Analysis Report

Author: Zhenqun Shao

Attention Head

1. Experimental Setup

I trained and analyzed a 2-layer Seq2Seq Transformer model with 4 attention heads per layer, applied to a symbolic addition task. During training, the padding token was set to −1 rather than 0, to avoid overlapping with valid digit tokens (e.g., the digit '0'). This ensures clear masking and prevents the model from attending to padded positions that could otherwise distort attention heatmaps or interfere with head importance estimation.

2. Encoder Attention Visualization

In the analysis, twelve different test samples were examined. For each sample, every layer produced one visualization, and each visualization contained four distinct attention heads. The four attention heads within the same encoder layer exhibit clearly differentiated attention patterns.

Specifically, the second encoder layer plays a more interpretable role—it tends to focus on the output-side positions, suggesting a higher-level abstraction of the addition process. Among its four heads, Head 2 consistently attends to position 10, corresponding to the “+” operator, while Heads 1, 3, and 4 display shifted attention before or after position 10. This observation aligns well with our intuition: different heads specialize in attending to the two operands, enabling more accurate symbolic computation through distributed focus. Such diversity among heads demonstrates functional specialization, where each head captures a unique structural dependency necessary for the symbolic addition task.

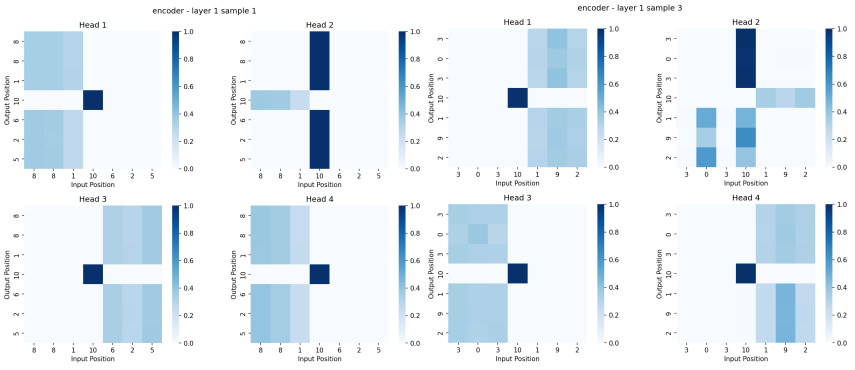

Figure 1:Layer 2

Turning to the first encoder layer, consistent attention patterns were observed in roughly 66% of the samples—particularly for Heads 2, 3, and 4, whose attention remained relatively stable across different examples. In contrast, Head 1 showed high variability with no clear positional regularity.

Notably, Heads 3 and 4 often concentrated attention at positions 1 and 5, which correspond to the leading digits of the operands. This pattern may stem from two theoretical factors: first, the leading digits are crucial for determining whether carry propagation is needed, thereby demanding stronger localized attention; second, the phenomenon reflects attention redundancy and optimization saturation. Once certain heads have minimized their contribution to the global loss or entropy objective, the optimization dynamics naturally reduce gradient updates to redundant heads. This results in stable yet functionally overlapping attention across lower-layer heads.

3. Decoder

Figures 2 visualize decoder self-attention maps for layers 0 and 1. The causal masking is clearly visible—each output token only attends to itself and earlier positions, preventing any information flow from future tokens. This autoregressive constraint ensures that digits are generated sequentially, with each position conditioned on all previously decoded outputs. In our implementation, the decoder input for attention visualization was constructed dynamically as the current ground-truth prefix (targets[:, :1] + targets[:, :-1]), without any padding tokens. As a result, the size of each self-attention matrix exactly matches the true output length of that sample (e.g., 3 × 3 for 3-digit results, 4 × 4 for 4-digit results). This design avoids spurious rows or columns that would appear if fixed-length padding were used. For cases where the correct answer has four digits (e.g., 663 + 497 = 1160), all four decoding positions are preserved, enabling a clear observation of carry propagation behavior between the last two digits. Conversely, when the prediction has only three digits, the visualization automatically truncates after the third output token, preventing meaningless attention patterns beyond the valid sequence boundary.

