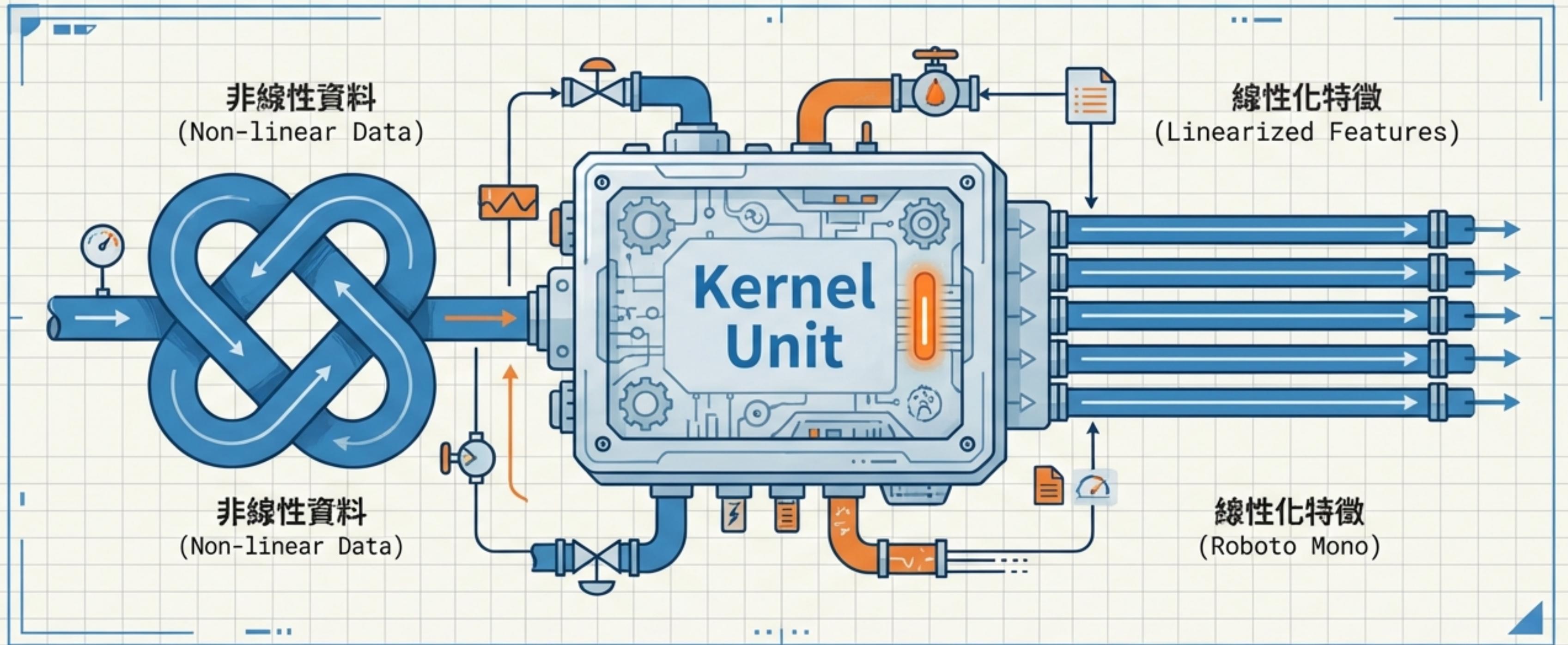


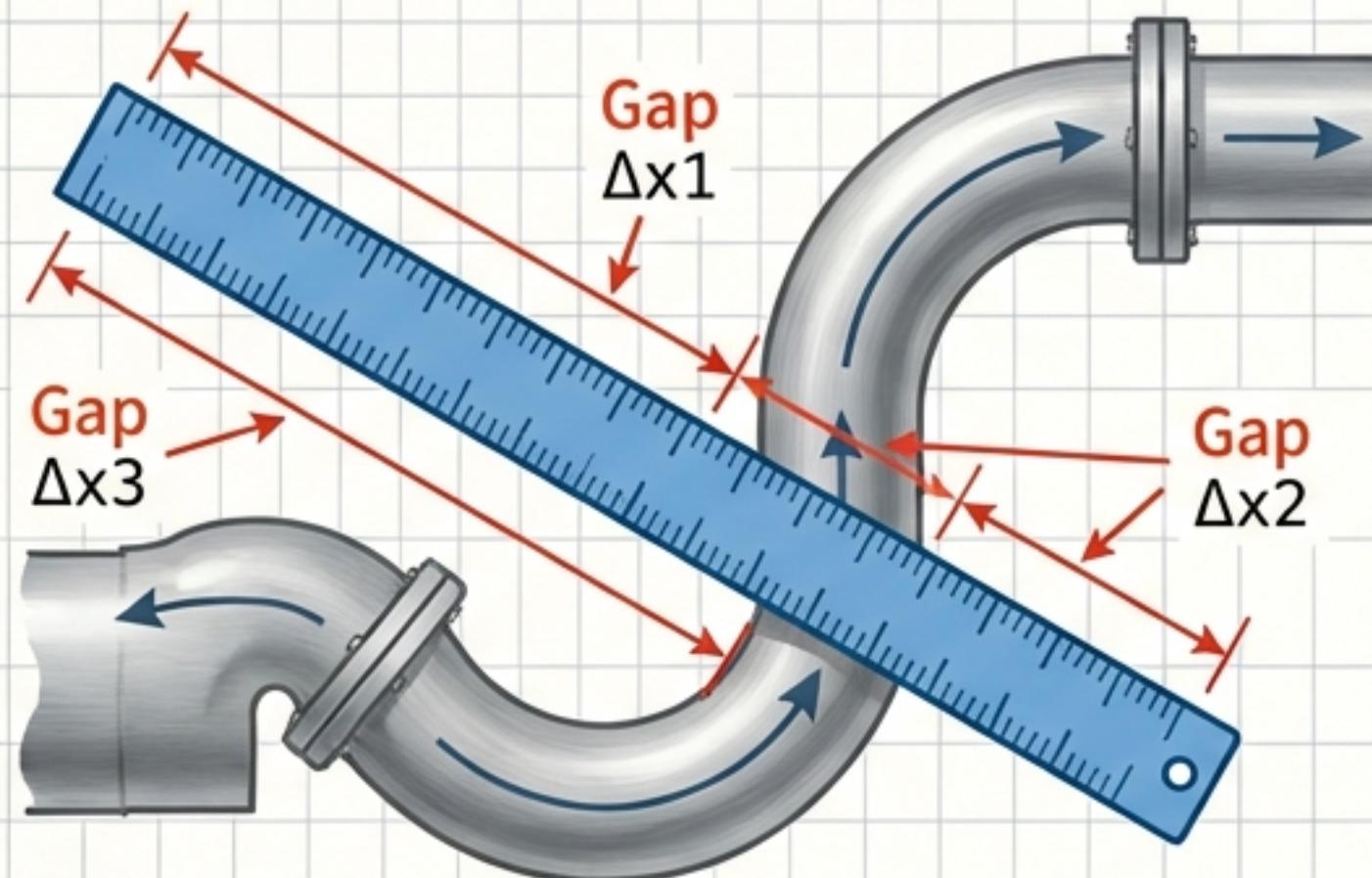
Unit 06: 核主成分分析 (Kernel PCA)

