Untitled8

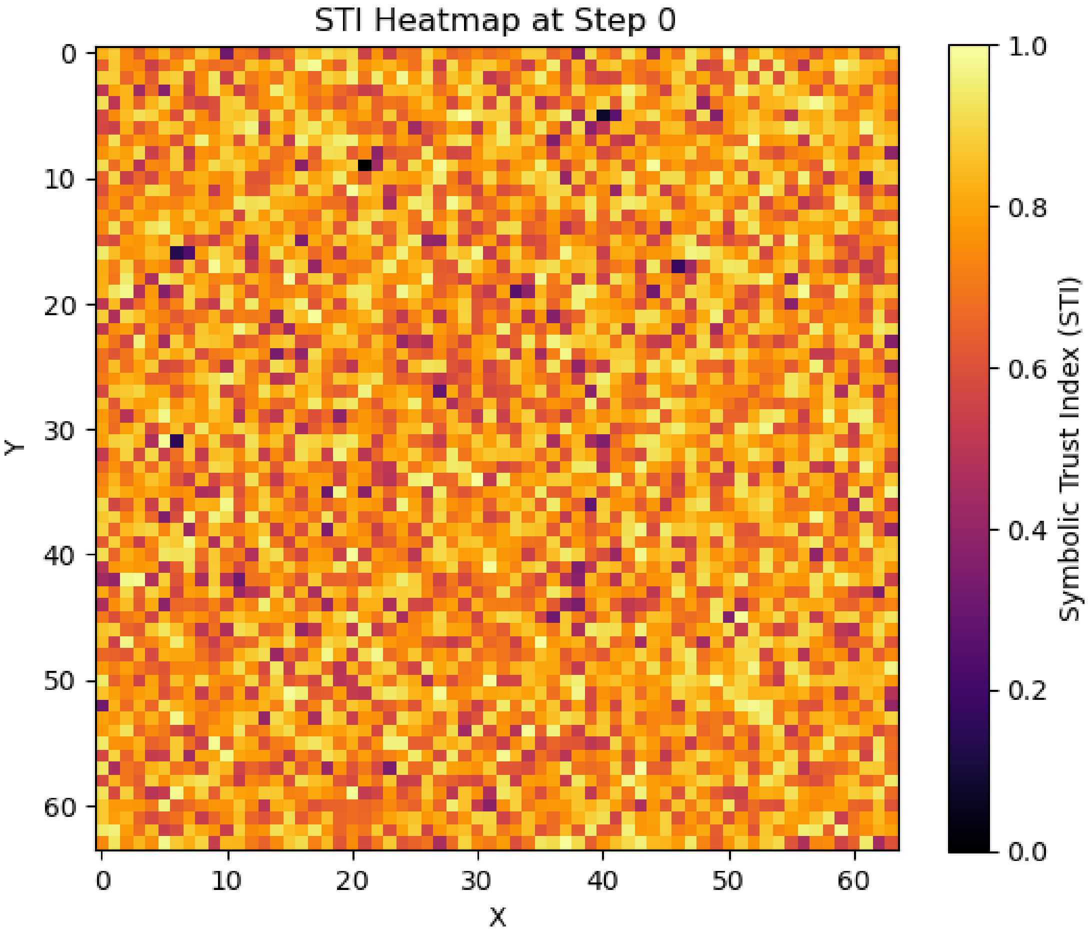
June 24, 2025

[1]:

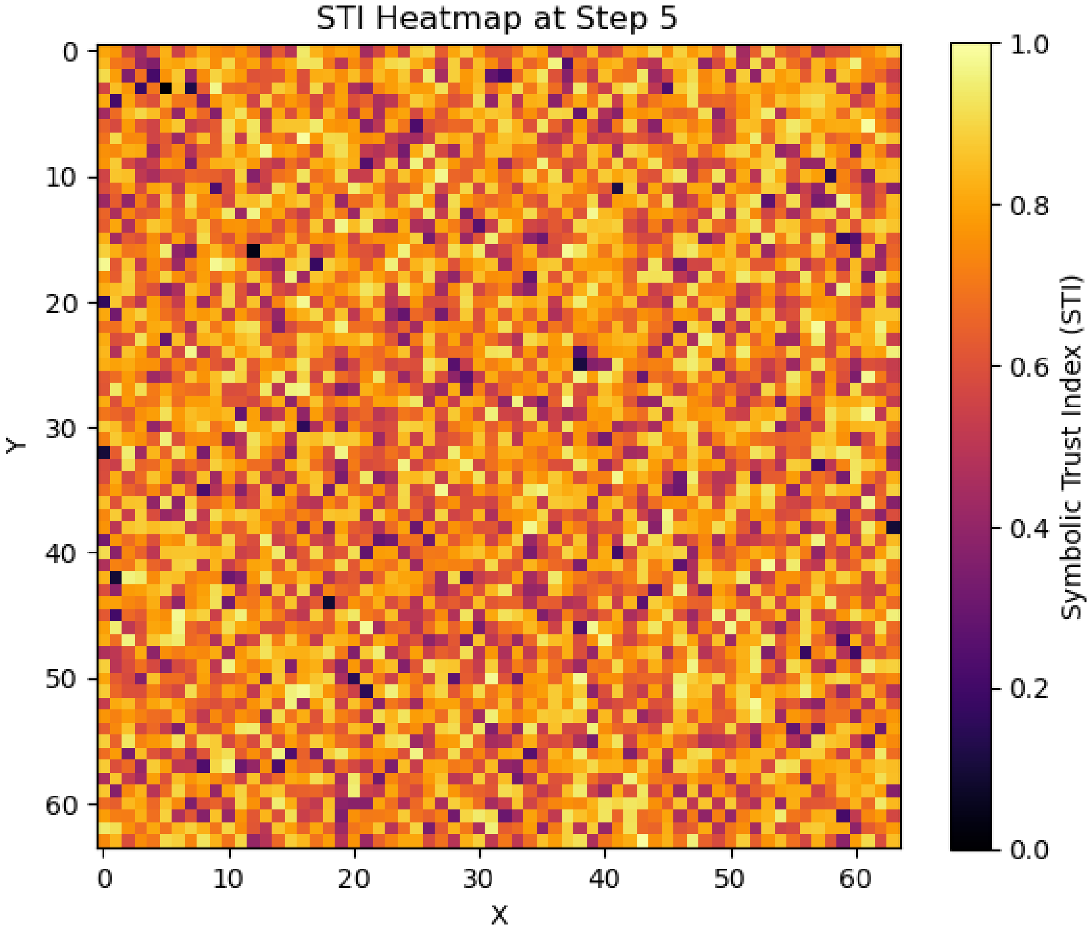
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.ndimage import uniform_filter

# Parameters
grid_size = 64              # Spatial grid resolution
time_steps = 100            # Number of time steps to simulate
delta_t = 0.01              # Time increment
viscosity = 0.1             # Fluid viscosity
jacobi_iterations = 20      # Pressure Poisson solver iterations
stability_threshold = 0.7 # STI threshold for instability
base_alpha = 0.35           # Base harmonic fold gain

# Initialize symbolic genome velocity field (Psi): shape (grid_size, grid_size,␣
 ↪2)
Psi = np.random.randn(grid_size, grid_size, 2) * 0.1

# Previous delta for STI calculation
prev_delta = np.zeros_like(Psi)

def ddx(f):
    return (np.roll(f, -1, axis=1) - np.roll(f, 1, axis=1)) / 2

def ddy(f):
    return (np.roll(f, -1, axis=0) - np.roll(f, 1, axis=0)) / 2

def navier_stokes_update_full(Psi, delta_t, viscosity=0.1, iterations=20):
    u = Psi[..., 0]
    v = Psi[..., 1]

    # Nonlinear advection
    u_x = ddx(u)
    u_y = ddy(u)
    v_x = ddx(v)
    v_y = ddy(v)
    adv_u = u * u_x + v * u_y
    adv_v = u * v_x + v * v_y
```