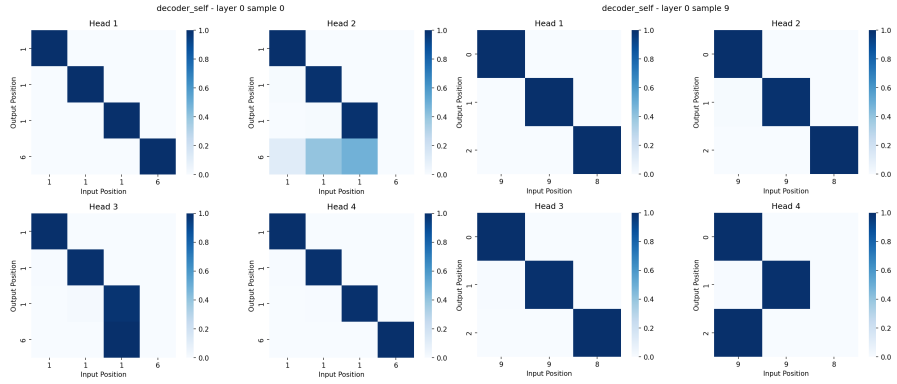

Figure 2: Decoder self

Figures 3 present decoder cross-attention maps, where each output token attends over the full encoder sequence (6 6 3 + 4 9 7). These heads reveal well-structured alignment patterns between input and output digits: lower decoder layers focus on local positional correspondence, while upper layers form broader connections that often capture carry propagation—linking output digits to the specific input pairs responsible for generating them.

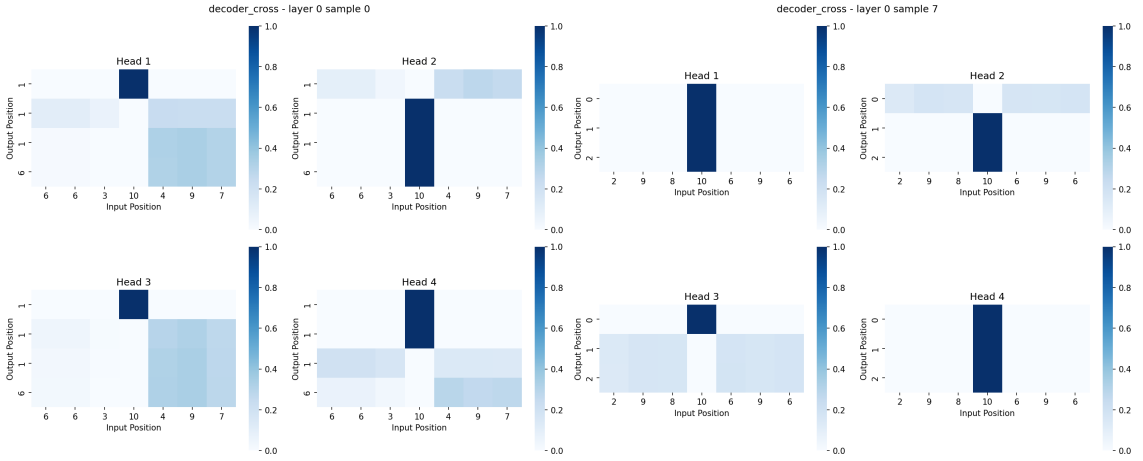

Figure 3: Decoder Cross

4. Head Ablation Study

Ablation experiments were conducted by disabling individual attention heads and measuring the corresponding change in test accuracy. The baseline model achieved 54.95% accuracy. Removing certain heads caused significant degradation, highlighting their critical roles in information encoding and reasoning.

Specifically, within encoder layer 0, heads h1 (3.5%), h2 (0.85%), and h3 (0.6%) (zero-indexed) led to near-collapse in model performance compared to h0 (54%). This indicates that these three heads are the core input-dependency heads: once removed, the model fails to correctly encode input structure and loses most of its addition capability. This finding is consistent with the earlier visualization analysis—heads 2, 3, and 4 (one-indexed) exhibited highly focused attention patterns with partial redundancy among them. They specialize in capturing local digit relationships and carry-relevant features, forming the foundation for the encoder's symbolic representation.

