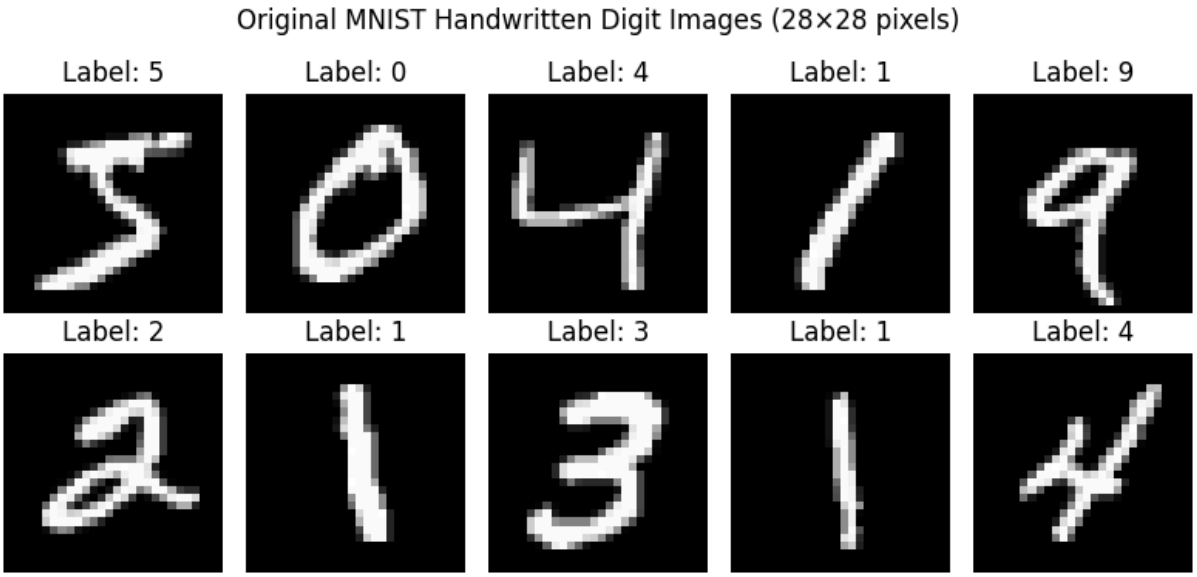
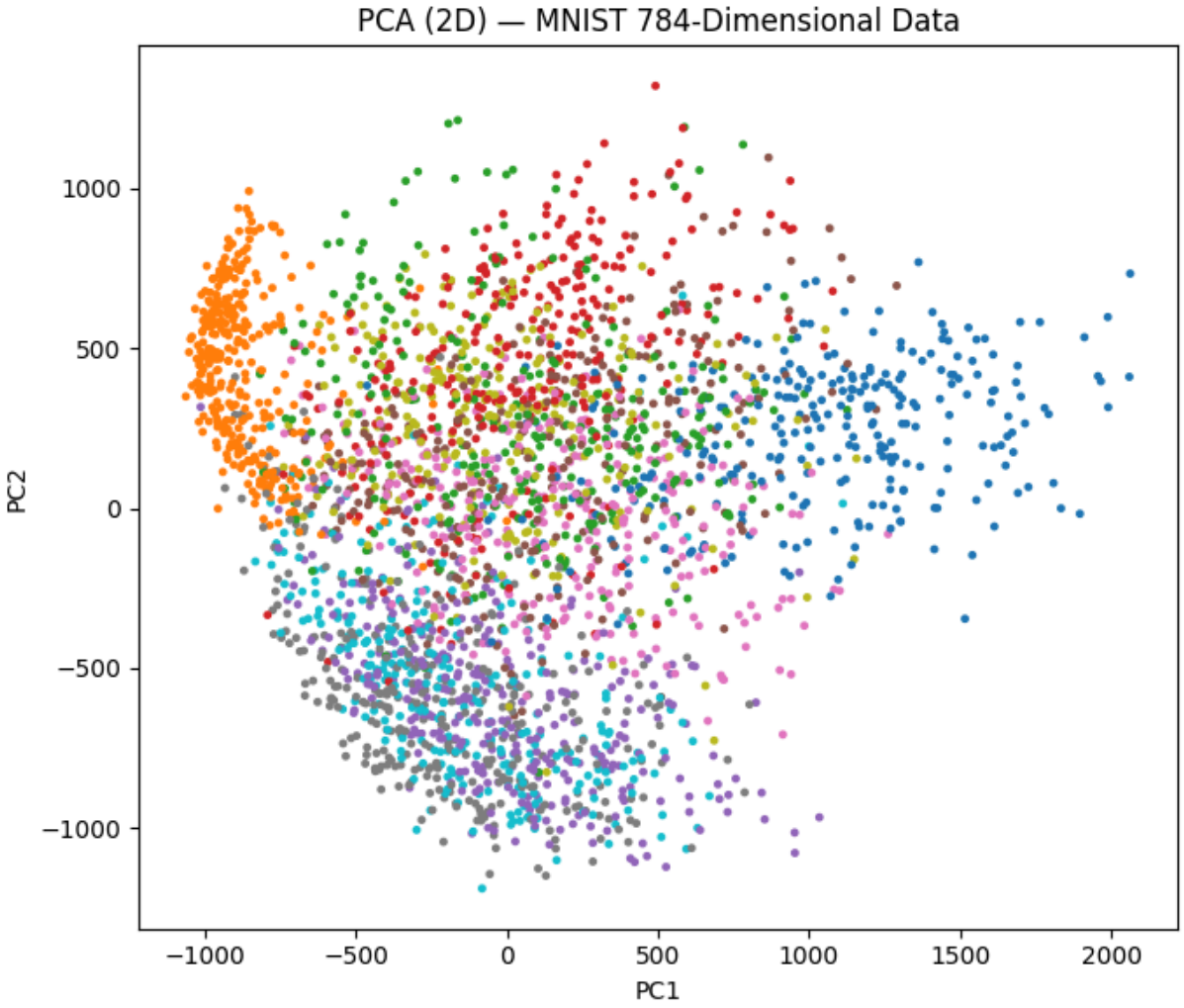
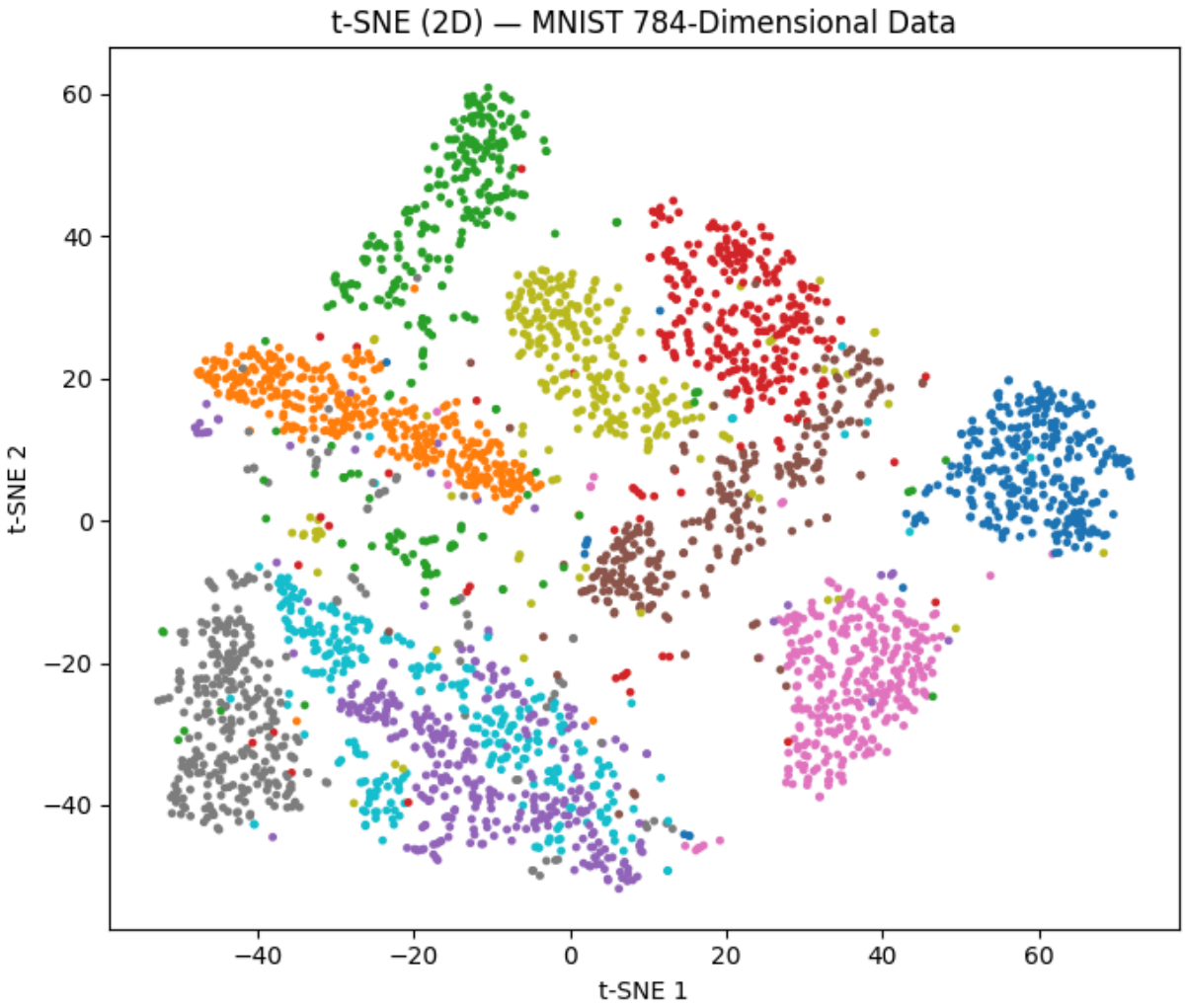
In [37]:

```
import warnings
warnings.filterwarnings("ignore")

import time
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE

# 1. Load MNIST (70,000 samples, 784 features)
X, y = fetch_openml('mnist_784', return_X_y=True, as_frame=False)
y = y.astype(int)

# Optional: take a subset (for faster demo)
X, y = X[:3000], y[:3000]

# 2. Show a few original digit images (28x28)
plt.figure(figsize=(8, 4))
for i in range(10):
    plt.subplot(2, 5, i + 1)
    plt.imshow(X[i].reshape(28, 28), cmap="gray")
    plt.title(f"Label: {y[i]}")
    plt.axis("off")
plt.suptitle("Original MNIST Handwritten Digit Images (28×28 pixels)", fonts
plt.tight_layout()
plt.savefig("MNIST_OriginalImages.png", dpi=300)
plt.show()

# 3. PCA: linear projection to 2D (time measurement)
start_pca = time.time()
pca = PCA(n_components=2, random_state=42)
X_pca = pca.fit_transform(X)
end_pca = time.time()
pca_time = end_pca - start_pca

# 4. t-SNE: nonlinear projection to 2D (time measurement)
start_tsne = time.time()
tsne = TSNE(
    n_components=2,
    perplexity=40,
    learning_rate=200,
    max_iter=1000,
