Appendix A. Accuracy in zero-shot MLM setting

While little work has explored few-shot knowledge completion, recent works have investigated performance of zero-shot knowledge graphs (**??**). Thus, we investigate the ability of T5-11B to complete commonsense knowledge in a zero-shot setting.

Model	BLEU-1	METEOR	ROUGE-L	CIDEr
T5 - Zero-shot	6.7	7.8	7.3	7.3
T5 - Few-shot ($n = 3$)	31.9	18.7	26.0	19.8
T5 - Fully-supervised	48.2	34.1	50.0	66.4

Table 1: Zero-shot performance of T5-11B

We use prompts to leverage the masking objective of the language model pretraining. Since the mask only predicts several tokens at a time, for relations with longer length tail entities, we allow the model to predict up to 7 mask tokens in succession, or until the model predicts an empty string for the mask. We suggest that this is still only a workaround, and masked models are poor predictors of longer length tail entities, as indicated by our results above.

Appendix B. Additional Results on the Effect of Relation Input Format

In Table 2, we provide further experimental results on the effect of relation input format on few-shot performance. Namely, we extend the results from Table **??** with the case $n = 300$. These results confirm that knowledge models can efficiently learn from fewer examples when relations are represented using natural language prompts. We also show the results from using paraphrases of the main prompts for training (original and paraphrased prompts can be found in Tables 3 and 5, respectively). We find that the paraphrased prompts do cause a slight drop in performance, but that this drop is generally close to the margin of error, indicating that while prompt formulation is an important consideration (**?**), fine-tuning on the prompts does make the model less sensitive to prompt variations.

# Ex	Input	BLEU-1	METEOR	ROUGE-L	CIDEr
3	Prompts	**24.2** ± 1.7	**14.4** ± 1.9	**21.3** ± 3.7	**18.1** ± 7.9
	Embedding	13.9 ± 1.3	11.4 ± 1.2	13.5 ± 0.9	7.4 ± 0.8
30	Prompts	**31.9** ± 0.3	**18.7** ± 0.5	**26.0** ± 0.6	**19.8** ± 1.2
	Embedding	18.1 ± 0.8	13.5 ± 1.1	16.7 ± 1.1	9.7 ± 1.7
	Shuffled Prompts	15.6 ± 0.8	11.3 ± 0.5	14.2 ± 0.5	8.3 ± 0.8
	Paraphrased	30.5 ± 1.3	18.3 ± 0.5	24.5 ± 0.8	18.4 ± 1.4
300	Prompts	**39.1** ± 0.9	**26.7** ± 0.9	**40.1** ± 1.5	**48.5** ± 2.8
	Embedding	25.2 ± 0.8	18.5 ± 0.9	26.4 ± 1.9	26.8 ± 4.3

Table 2: Effect of relation input format on few-shot performance. Prompts accelerate few-shot commonsense interface learning. We show mean performance over 5 random splits of training examples, and standard deviation (±) between splits.

Relation	Template
ObjectUse	{} is used for
AtLocation	You are likely to find {} in
MadeUpOf	{} is made up of
HasProperty	{} is
CapableOf	{} can
Desires	{} wants
NotDesires	{} does not want
isAfter	Something that happens after {} is
HasSubEvent	Something you might do while {} is
isBefore	Something that happens before {} is
HinderedBy	{} is hindered by
Causes	Sometimes {} causes
xReason	{}. The reason for PersonX doing this is
isFilledBy	{} can be filled by
xNeed	But before {}, PersonX needed
xAttr	{} is seen as
xEffect	As a result of {}, PersonX will
xReact	As a result of {}, PersonX feels
xWant	After {}, PersonX would want
xIntent	Because of {}, PersonX wanted
oEffect	as a result of {}, others will
oReact	as a result of {}, others would feel
oWant	as a result of {}, others would want

Table 3: Prompts used for relations in ATOMIC2020.

Something you might do while {} is ___
Something you might do while design software is determine deliverables
Something you might do while scuba dive is take off scuba gear
Something you might do while play ball is put on mitt

Table 4: Example of augmentation experiments, following (**?**). {} indicates the location of the head, and the language model is asked to complete the tail by finishing the sentence.

Relation	Template
ObjectUse	a {} can be used for
AtLocation	You could find {} in the location
MadeUpOf	{} is made up of
HasProperty	{} will have
CapableOf	{} is capable of
Desires	a {} desires
NotDesires	a {} does not desire
isAfter	Before {},
HasSubEvent	You might do {} while doing
isBefore	After {},
HinderedBy	{}. This is hindered by
Causes	Sometimes {} causes
xReason	{}. PersonX did this because
isFilledBy	{} is filled
xNeed	Before {}, PersonX needs to
xAttr	{}. An attribute of PersonX is
xEffect	The effect of {} PersonX will be
xReact	As a result of {}. PersonX will be
xWant	After {}, PersonX will want to
xIntent	For {}, PersonX did this to
oEffect	An effect of {} on others will be
oReact	As a result of {}, other feel
oWant	After {}, others will want to

Table 5: Paraphrased version of the prompts used (for paraphrased experiment in the Appendix.)

Appendix C. Additional Experiments on Parameter Change Measures

In addition, we provide extensive results of the ℓ_1 and angular distances, as well as the distributional parameter change metric (AUC), for both the encoder and decoder of various model sizes (Small, Large, and 11B) and different example budget ($n \in \{3, 30, 300\}$) (Figures 1-30). When computing AUC diagrams, we round each weight change to the nearest 10^{-5}.

