A Appendix

A.1 Probing Implementation Details

For initial experiments, we trained a linear probe mapping our 1024-dimension token representations to the number of output classes (e.g. 45 POS tags). For subsequent MLP probing, we use an MLP with one 1024-neuron hidden layer with ReLU activation. All weights are trained using Adam with learning rate 10^{-3} for at most 50 epochs with 3 epochs of early stopping patience.

For more complete task descriptions, please refer to Liu et al. (2019). Of the eighteen tasks, five are pairwise, *i.e.* they involve predicting a property about a pair of tokens. These tasks are syntactic arc prediction, syntactic arc classification, semantic arc prediction, semantic arc classification, and coreference resolution (note that prediction refers to binary link identification while classification concerns the type of link). For these prediction tasks involving pairs of tokens, we input the two token embeddings $\mathbf{w}_i$ and $\mathbf{w}_j$ in addition to their elementwise product $\mathbf{w}_i \odot \mathbf{w}_j$ (as in Liu et al. (2019)).

Because our model uses Moses tokenization and byte-pair encoding, the source tokens in our preprocessed probing datasets are further split into subtokens by our model. We aggregate subtoken representations by averaging representations. Finally, we noticed that tasks with smaller train/test sets displayed some run-to-run variability, so for these tasks (PS-Fxn, PS-Role, Coref), we report averaged metrics across five replicate runs with different random seeds (for both the linear probe and for the MLP probe).

Table A1: Sparsities and BLEUs for the pruned models at each pruning iteration. LTHk refers to the name of the model pruned using learning rate rewinding, after k pruning iterations. Because we don't prune embedding weights, we computed two sparsity values, one including all weights and the other excluding the embedding weights. MP BLEU refers to the models pruned using magnitude pruning (*i.e.* the models we perform analysis of), while Random BLEU refers to the baseline using iterative *random* pruning with LR rewinding.

Model Name	Sparsity (incl. emb)	Sparsity (excl. emb)	MP BLEU	Random BLEU
LTH0	0.000	0.000	27.77	27.77
LTH1	0.168	0.200	28.04	27.59
LTH2	0.302	0.360	28.00	27.81
LTH3	0.410	0.488	27.70	27.46
LTH4	0.496	0.590	27.93	27.24
LTH5	0.565	0.672	27.80	26.90
LTH6	0.620	0.738	27.76	26.51
LTH7	0.664	0.790	27.61	26.14
LTH8	0.669	0.832	27.19	25.82
LTH9	0.727	0.865	27.16	25.33

Table A2: Results using the linear probe, for the subset of tasks whose performances vary with sparsity.

Model	PS-Fxn	PS-Role	Coref	SynPred	SemPred	NER	GED	EF
LTH0	0.858	0.740	0.771	0.905	0.892	0.718	0.302	0.712
LTH1	0.859	0.760	0.778	0.906	0.894	0.723	0.307	0.714
LTH2	0.847	0.756	0.743	0.908	0.895	0.726	0.305	0.719
LTH3	0.840	0.744	0.764	0.909	0.898	0.727	0.306	0.722
LTH4	0.847	0.737	0.766	0.912	0.903	0.728	0.309	0.722
LTH5	0.841	0.726	0.747	0.915	0.908	0.731	0.310	0.722
LTH6	0.833	0.717	0.724	0.916	0.911	0.719	0.305	0.717
LTH7	0.826	0.717	0.748	0.919	0.913	0.719	0.297	0.716
LTH8	0.828	0.721	0.749	0.921	0.917	0.717	0.299	0.709

Table A3: Results using the multilayer perceptron probe, for the subset of tasks whose linear probe performances varied with sparsity.

Model	PS-Fxn	PS-Role	Coref	SynPred	SemPred	NER	GED	EF
LTH0	0.866	0.764	0.832	0.969	0.964	0.790	0.427	0.735
LTH1	0.867	0.765	0.835	0.969	0.963	0.800	0.439	0.737
LTH2	0.869	0.762	0.822	0.969	0.963	0.796	0.428	0.742
LTH3	0.861	0.758	0.831	0.969	0.963	0.797	0.435	0.745
LTH4	0.853	0.749	0.834	0.968	0.963	0.796	0.438	0.736
LTH5	0.853	0.752	0.823	0.969	0.963	0.795	0.434	0.746
LTH6	0.841	0.740	0.798	0.969	0.963	0.787	0.433	0.739
LTH7	0.851	0.741	0.813	0.970	0.964	0.791	0.426	0.738
LTH8	0.846	0.722	0.814	0.971	0.965	0.782	0.431	0.732

