Positional Encoding

Sinusoidal and Learned Absolute Positional Encodings

Positional encoding injects position information into transformer inputs to break the permutation invariance of self-attention. Sinusoidal encodings use geometrically spaced frequencies to enable relative position learning.

Key properties demonstrated:

- PE(pos+k) = M_k @ PE(pos) via rotation matrices
- Dot products depend only on relative distance (Toeplitz)
- Self-dot = d_model/2 (from sin^2 + cos^2 = 1)
- Wavelengths form geometric progression: 2pi to ~10000*2pi

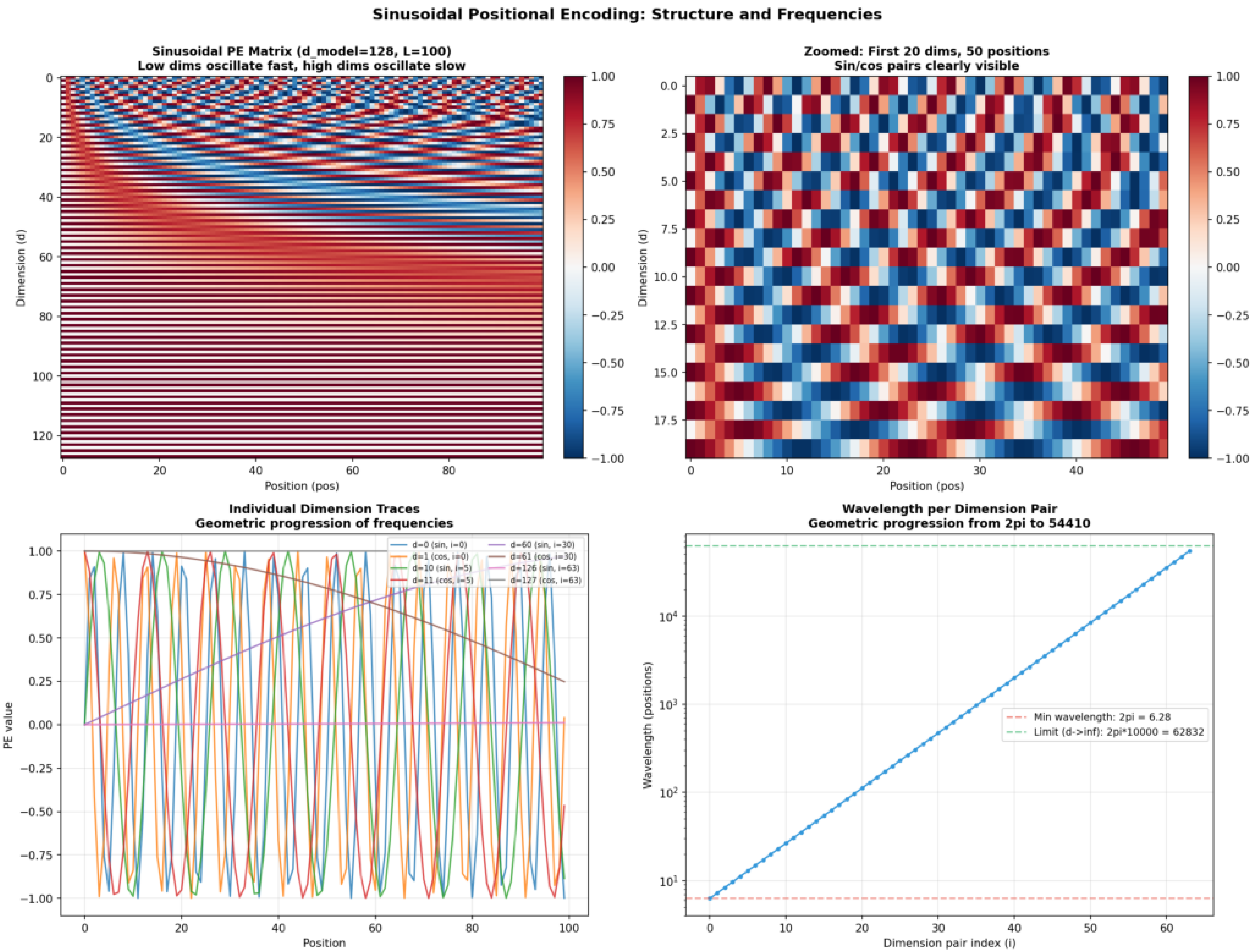
Random seed: 42
Number of visualizations: 6
Examples: 6