    init="pca",
    random_state=42
)
X_tsne = tsne.fit_transform(X)
end_tsne = time.time()
tsne_time = end_tsne - start_tsne

# 5. Plot PCA
plt.figure(figsize=(7,6))
plt.scatter(X_pca[:,0], X_pca[:,1], c=y, cmap="tab10", s=6)
plt.title("PCA (2D) – MNIST 784-Dimensional Data")
plt.xlabel("PC1"); plt.ylabel("PC2")
```

```
plt.tight_layout()
plt.savefig("PCA_MNIST.png", dpi=300)
plt.show()

# 6. Plot t-SNE
plt.figure(figsize=(7,6))
plt.scatter(X_tsne[:,0], X_tsne[:,1], c=y, cmap="tab10", s=6)
plt.title("t-SNE (2D) - MNIST 784-Dimensional Data")
plt.xlabel("t-SNE 1"); plt.ylabel("t-SNE 2")
plt.tight_layout()
plt.savefig("tSNE_MNIST.png", dpi=300)
plt.show()

# 7. Print time comparison
print("Time Comparison:")
print(f"   PCA  time: {pca_time:.2f} seconds")
print(f"   t-SNE time: {tsne_time:.2f} seconds")
print("\nFigures saved as:")
print(" - MNIST_OriginalImages.png (original digit samples)")
print(" - PCA_MNIST.png (linear projection)")
print(" - tSNE_MNIST.png (nonlinear embedding)")
```




PCA (2D) — MNIST 784-Dimensional Data
PC2
PC1
1000
500
0
−500
−1000
−1000
−500
0
500
1000
1500
2000




```
Time Comparison:
   PCA  time: 0.03 seconds
   t-SNE time: 4.97 seconds

Figures saved as:
 - MNIST_OriginalImages.png (original digit samples)
 - PCA_MNIST.png (linear projection)
 - tSNE_MNIST.png (nonlinear embedding)
```

In [30]:

```python
import pandas as pd
from IPython.display import display

# Define data
data = {
    "Concept": [
        "Input space",
        "Mapping",
        "Focus",
        "Output",
        "Handles high-D curvature",
        "Speed",
        "Use case"
    ],
    "PCA": [
        "784-D (pixels)",
```

```
        "Linear",
        "Global variance",
        "Overlapping clouds",
        "No",
        "Fast",
        "Compression, preprocessing"
    ],
    "t-SNE": [
        "784-D (pixels)",
        "Nonlinear",
        "Local neighborhoods",
        "10 distinct clusters",
        "Yes",
        "Slower (O(n²))",
        "Visualization, manifold discovery"
    ]
}

# Create DataFrame and display
df = pd.DataFrame(data)
display(df)
```

	Concept	PCA	t-SNE
0	Input space	784-D (pixels)	784-D (pixels)
1	Mapping	Linear	Nonlinear
2	Focus	Global variance	Local neighborhoods
3	Output	Overlapping clouds	10 distinct clusters
4	Handles high-D curvature	No	Yes
5	Speed	Fast	Slower (O(n²))
6	Use case	Compression, preprocessing	Visualization, manifold discovery

In [ ]: