We have included the results omitted in the main text due to space constraints in the 'results/figures' folder of our replication package. Here is a brief description of each table and figure:

IX. Corresponding Results for Empirical Analysis

The top tokens from machine-authored code at $T = 1.0$ are shown in Table VII, corresponding to the results in Table II in the main text. Similar attention on exception handling and object-oriented programming tokens can also be observed at $T = 1.0$.

Table VII: Top tokens from machine-authored code at $T = 1.0$

Rank	Machine-Authored Tokens
1-10	**. ' ( ) self : , = " if**
11-20	**return** def [ ] None in == not is import
21-30	for { **raise** **__name__** from } - data else get
31-40	elif value path isinstance True + os **ValueError** key text
41-50	x type 0 1 len append replace str @ f

Figure 7 and Table VIII provide the distribution of naturalness scores and the naturalness of different categories of syntax elements, respectively. They correspond to the empirical analysis in Figure 4 and Table VIII in the main text.

We can observe from Figure 7 that the trend of naturalness scores is similar to that at $T = 0.2$. although the overlap between machine- and human-authored code is greater. The naturalness of different categories of syntax elements in Table VIII shows that whitespace tokens is still the most effective feature as in $T = 0.2$.

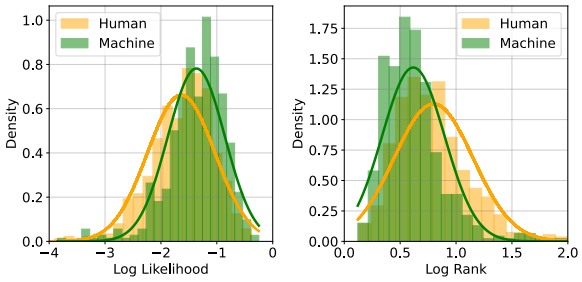


Figure 7: Distribution of naturalness scores with $T = 1.0$

Table VIII: The naturalness of different categories of syntax elements with $T = 1.0$

Category	Log Likelihood			Log Rank		
	Machine	Human	Δ	Machine	Human	Δ
keyword	-2.025	-2.128	0.103	0.968	1.053	0.085
identifier	-0.787	-0.874	0.087	0.328	0.378	0.050
literal	-1.059	-1.364	0.305	0.454	0.630	0.176
operator	-1.614	-1.835	0.221	0.733	0.872	0.139
symbol	-0.968	-1.639	0.671	0.321	0.781	0.460
comment	-2.395	-3.028	0.633	1.180	1.610	0.430
whitespace	-2.058	-2.740	**0.682**	0.946	1.441	**0.495**
ALL	-1.367	-1.658	0.291	0.618	0.811	0.193

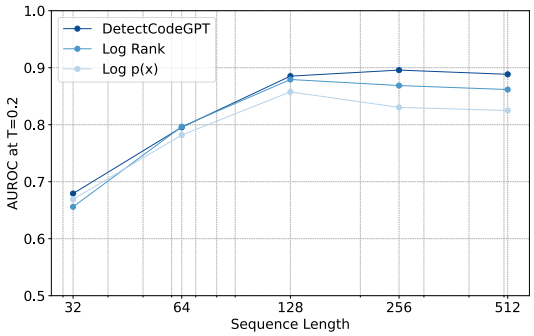
X. Cross-Model Detection Performance at $T = 1.0$

Figure 8 illustrates the cross-model detection performance at $T = 1.0$, corresponding to the results in Figure 6 in the main text. The results show that the cross-model detection performance of DetectCodeGPT has a similar trend across different temperatures.

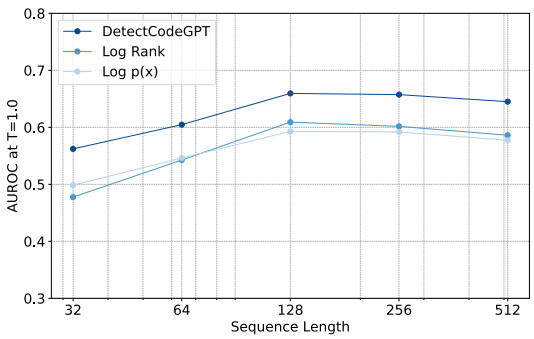
Figure 8: Cross-model detection performance at $T = 1.0$

XI. Impact of Varying Length for Detection

Figure 9 illustrates the trend in detection performance as the length of code snippets varies across different temperatures, compared against two of the most competitive baselines. This experiment uses the Stack dataset with CodeLlama. The results show that trimming the first 128 tokens of the snippets is sufficient to maintain high detection performance.



(a) AUROC at $T = 0.2$



(b) AUROC at $T = 1.0$

Figure 9: AUROC with different code trimming length