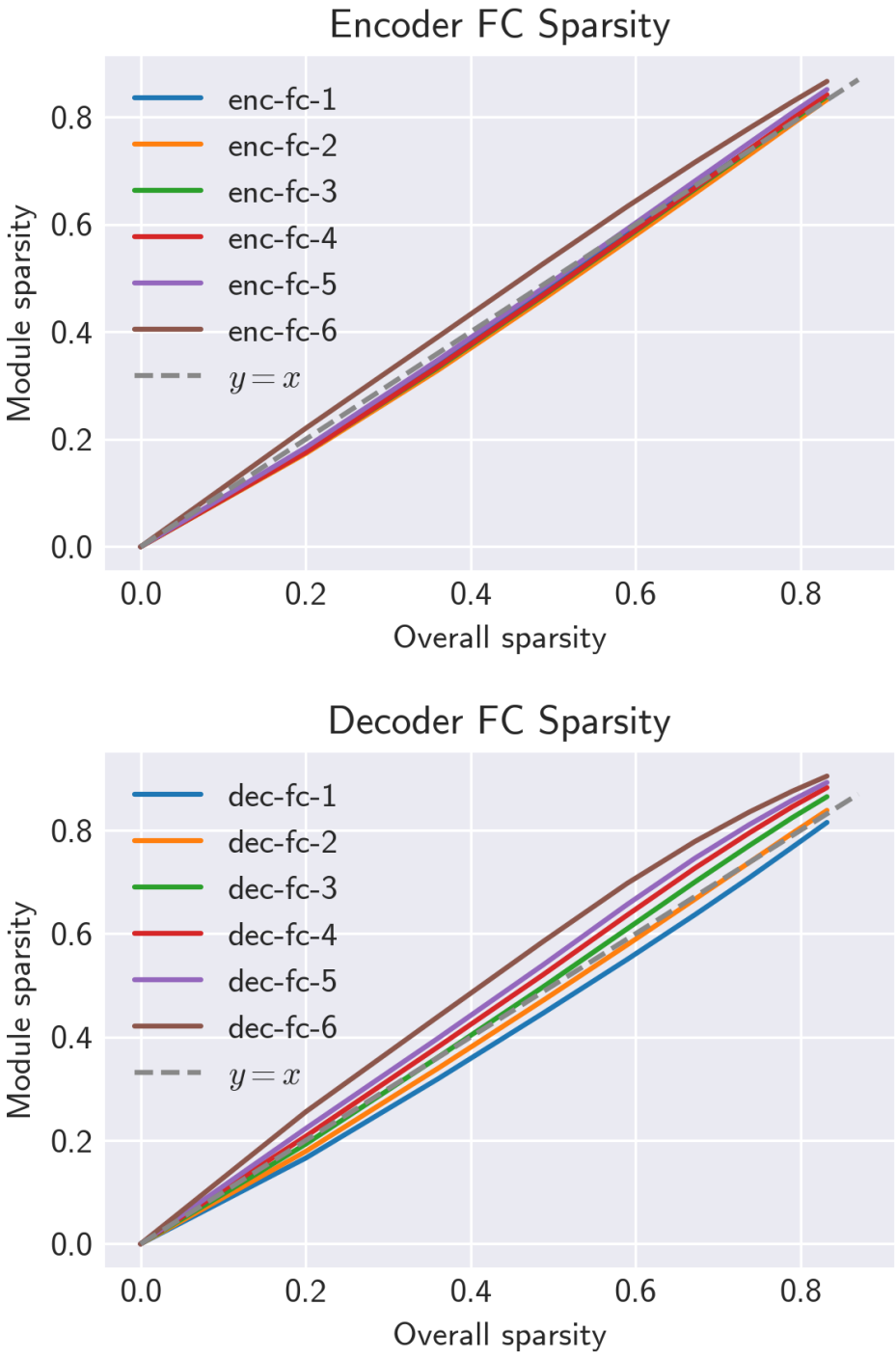

Figure A1: Sparsities of the encoder FC layers (top) and decoder FC layers (bottom). x-axis shows sparsity of the overall model, excluding embedding weights.