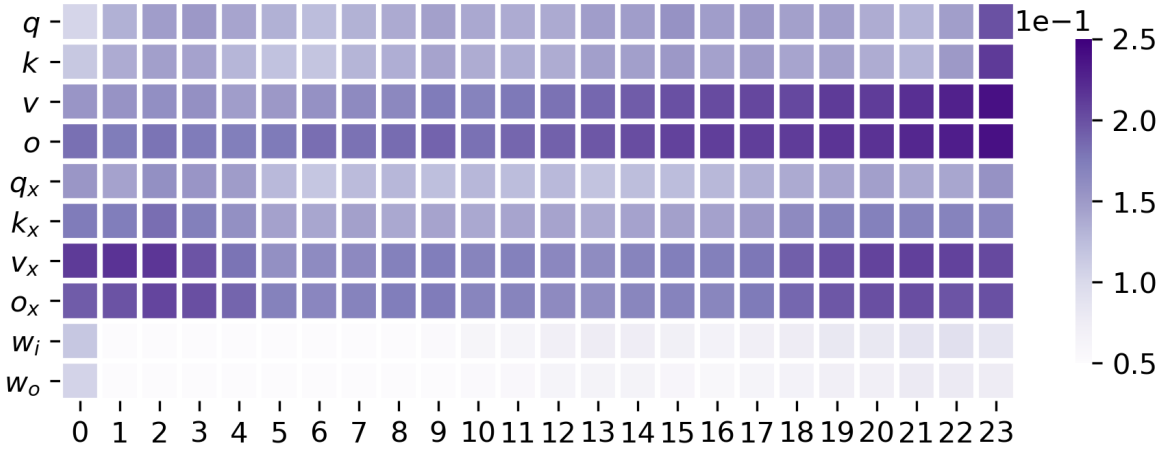

Figure 1: Area Under Curve, Decoder, T5-11B, $n = 30$

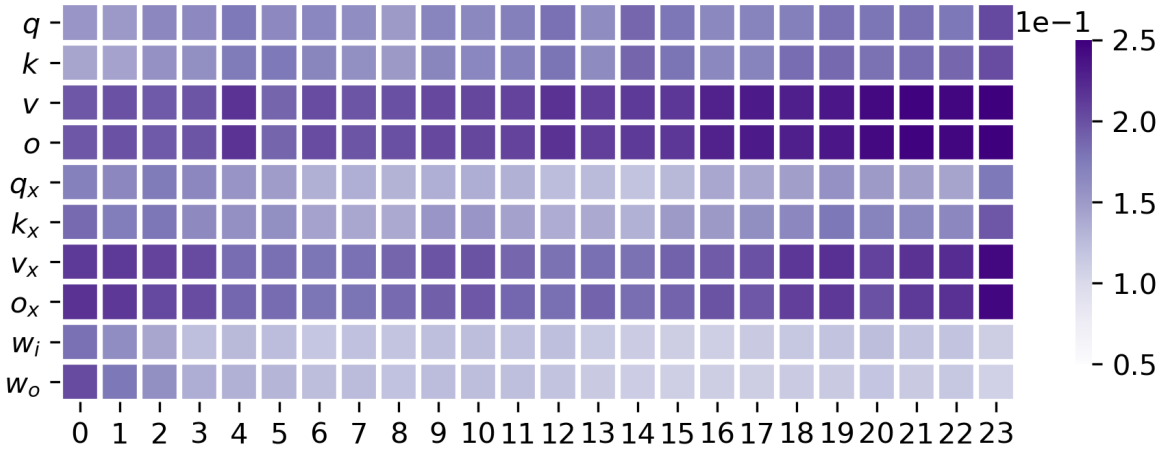

Figure 2: Area Under Curve, Decoder, T5-Large, $n = 30$

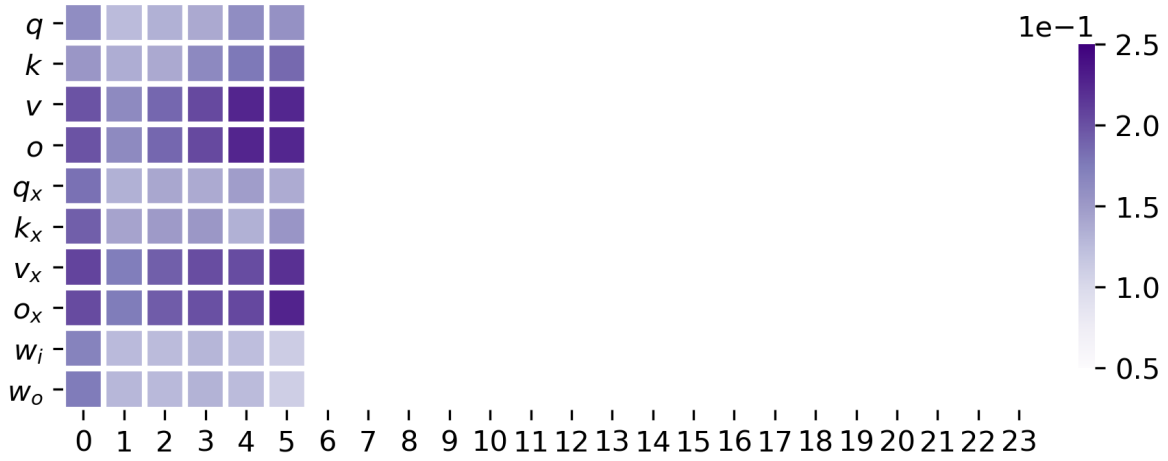

Figure 3: Area Under Curve, Decoder, T5-Small, $n = 30$

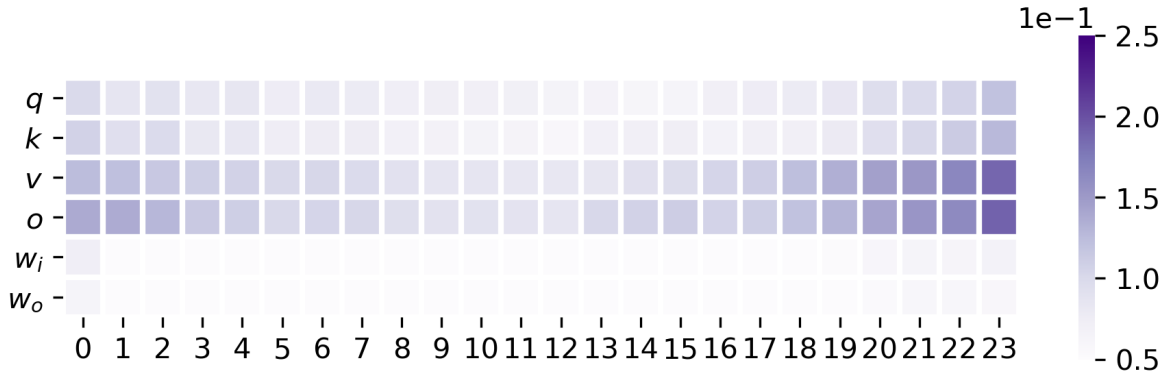

Figure 4: Area Under Curve, Encoder, T5-11B, $n = 30$

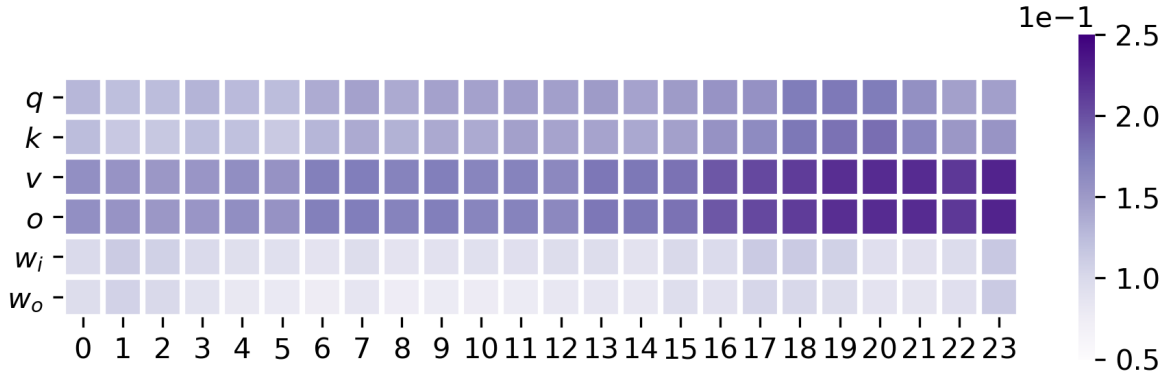