```python
    # Diffusion (viscous term)
    laplace_u = (np.roll(u, 1, axis=0) + np.roll(u, -1, axis=0) +
                 np.roll(u, 1, axis=1) + np.roll(u, -1, axis=1) - 4 * u)
    laplace_v = (np.roll(v, 1, axis=0) + np.roll(v, -1, axis=0) +
                 np.roll(v, 1, axis=1) + np.roll(v, -1, axis=1) - 4 * v)

    u_new = u + delta_t * (viscosity * laplace_u - adv_u)
    v_new = v + delta_t * (viscosity * laplace_v - adv_v)

    # Pressure projection to enforce incompressibility
    div = ddx(u_new) + ddy(v_new)
    p = np.zeros_like(u_new)
    for _ in range(iterations):
        p = (np.roll(p, 1, axis=0) + np.roll(p, -1, axis=0) +
             np.roll(p, 1, axis=1) + np.roll(p, -1, axis=1) - div) / 4

    u_proj = u_new - delta_t * ddx(p)
    v_proj = v_new - delta_t * ddy(p)

    return np.stack([u_proj, v_proj], axis=-1)

def symbolic_trust_index(delta, prev_delta):
    drift = np.linalg.norm(delta - prev_delta, axis=2)
    max_drift = np.max(drift) if np.max(drift) > 0 else 1.0
    sti = 1 - drift / max_drift
    return sti

def plot_sti_heatmap(sti, step):
    plt.figure(figsize=(6,5))
    plt.imshow(sti, cmap='inferno', vmin=0, vmax=1)
    plt.colorbar(label='Symbolic Trust Index (STI)')
    plt.title(f'STI Heatmap at Step {step}')
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.tight_layout()
    plt.show()

def multi_scale_fold(Psi, delta, sti, base_alpha=0.35):
    from scipy.ndimage import uniform_filter
    scales = [1, 2, 4, 8]
    Psi_corrected = Psi.copy()
    for scale in scales:
        # Aggregate delta and sti at current scale
        delta_avg = uniform_filter(delta, size=scale, mode='reflect')[::scale, ::scale, :]
        sti_avg = uniform_filter(sti, size=scale, mode='reflect')[::scale, ::
 ↪scale]
        gain = base_alpha * (1 + 1.5 * (1 - sti_avg))
        gain = np.clip(gain, 0, 1)[..., None]
        correction = -gain * delta_avg
        # Broadcast correction back to full grid
        correction_full = np.repeat(np.repeat(correction, scale, axis=0),␣
 ↪scale, axis=1)
        Psi_corrected += correction_full[:Psi.shape[0], :Psi.shape[1], :]
    return Psi_corrected

# Main simulation loop
for t in range(time_steps):
    Psi_new = navier_stokes_update_full(Psi, delta_t, viscosity,␣
 ↪jacobi_iterations)
    delta = Psi_new - Psi
    sti = symbolic_trust_index(delta, prev_delta)

    unstable_mask = sti < stability_threshold
    delta_corrected = np.where(unstable_mask[..., None], delta, 0)
    Psi_new = multi_scale_fold(Psi_new, delta_corrected, sti, base_alpha)