突破線性限制：化工製程的非線性特徵提取



線性假設 vs. 化工現實 (The Linear Limit)

假設 (The Assumption)



傳統 PCA 假設數據結構是平直的 (Linear)。

現實 (The Reality)



- 反應動力學：
Arrhenius Equation
(速率 vs 溫度呈指數關係)

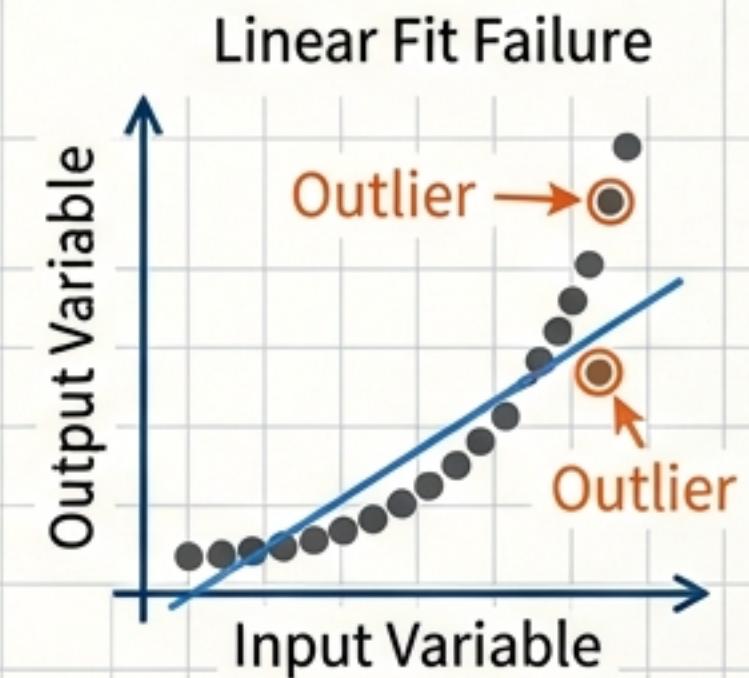


- 相平衡：
VLE Curves,
Azeotropes



- pH 控制：
對數關係與緩衝區
 $pH = -\log[H^+]$

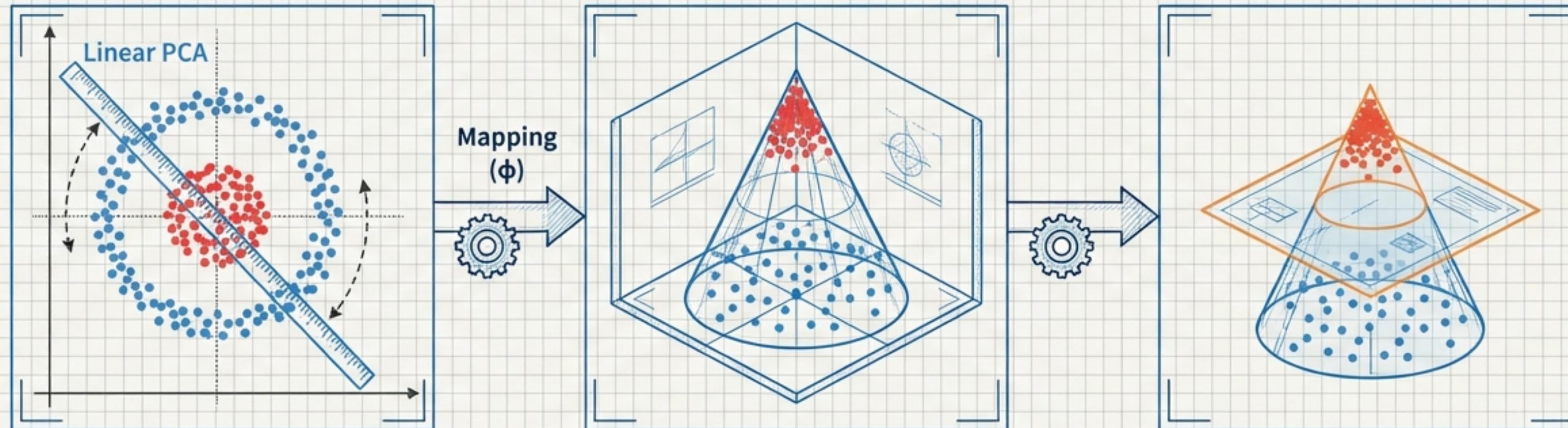
$$k = A * \exp\left(-\frac{Ea}{R*T}\right)$$



化工廠不是線性的，為何我們使用線性工具？

什麼是 Kernel PCA? (Bending Space)

