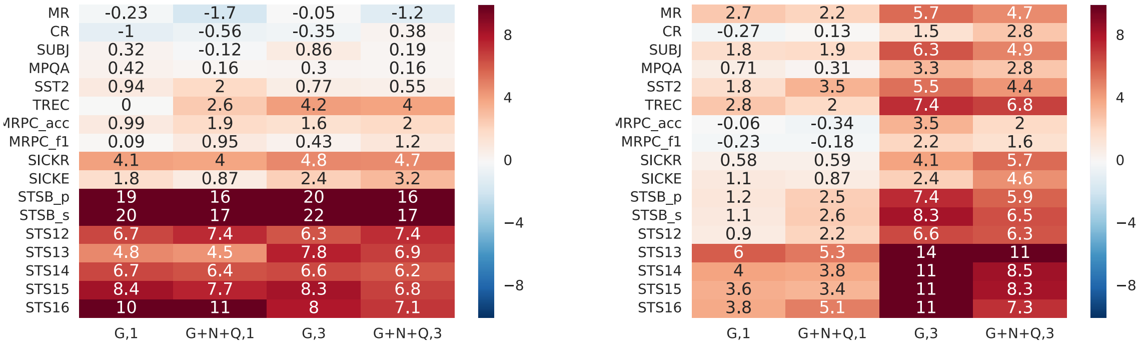
Table 5: Comparison of embedding methods on downstream evaluations. Each set of results is a) bag-of-words b) RNN and Transformer c) ours, filter length 1 and d) ours, filter length 3. Last column is training time in hours.

Model	SICK-R	SICK-E	STSB	STS12	STS13	STS14	STS15	STS16
semhash,256	79.9	79.0	67.9/67.6	56.5	49.1	61.2	67.3	65.6
semhash,1024	82.8	82.9	64.9/64.9	59.3	51.5	67.0	71.5	70.8
semhash,4096	81.2	83.4	63.4/63.3	61.4	53.4	68.0	71.9	70.9
glove,3	**88.1**	**85.5**	**78.4/78.3**	**61.9**	**61.3**	**71.7**	**74.5**	**71.2**

Table 6: Comparison of embedding methods on downstream evaluations. Each sets of results are a) bag-of-words b) RNN encoders c) ours, filter length 1 d) ours, filter length 3.

Model	Len	WC	Depth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
BoV-fastText	54.8	91.6	32.3	63.1	50.8	**87.8**	**81.9**	**79.3**	50.3	52.7
BiLSTM-Max NLI	65.1	87.3	38.5	67.9	63.8	86.0	78.9	78.5	59.5	64.9
GatedConv NLI	**70.9**	29.2	**38.8**	59.3	**66.8**	80.1	77.7	72.8	**69.0**	**69.1**
glove,1	63.5	92.9	33.3	74.1	49.8	84.4	77.8	74.0	51.6	54.8
glove+news,1	62.8	95.1	33.8	74.5	50.6	84.7	78.1	75.6	51.0	55.1
glove+query,1	62.2	95.1	33.6	75.1	50.3	83.7	77.4	74.1	51.2	55.3
glove+news+query,1	63.3	**95.2**	34.0	75.0	50.7	84.5	78.0	75.3	51.3	54.2
glove,3	63.3	88.3	33.5	75.9	57.2	84.0	78.6	74.0	52.1	61.2
glove+news,3	63.4	90.3	34.1	77.8	59.7	84.8	78.4	74.7	51.4	60.8
glove+query,3	62.7	90.5	34.4	77.5	59.6	85.4	77.7	73.6	51.8	60.6
glove+news+query,3	62.3	90.7	33.6	**78.1**	59.6	83.6	78.7	74.9	51.7	60.5

Table 7: Results on probing tasks. All comparisons use logistic regression for training on each task.

