GEDU501: Scientific Writing **POSLEC**

Name: Date:

Tables and Figures Survey Worksheet

Identifying Styles in Your Reading

Instructions

1. *Insert information from one research article read in the **Article Information** table.*
2. ***Erase the examples first,** and then using the appropriate sections from one of the research articles you are reading, copy and paste images and 1~2 example sentences that perform the move described in the **Reading Table.***
3. *If the information is not available, put N/A (N/A or not applicable) in the example space. i.e. if "we" or "our" is not used in the section you are reading, you can place N/A in that example space in the table.*

Article Information

Title	Error Correction Code Transformer.
Author (s)	Choukroun, Yoni, and Lior Wolf.
Journal Title	*arXiv preprint arXiv:2203.14966* (2022).
Year of Publishing	
Volume/Issue	
Pages	
Keywords / Search Terms	Transformer, ldpc, decoding, encoding

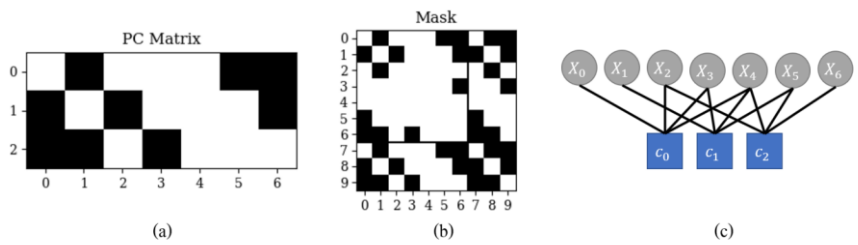
Table and Figures Reading Table

Table	
Moves	**Example from your article**
Image of table and it's label	Table 1: A comparison of the negative natural logarithm of Bit Error Rate (BER) for three normalized SNR values of our method with literature baselines. Higher is better. Concurrent results are obtained after $L = 5$ BP iterations in first row (i.e. 10 layers neural network) and *at convergence* results in second row obtained after $L = 50$ BP iterations (i.e. 100 layers neural network). Best results in **bold**, second best is underlined, and the minimal Transformer architecture to outperform every other competing method is in *italic*. Our performance are presented for seven different architectures: for $N = \{2, 6\}$, we present results for $d = \{32, 64, 128\}$ (first to third row), and for $N = 10$ we run only the $d = 128$ configuration.
Where is the label located? Underline or Highlight	Above the Table
Copy and paste all text from label here.	Table 1: A comparison of the negative natural logarithm of Bit Error Rate (BER) for three normalized SNR values of our method with literature baselines. Higher is better. Concurrent results are obtained after L = 5 BP iterations in first row (i.e. 10 layers neural network) and at convergence results in second row obtained after L = 50 BP iterations (i.e. 100 layers neural network). Best results in bold, second best is underlined, and the minimal Transformer architecture to outperform every other competing method is in