PCA 的非線性擴展，透過將數據映射至高維特徵空間 (\mathcal{F}) 來提取結構。



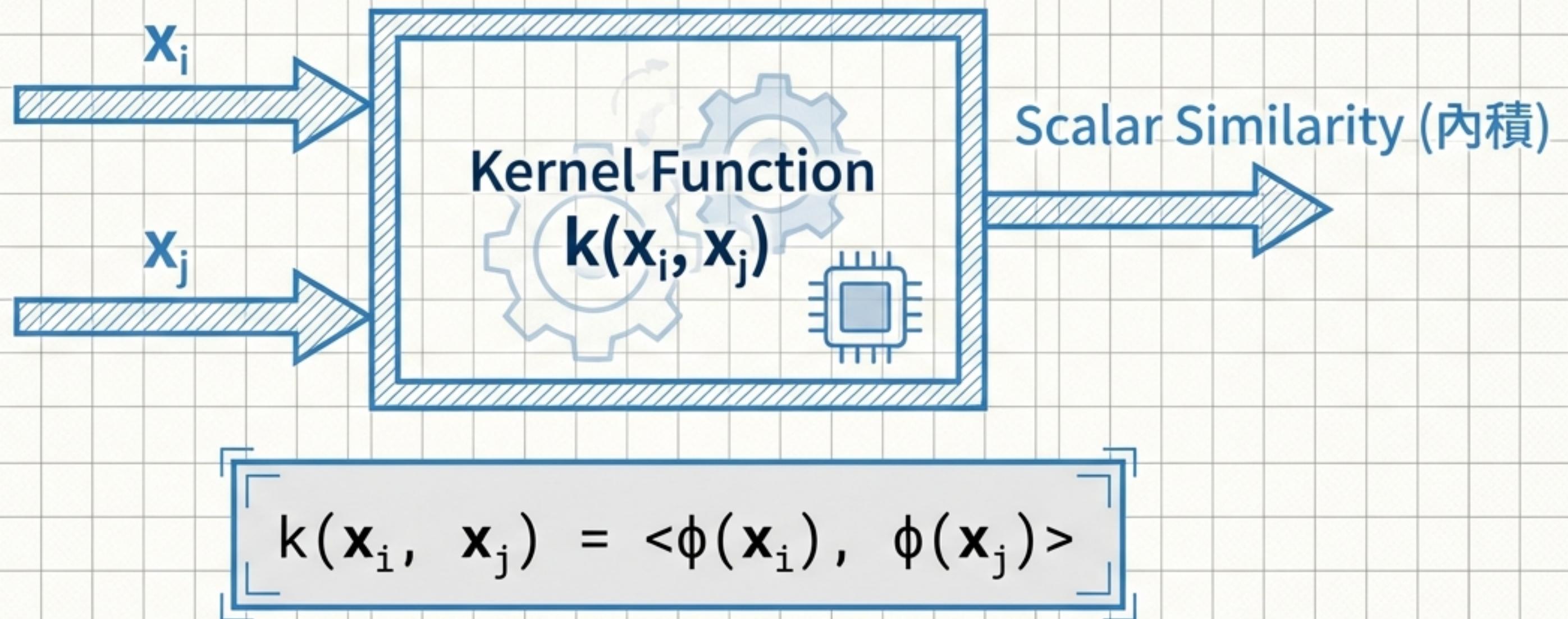
1. Input Space (2D)

2. Mapping (ϕ)

3. Feature Space

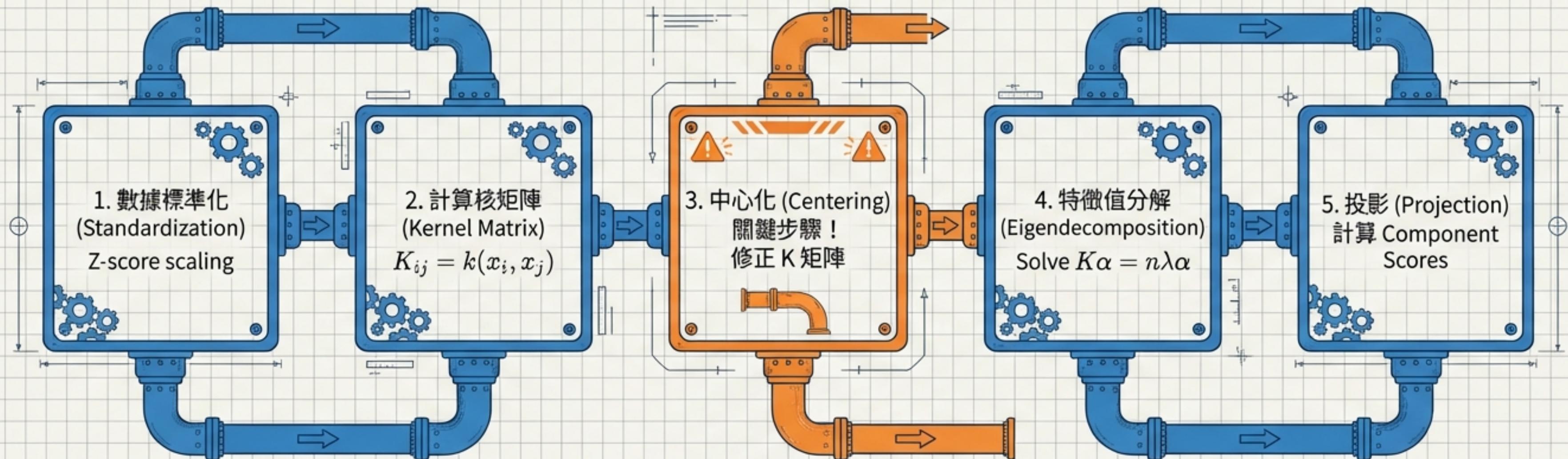
$$\mathbf{x} \in \mathbb{R}^p \rightarrow \phi(\mathbf{x}) \in \mathbb{F}$$

核技巧 (The Kernel Trick)：計算捷徑



我們不需要真的「去」高維空間，我們只需要計算「距離」。
就像不需拆開反應器，只需測量入口與出口的溫差。

Kernel PCA 演算法流程圖 (Process Blueprint)

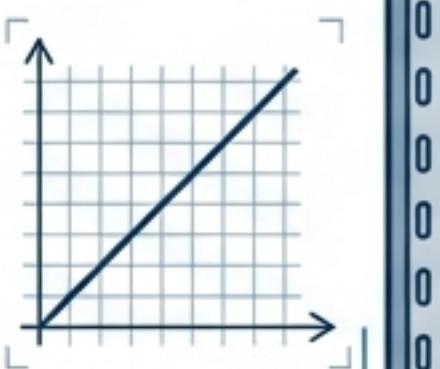


核函數工具箱 (Kernel Toolbox)