Figure 5: Area Under Curve, Encoder, T5-Large, $n = 30$

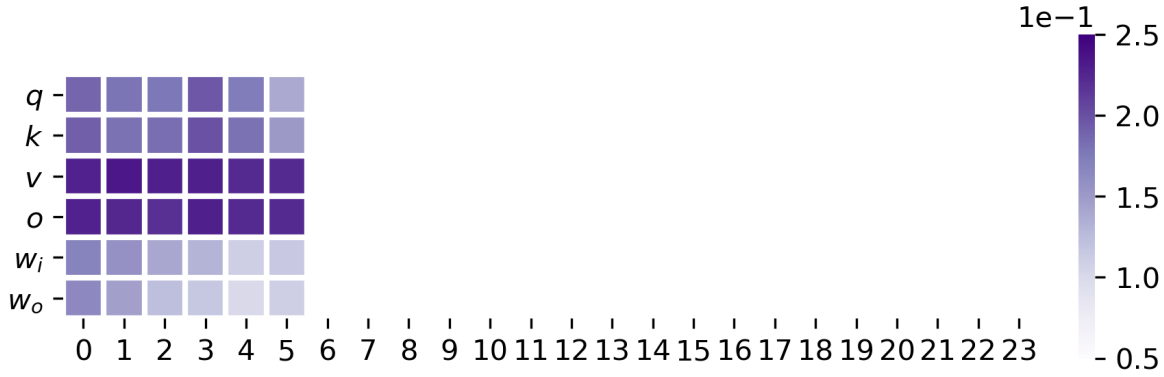

Figure 6: Area Under Curve, Encoder, T5-Small, $n = 30$

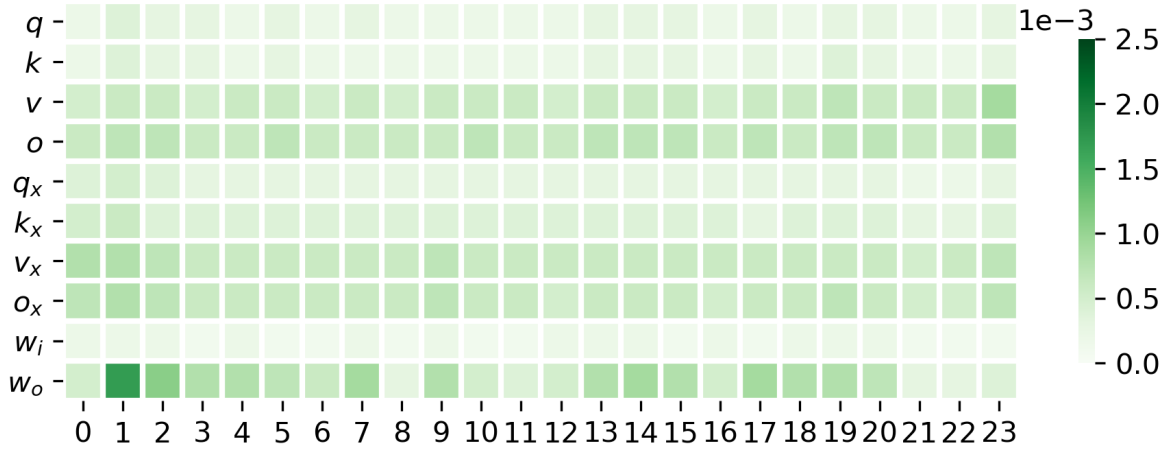

Figure 7: Angular change, Decoder, T5-11B, $n = 3$

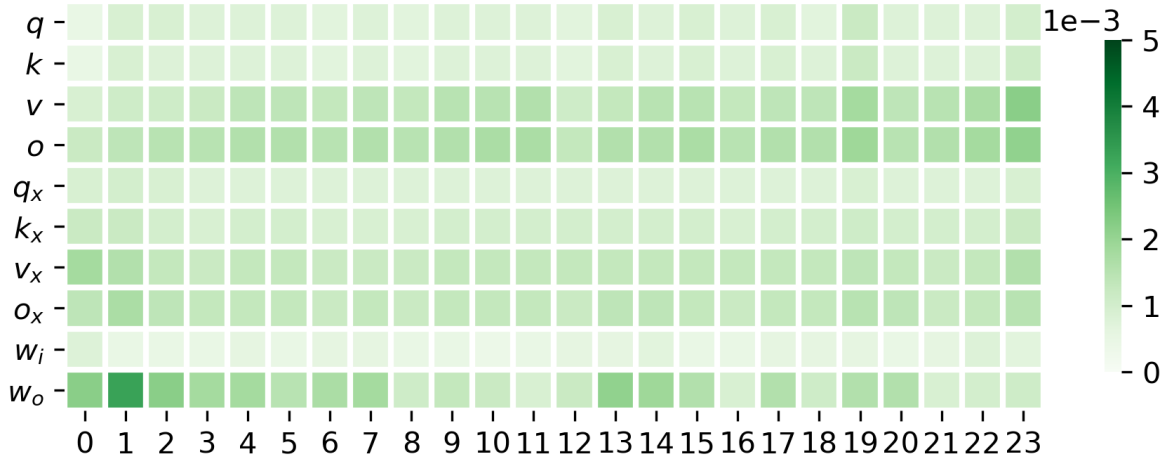

Figure 8: Angular change, Decoder, T5-11B, $n = 30$

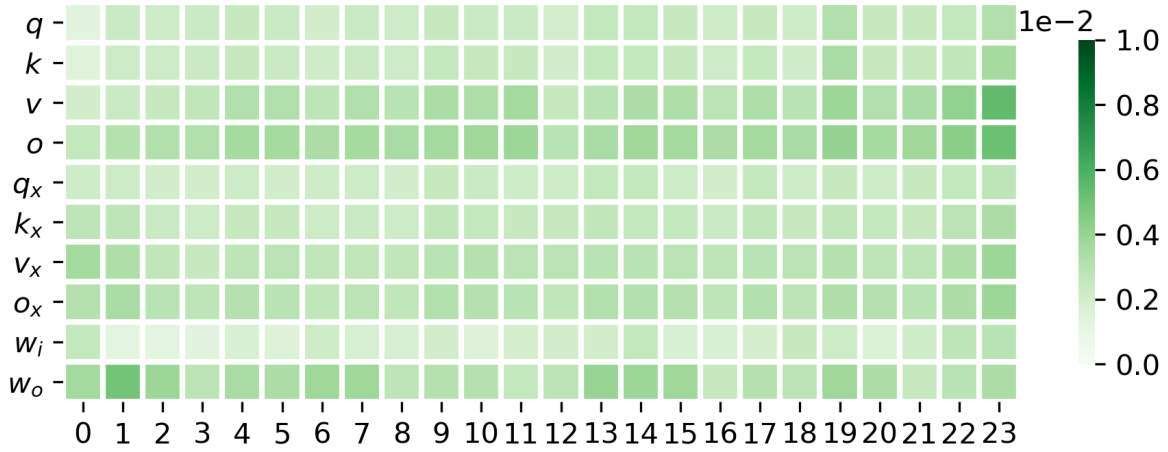

Figure 9: Angular change, Decoder, T5-11B, $n = 300$