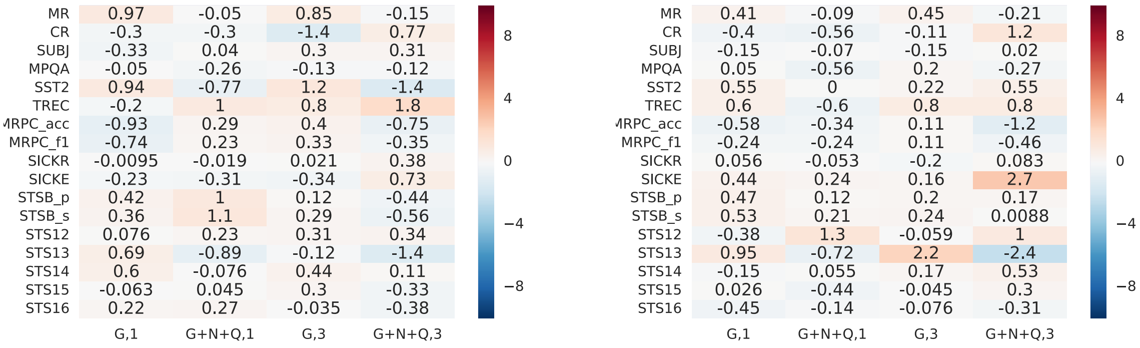


	G,1	G+N+Q,1	G,3	G+N+Q,3
MR	-0.23	-1.7	-0.05	-1.2
CR	-1	-0.56	-0.35	0.38
SUBJ	0.32	-0.12	0.86	0.19
MPQA	0.42	0.16	0.3	0.16
SST2	0.94	2	0.77	0.55
TREC	0	2.6	4.2	4
RPC_acc	0.99	1.9	1.6	2
MRPC_f1	0.09	0.95	0.43	1.2
SICKR	4.1	4	4.8	4.7
SICKE	1.8	0.87	2.4	3.2
STSB_p	19	16	20	16
STSB_s	20	17	22	17
STS12	6.7	7.4	6.3	7.4
STS13	4.8	4.5	7.8	6.9
STS14	6.7	6.4	6.6	6.2
STS15	8.4	7.7	8.3	6.8
STS16	10	11	8	7.1

(a) Mean pooling

	G,1	G+N+Q,1	G,3	G+N+Q,3
MR	2.7	2.2	5.7	4.7
CR	-0.27	0.13	1.5	2.8
SUBJ	1.8	1.9	6.3	4.9
MPQA	0.71	0.31	3.3	2.8
SST2	1.8	3.5	5.5	4.4
TREC	2.8	2	7.4	6.8
RPC_acc	-0.06	-0.34	3.5	2
MRPC_f1	-0.23	-0.18	2.2	1.6
SICKR	0.58	0.59	4.1	5.7
SICKE	1.1	0.87	2.4	4.6
STSB_p	1.2	2.5	7.4	5.9
STSB_s	1.1	2.6	8.3	6.5
STS12	0.9	2.2	6.6	6.3
STS13	6	5.3	14	11
STS14	4	3.8	11	8.5
STS15	3.6	3.4	11	8.3
STS16	3.8	5.1	11	7.3

(b) No Skip connection

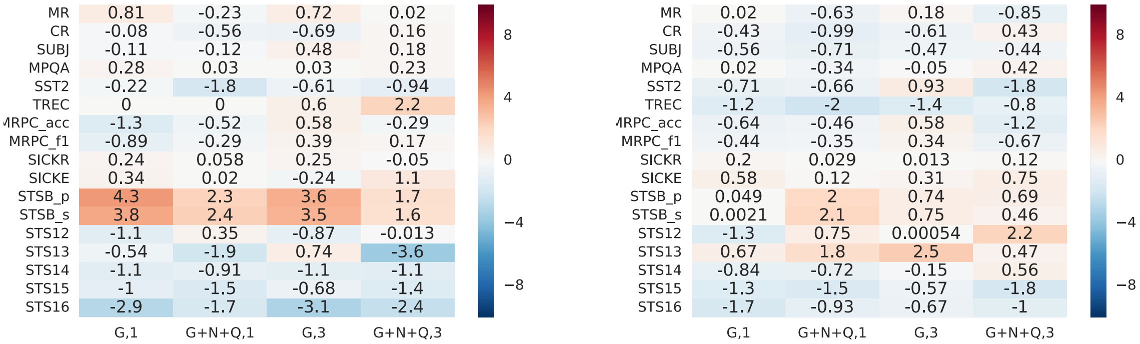


	G,1	G+N+Q,1	G,3	G+N+Q,3
MR	0.97	-0.05	0.85	-0.15
CR	-0.3	-0.3	-1.4	0.77
SUBJ	-0.33	0.04	0.3	0.31
MPQA	-0.05	-0.26	-0.13	-0.12
SST2	0.94	-0.77	1.2	-1.4
TREC	-0.2	1	0.8	1.8
RPC_acc	-0.93	0.29	0.4	-0.75
MRPC_f1	-0.74	0.23	0.33	-0.35
SICKR	-0.0095	-0.019	0.021	0.38
SICKE	-0.23	-0.31	-0.34	0.73
STSB_p	0.42	1	0.12	-0.44
STSB_s	0.36	1.1	0.29	-0.56
STS12	0.076	0.23	0.31	0.34
STS13	0.69	-0.89	-0.12	-1.4
STS14	0.6	-0.076	0.44	0.11
STS15	-0.063	0.045	0.3	-0.33
STS16	0.22	0.27	-0.035	-0.38