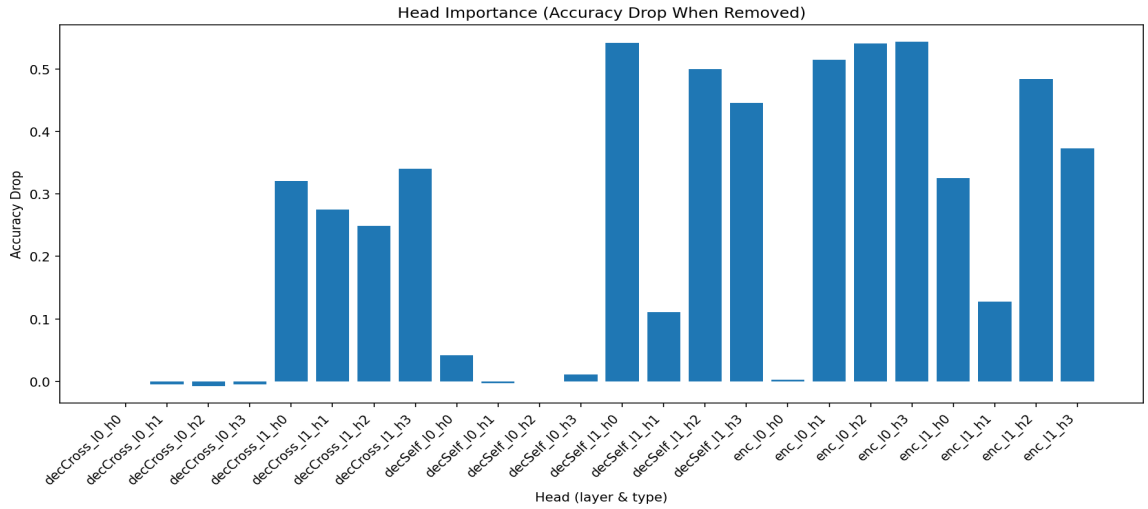

Figure 4 : Head Ablation

In encoder layer 1, heads h0 (22.45%) and h3 (17.65%) caused around 30–35% accuracy drop when ablated, suggesting that upper-layer heads act as global integrators, bridging positional and carry information across the sequence. In contrast, h1 (42.2%) showed minimal effect, implying its role is largely substitutable by other heads. Visual inspection supports this interpretation: head 1 primarily distinguishes the two operands or encodes the symbolic meaning of the "+" operator. This redundancy likely arises from task-specific overfitting—if subtraction examples were included, such a head would become indispensable.

Analysis of Positional Encodings

1. Overview

This experiment investigates how different positional encoding strategies — Sinusoidal, Learned, and None — affect a Transformer encoder's ability to generalize beyond training sequence lengths. The model performs binary classification (sorted = 1, unsorted = 0) on integer sequences of lengths 8–16 (training) and is evaluated on longer lengths (32, 64, 128, 256) to analyze extrapolation behavior.

2. Visualization of Positional Encodings

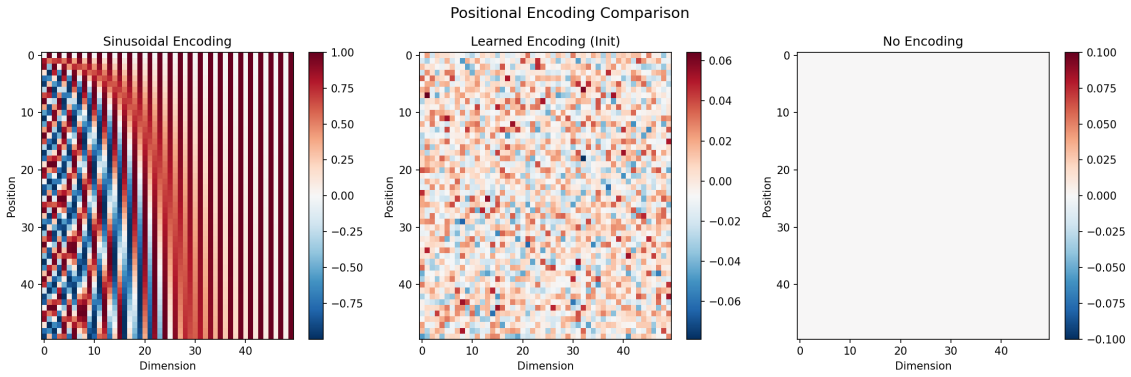

Figure 5: Visualization of three positional encoding strategies.

Sinusoidal encoding shows smooth periodic patterns across dimensions, encoding relative positions by sine and cosine waves of varying frequencies. Learned encoding starts as random noise (before training) with no structured pattern, and its representation depends entirely on the training length range. No encoding provides no positional signal, resulting in permutation invariance.