    prev_delta = delta
    Psi = Psi_new

    if t % 5 == 0:
        avg_sti = np.mean(sti)
        print(f"Step {t}: Avg STI = {avg_sti:.4f}")
        plot_sti_heatmap(sti, t)
```

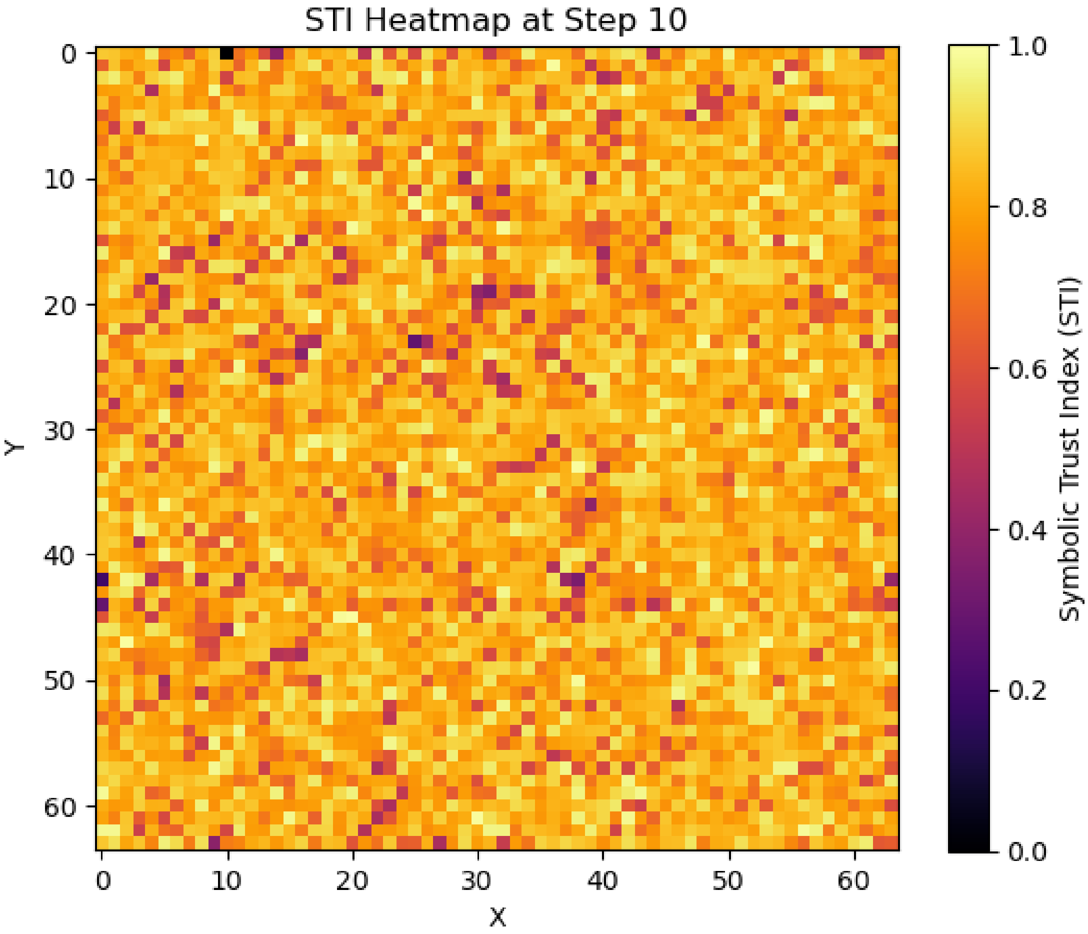
```