Method	BP[24]			Hyp BP[21]			AR BP[22]			Ours N=2			Ours N=6			Ours N=10		
	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6	4	5	6
Polar(64,32)	3.52	4.04	4.48	4.25	5.49	7.02	4.77	6.30	8.19	4.27	5.44	6.95	*5.71*	*7.63*	*9.94*			
	4.26	5.38	6.50	4.59	6.10	7.69	5.57	7.43	9.82	4.57	5.86	7.50	6.48	8.60	11.43			
										4.87	6.2	7.93	**6.99**	**9.44**	**12.32**			
Polar(64,48)	4.15	4.68	5.31	4.91	6.48	8.41	5.25	6.96	9.00	4.92	6.46	8.41	*5.82*	*7.81*	10.24			
	4.74	5.94	7.42	4.92	6.44	8.39	5.41	7.19	9.30	5.14	6.78	8.9	6.15	8.20	10.86			
										5.36	7.12	*9.39*	**6.36**	**8.46**	**11.09**			
Polar(128,64)	3.38	3.80	4.15	3.89	5.18	6.94	4.02	5.48	7.55	3.51	4.52	5.93	4.47	6.34	8.89			
	4.10	5.11	6.15	4.52	6.12	8.25	4.84	6.78	9.30	3.83	5.16	7.04	*5.12*	*7.36*	*10.48*			
										4.04	5.52	7.62	**5.92**	**8.64**	**12.18**			
Polar(128,86)	3.80	4.19	4.62	4.57	6.18	8.27	4.81	6.57	9.04	4.30	5.58	7.34	5.36	*7.45*	*10.22*			
	4.49	5.65	6.97	4.95	6.84	9.28	5.39	7.37	10.13	4.49	5.90	7.75	5.75	8.16	11.29			
										4.75	6.25	8.29	**6.31**	**9.01**	**12.45**			
Polar(128,96)	3.99	4.41	4.78	4.73	6.39	8.57	4.92	6.73	9.30	4.56	5.98	7.93	*5.39*	*7.62*	*10.45*			
	4.61	5.79	7.08	4.94	6.76	9.09	5.27	7.44	10.2	4.69	6.20	8.30	5.88	8.33	11.49			
										4.88	6.58	8.93	**6.31**	**9.12**	**12.47**			
LDPC(49,24)	5.30	7.28	9.88	5.76	7.90	11.17	6.05	8.13	11.68	4.33	5.79	7.69	5.45	7.65	10.58			
	6.23	8.19	11.72	6.23	8.54	11.95	**6.58**	**9.39**	12.39	4.40	5.89	7.99	5.60	7.93	11.13	6.35	9.01	***12.43***
										4.53	6.10	8.33	5.79	8.13	11.40			
LDPC(121,60)	4.82	7.21	10.87	5.22	8.29	13.00	5.22	8.31	13.07	3.80	5.32	7.79	4.77	7.52	11.99			
	-	-	-	-	-	-	-	-	-	3.81	5.36	7.87	4.88	7.77	12.39	***5.51***	***8.89***	***14.51***
										3.80	5.45	8.06	5.01	7.99	12.78			
LDPC(121,70)	5.88	8.76	13.04	6.39	9.81	14.04	6.45	10.01	14.77	4.49	6.38	9.24	5.80	9.11	13.77			
	-	-	-	-	-	-	-	-	-	4.52	6.44	9.36	6.04	9.54	14.65	***6.86***	***11.02***	**16.85**
										4.53	6.49	9.39	6.19	9.89	*15.58*			
LDPC(121,80)	6.66	9.82	13.98	6.95	10.68	15.80	7.22	11.03	15.90	5.11	7.18	10.18	6.63	10.28	15.10			
	-	-	-	-	-	-	-	-	-	5.13	7.26	10.28	6.84	10.56	15.68	***7.76***	***12.30***	**17.82**
										5.24	7.46	10.66	7.07	10.96	*16.25*			
MacKay(96,48)	6.84	9.40	12.57	7.19	10.02	13.16	7.43	10.65	14.65	4.92	6.62	8.88	6.78	9.61	13.31			
	-	-	-	-	-	-	-	-	-	4.98	6.72	9.04	7.02	10.12	14.18	***8.39***	***12.24***	***16.41***
										5.12	6.98	9.44	7.23	10.42	14.12			
CCSDS(128,64)	6.55	9.65	13.78	6.99	10.57	15.27	7.25	10.99	16.36	4.27	5.97	8.18	6.29	9.59	13.95			
	-	-	-	-	-	-	-	-	-	4.39	6.03	8.43	6.49	10.18	14.76	***8.02***	***12.60***	***17.75***
										4.47	6.22	8.74	6.77	10.55	15.9			
BCH(31,16)	4.63	5.88	7.60	5.05	6.64	8.80	5.48	7.37	9.61	4.51	5.74	7.35	*5.74*	*7.42*	*9.59*			
	-	-	-	-	-	-	-	-	-	4.78	6.15	7.98	5.85	7.52	10.08			
										5.18	6.82	8.91	**6.39**	**8.29**	**10.66**			
BCH(63,36)	3.72	4.65	5.66	3.96	5.35	7.20	4.33	5.94	8.21	3.79	4.87	6.35	4.42	5.91	8.01			
	4.03	5.42	7.26	4.29	5.91	8.01	4.57	6.39	8.92	4.05	5.28	7.01	*4.62*	6.24	8.44			
										4.21	5.50	7.25	**4.86**	***6.65***	***9.10***			
BCH(63,45)	4.08	4.96	6.07	4.48	6.07	8.45	4.80	6.43	8.69	4.47	5.88	7.81	*5.16*	*7.02*	*9.75*			
	4.36	5.55	7.26	4.64	6.27	8.51	4.97	6.90	9.41	4.66	6.16	8.17	5.41	7.49	10.25			
										4.79	6.39	8.49	**5.60**	**7.79**	**10.93**			
BCH(63,51)	4.34	5.29	6.35	4.64	6.08	8.16	4.95	6.69	9.18	4.60	6.05	8.05	*5.20*	7.08	*9.65*			
	4.5	5.82	7.42	4.80	6.44	8.58	5.17	7.16	9.53	4.78	6.34	8.49	5.46	*7.57*	10.51			
										5.01	6.72	9.03	**5.66**	**7.89**	**11.01**			