1. 線性核 (Linear)

$$k(x, y) = \mathbf{x}^T \mathbf{y}$$

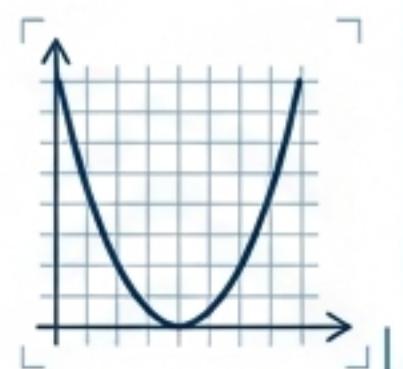
即標準 PCA，基準模型。



2. 多項式核 (Polynomial)

$$k(x, y) = (\gamma \mathbf{x}^T \mathbf{y} + c_0)^d$$

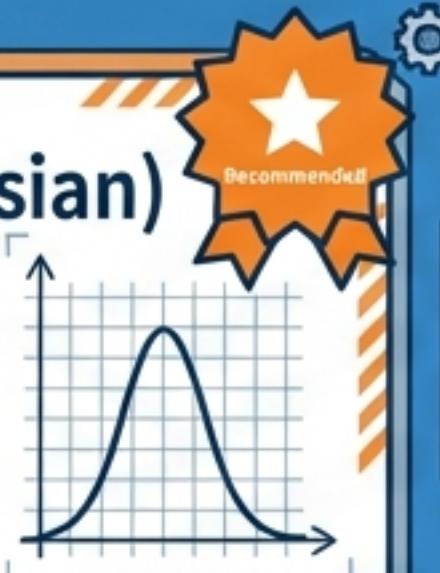
適合反應動力學 (Kinetics)、
交互作用項。



3. 徑向基函數核 (RBF/Gaussian)

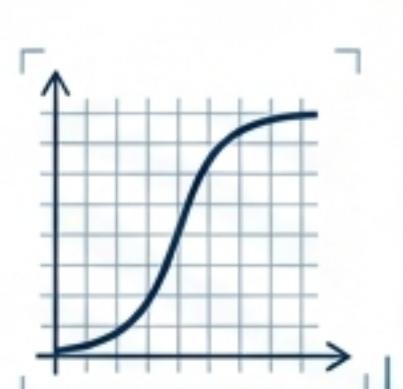
$$k(x, y) = \exp(-\gamma ||\mathbf{x} - \mathbf{y}||^2)$$

通用性最強，可處理大多數
非線性。



4. Sigmoid

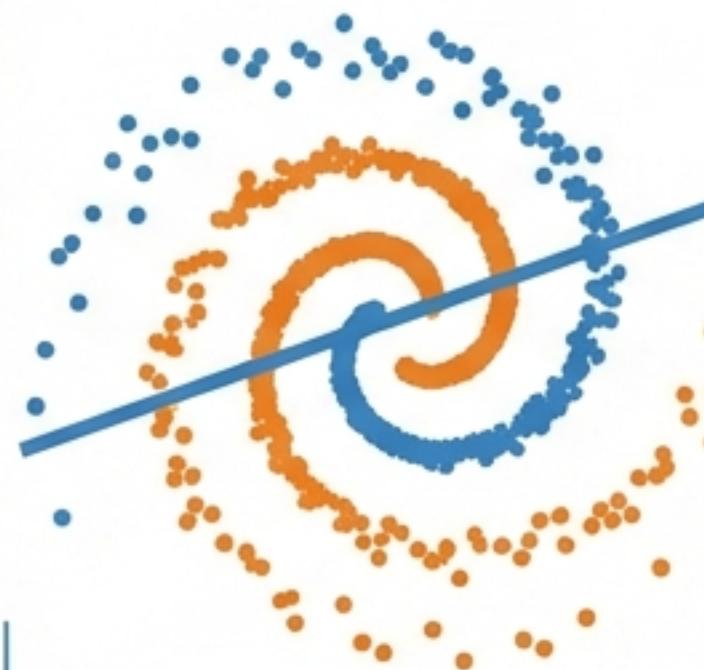
類神經網絡激活函數，較少
用。



參數調校：尋找最佳設定 (Hyperparameters)

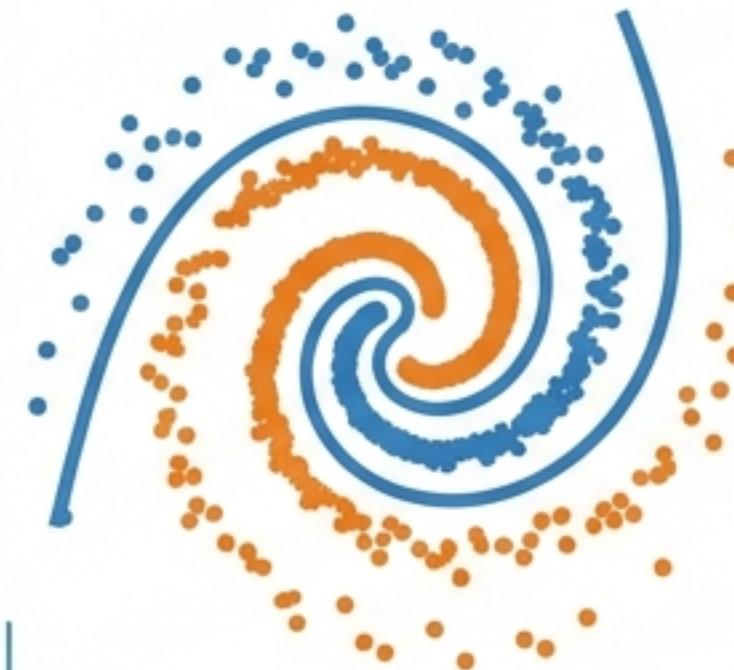
RBF Kernel 的 Gamma (γ) 參數影響

Low γ (0.001)