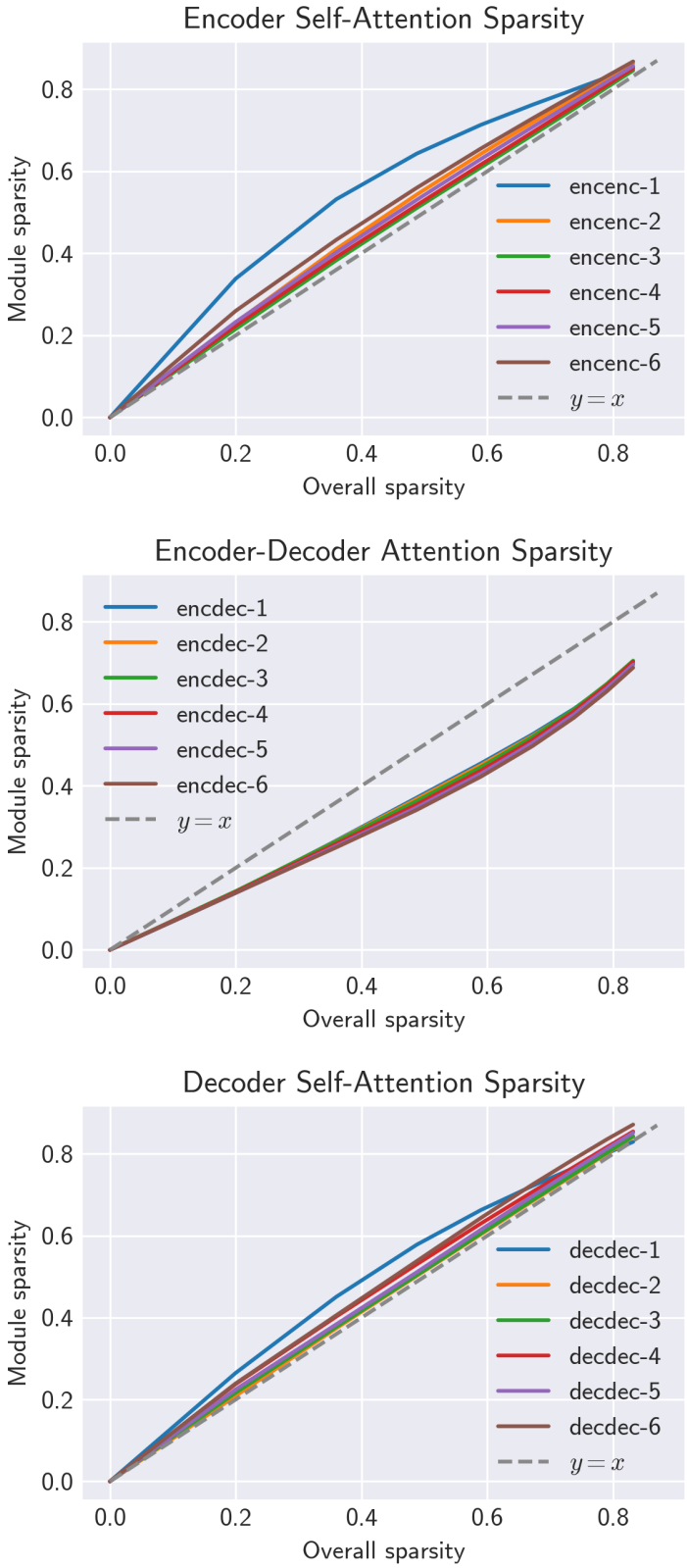

Figure A2: Sparsities of encoder self-attention (top), encoder-decoder attention (middle), and decoder self-atttention (bottom). Sparsity is aggregated across the query, key, value, and out projection matrices. x-axis shows sparsity of the overall model, excluding embedding weights.

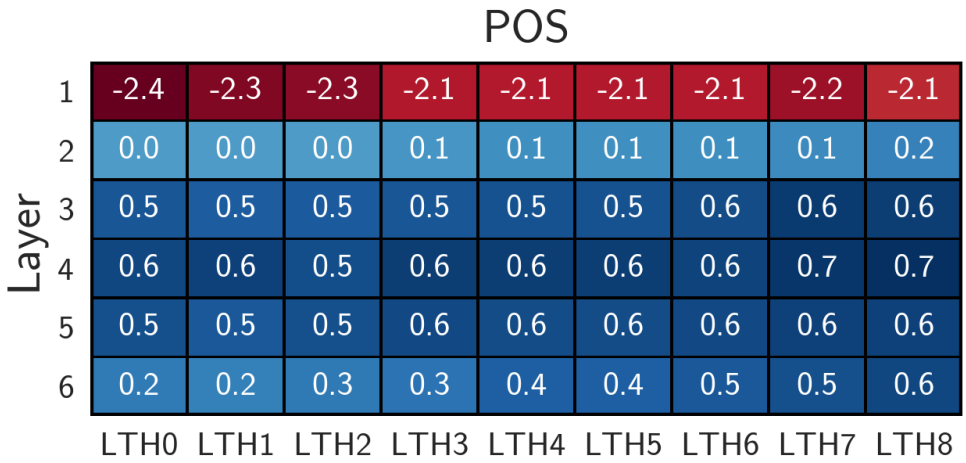

Figure A3: Each cell shows, for a particular layer of a particular model, that layer's accuracy z-score for the POS tagging probing task.

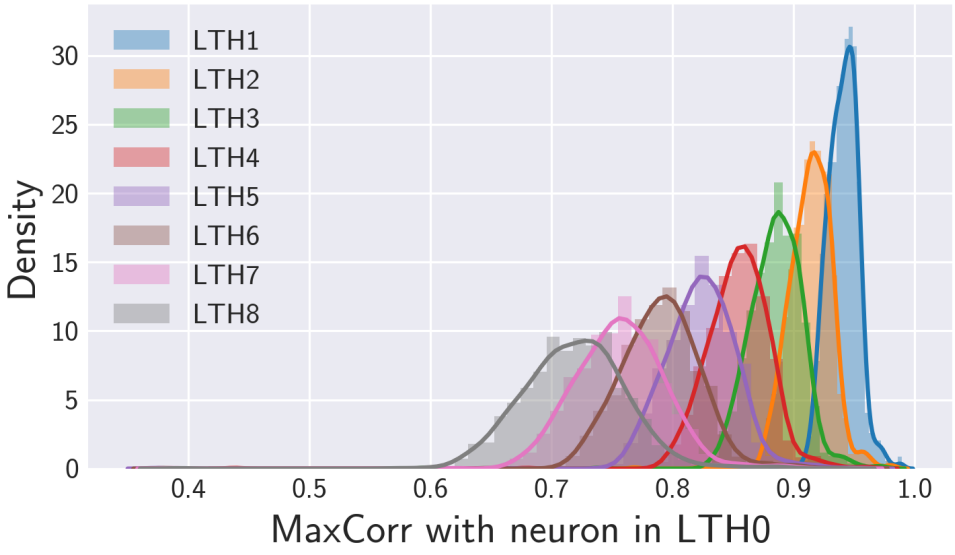

Figure A4: For each of our pruned models, we show its distribution of maximum correlations with a neuron in LTH0 (unpruned). Rather than e.g. becoming bimodal, the distributions gradually shift to the right, suggesting that all neurons slowly become less similar to their counterparts in the dense model.