Step 0: Avg STI = 0.7499
```



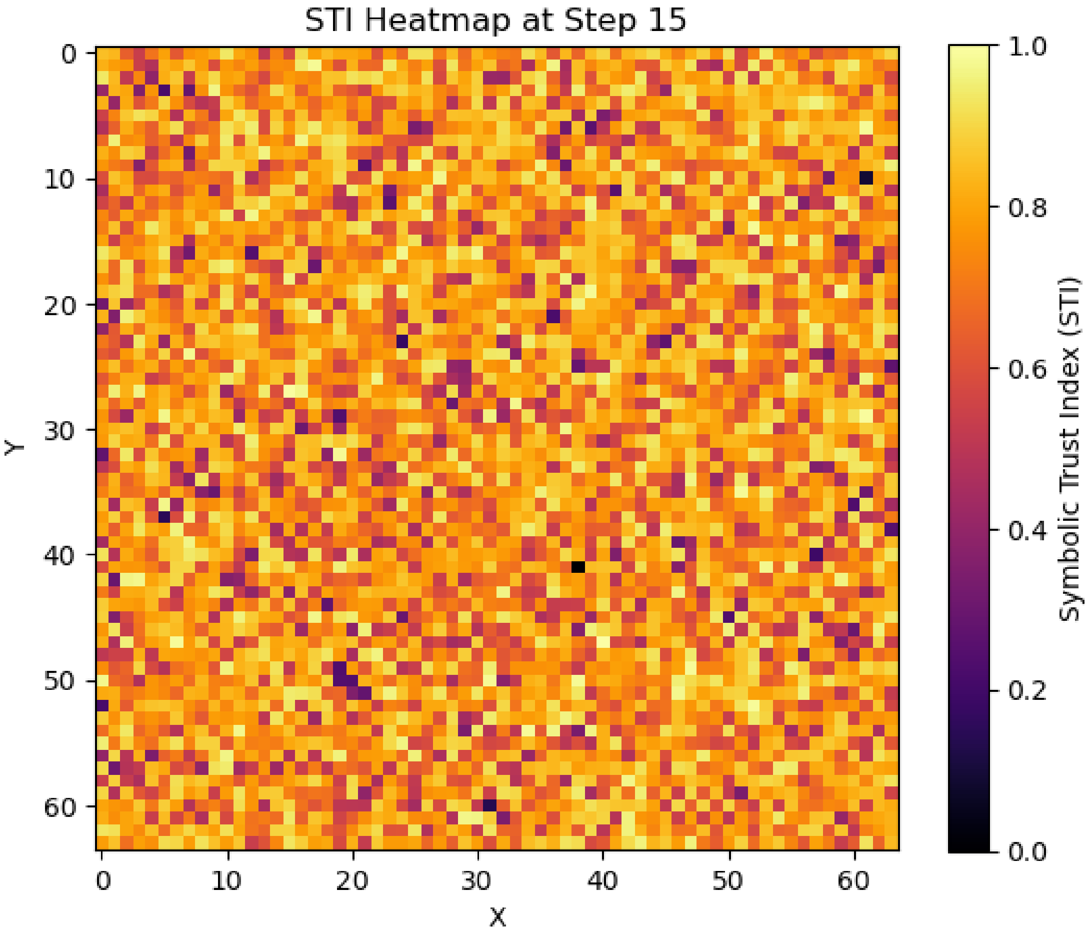
```
Step 5: Avg STI = 0.7022
```



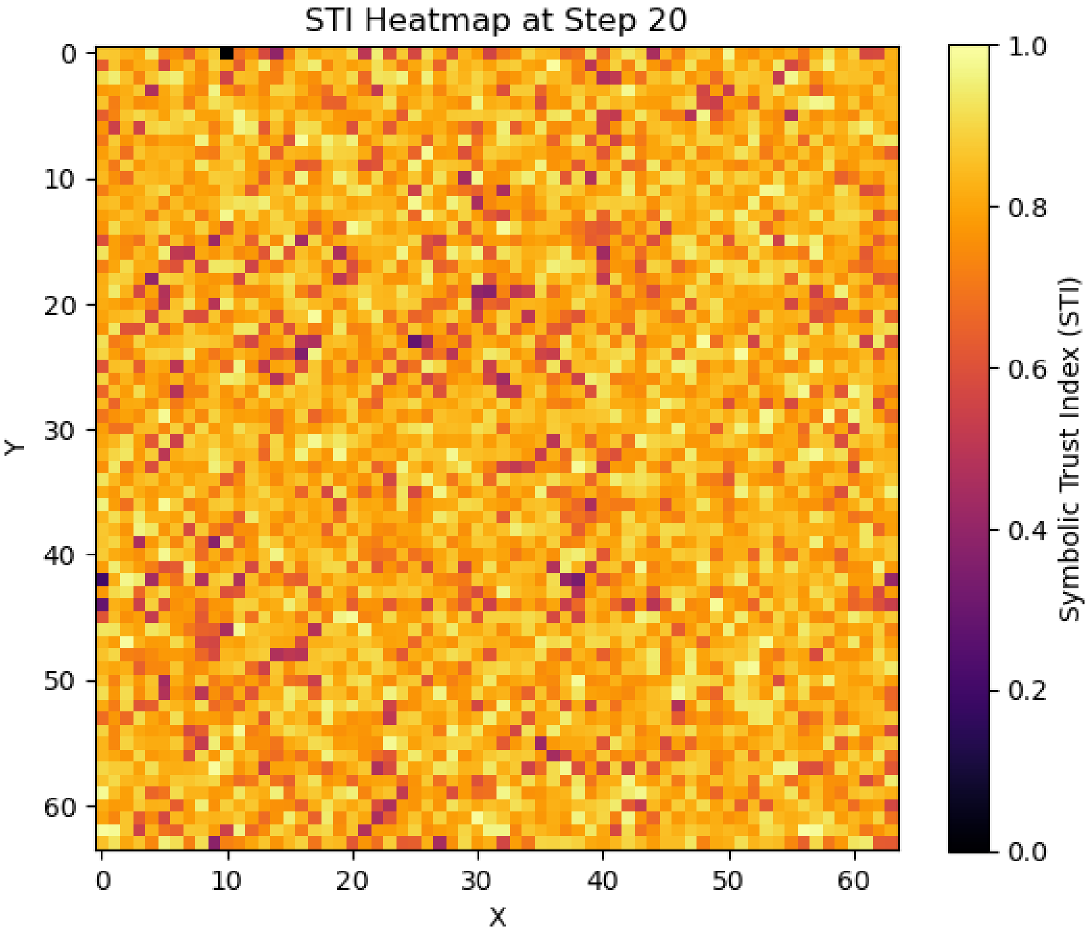
```
Step 10: Avg STI = 0.8014
```



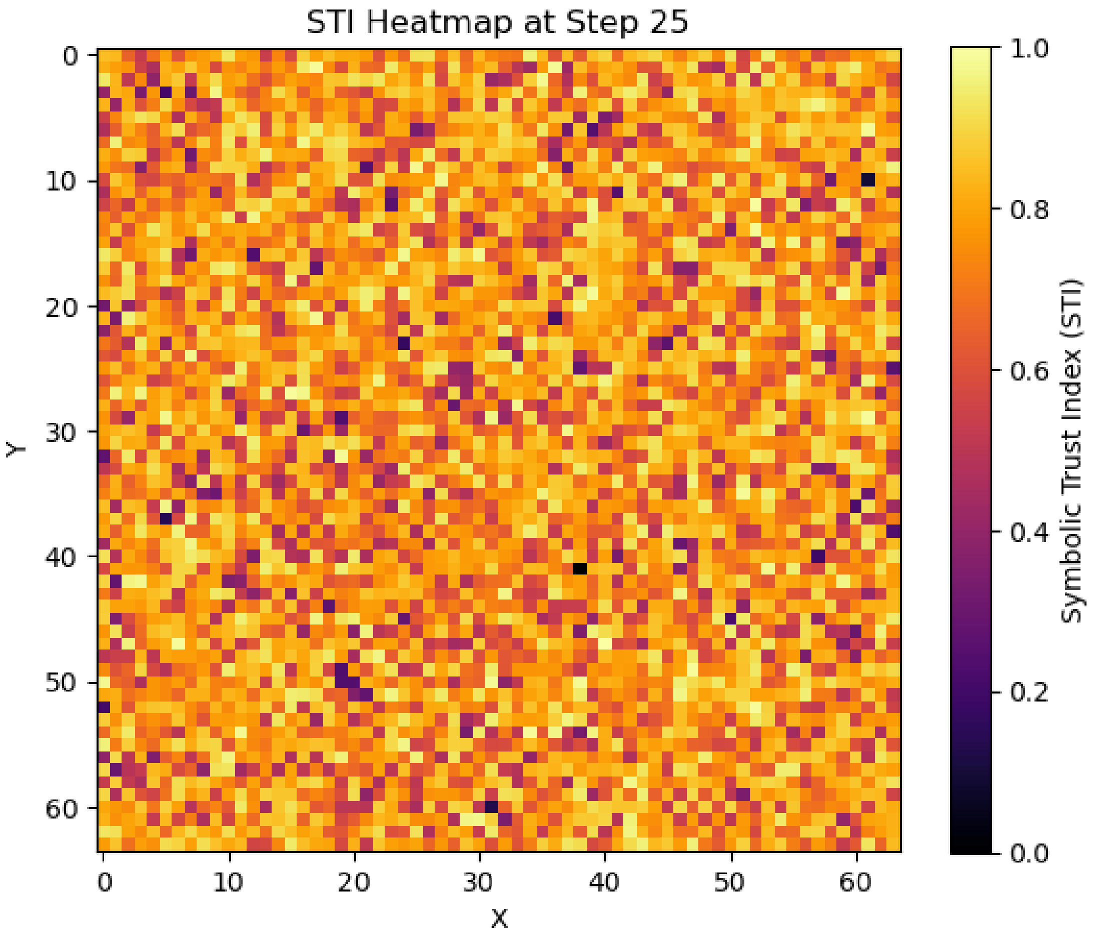
```
Step 15: Avg STI = 0.7417
