	Before Perturb					After Perturb			
	dev	test	$\text{test}_{\text{simple}}$	$\text{test}_{\text{complex}}$	$\text{test}_{\text{small}}$	test	$\text{test}_{\text{simple}}$	$\text{test}_{\text{complex}}$	$\text{test}_{\text{small}}$
Eisenschlos et al. (2020)	81.0	81.0	92.3	75.6	83.9	–	–	–	–
Eisenschlos et al. (2021)	–	81.4	–	–	–	–	–	–	–
TAPAS$_{\text{BASE}}$	72.8	72.3	84.8	66.2	74.4	71.2	83.4	65.2	72.5
TABLEFORMER$_{\text{BASE}}$	**75.1**	**75.0**	**88.2**	**68.5**	**77.1**	**75.0**	**88.2**	**68.5**	**77.1**
TAPAS$_{\text{LARGE}}$	74.7	74.5	86.6	68.6	76.8	73.7	86.0	67.7	76.1
TABLEFORMER$_{\text{LARGE}}$	**77.2**	**77.0**	**90.2**	**70.5**	**80.3**	**77.0**	**90.2**	**70.5**	**80.3**
TAPAS$_{\text{BASE}}$ inter	78.4	77.9	90.1	71.9	80.5	76.8	89.5	70.5	79.7
TABLEFORMER$_{\text{BASE}}$ inter	**79.7**	**79.2**	**91.6**	**73.1**	**81.7**	**79.2**	**91.6**	**73.1**	**81.7**
TAPAS$_{\text{LARGE}}$ inter	80.6	80.6	92.0	74.9	83.1	79.2	91.7	73.0	83.0
TABLEFORMER$_{\text{LARGE}}$ inter	**82.0**	**81.6**	**93.3**	**75.9**	**84.6**	**81.6**	**93.3**	**75.9**	**84.6**

Table 2: Binary classification accuracy on TABFACT development and 4 splits of test set, as well as performance on test sets with our perturbation evaluation. Median of 5 independent runs are reported. Missing values are those not reported in the original paper.

Model	dev	test
Herzig et al. (2020)	–	48.8
Eisenschlos et al. (2021)	–	51.5
TAPAS$_{\text{BASE}}$	23.6	24.1
TABLEFORMER$_{\text{BASE}}$	**34.4**	**34.8**
TAPAS$_{\text{LARGE}}$	40.8	41.7
TABLEFORMER$_{\text{LARGE}}$	**42.5**	**43.9**
TAPAS$_{\text{BASE}}$ inter-sqa	44.8	45.1
TABLEFORMER$_{\text{BASE}}$ inter-sqa	**46.7**	**46.5**
TAPAS$_{\text{LARGE}}$ inter-sqa	49.9	50.4
TABLEFORMER$_{\text{LARGE}}$ inter-sqa	**51.3**	**52.6**

Table 3: Denotation accuracy on WTQ development and test set. Median of 5 independent runs are reported.

on similar perturbations, both in terms of performance and robustness?

- Can we use perturbation based data augmentation to achieve robustness at test time?
- Which attention biases in TABLEFORMER contribute the most to performance?

5.1 Main Results

Table 1, 2, and 3 shows TABLEFORMER performance on SQA, TABFACT, and WTQ, respectively. As can be seen, TABLEFORMER outperforms corresponding TAPAS baseline models in all settings on SQA and WTQ datasets, which shows the general effectiveness of TABLEFORMER's structural biases in Table QA datasets. Specifically, TABLEFORMER$_{\text{LARGE}}$ combined with intermediate pretraining achieves new state-of-the-art performance on SQA dataset.

Similarly, Table 2 shows that TABLEFORMER also outperforms TAPAS baseline models in all settings, which shows the effectiveness of TABLEFORMER in the table entailment task. Note that, Liu et al. (2021) is not comparable to our results, because they used different pretraining data, different pretraining objectives, and BART NLG model instead of BERT NLU model. But TABLEFORMER attention bias is compatible with BART model.

5.2 Perturbation Results

One of our major contributions is to systematically evaluate models' performance when facing row and column order perturbation in the testing stage.

Ideally, model predictions should be consistent on table QA and entailment tasks when facing such perturbation, because the table semantics remains the same after perturbation.

However, in Table 1 and 2, we can see that in our perturbed test set, performance of all TAPAS models drops significantly in both tasks. TAPAS models drops by at least 3.7% and up to 6.5% in all settings on SQA dataset in terms of ALL accuracy, while our TABLEFORMER being strictly invariant to such row and column order perturbation leads to no drop in performance.[4] Thus, in the perturbation setting, TABLEFORMER outperforms all TAPAS baselines even more significantly, with at least 6.2% and 2.4% improvements on SQA and TABFACT dataset, respectively. In the instance level, we can see that, with TAPAS, there are many example predictions changed due to high VP, while there is nearly no

[4] In SQA dataset, there is at most absolute 0.1% performance drop because of some bad data point issues. Specifically, some columns in certain tables are exactly the same, but the ground-truth selected cells are in only one of such columns. TABLEFORMER would select from one column randomly.