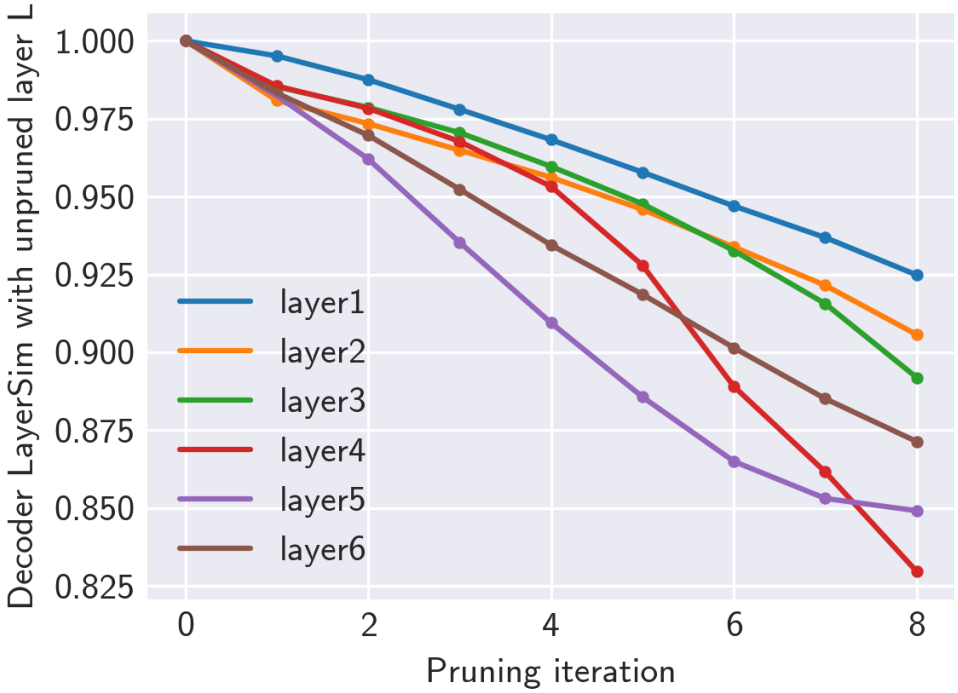

Figure A5: Each decoder layer's LayerSim with the corresponding layer in LTH0. Sparsity increases from left to right.

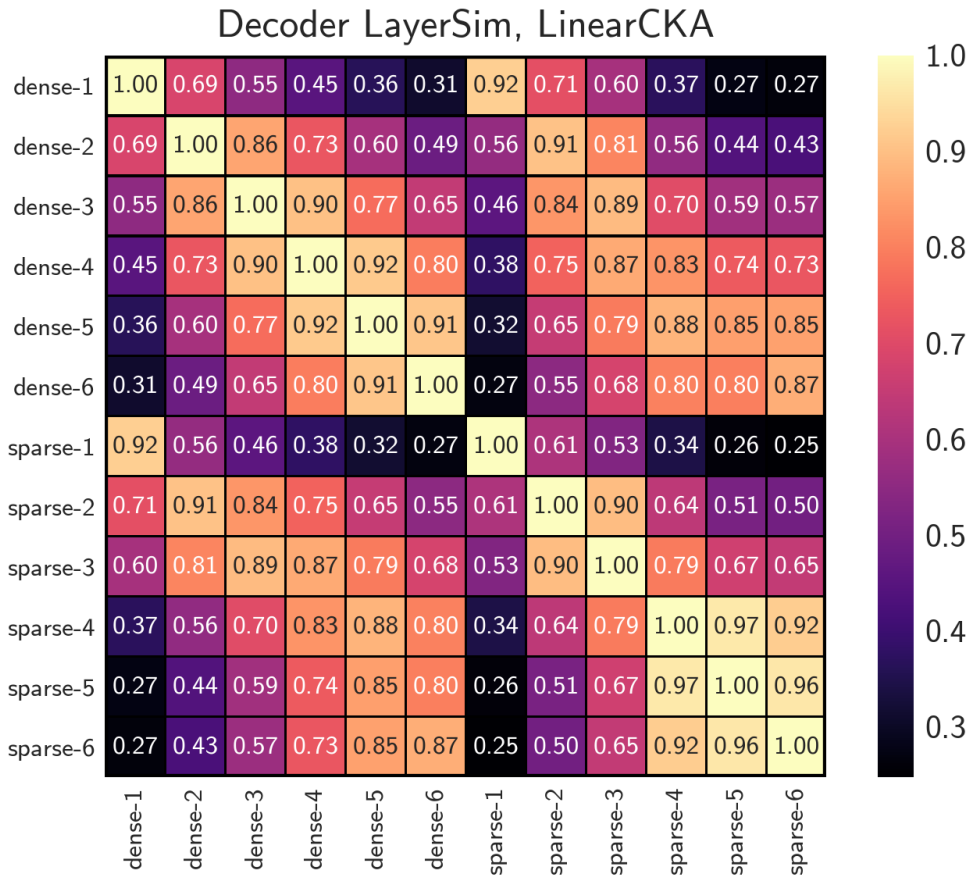

Figure A6: Decoder layer representation similarities for pairs of layers in LTH0 (dense) and LTH8 (70% sparse).