```



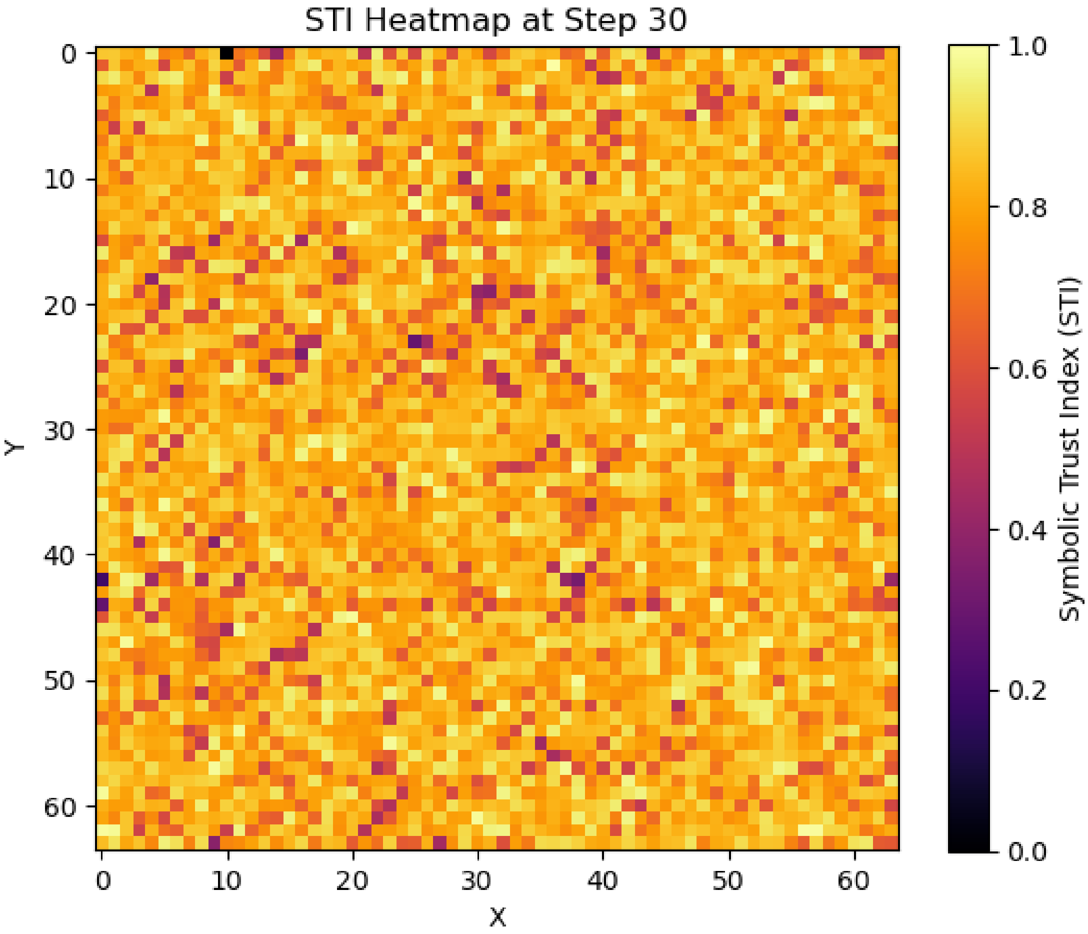
```
Step 20: Avg STI = 0.8006
```



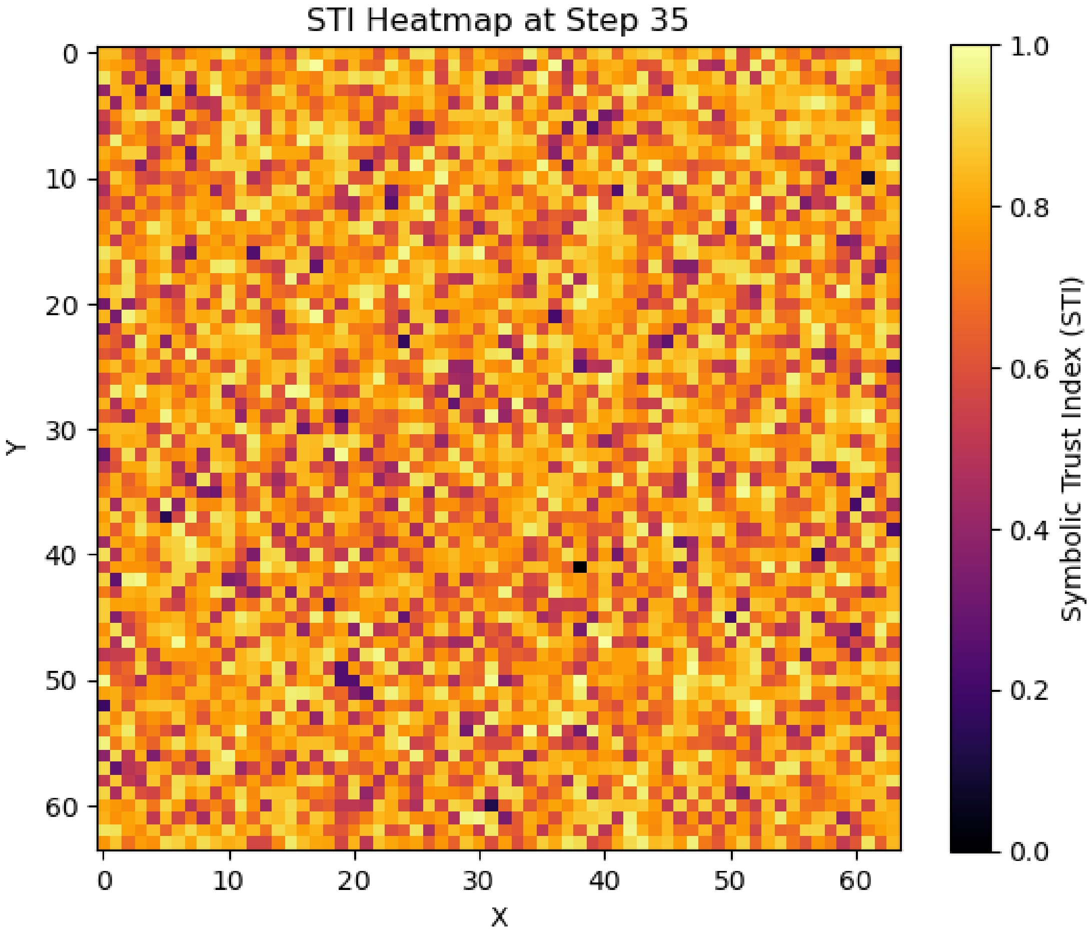
```
Step 25: Avg STI = 0.7410
```



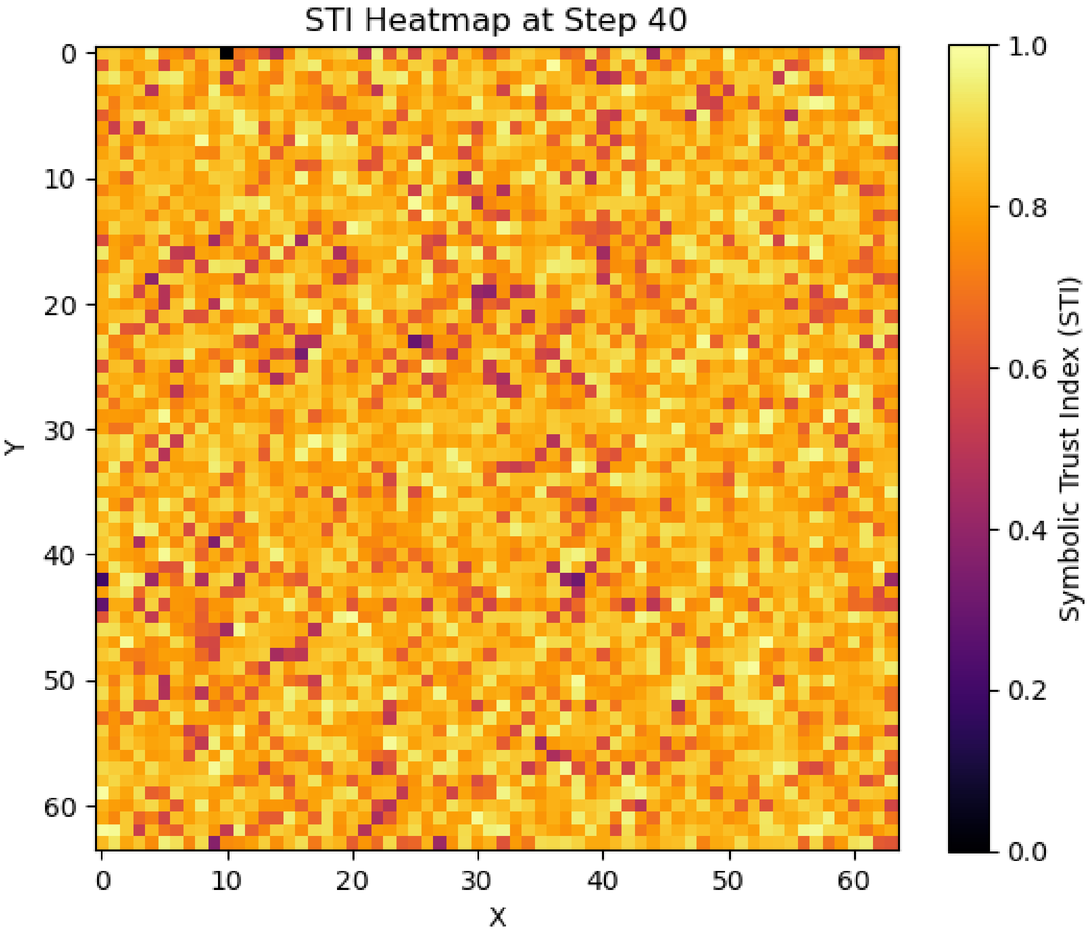
```
Step 30: Avg STI = 0.8024
