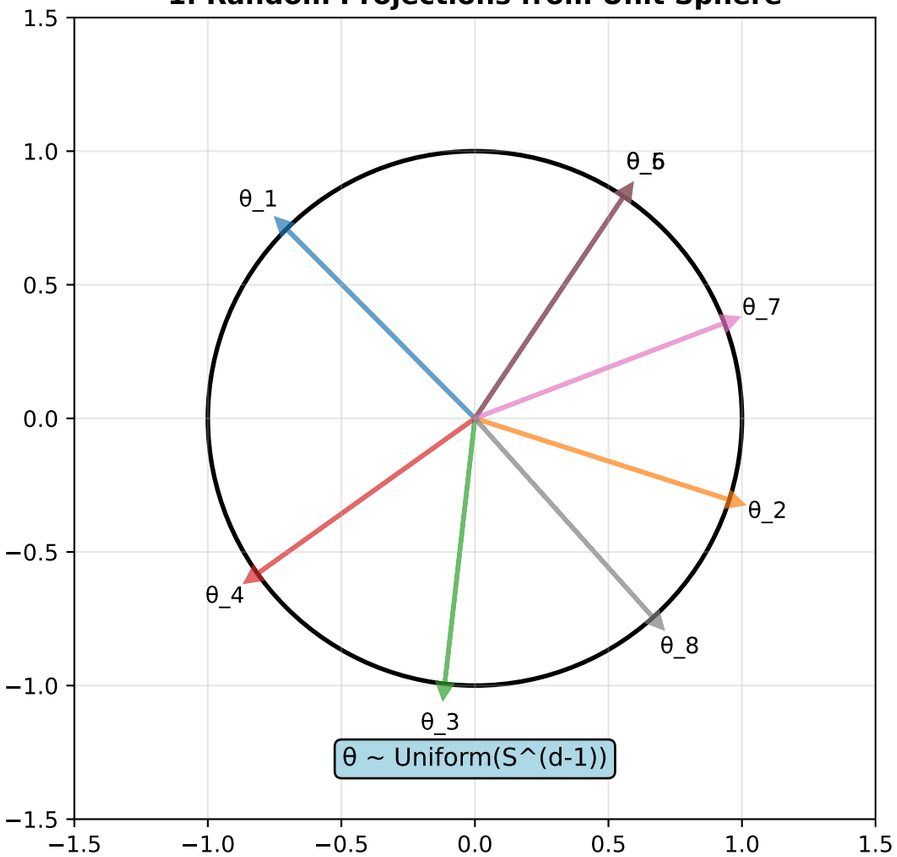
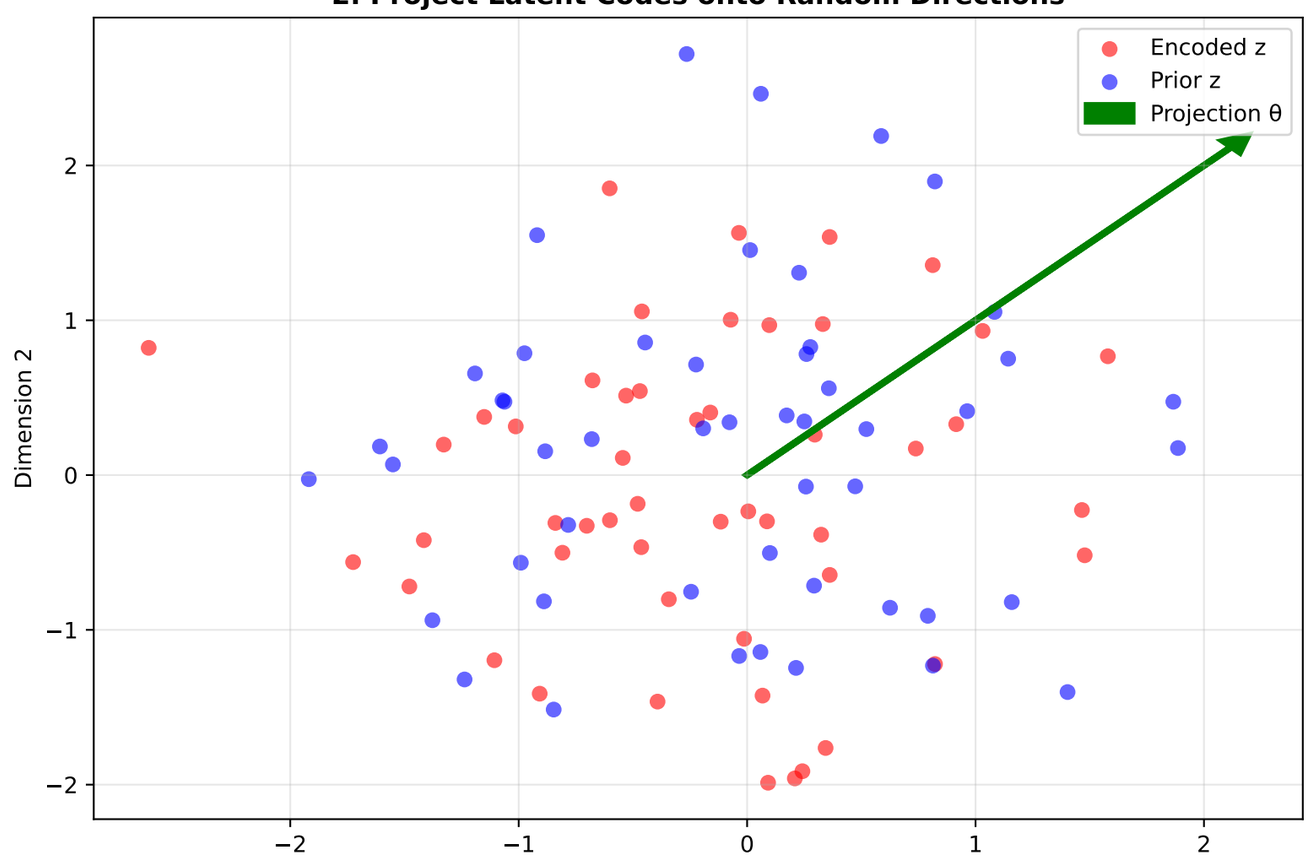
Sliced Wasserstein Distance in SWAE