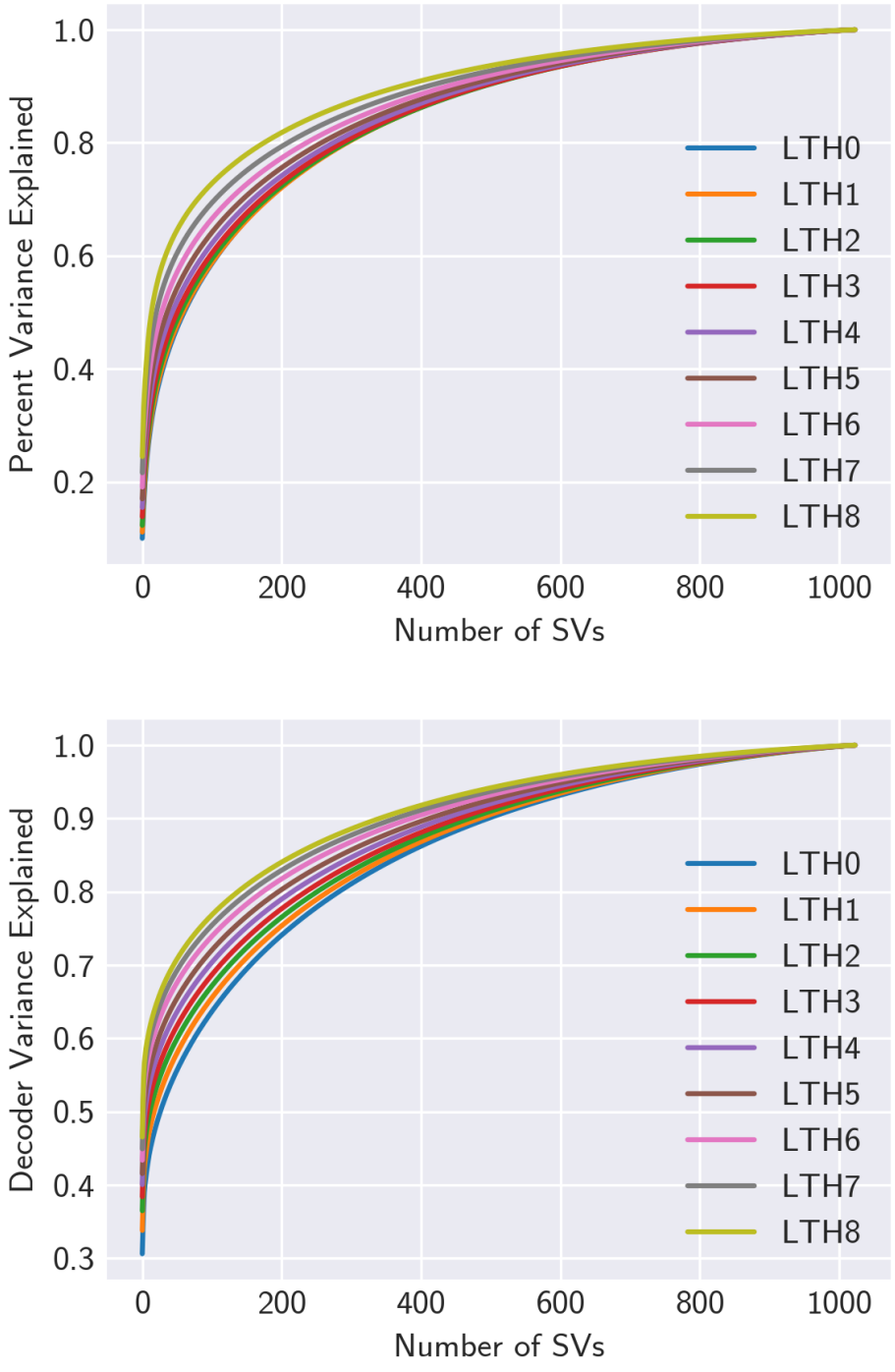

Figure A7: Percent variance explained by retaining top k singular vectors in the singular value decomposition of the final layer representation matrix for the encoder (top) and the decoder (bottom). We find that sparser models have more variance explained by fewer components, implying less representational complexity.