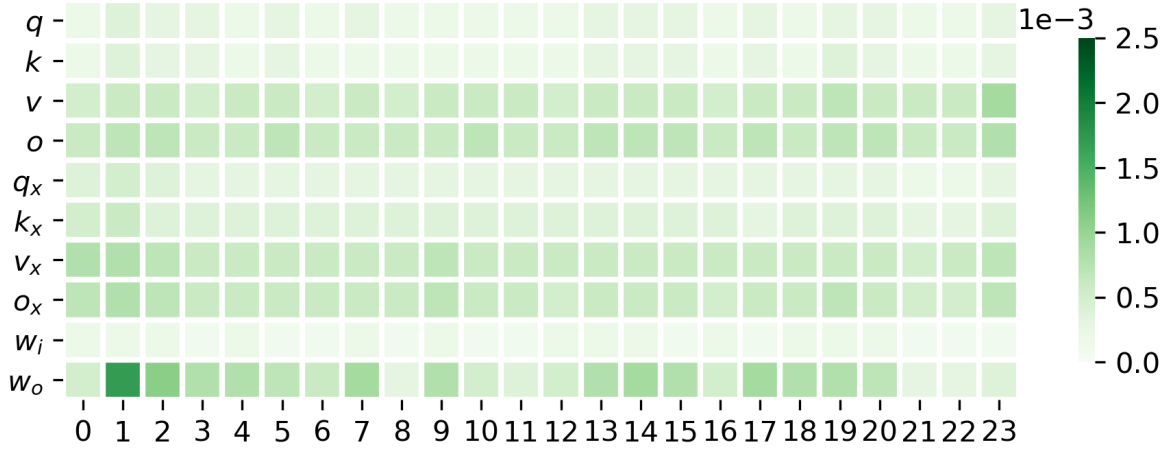

Figure 10: Angular change, Encoder, T5-11B, $n = 3$

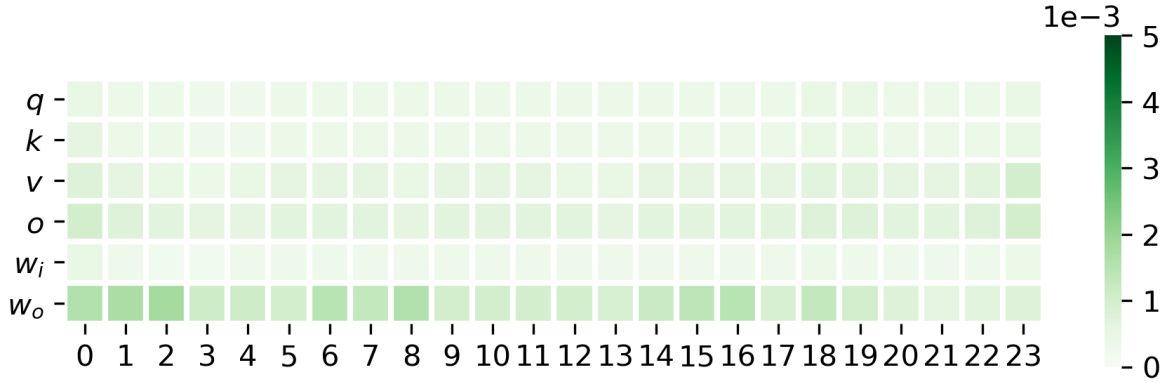

Figure 11: Angular change, Encoder, T5-11B, $n = 30$

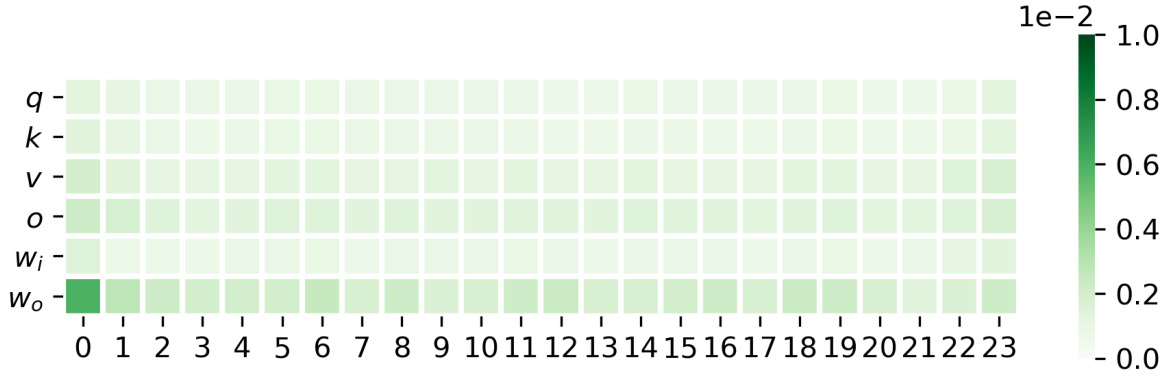

Figure 12: Angular change, Encoder, T5-11B, $n = 300$

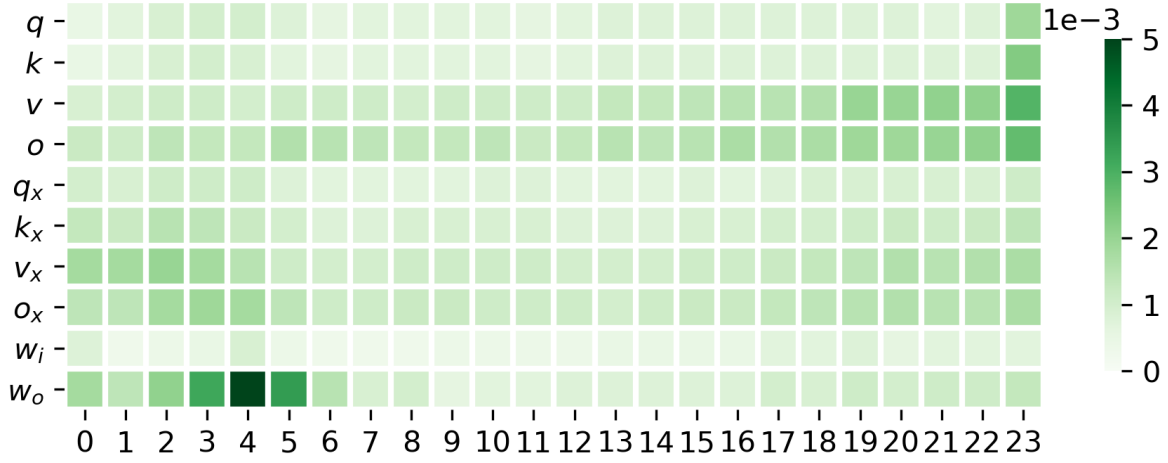

Figure 13: Angular change, Decoder, T5-11B (relation embedding), $n = 30$