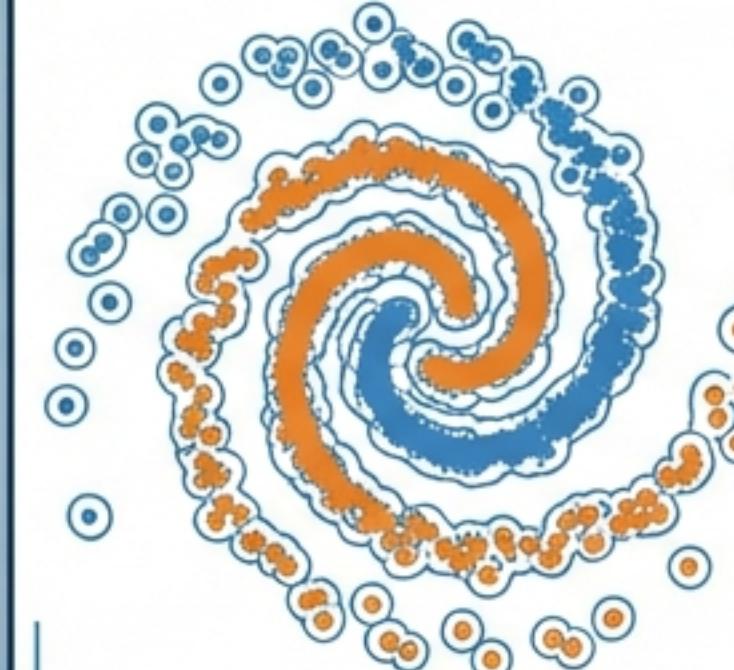
欠擬合 (Underfit)

Optimal γ (0.1 - 1.0)



✓ 最佳擬合 (Optimal)

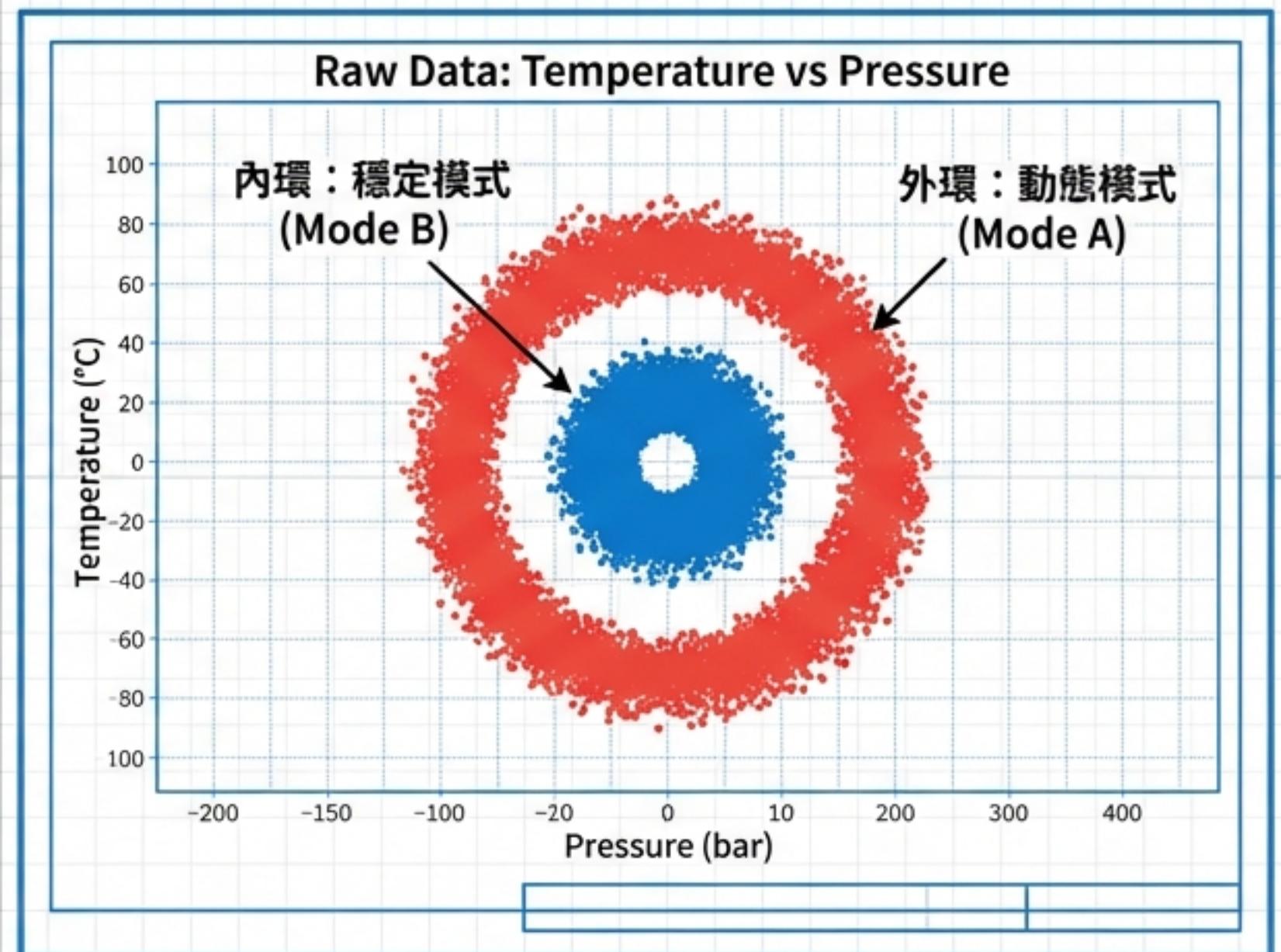
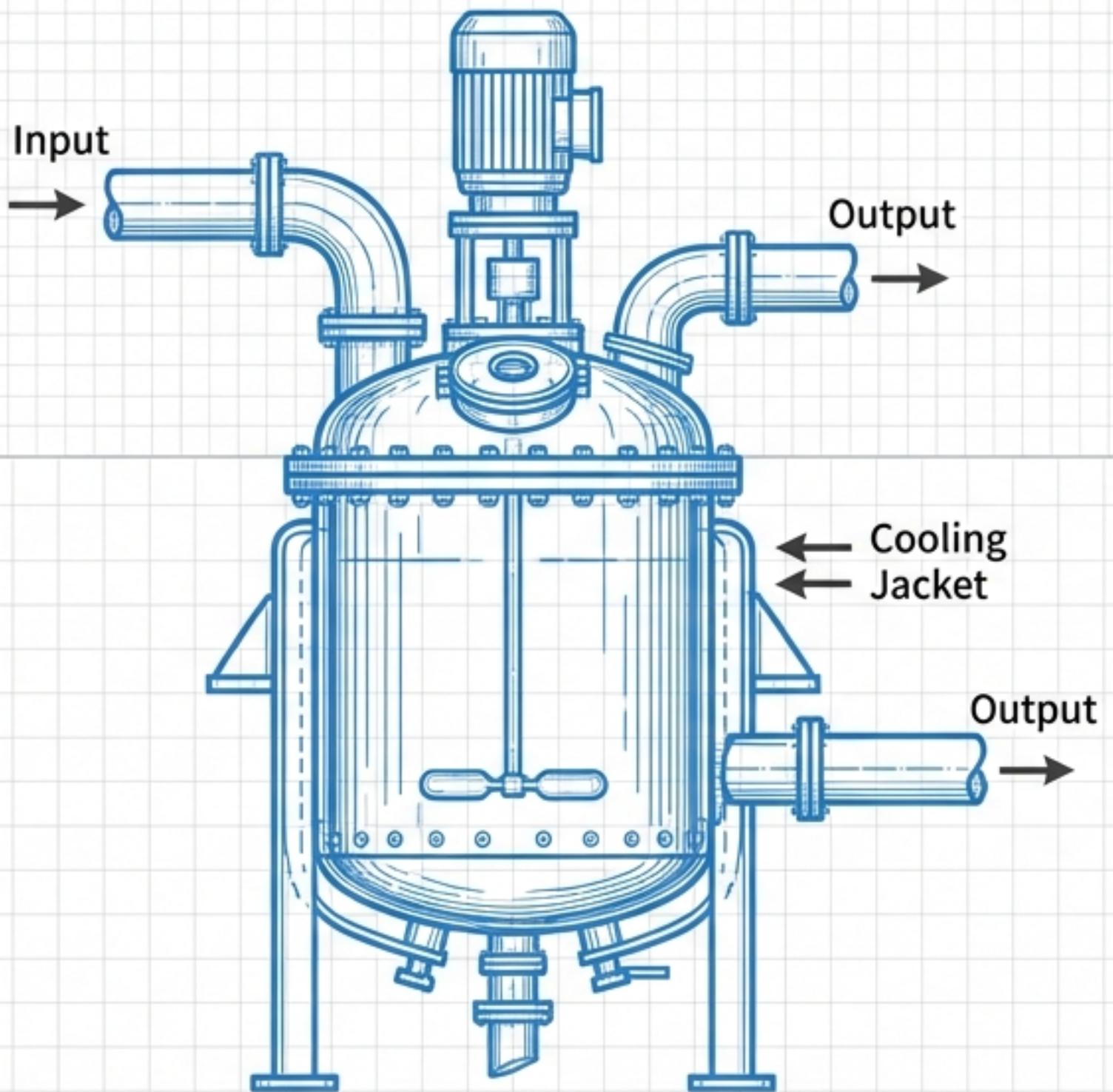
High γ (10+)