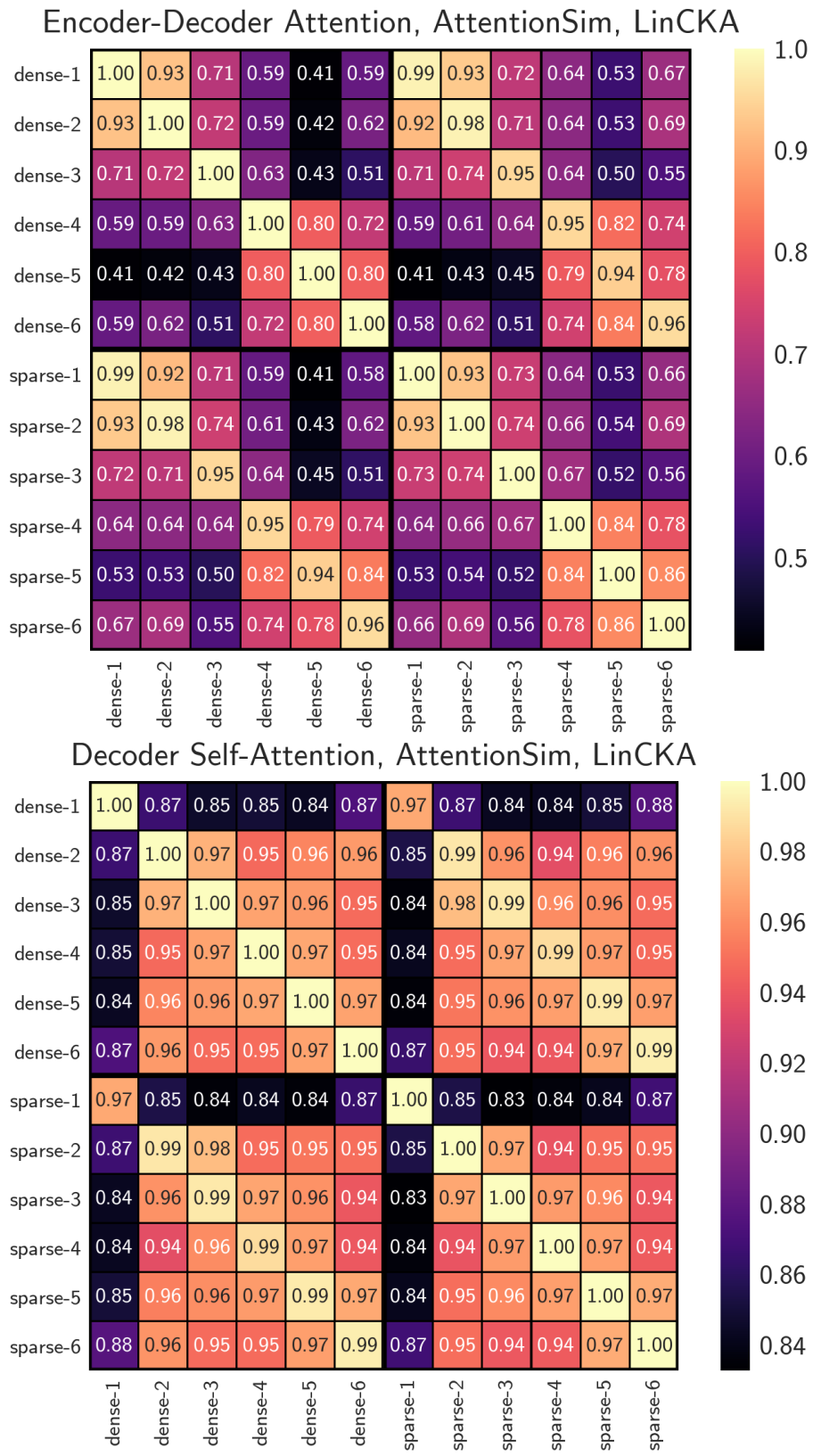

Figure A8: AttentionSim similarity for pairs of layers in LTH0 (dense) and LTH8 (sparse) for encoder-decoder attention and decoder self-attention.