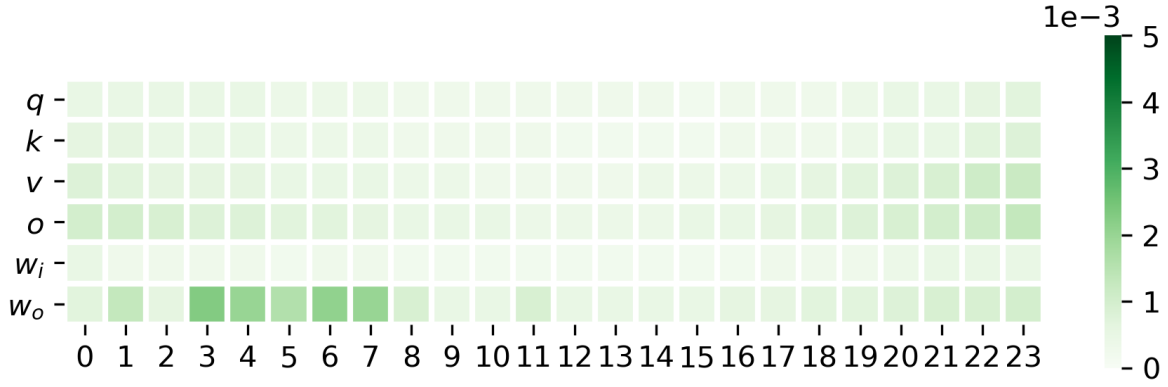

Figure 14: Angular change, Encoder, T5-11B (relation embedding), $n = 30$

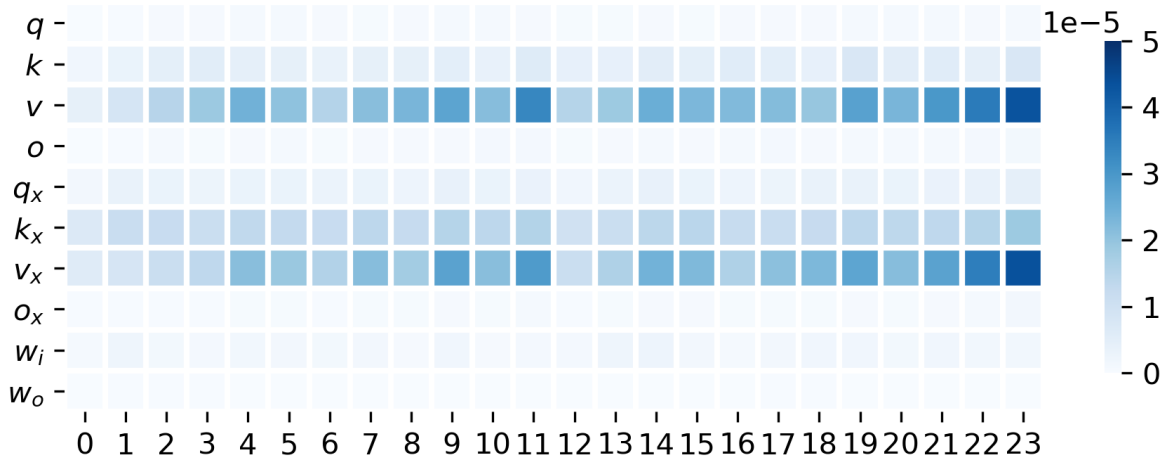

Figure 15: L1 change, Decoder, T5-11B, $n = 3$

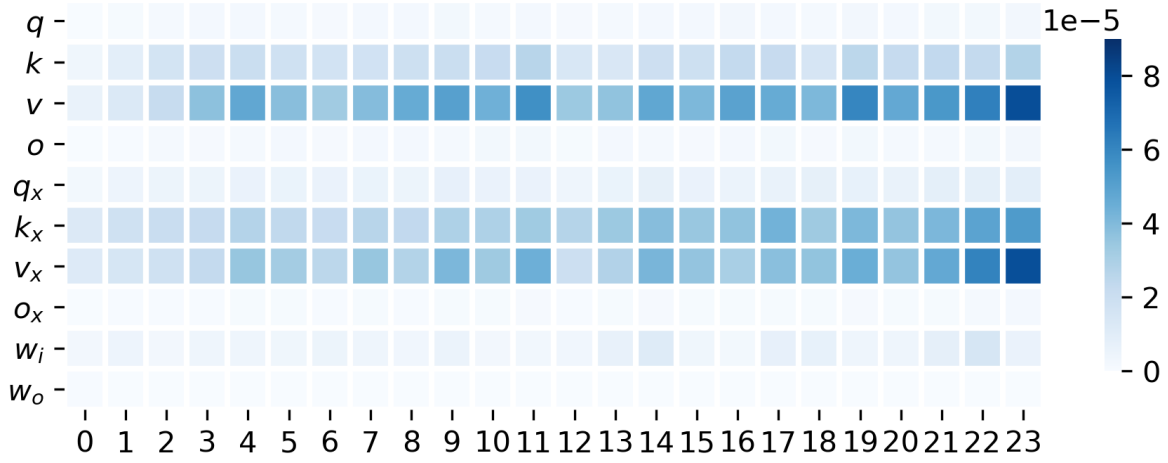

Figure 16: L1 change, Decoder, T5-11B, $n = 30$

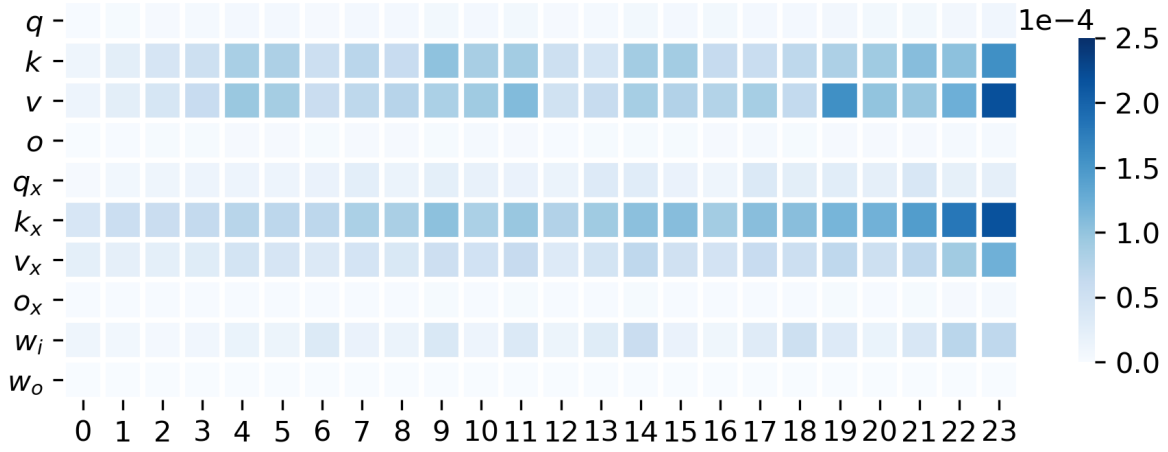

Figure 17: L1 change, Decoder, T5-11B, $n = 300$

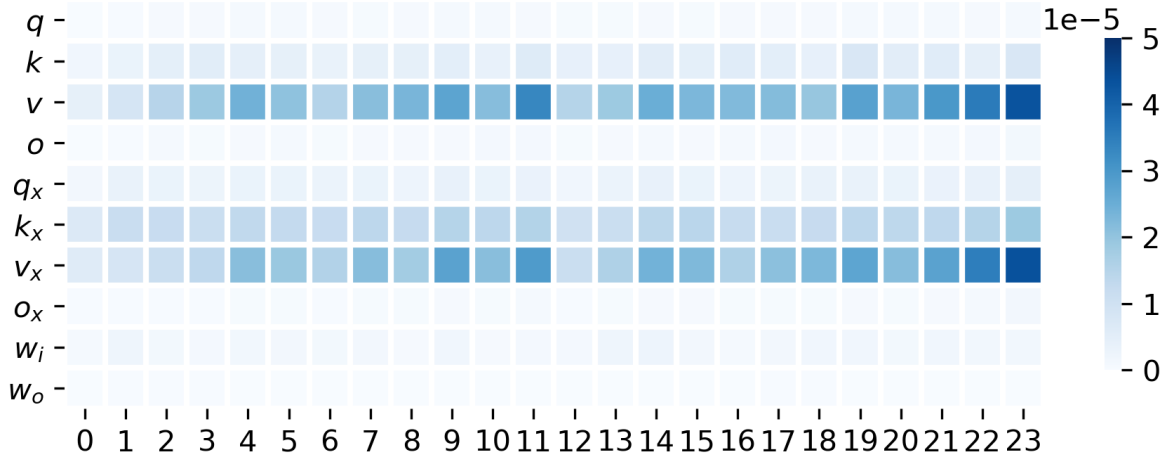