(c) No gating

	G,1	G+N+Q,1	G,3	G+N+Q,3
MR	0.41	-0.09	0.45	-0.21
CR	-0.4	-0.56	-0.11	1.2
SUBJ	-0.15	-0.07	-0.15	0.02
MPQA	0.05	-0.56	0.2	-0.27
SST2	0.55	0	0.22	0.55
TREC	0.6	-0.6	0.8	0.8
RPC_acc	-0.58	-0.34	0.11	-1.2
MRPC_f1	-0.24	-0.24	0.11	-0.46
SICKR	0.056	-0.053	-0.2	0.083
SICKE	0.44	0.24	0.16	2.7
STSB_p	0.47	0.12	0.2	0.17
STSB_s	0.53	0.21	0.24	0.0088
STS12	-0.38	1.3	-0.059	1
STS13	0.95	-0.72	2.2	-2.4
STS14	-0.15	0.055	0.17	0.53
STS15	0.026	-0.44	-0.045	0.3
STS16	-0.45	-0.14	-0.076	-0.31

(d) $M = 1$ convolutional layer



	G,1	G+N+Q,1	G,3	G+N+Q,3
MR	0.81	-0.23	0.72	0.02
CR	-0.08	-0.56	-0.69	0.16
SUBJ	-0.11	-0.12	0.48	0.18
MPQA	0.28	0.03	0.03	0.23
SST2	-0.22	-1.8	-0.61	-0.94
TREC	0	0	0.6	2.2
RPC_acc	-1.3	-0.52	0.58	-0.29
MRPC_f1	-0.89	-0.29	0.39	0.17
SICKR	0.24	0.058	0.25	-0.05
SICKE	0.34	0.02	-0.24	1.1
STSB_p	4.3	2.3	3.6	1.7
STSB_s	3.8	2.4	3.5	1.6
STS12	-1.1	0.35	-0.87	-0.013
STS13	-0.54	-1.9	0.74	-3.6
STS14	-1.1	-0.91	-1.1	-1.1
STS15	-1	-1.5	-0.68	-1.4
STS16	-2.9	-1.7	-3.1	-2.4

(e) No embedding norm

	G,1	G+N+Q,1	G,3	G+N+Q,3
MR	0.02	-0.63	0.18	-0.85
CR	-0.43	-0.99	-0.61	0.43
SUBJ	-0.56	-0.71	-0.47	-0.44
MPQA	0.02	-0.34	-0.05	0.42
SST2	-0.71	-0.66	0.93	-1.8
TREC	-1.2	-2	-1.4	-0.8
RPC_acc	-0.64	-0.46	0.58	-1.2
MRPC_f1	-0.44	-0.35	0.34	-0.67
SICKR	0.2	0.029	0.013	0.12
SICKE	0.58	0.12	0.31	0.75
STSB_p	0.049	2	0.74	0.69
STSB_s	0.0021	2.1	0.75	0.46
STS12	-1.3	0.75	0.00054	2.2
STS13	0.67	1.8	2.5	0.47
STS14	-0.84	-0.72	-0.15	0.56
STS15	-1.3	-1.5	-0.57	-1.8
STS16	-1.7	-0.93	-0.67	-1

(f) No time-distributed layers

Figure 3: Ablation study results. **G** stands for Glove, **N** stands for News and **Q** stands for Query embeddings.

No skip connection. In the fusion layer, the skip connection is removed:

$$F' = \sum_{k=1}^{K} H_M^k \odot G_M^k \quad (10)$$

$$F = \phi_f(W_f F' + b_f) \quad (11)$$

While this is a simple modification, we show the resulting fusion layer has a dramatic effect on downstream performance.

No gating. The embeddings of each individual component are summed without applying any multiplicative gating:

$$F' = \left(\sum_{k=1}^{K} H_M^k \right) + G_0^c \quad (12)$$

$$F = \phi_f(W_f F' + b_f) \quad (13)$$

In this case, the entire gating module is removed with the exception of G_0^c.

1 convolutional layer. We use $M = 1$ convolutional layers instead of $M = 3$.

No embedding normalization. All embeddings are left unnormalized.

No translation (time-distributed) layers. All time-distributed layers originating from the input embeddings are removed.

The results of this study are in Figure 3. We plot the performance difference of InferLite with the ablation version. Positive results indicate that the original InferLite model performs better, while negative results indicate that the ablation performs better. We observe that max-pooling and the skip connection is critical to the InferLite model. The effect of gating and convolutional (M = 3) generally helps overall. However, embedding normalization and time-distributed layers help in some tasks but hurt in others.

F Inference speed

For this experiment, we report timing results for encoding 1 million sentences from the 1 Billion word benchmark (Chelba et al., 2013), which has an average word length of 25. We use the semantic hashing version of InferLite (glove,3), running locally on a single TitanXp GPU with a batch size of 512 sequences. Table 8 reports the InferLite inference speeds for various dimensions. Reported times do not include time to sort sequences by word length.