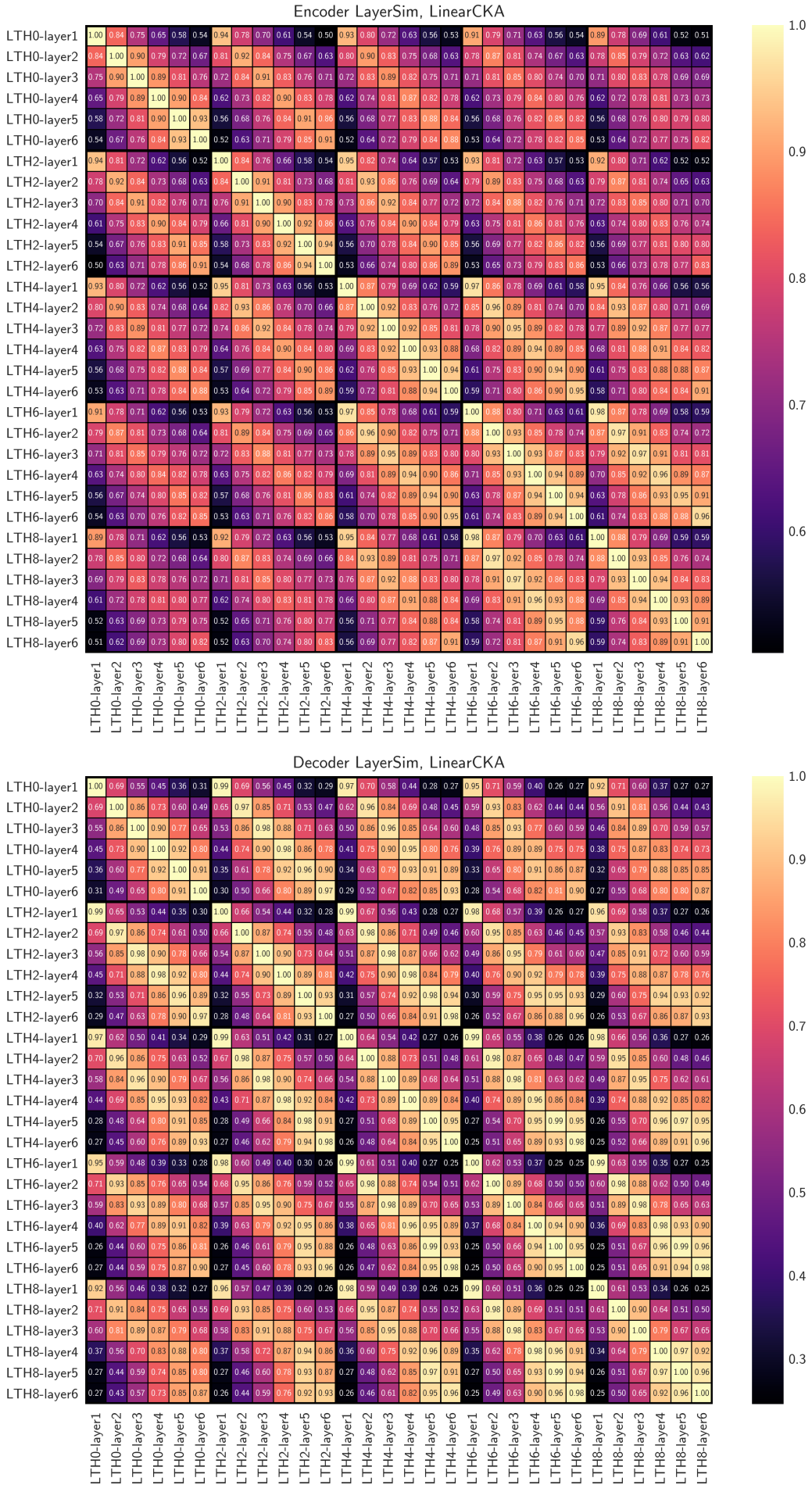

Figure A9: Full LayerSim representation similarity heatmaps for models from even-numbered pruning iterations. Top: Encoder, Bottom: Decoder.