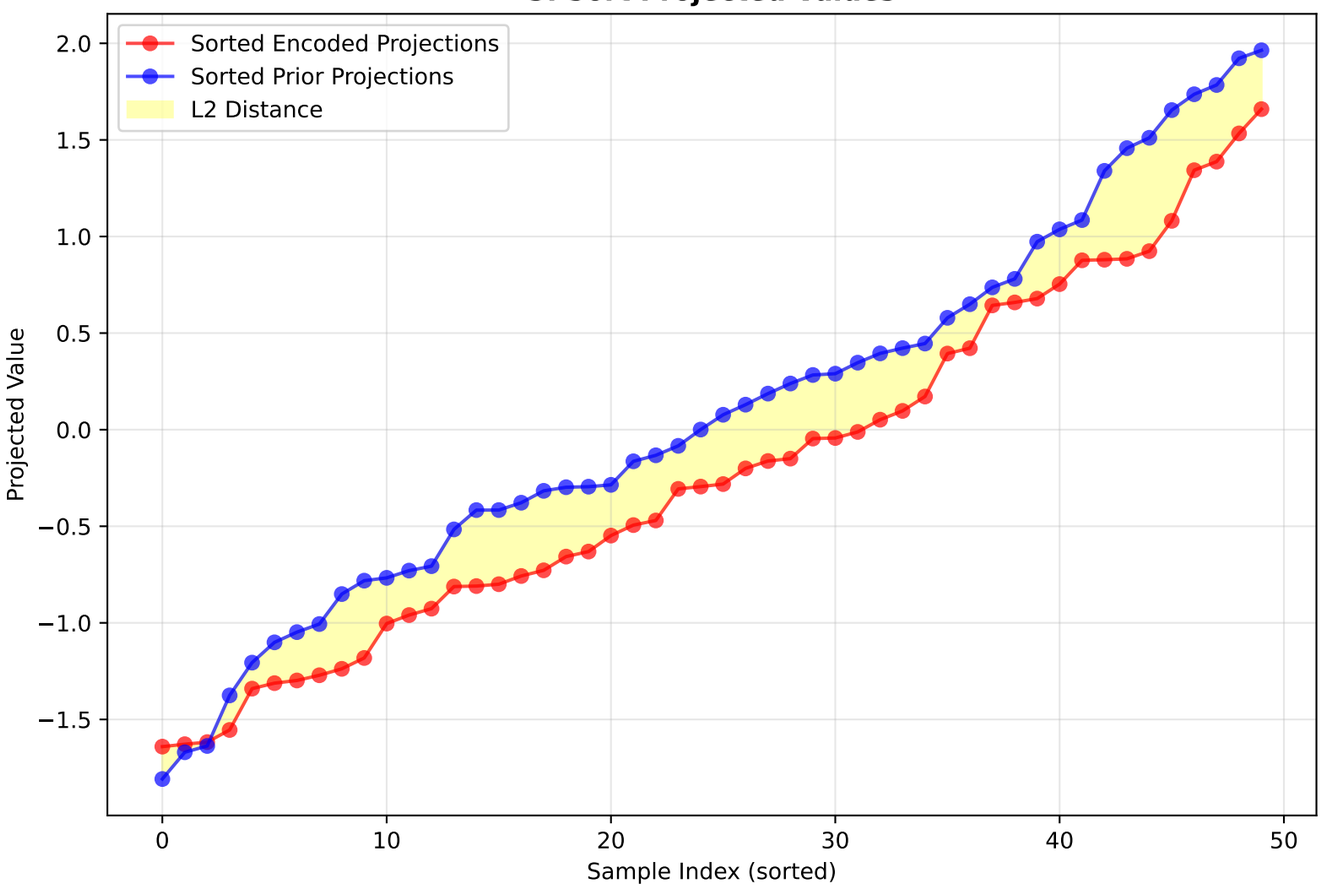
1. Random Projections from Unit Sphere



2. Project Latent Codes onto Random Directions



3. Sort Projected Values



4. Sliced Wasserstein Algorithm

```
Algorithm: Sliced Wasserstein Distance

Input: z_encoded, z_prior, num_projections=50

1. For l = 1 to num_projections:
   • Sample θ_l ~ Uniform(S^(d-1))
   • Compute p_l^enc = θ_l^T · z_encoded
   • Compute p_l^prior = θ_l^T · z_prior
   • Sort: p̃_l^enc = sort(p_l^enc)
   • Sort: p̃_l^prior = sort(p_l^prior)
   • Compute: d_l = ||p̃_l^enc - p̃_l^prior||²

2. Return: SW = (1/L) Σ d_l

Complexity: O(n log n) per projection
Total: O(L·n log n) where L=50, n=batch_size
```