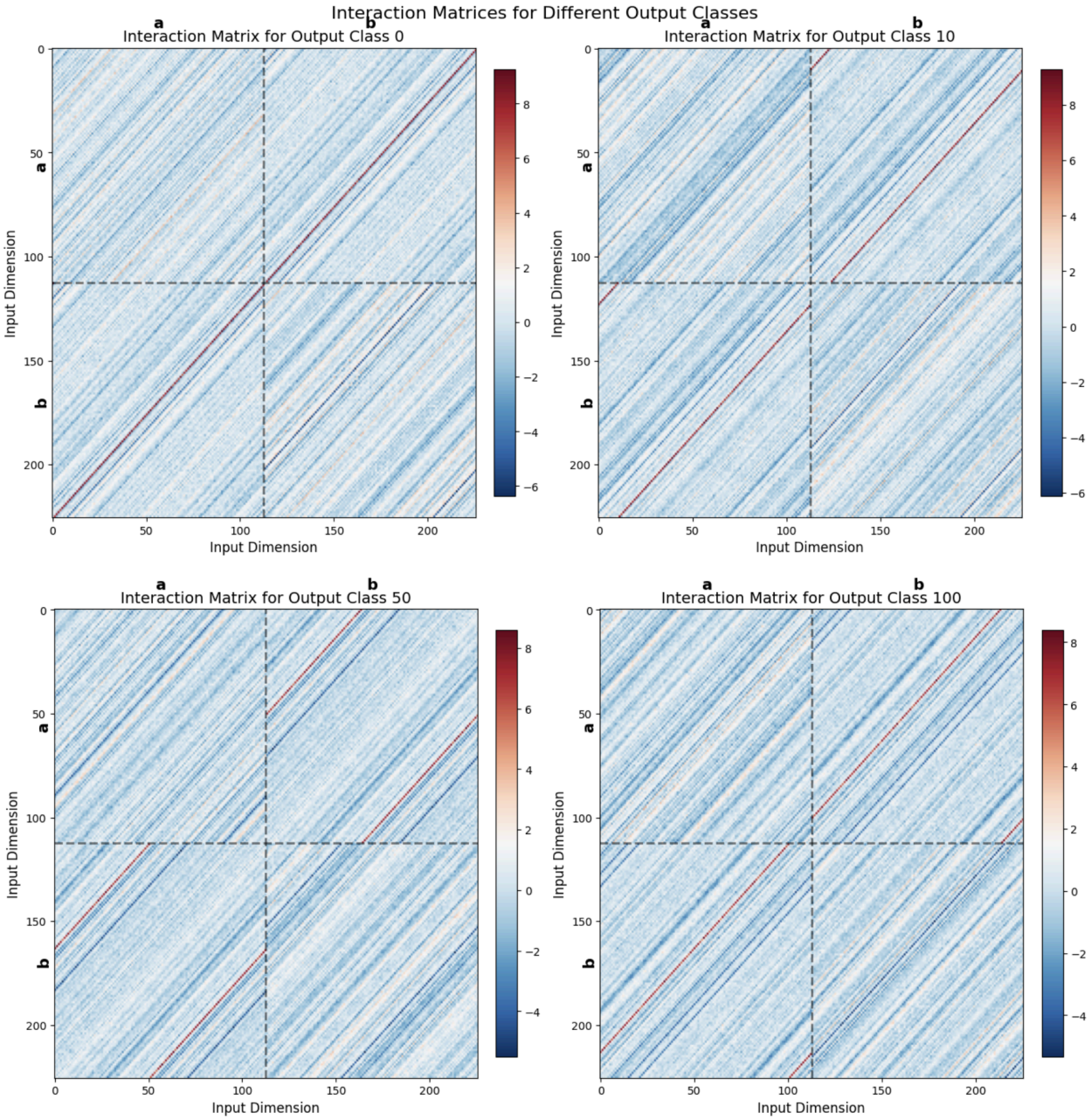
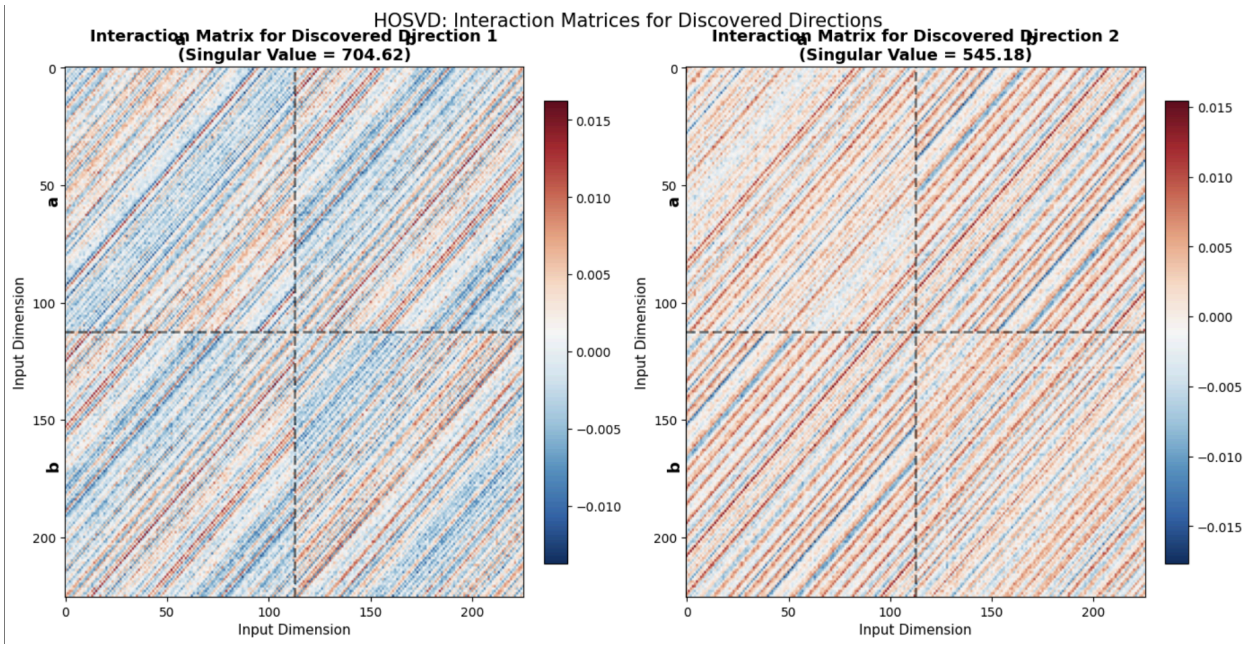
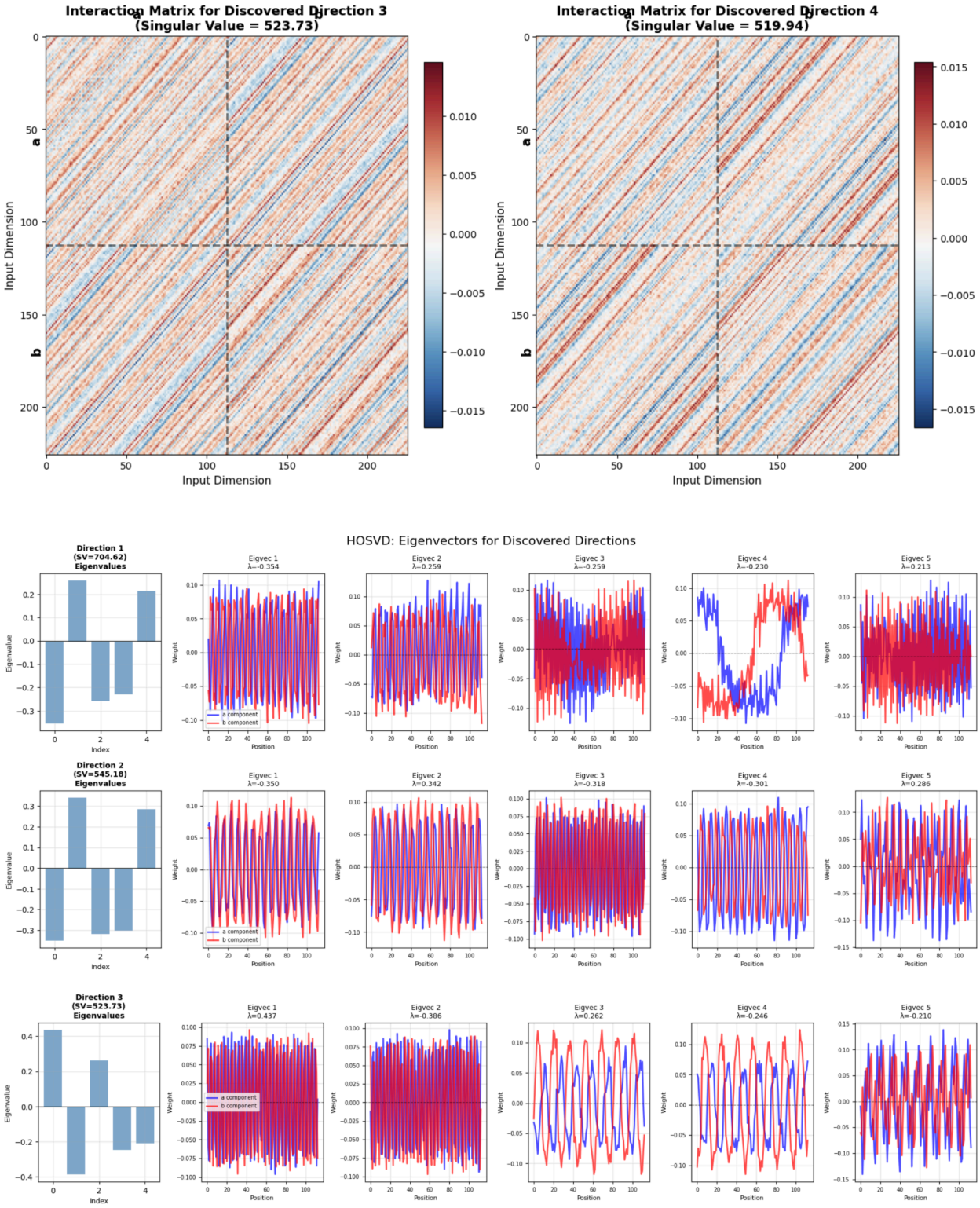
I trained a bilinear layer with AdamW, weight_decay=0.5 for 1000 epochs.
Code: https://github.com/Ryan-Rong-24/Bilinear-Mod