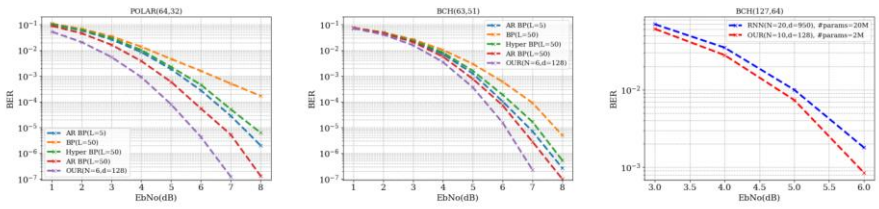
	italic. Our performance are presented for seven different architectures: for N = {2, 6}, we present results for d = {32, 64, 128} (first to third row), and for N = 10 we run only the d = 128 configuration
Reference to table in text	The results are reported in **Tab. 1** where we present the negative natural logarithm of the BER. For each code we present the results of the BP based competing methods for 5 and 50 iterations (first and second row), corresponding to 10 and 100 layers neural network respectively.
Additional Notes / Unusual features	Although it is mentioned in the paper's other section, contents of parameters are omitted in the label.

Figure (image)

Moves	**Example from your article**
Image of Figure and it's label	 Figure 2: We present the corresponding parity-check matrix (a), the induced Tanner graph (c) and the proposed masking (b) on the Hamming(7,4) code.
Where is the label located? Underline or Highlight	Underneath the Figure
Copy and paste all text from label here.	Figure 1: Left: Feedforward neural network exhibiting the dependency "if n1,2 is inactive, then n2,1 is inactive". Right: Depiction of the configuration space reduction induced by the dependency
Reference to Figure in text	This construction enables more freedom in decoding than the relations enabled by the Tanner graph, since related bits may have an impact on each other beyond the parity check equations as depicted in **Figure 2**. While regular Transformer can be assimilated to a neural network applied on a complete graph, the proposed mask can be seen as the adjacency matrix of the Tanner graph extended to a two rings connectivity.

Additional Notes / Unusual features	

Figure (graph)

Moves	**Example from your article**
Image of Figure and it's label	 (a) (b) (c) Figure 4: BER for various values of SNR for (a) Polar(64,32) and (b) BCH(63,51) codes. Comparison with the best model of Bennatan et al. [2] on BCH(127,64) code (c).
Where is the label located? Underline or Highlight	Above the Figure
Copy and paste all text from label here.	Figure 4: BER for various values of SNR for (a) Polar(64,32) and (b) BCH(63,51) codes. Comparison with the best model of Bennatan et al. [2] on BCH(127,64) code (c).
Reference to Figure in text	We provide plots of more BER values for some of the codes in **Figure 4(a,b)** for Polar(64,32) and BCH(63,51) codes, respectively. Our method is able to outperform the converged AR-BP model by up to two orders of magnitudes for high SNR values.

Additional Notes / Unusual features	

Equation / Algorithm (underline the appropriate one)	
Moves	**Example from your article**
Image of equation/algorthim and it's label	Thus, for a given algebraic code defined by the matrix H we propose to defined a function $g(H) : \{0,1\}^{(n-k)\times k} \to \{-\infty, 0\}^{2n-k\times 2n-k}$ defining the mask to be applied to the self-attention mechanism such that $$A_H(Q,K,V) = \text{Softmax}\left(\frac{QK^T + g(H)}{\sqrt{d}}\right)V. \quad (6)$$
Where is the label located? Underline or Highlight	No label
Copy and paste the Descriptive Label (for algorithm)	N/A
Reference to equation in text	N/A
Additional Notes / Unusual features	There is no numerical label for this equation, perhaps because it is the only self-standing equation in the paper. (Because it's a formula that's too widely used.)

Equation / Algorithm	
Moves	**Example from your article**
Image of equation/algorithm and it's label	

Algorithm 1: Mask construction Pseudo Code

```
function g(H)
    l,n = H.shape
    k = n-l
    mask = eye(2n-k)
    for i in range(0,n-k) do
        idx = where(H[i]==1)
        for j in idx do
            mask[n+i,j] = mask[j,n+i] = 1
            for k in idx do
                mask[j,k] = mask[k,j] = 1
    return −∞(¬ mask)
```

Where is the label located? Underline or Highlight	No label
Copy and paste the Descriptive Label (for algorithm)	N/A
Reference to equation in text	We summarize the construction of the mask in **Algorithm 11**. This construction enables more freedom in decoding than the relations enabled by the Tanner graph, since related bits may have an impact on each other beyond the parity check equations as depicted in Figure 2.
Additional Notes / Unusual features	The author made a typo. (Algorithm11 -> Algorithm1)