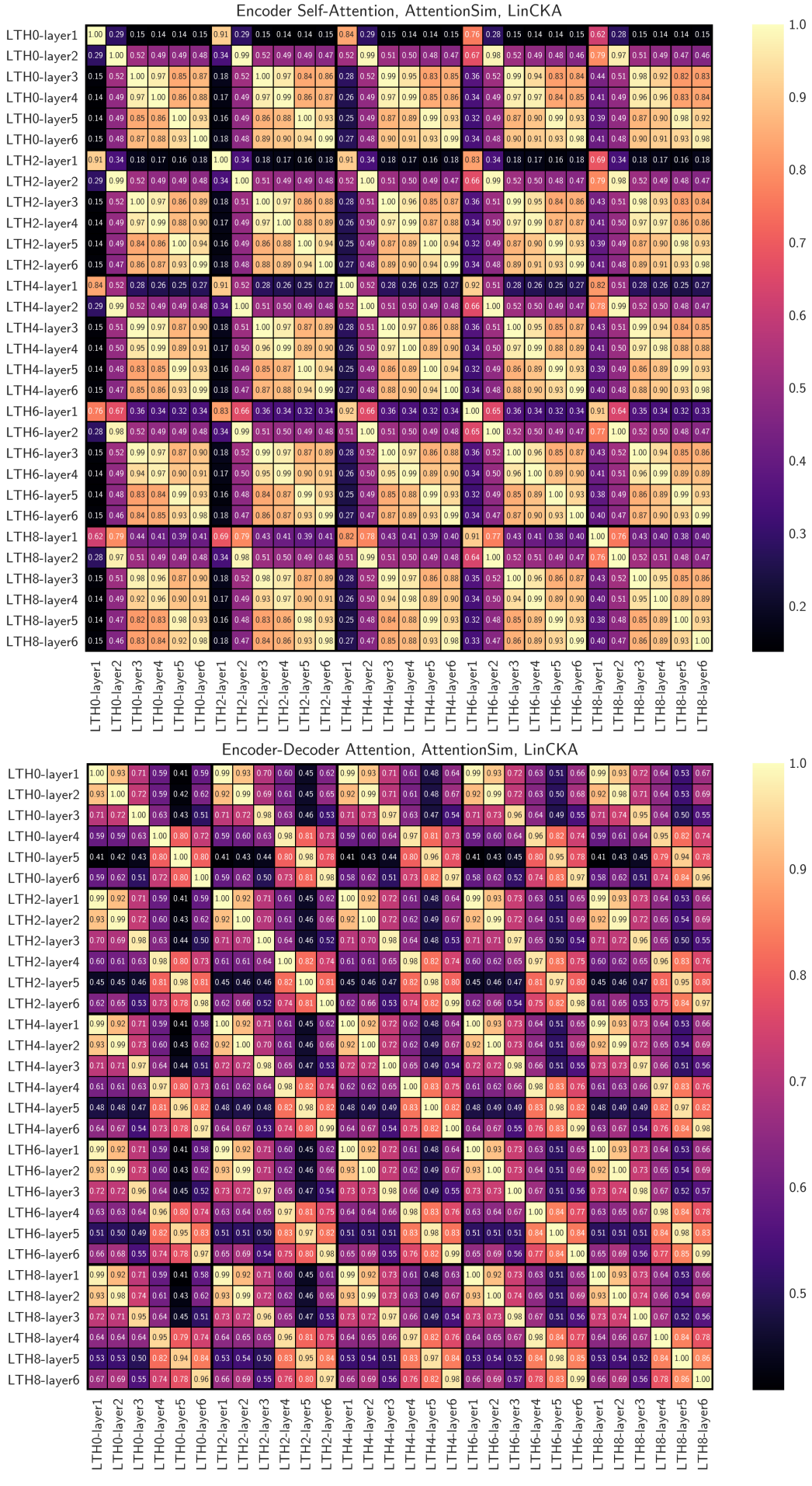

Figure A10: Full AttentionSim heatmaps for models from even-numbered pruning iterations. Top: encoder self-attention. Bottom: encoder-decoder attention.

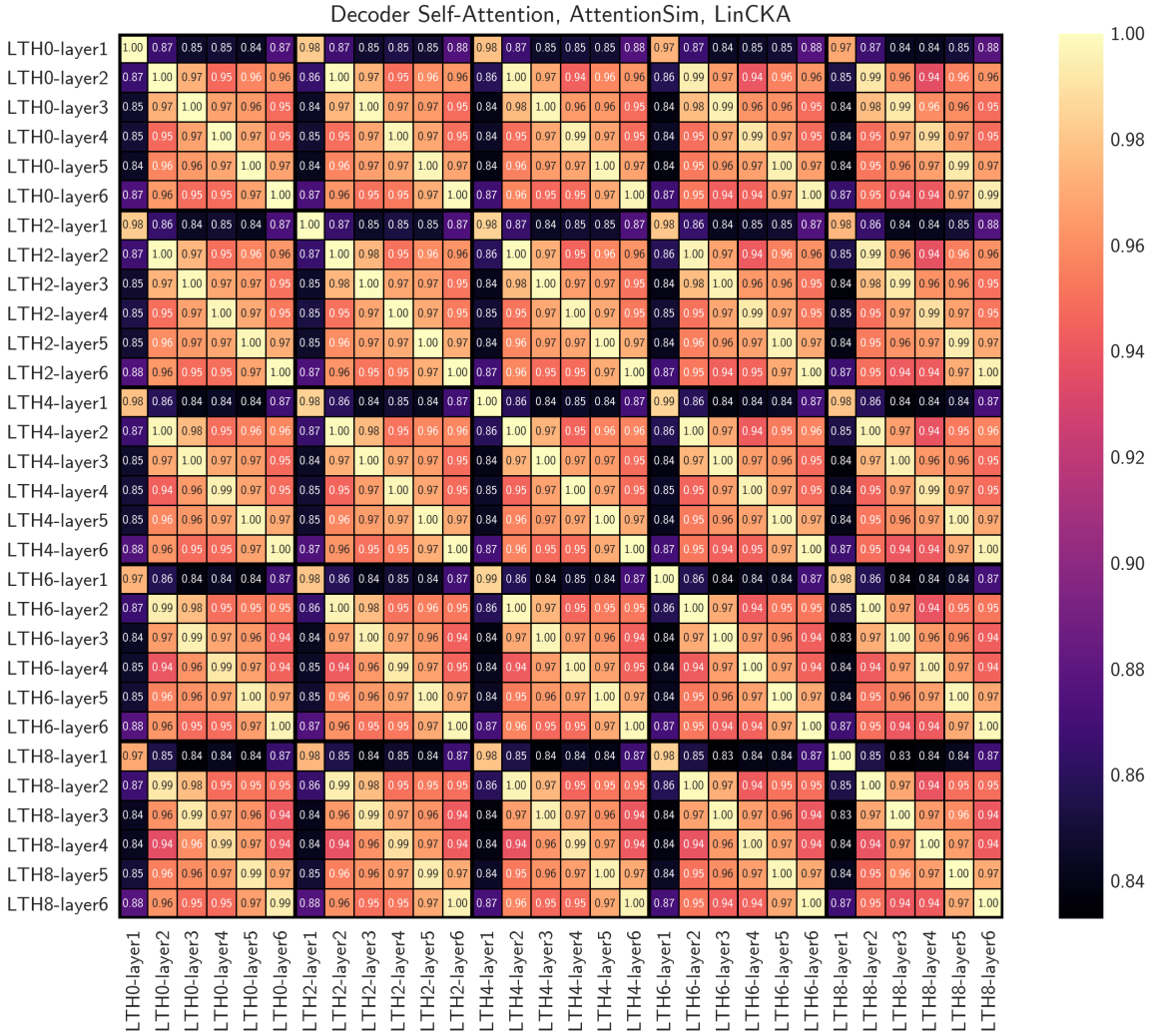

Figure A11: Decoder self-attention AttentionSim between models from even-numbered pruning iterations.