Figure 18: L1 change, Encoder, T5-11B, $n = 3$

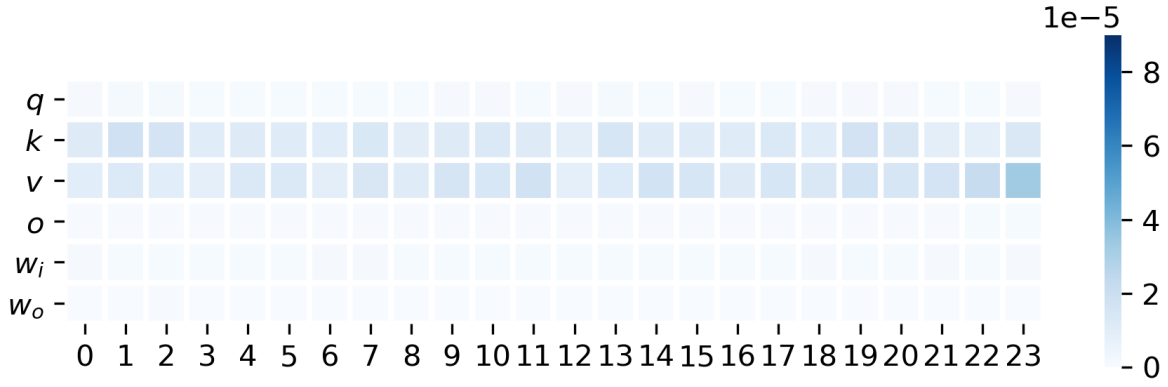

Figure 19: L1 change, Encoder, T5-11B, $n = 30$

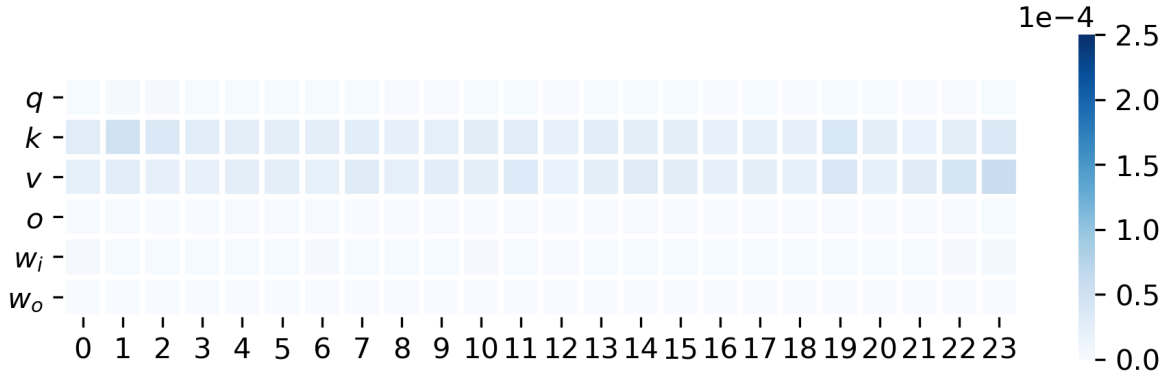

Figure 20: L1 change, Encoder, T5-11B, $n = 300$

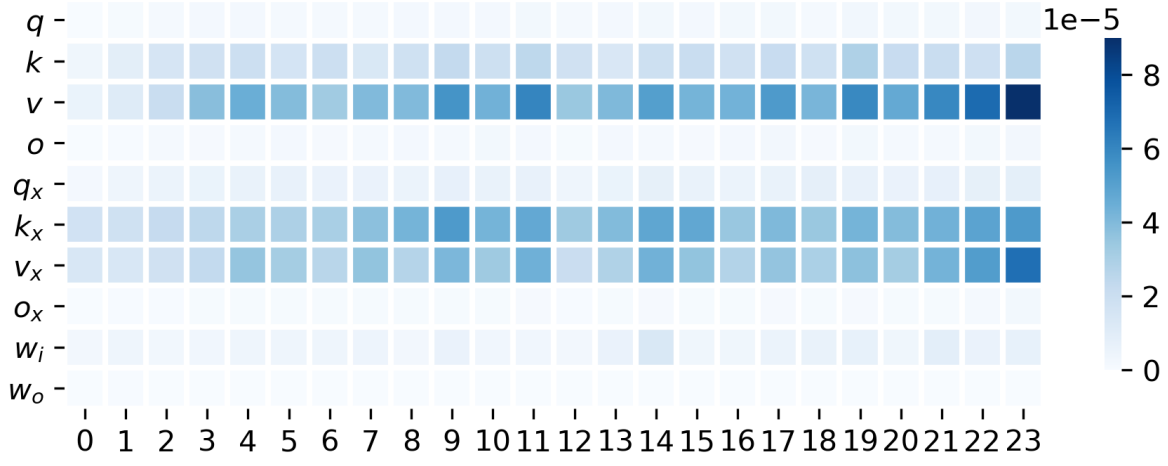

Figure 21: L1 change, Decoder, T5-11B (relation embedding), $n = 30$

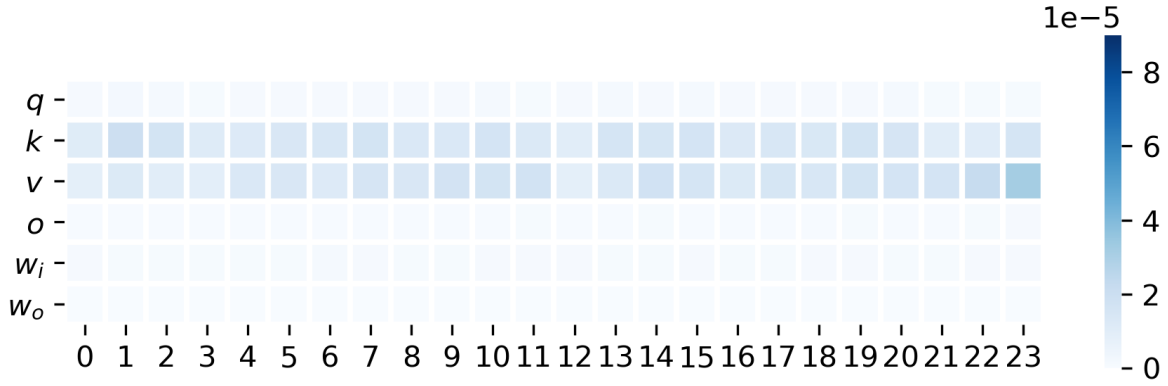