Generated by demo.py

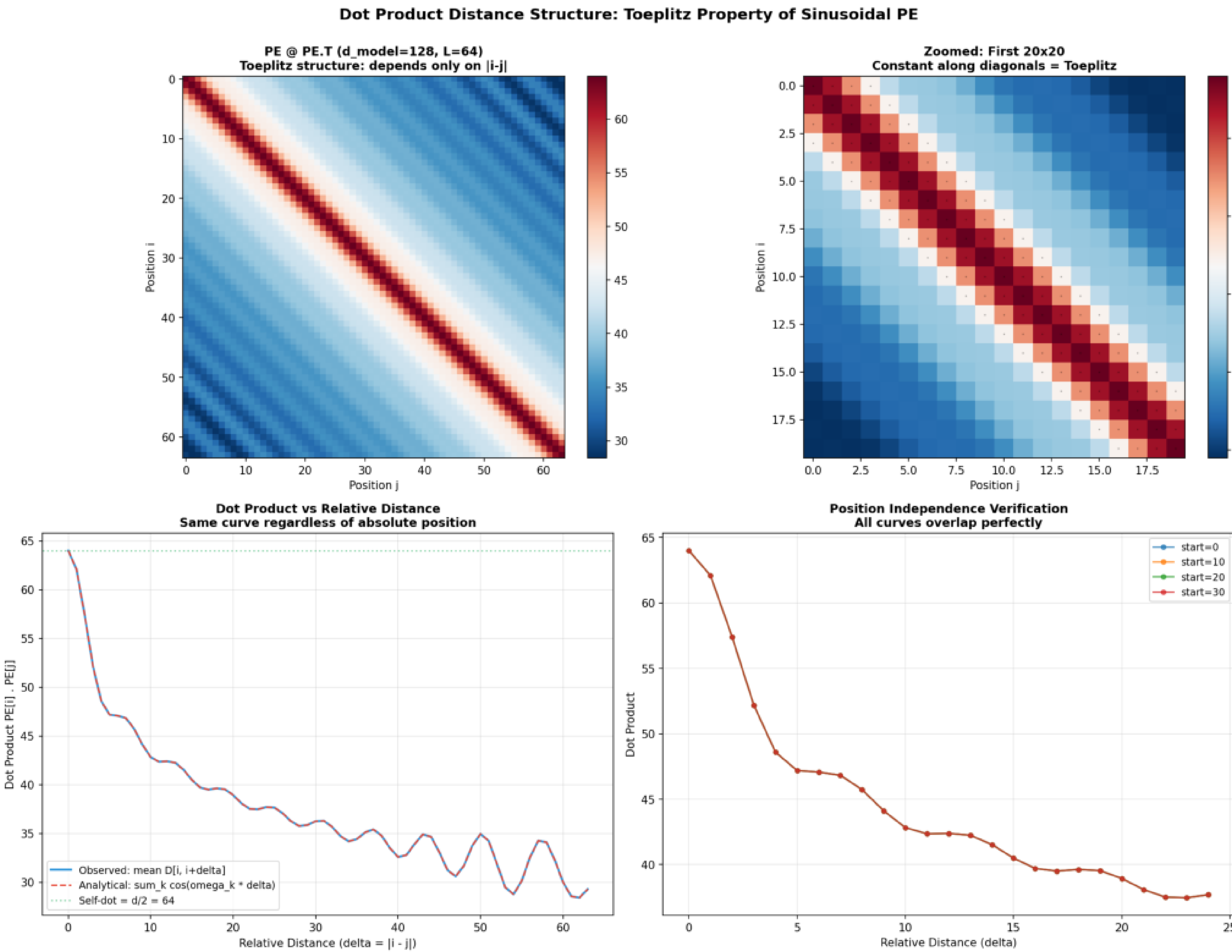
Summary of Findings

```
1. PE Matrix Structure: Sinusoidal PE shows characteristic wave patterns.
   Low dimensions (i=0) oscillate every ~6 positions (wavelength=2*pi).
   High dimensions (i=63) have wavelength ~54000 positions (d=128).
   Frequencies form a geometric progression with ratio 10000^(2/d).

2. Dot Product Distance: PE @ PE.T has perfect Toeplitz structure.
   D[i,j] = sum_k cos(omega_k * (i-j)), depends ONLY on relative distance.
   Self-dot product = d_model/2 exactly (from sin^2 + cos^2 = 1 per pair).
   Variation at same distance: ~1e-14 (floating-point rounding only).

3. Relative Position Property: PE(pos+k) = M_k @ PE(pos) EXACTLY.
   M_k is block-diagonal with d/2 rotation blocks R_i(k).
   Reconstruction error: ~1e-14 (analytically exact, limited by float64).
   This enables attention to learn relative positions via linear transforms.

4. Sinusoidal vs Learned: Sinusoidal has constant norm sqrt(d/2),
   perfect Toeplitz dot products, and unlimited extrapolation.
   Learned (random init) has variable norms, no Toeplitz structure,
   and is capped at max_seq_len.

5. Attention Impact: Without PE, attention is permutation-equivariant
   (diff ~0). With PE, permuting input changes output significantly,
   making attention position-aware.

6. Frequency Analysis: Wavelengths span 2*pi to ~10000*2*pi (exact max depends on d).
   Variance drops from ~0.5 (fast dims) to ~0 (slow dims).
   Crossover where wavelength = seq_len determines which dims are
   informative for a given sequence length.
```