```



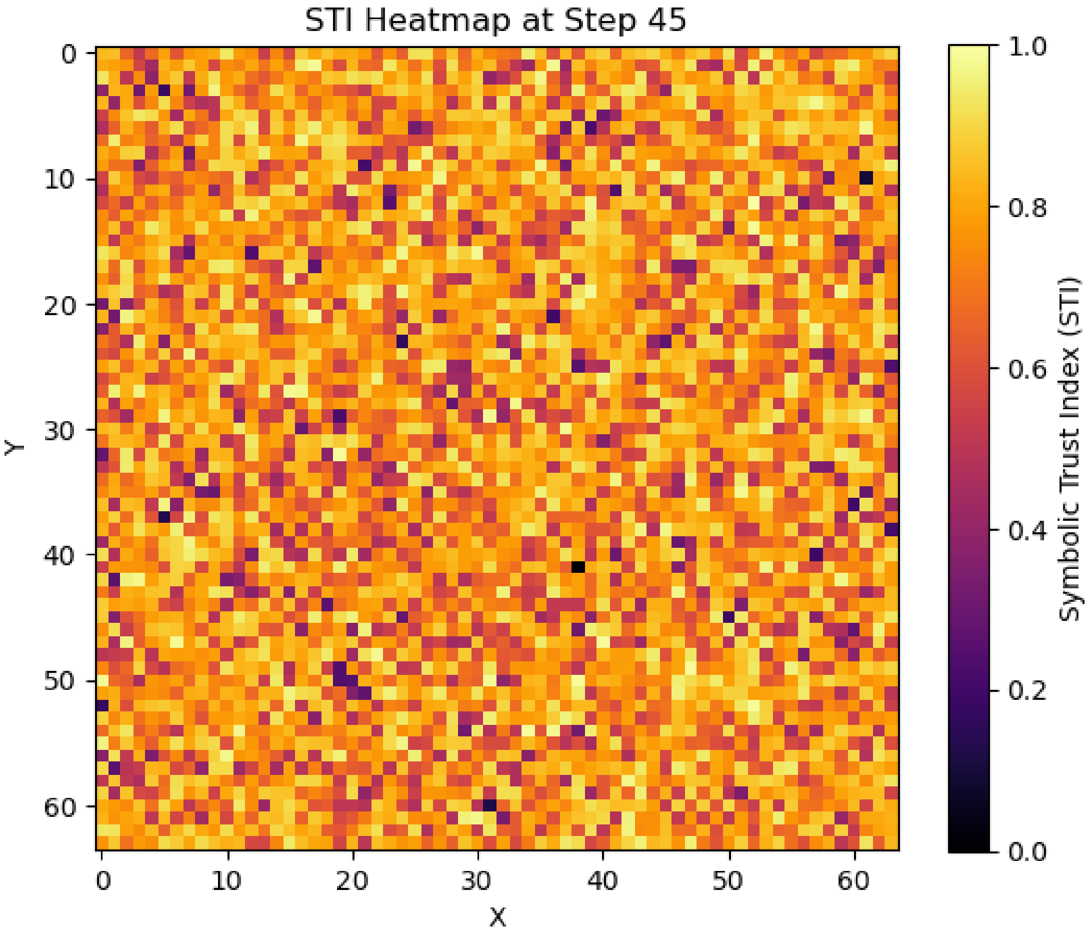
```
Step 35: Avg STI = 0.7396
```



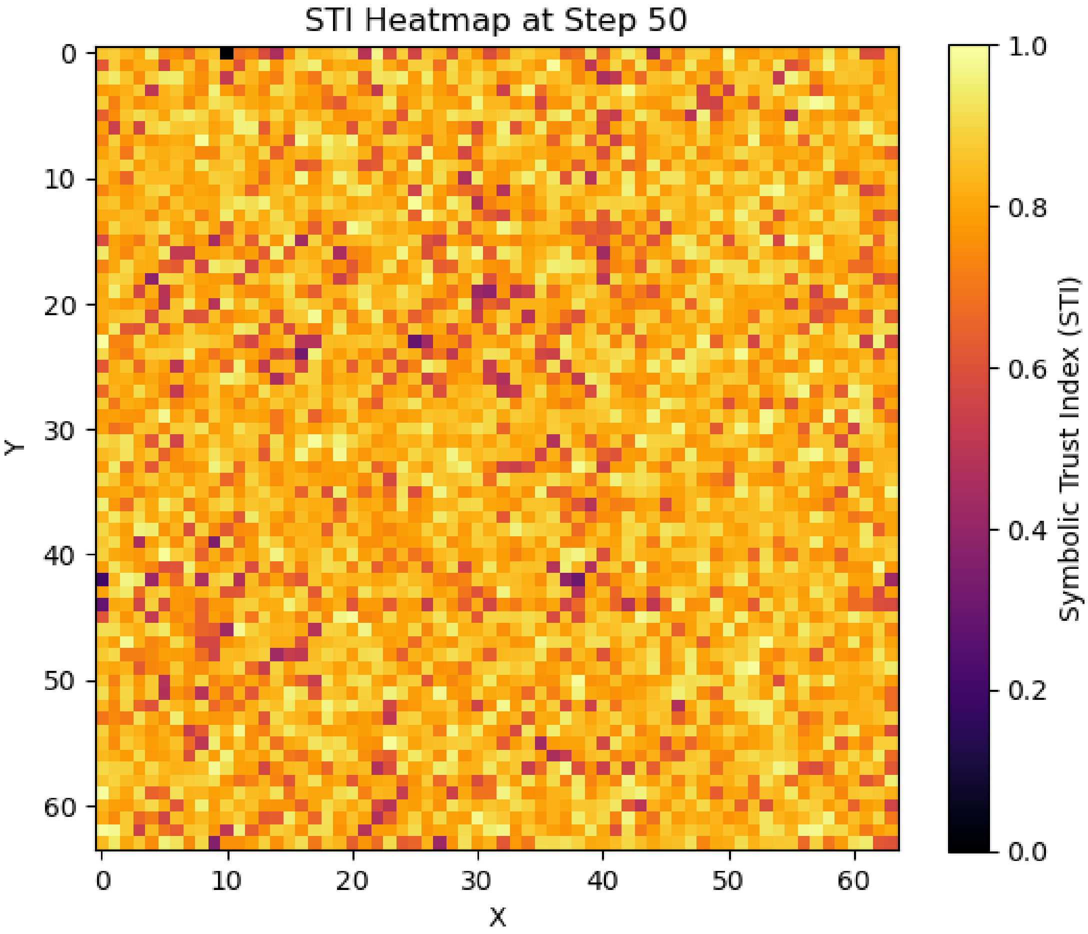
```
Step 40: Avg STI = 0.8039
```



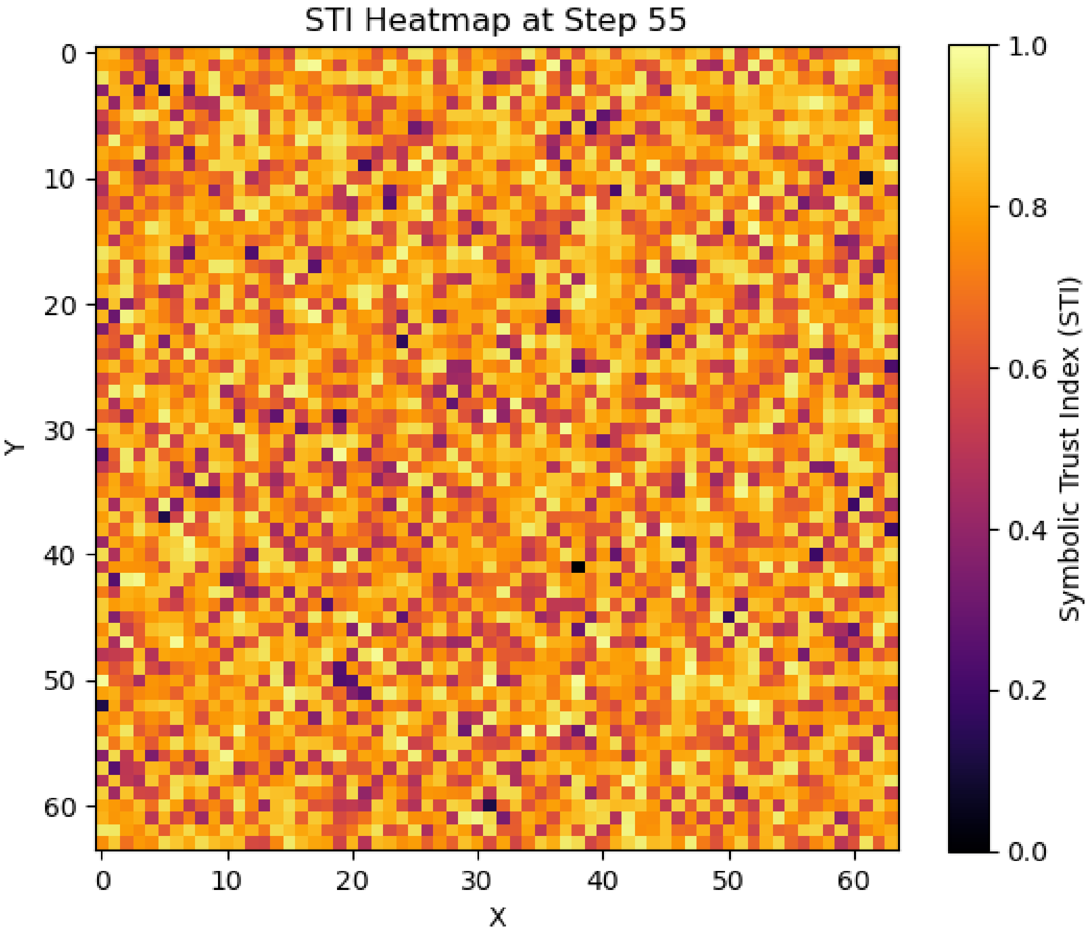
```
Step 45: Avg STI = 0.7383