3. Length Extrapolation Performance

Sinusoidal and learned encodings both maintain high accuracy within the training range(green range in Figure 6). However, neither generalizes to longer sequences: their performance drops to approximately 0.5, equivalent to random guessing. For the None condition, the model also fails to achieve meaningful accuracy even on the training set. This is expected, since without positional encoding, the Transformer's self-attention and feed-forward operations are permutation-invariant.

The model treats the input as an unordered set of elements rather than a sequence, making it impossible to distinguish sorted from shuffled inputs.

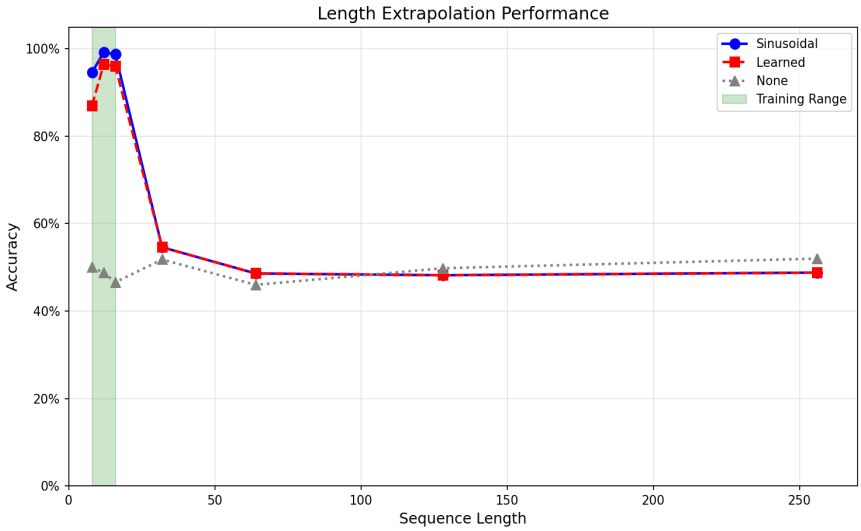

Figure 6. Accuracy vs. sequence length for all encoding methods.

For the learned encoding, each position index corresponds to a unique, trainable embedding vector. Positions beyond the training range (e.g., beyond length 16) have no learned representations. Even within the seen range, the model essentially learns an absolute lookup table rather than a continuous function of position, which prevents it from inferring unseen patterns or extrapolating to longer sequences.

The sinusoidal encoding, in theory, is designed to extrapolate: each embedding dimension represents a sinusoid with a distinct wavelength, allowing positions beyond the training range (e.g., 64, 128, 256) to be represented by the same analytical function. However, in this experiment, the sinusoidal model did not exhibit the expected extrapolation capability. After analysis, it became clear that the provided baseline — a standard Transformer encoder with absolute positional encodings — is theoretically incapable of true length extrapolation.

The reason lies in the structure of the attention mechanism:

$$Q = (X + PE)W_Q, K = (X + PE)W_K$$

The attention scores therefore depend on the linear combination of absolute position vectors:

$$QK^T = (PE_i W_Q)(PE_j W_K)^T$$

When unseen positions introduce new sinusoidal phases, these dot-product patterns differ drastically from those in the training range, leading to unpredictable attention behavior.

A suitable solution is to replace absolute position encodings with relative positional encodings, where the attention depends explicitly on the relative distance $(j - i)$ between tokens.
This approach, as used in models such as Transformer-XL and T5, allows consistent relational reasoning across arbitrary sequence lengths and supports true extrapolation.

4. Conclusion

This experiment systematically compared three positional encoding strategies — sinusoidal, learned, and none — for a Transformer-based sorting classifier. The results demonstrate that both sinusoidal and learned encodings achieve near-perfect accuracy within the training range (sequence lengths 8–16), while performance degrades sharply when evaluated on longer sequences. The "none" configuration fails entirely, confirming that positional information is essential for order-sensitive reasoning tasks.

The sinusoidal encoding, although theoretically capable of extrapolating due to its continuous analytical form, does not generalize in practice. This failure arises because the standard Transformer's attention mechanism depends on the absolute positional vectors through linear projections; the learned parameters are optimized only for the positions seen during training and cannot interpret unseen positional phases correctly. Overall, the findings highlight a key limitation of absolute positional encodings: they not true extrapolation.