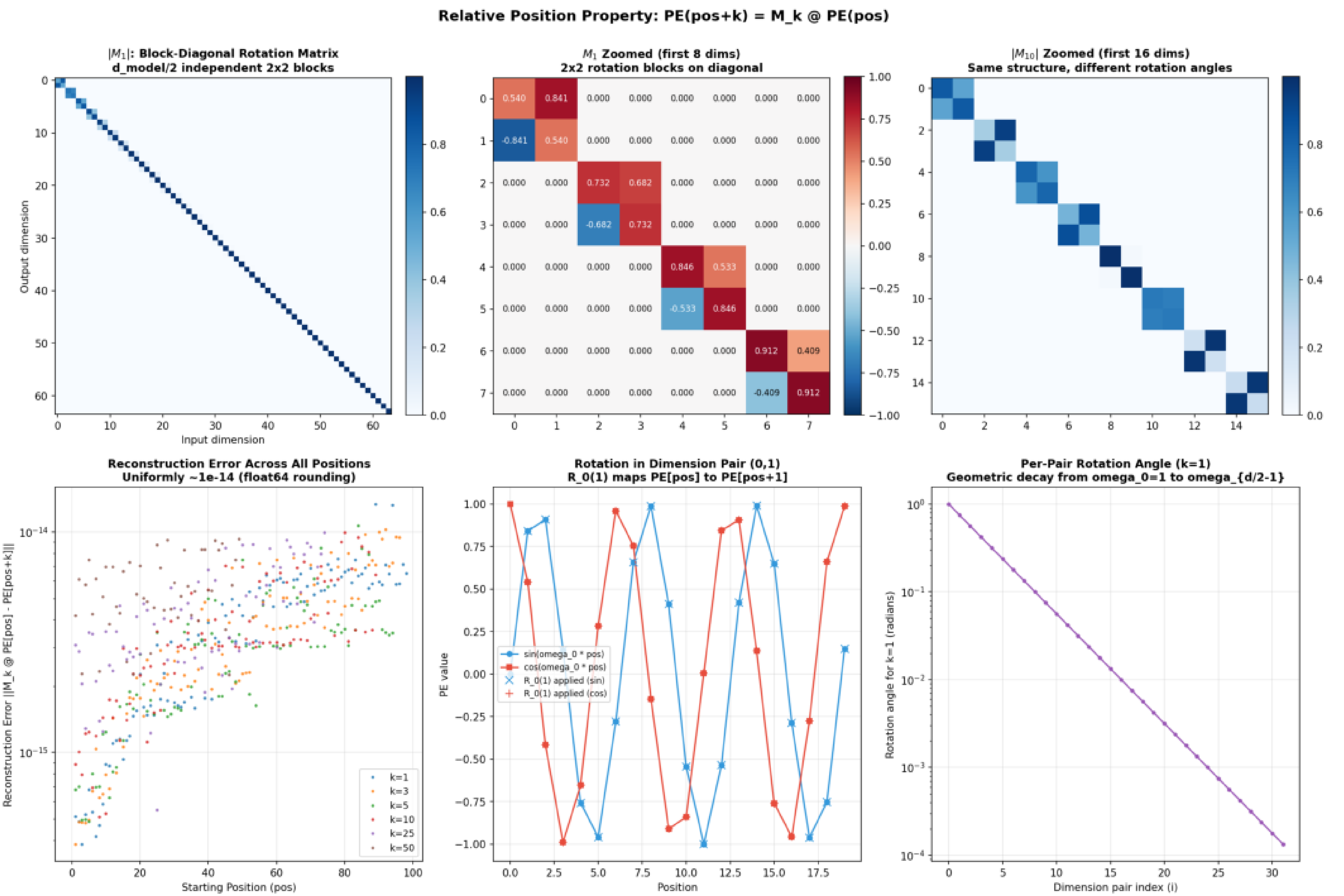
Example 1: PE Matrix Heatmap and Frequency Analysis