```



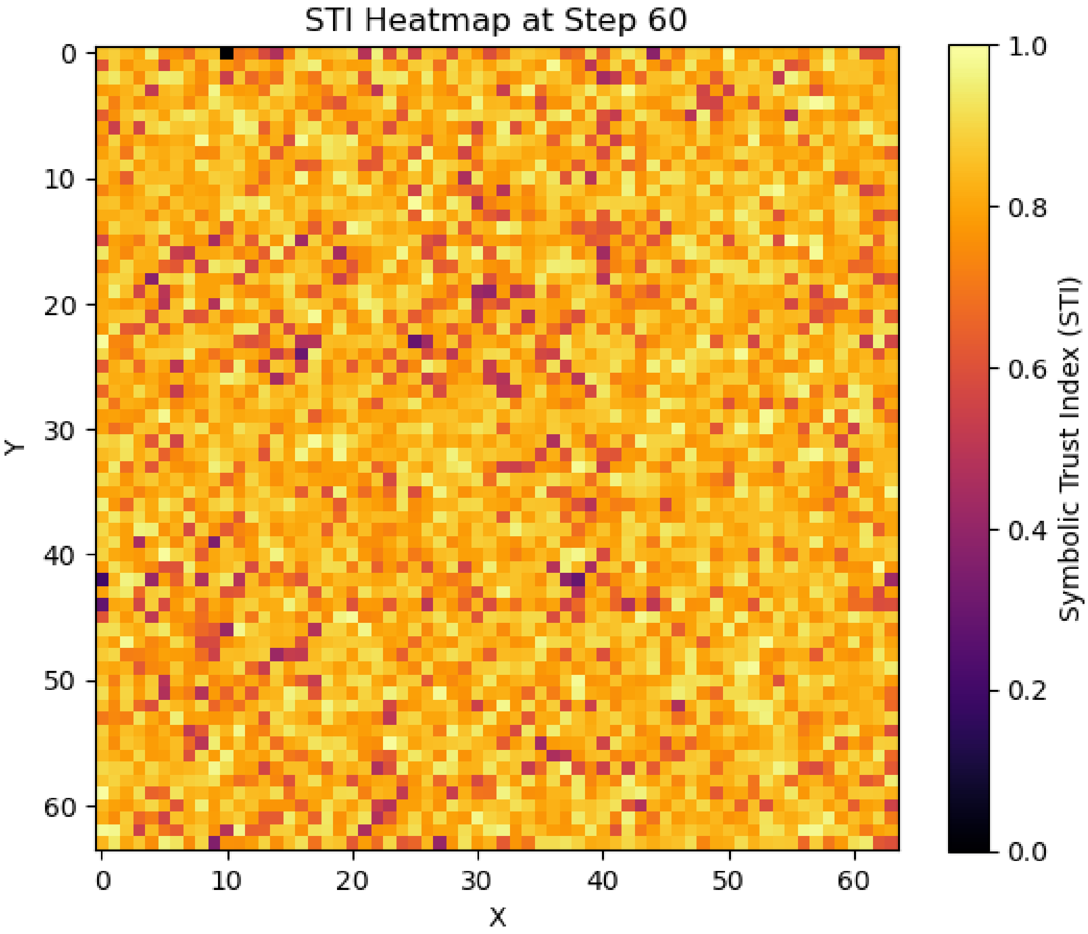
```
Step 50: Avg STI = 0.8057
```



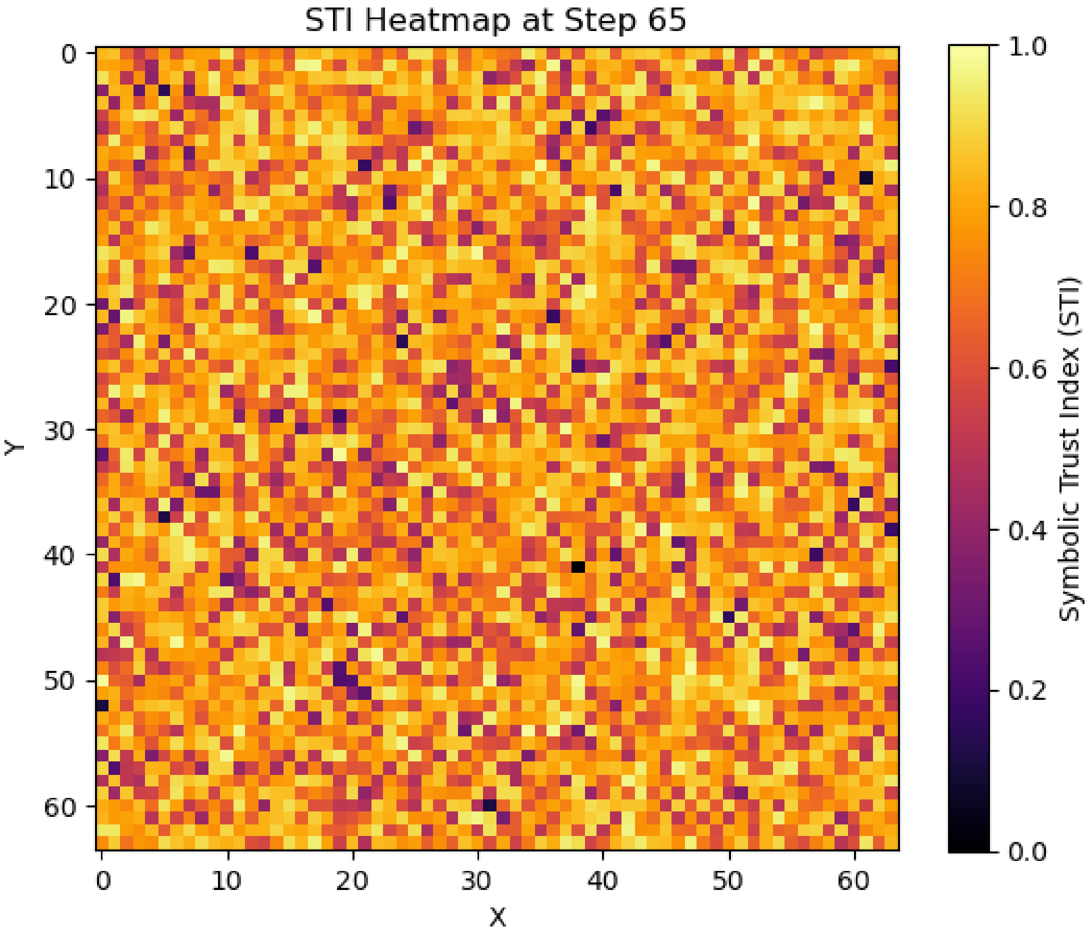
```
Step 55: Avg STI = 0.7366
```



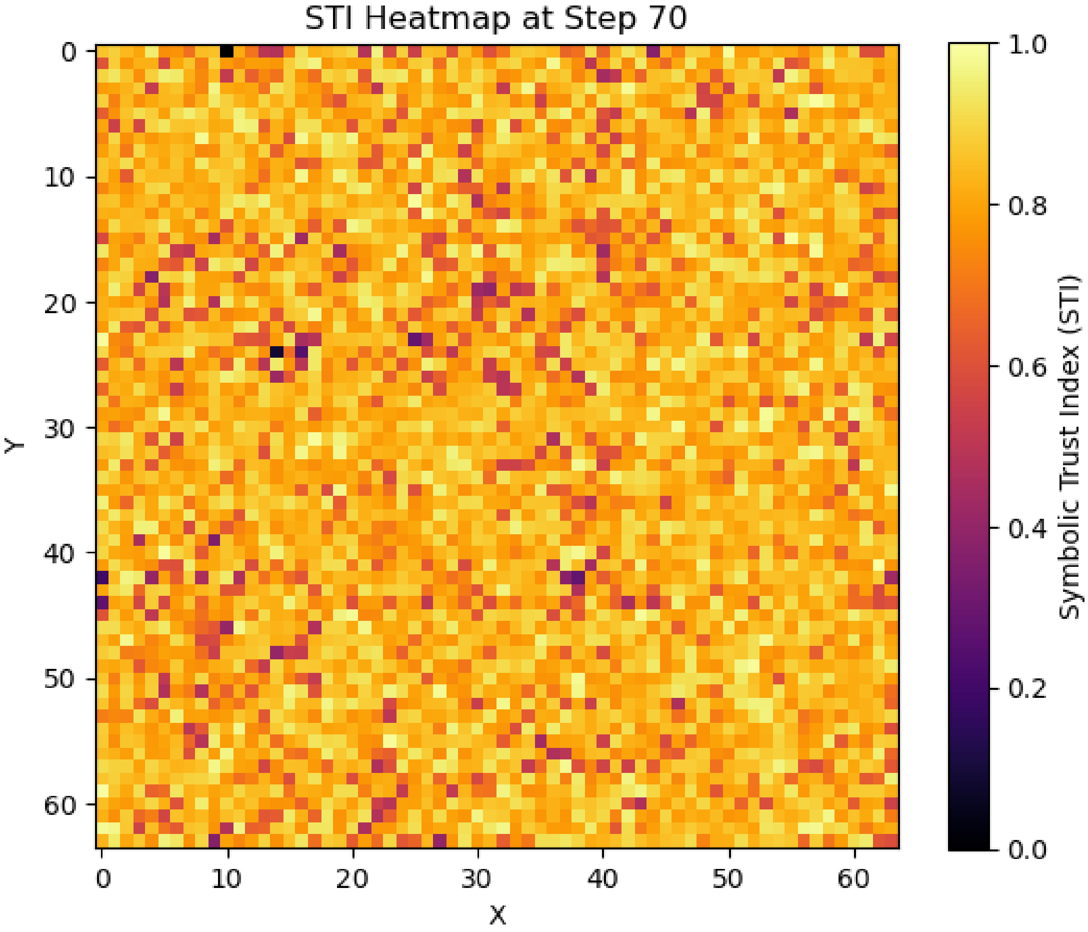
```
Step 60: Avg STI = 0.8070