The interaction matrices reveal how the bilinear layer encodes modular addition directly in its weights. Each matrix Q corresponds to an output class c, where position (i, j) indicates the interaction strength between inputs a=i and b=j for predicting that output. The prominent red diagonal stripes in the cross-interaction quadrants (a×b and b×a) represent the exact solutions to a + b ≡ c (mod 113). As the output class increases from 0 to 100, the diagonal systematically shifts position, reflecting the modular arithmetic structure. The model learns to create constructive interference (red) along correct solution pairs and destructive interference (blue) elsewhere.

Section 3.3: Top eigen-vectors of 3rd-order tensor





Higher-Order SVD automatically discovered 5 dominant computational directions (singular values: 704.62–516.11) that organize modular addition without prior knowledge of output meanings. Each discovered direction combines multiple output classes with shared structure, as shown by their output contribution patterns. The interaction matrices for these directions exhibit

diagonal stripes encoding addition operations, while eigendecomposition reveals low-rank structure with 2-3 dominant eigenvalues. The eigenvectors display periodic oscillations in a and b components, similar to Section 3.2 but for discovered meta-patterns rather than individual output classes. This unsupervised decomposition exposes the model's internal hierarchical organization of computational space.

Additional:

Section 3.2: Output features → EIGENDECOMPOSITION