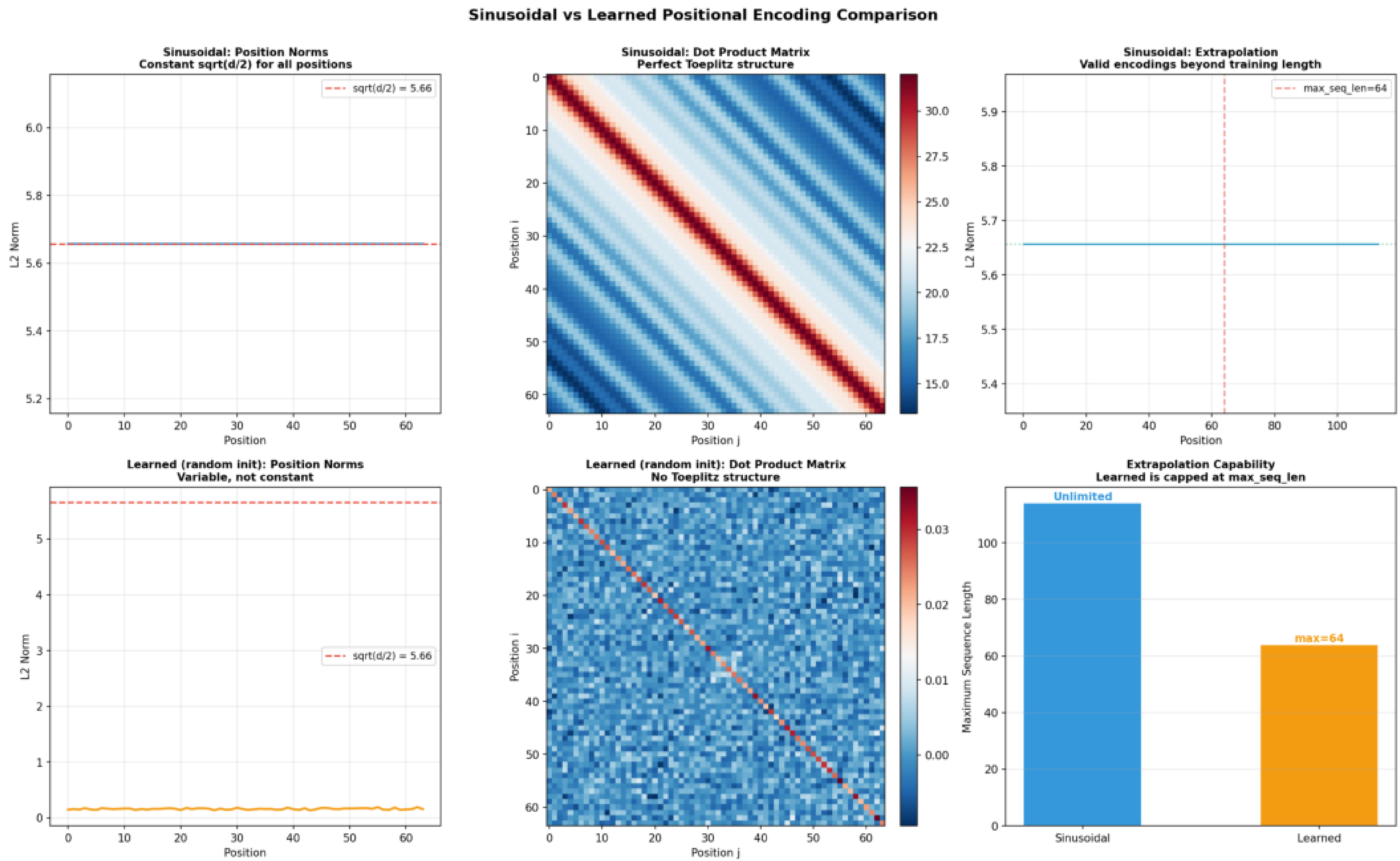
Example 2: Dot Product Distance Structure (Toeplitz Property)



