---

# ISDO Training Regimes — Implementation Plan & Tracking Summary

## Scope

These regimes **build on Direction 1 (ISDO)**, which is **closed**. They **do not modify**:

- the ISDO observable
- the ISDO circuit (B )

They only define **how class states are constructed and updated**.

---

## Shared Foundations (Applies to All Regimes)

### Fixed Components (Never Trained)

- **Quantum circuit**: ISDO-B  (transition-based interference)
- **Measurement**: ancilla $\langle Z \rangle$
- **Decision rule**:

$$\hat{y}(x) = \text{sign}(\text{Re}\langle \chi | \psi(x) \rangle)$$

### Learned Objects

- **Quantum class state(s)** $|\chi\rangle$
- These are the **only trainable parameters**

### Data Pipeline

```
image
  -> CNN
  -> embedding ($\mathbb{R}^{32}$)
  -> L2 normalize
  -> $|\psi\rangle$
  -> ISDO($|\psi\rangle$, $|\chi\rangle$)
```

---

# Regime 1 — Static Prototype Aggregation (Baseline)

### Purpose

- Non-iterative baseline

- Fast, stable, interpretable
- Already implemented and validated

---

### Definition

Given labeled embeddings:

- Class +1: $\{|\phi_k^+\rangle\}$
- Class −1: $\{|\phi_k^-\rangle\}$

Construct:

$$|\chi\rangle = \text{normalize}\left(\sum_k |\phi_k^+\rangle - \sum_k |\phi_k^-\rangle\right)$$

---

### Implementation Steps

1. Collect embeddings per class
2. Optionally cluster (KMeans) -> prototypes
3. Sum positive prototypes
4. Subtract negative prototypes
5. Normalize -> class state $|\chi\rangle$

### Inference

For each test sample:

$$s = \text{Re}\langle\chi|\psi\rangle \quad \Rightarrow \quad \hat{y} = \text{sign}(s)$$

---

### Properties

| Aspect | Status |
|---|---|
| Training | One-shot |
| Updates | None |
| Stability | Very high |
| Circuit depth | Fixed |
| QML validity | Yes |
| Novelty | Moderate |

---

### When to Use

- Baseline comparisons
- Low-data regimes
- Cold-start initialization for Regime 2 or 3

---

## Regime 2 — Online / Incremental ISDO Training (Recommended Core)

### Purpose

- Enable **learning over time**
- Adapt class state to data distribution
- Avoid gradients, shots, or parameterized circuits

This is the **quantum perceptron regime**.

---

### Initialization (Cold Start)

Option A — Zero state:

$$|\chi_0\rangle = |0\rangle^{\otimes n}$$

Option B — Bootstrap (recommended):

$$|\chi_0\rangle = \text{normalize}\left(\sum_{i=1}^{K} y_i|\psi_i\rangle\right)$$

---

### Online Update Rule

For each training sample $(|\psi\rangle, y)$, with $y \in \{+1, -1\}$:

1. Measure:

$$s = \text{Re}\langle\chi_t|\psi\rangle$$

2. If correctly classified:

```
y * s >= 0  ->  no update
```

3. If misclassified:

$$|\chi_{t+1}\rangle = \text{normalize}\left(|\chi_t\rangle + \eta \cdot y|\psi\rangle\right)$$

---

#### Implementation Notes

- Update is **classical vector arithmetic**
- Renormalize after every update
- Learning rate $\eta$ can be:
  - constant
  - decaying
  - adaptive

---

#### Properties

| Aspect | Status |
|---|---|
| Training | Online |
| Updates | Deterministic |
| Gradients | None |
| Circuit | Fixed |
| Shot cost | Minimal |
| Interpretability | High |
| QML validity | Strong |

---

#### When to Use

- Continual learning
- Streaming data
- Non-stationary datasets
- As the **main learning regime**

---

## Regime 3 — Multi-State Quantum Memory (Advanced / Nonlinear)

#### Purpose

- Increase expressivity
- Approximate nonlinear decision boundaries
- Avoid deep circuits or kernels

This is **ensemble learning in Hilbert space**.

---

### Model Structure

Maintain a set of class states:

$$\mathcal{M} = \{|\chi^{(1)}\rangle, \dots, |\chi^{(M)}\rangle\}$$

Each state:

- initialized independently
- trained independently (Regime 2 rule)

### Inference Rule

For input $|\psi\rangle$:

$$s_j = \mathrm{Re}\langle\chi^{(j)}|\psi\rangle$$

Decision:

$$\hat{y} = \mathrm{sign}\left(\max_j s_j\right)$$

---

### Training Variants

**Variant A — Winner-updates**

- Only the best-matching (|\chi^{(j)}\rangle) is updated

**Variant B — All-updates**

- All memory states updated with different learning rates

**Variant C — Pruned memory**

- Periodically discard low-utility states

---

### Properties

| Aspect | Status |
|---|---|
| Expressivity | High |
| Circuit depth | Fixed |
| Parameters | Quantum states |
| Interpretability | Medium |
| QML novelty | Very high |
| Complexity | Moderate |

---

### When to Use

- Nonlinear class boundaries
- Multimodal class distributions
- Few-shot / continual learning

---

## Relationship Between Regimes

```
Regime 1
  ↓ (initialization)
Regime 2
  ↓ (parallelization)
Regime 3
```

- Regime 1 feeds Regime 2
- Regime 2 generalizes to Regime 3
- No regime invalidates ISDO

---

## Key Constraints (Must Not Violate)

- Do NOT train gate parameters
- Do NOT introduce variational ansätze
- Do NOT change the ISDO observable
- Always renormalize class states
- Updates must be linear combinations of embeddings

---

## Final Status Summary

| Regime | Purpose | Status |
|---|---|---|
| Regime 1 | Static baseline | Implemented |
| Regime 2 | Online learning | Ready to implement |
| Regime 3 | Nonlinear extension | Defined |

---

### Recommended Next Action

**Implement Regime 2 end-to-end** on your existing embeddings and log:

- accuracy vs time

- number of updates
- stability of $|\chi\rangle$

Once Regime 2 is stable, Regime 3 becomes trivial.