⚠ 過擬合 (Overfit)

建議策略: 網格搜索 (Grid Search) + 交叉驗證 (Cross-Validation)

案例研究：反應器操作模式識別

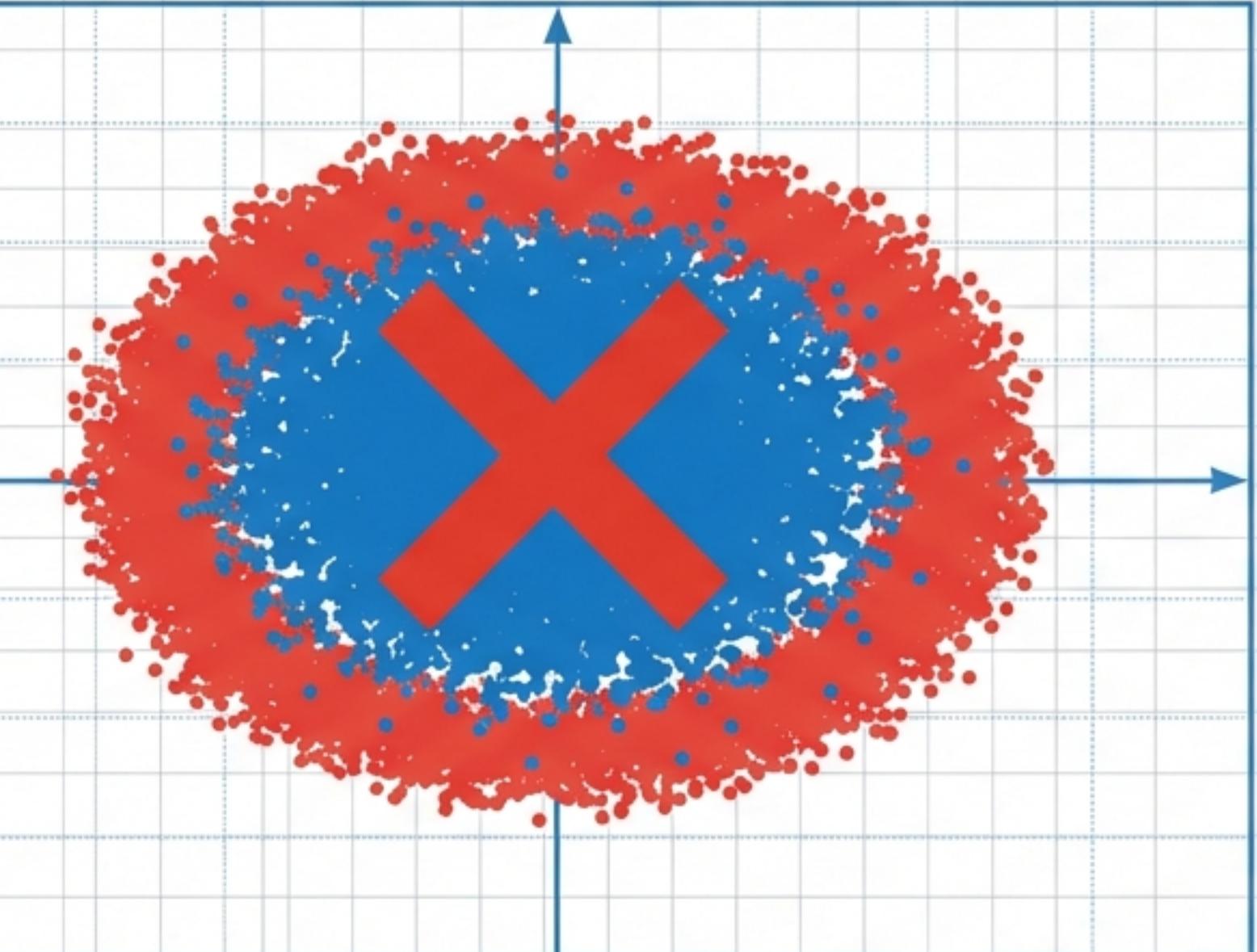


挑戰：數據呈現「同心圓」結構。線性投影會將內環與外環壓扁在一起，導致無法分辨。

傳統 PCA 的困境 (The Trap)