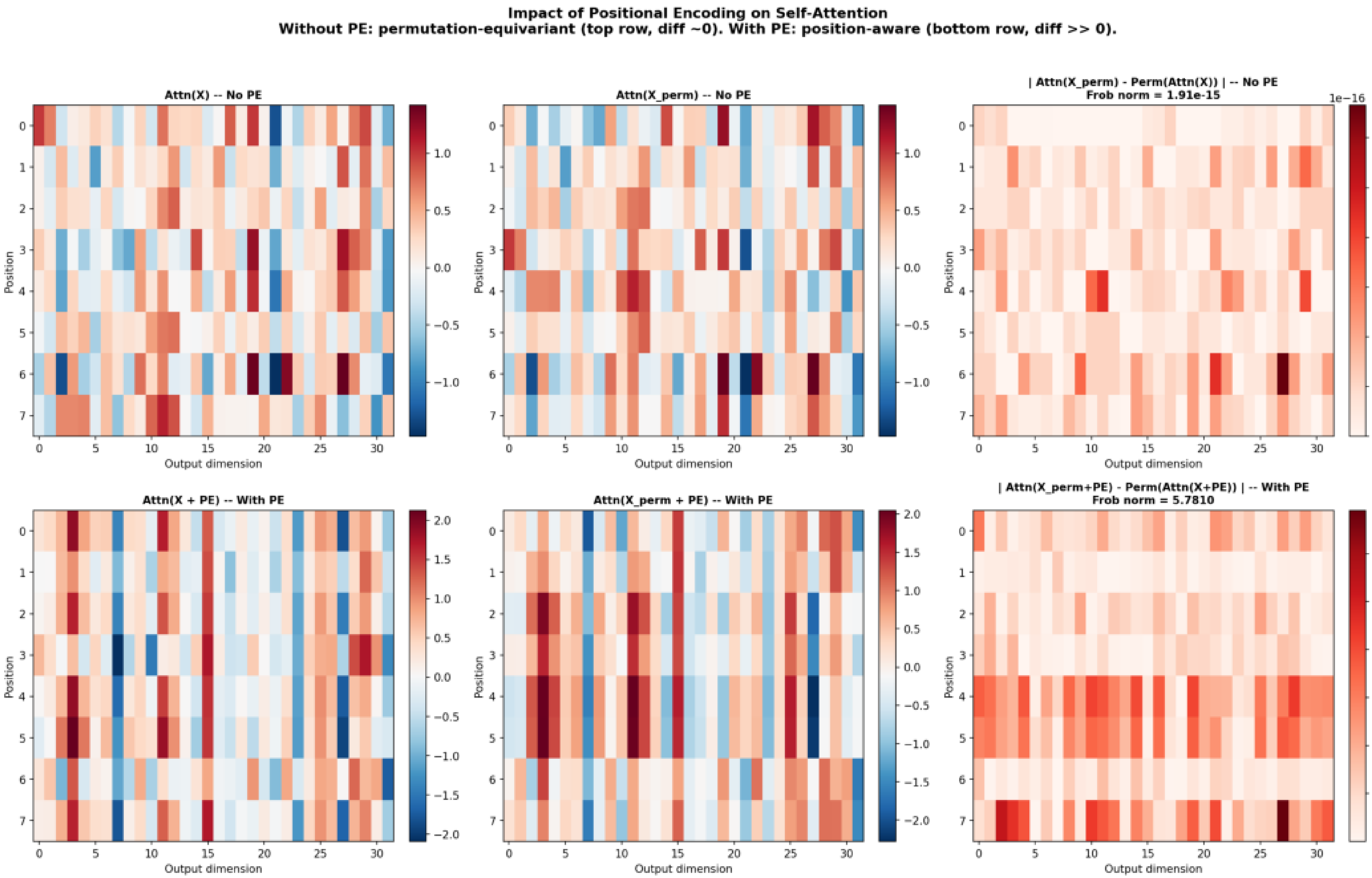
Example 3: Relative Position Property (Rotation Matrices)



Example 4: Sinusoidal vs Learned Encoding Comparison



Example 5: Impact on Self-Attention (Permutation Invariance)



Example 6: Frequency Structure and Variance Analysis