Figure 22: L1 change, Encoder, T5-11B (relation embedding), $n = 30$

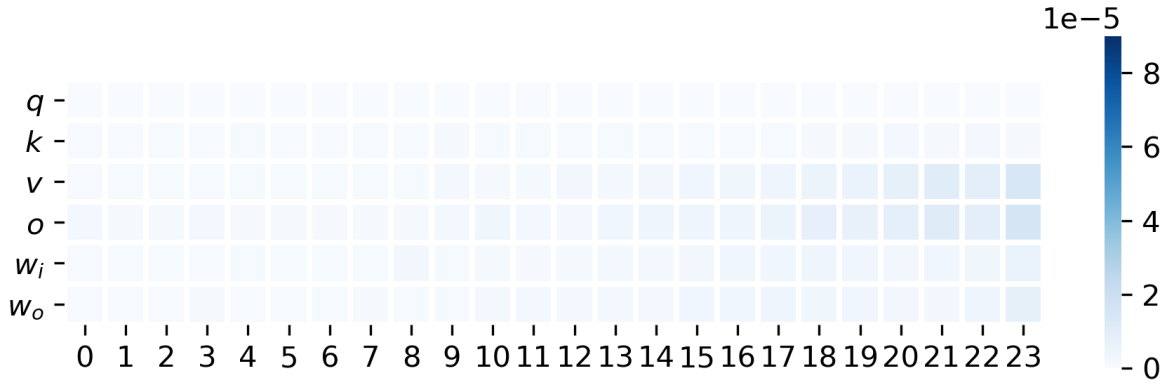

Figure 23: L1 change, Encoder, T5-Large, $n = 30$

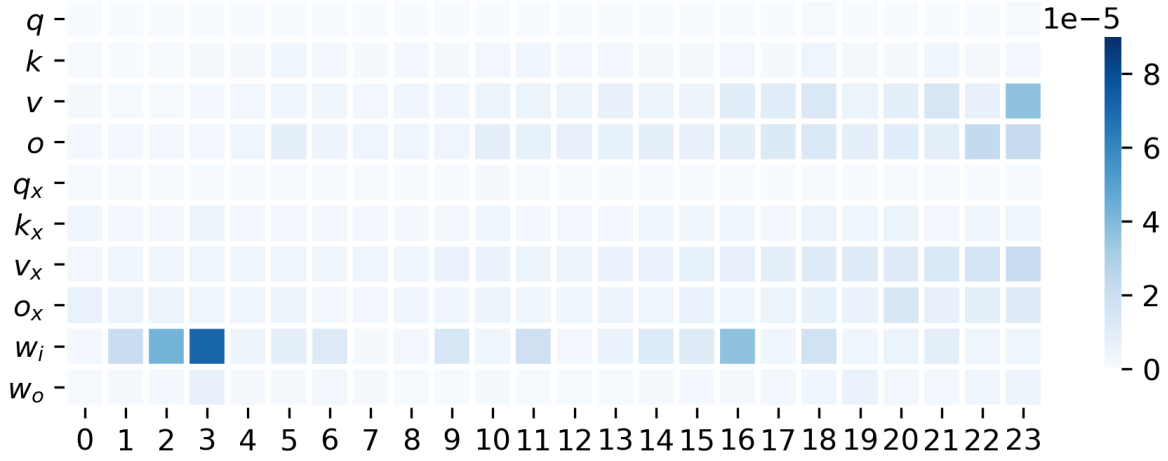

Figure 24: L1 change, Decoder, T5-Large, $n = 30$

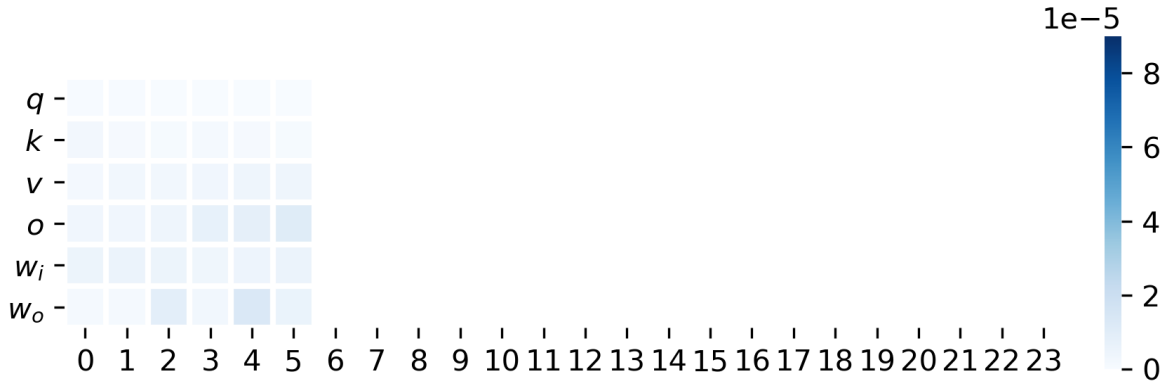

Figure 25: L1 change, Encoder, T5-Small, $n = 30$

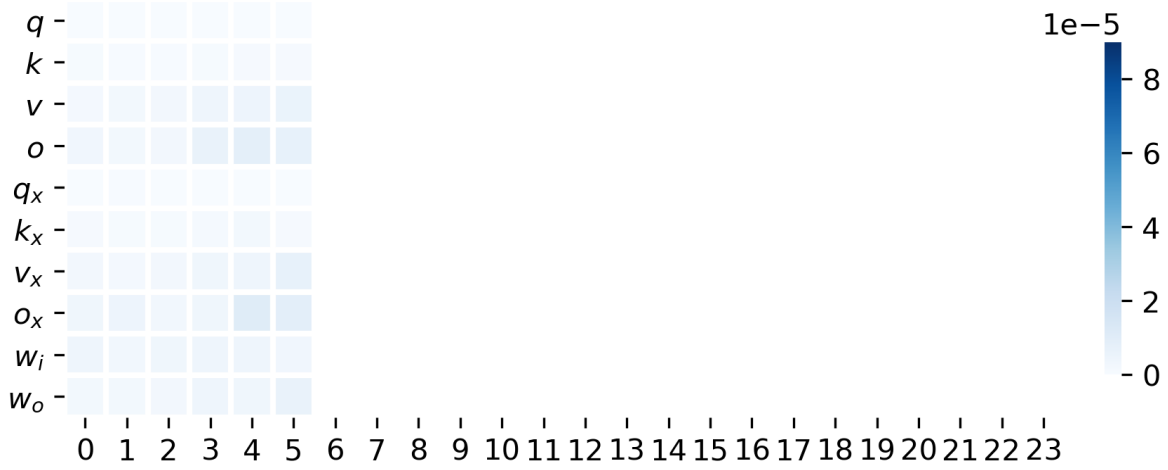

Figure 26: L1 change, Decoder, T5-Small, $n = 30$