Bad Code 

PCA Result (PC1 vs PC2)

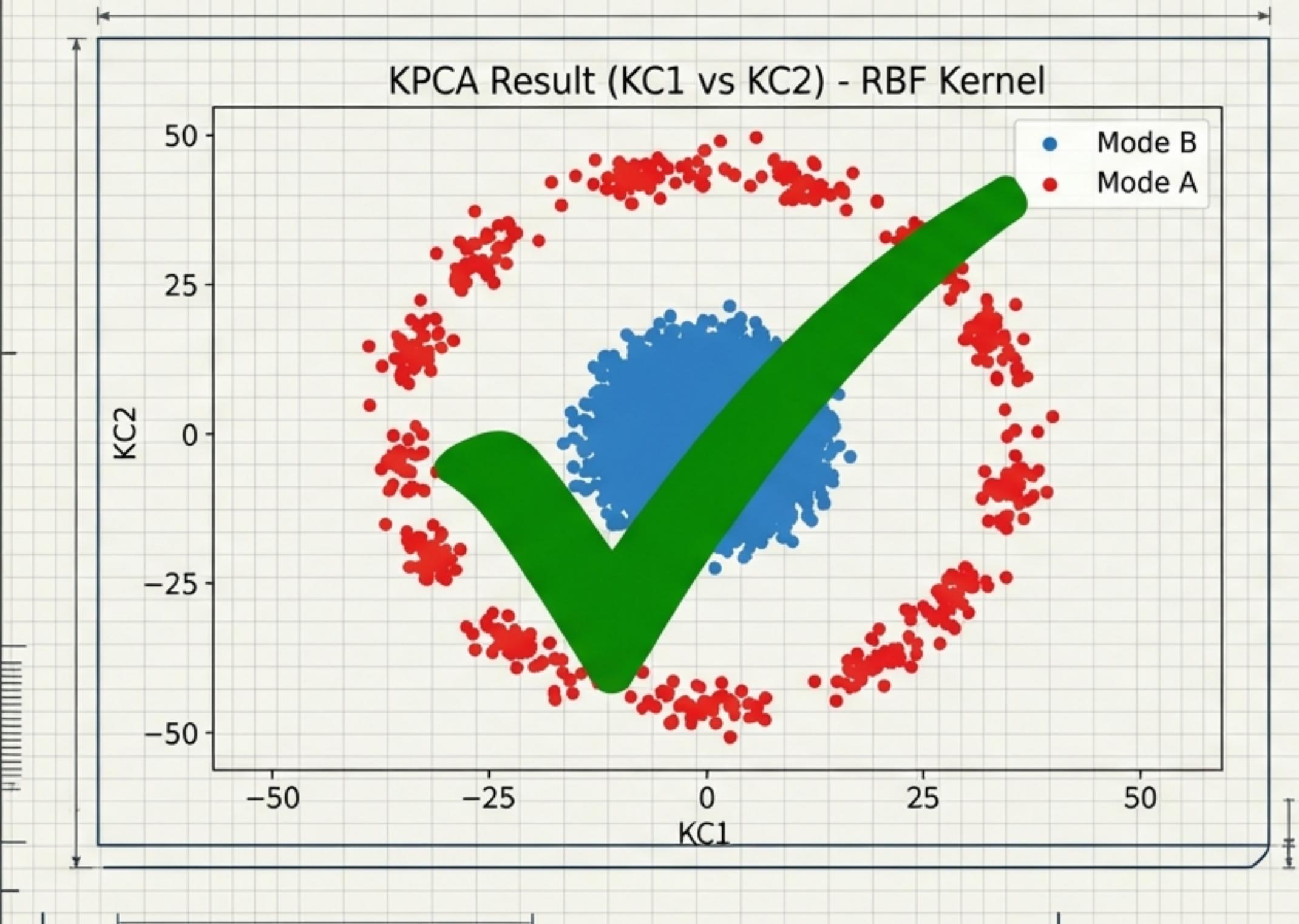


Explained Variance : 86 . 70%

Result : Failed Separation 

 警示：高解釋變異數 ≠ 良好的分類效果。PCA 捕捉了變異，但沒有解開糾纏的結構。

Kernel PCA 的突破 (The Solution)