Dimension	Samples / Second
256	11.9k/sec
1024	11.4k/sec
4096	9.8k/sec

Table 8: InferLite inference speed.

G Gating visualizations

Figure 4 shows examples of mean gate activations across words for 6 sentences, using our (glove+news+query,3) model. In all cases, news embeddings tend to have the highest mean activation. Note how the mean activation values change across words for different sentences, for example 'man', 'a' and '.' all tend to have different activation values depending on the sentence. Models that do not use context would have fixed activation values for each word independent of other words in the sentence.

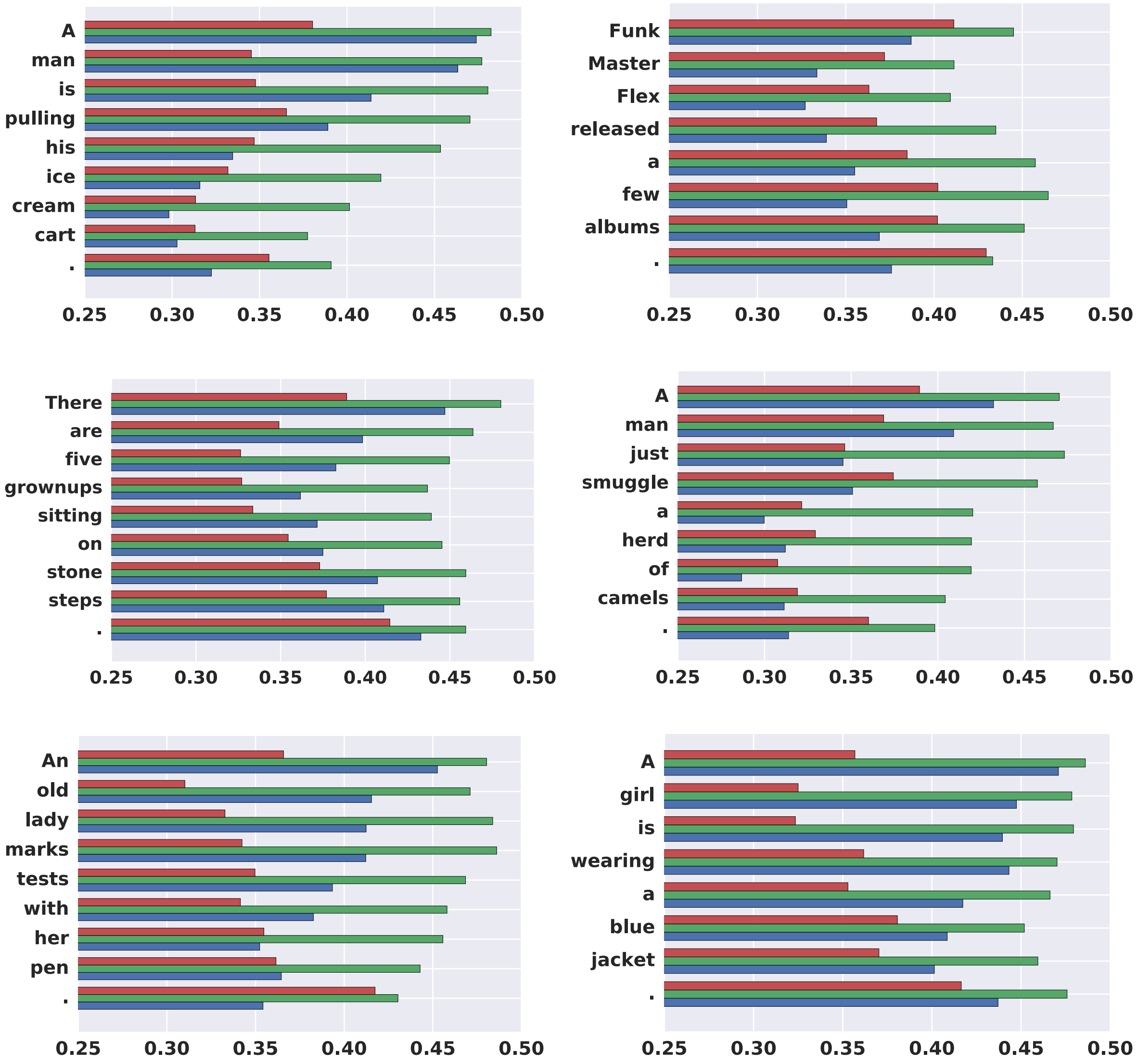


Figure 4: Visualizations of mean gate activations for each word in several sentences from the NLI development set. First (red) bar is Glove, second (green) bar is News and the third (blue) bar is Query.