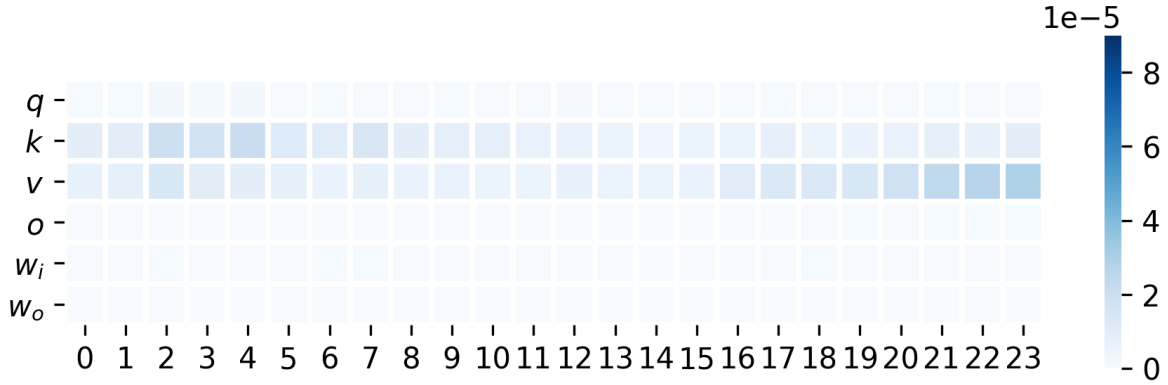

Figure 27: L1 change, Encoder, T5-11B, Shuffled Prompts, $n = 30$

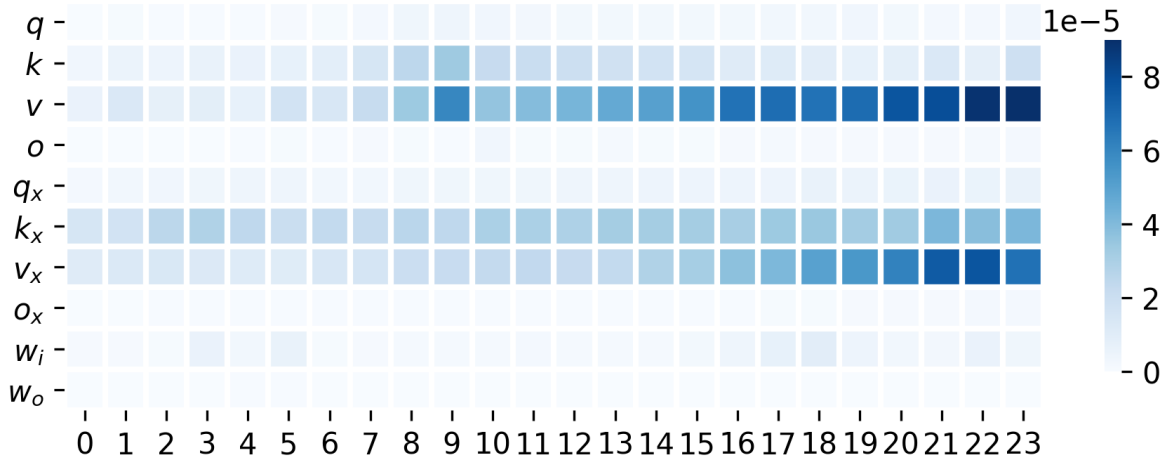

Figure 28: L1 change, Decoder, T5-11B, Shuffled Prompts, $n = 30$

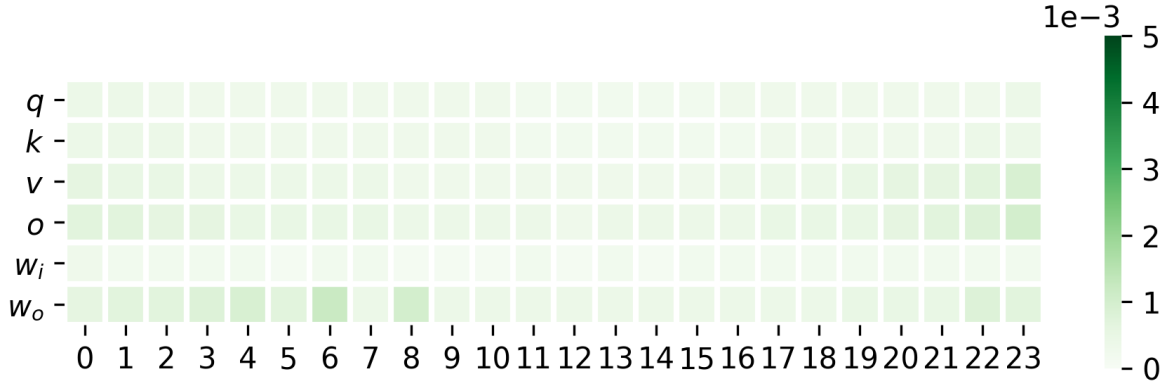

Figure 29: Angular change, Encoder, T5-11B, Shuffled Prompts, $n = 30$

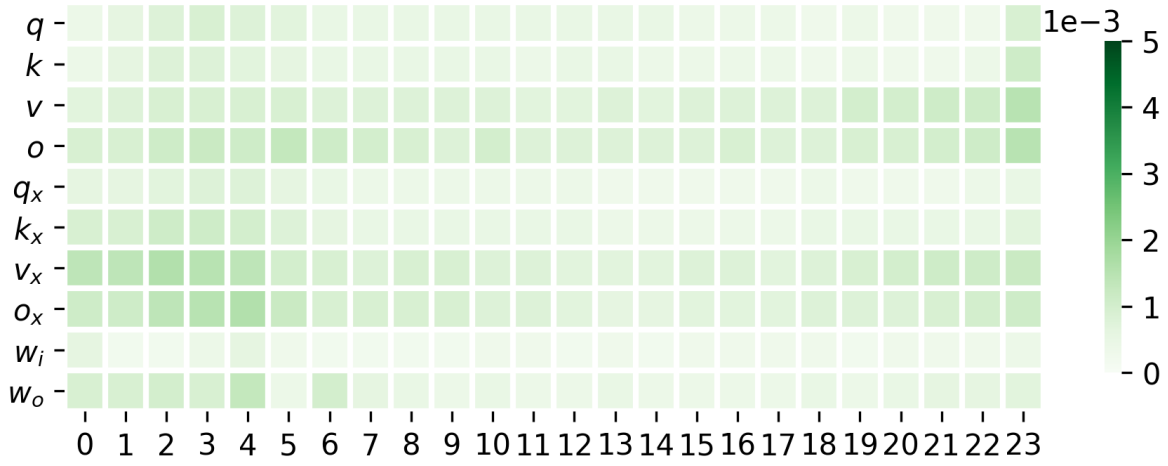

Figure 30: Angular change, Decoder, T5-11B, Shuffled Prompts, $n = 30$