Kernel: RBF (Gamma=0.1)
Result: Perfect Separation

RBF 核成功利用「徑向距離」特徵，將同心圓結構轉化為中心-周邊分佈，完全解開結構構。

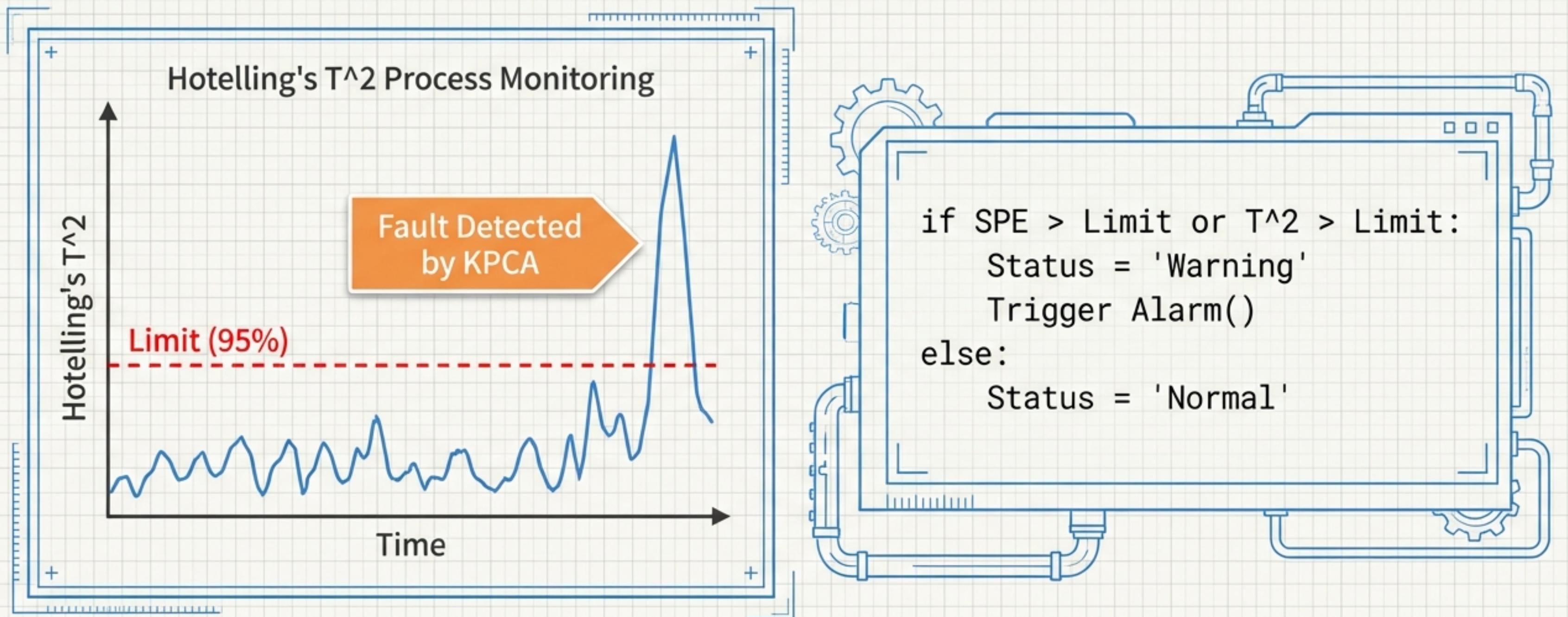


綜合比較：PCA vs. Kernel PCA

| 特性 | 傳統 PCA | Kernel PCA |
|------|--------------------|-------------------|
| 結構捕捉 | 線性 (Linear) | 非線性 (Non-linear) |
| 可解釋性 | 高 (Loadings 有物理意義) | 低 (黑盒子特徵) |
| 計算成本 | 低 | 高 (隨樣本數 n^2 增加) |
| 重建能力 | 直接重建 | 困難 (Pre-image 問題) |
| 適用場景 | 穩態、簡單相關性 | 複雜反應、故障偵測 |

Bottom Line: 簡單問題用 PCA，複雜結構用 Kernel PCA。

應用場景：非線性製程監控



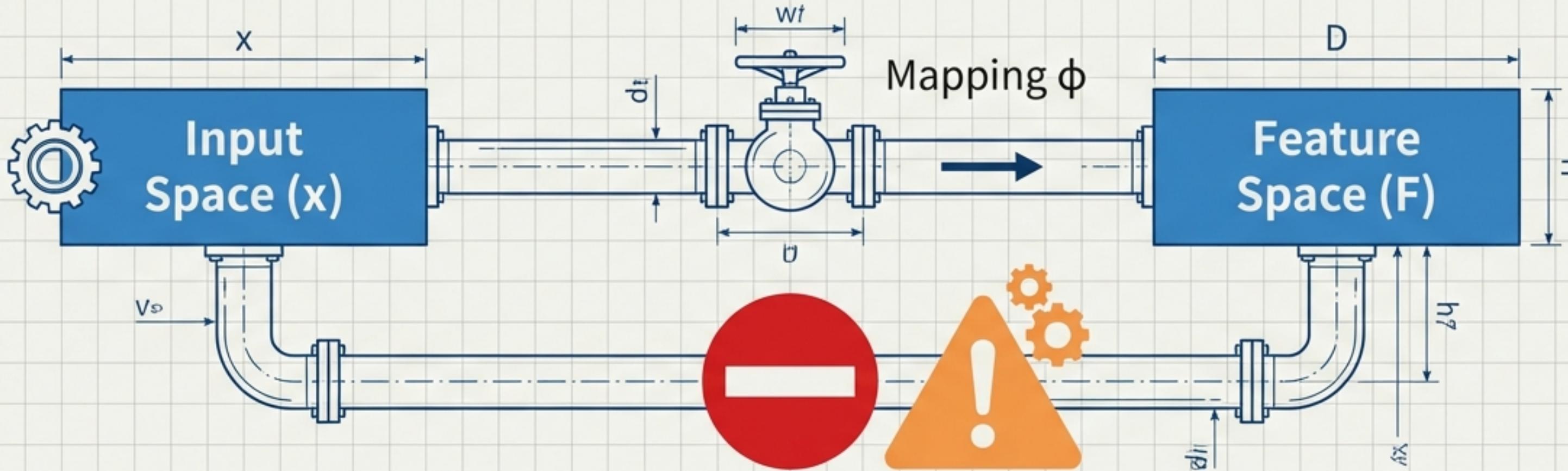
實務操作清單 (Best Practices)

CHECKLIST

- 1. 數據標準化 (Scale)：必須做 Z-score，確保距離計算正確。
- 2. 核函數 (Kernel)：優先嘗試 RBF 核。
- 3. 調校 (Tune)：使用 Cross-Validation 尋找最佳 Gamma。
- 4. 評估 (Evaluate)：不要只看變異數，要看分群效果。
- 5. 樣本數 (Sample Size)：若 $N > 10,000$ ，需考慮記憶體限制。

Ready for Deployment

限制與挑戰：Pre-image 問題



Exception: Inverse Mapping Hard

- 適合：故障檢測 (Detection)、分群 (Clustering)
- 困難：去噪重建 (Reconstruction)、Loadings 解釋

總結：您的非線性工具箱

核心公式 (The Core)

$$f(x) = \left(\sum_{i=1}^m \frac{a_i \mathbf{x}^T \mathbf{v}_i}{2n} \right)^2$$
$$= (e^{-\|\mathbf{x}\|^2/2})$$

Kernel PCA = PCA + Kernel Trick。解決化工製程的非線性難題。

權衡 (Trade-off)



犧牲了可解釋性與速度，換取了強大的結構洞察力。

下一步 (Next Step)



開啟 Python 練習 scikit-learn 的 KernelPCA 模組。

Unit 06 結束 | 前往 Unit 07