```



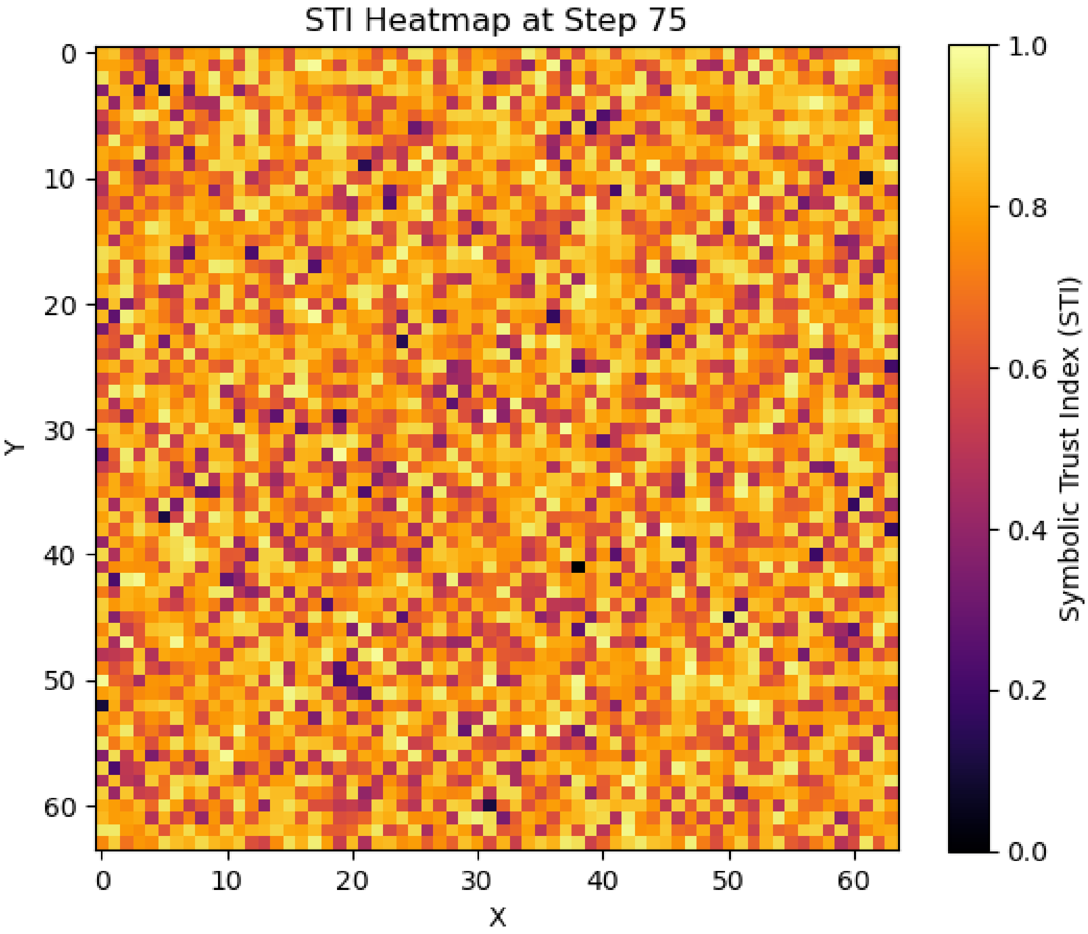
```
Step 65: Avg STI = 0.7359
```



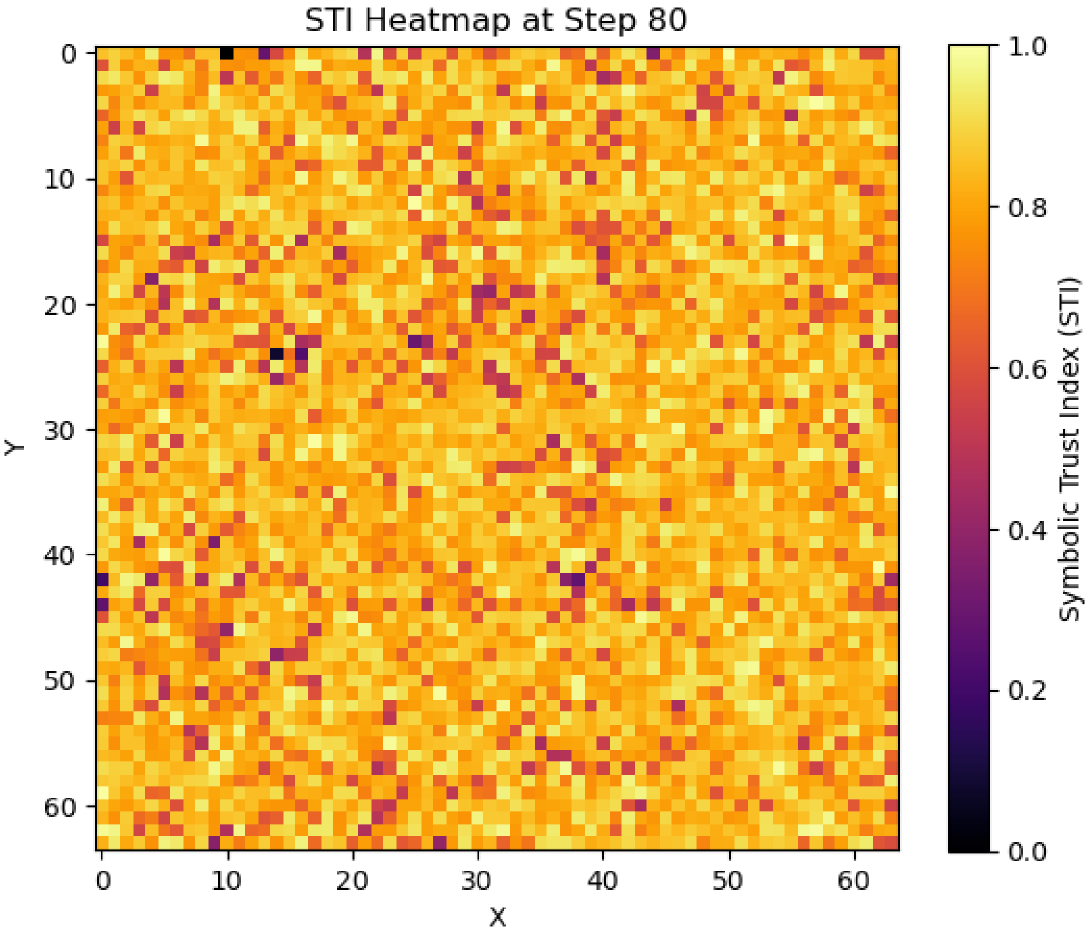
```
Step 70: Avg STI = 0.8075
```



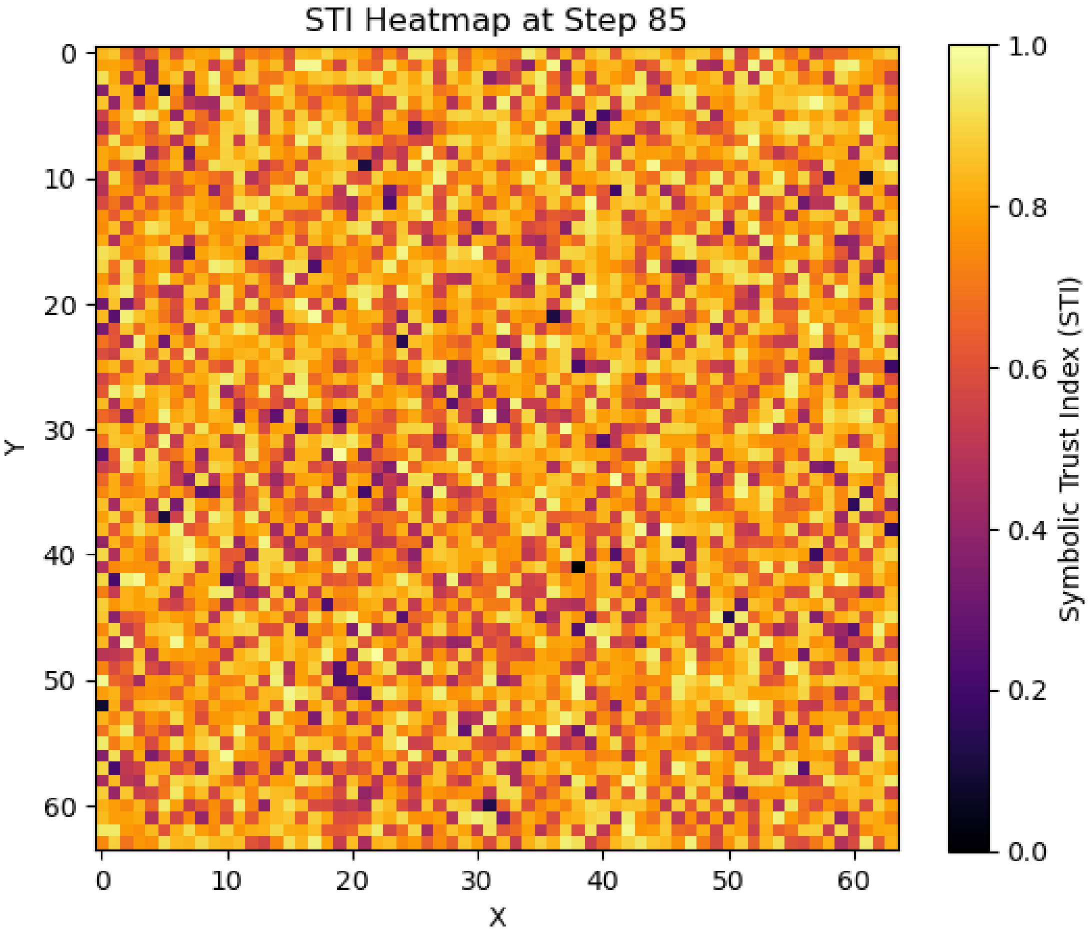
```
Step 75: Avg STI = 0.7357
```



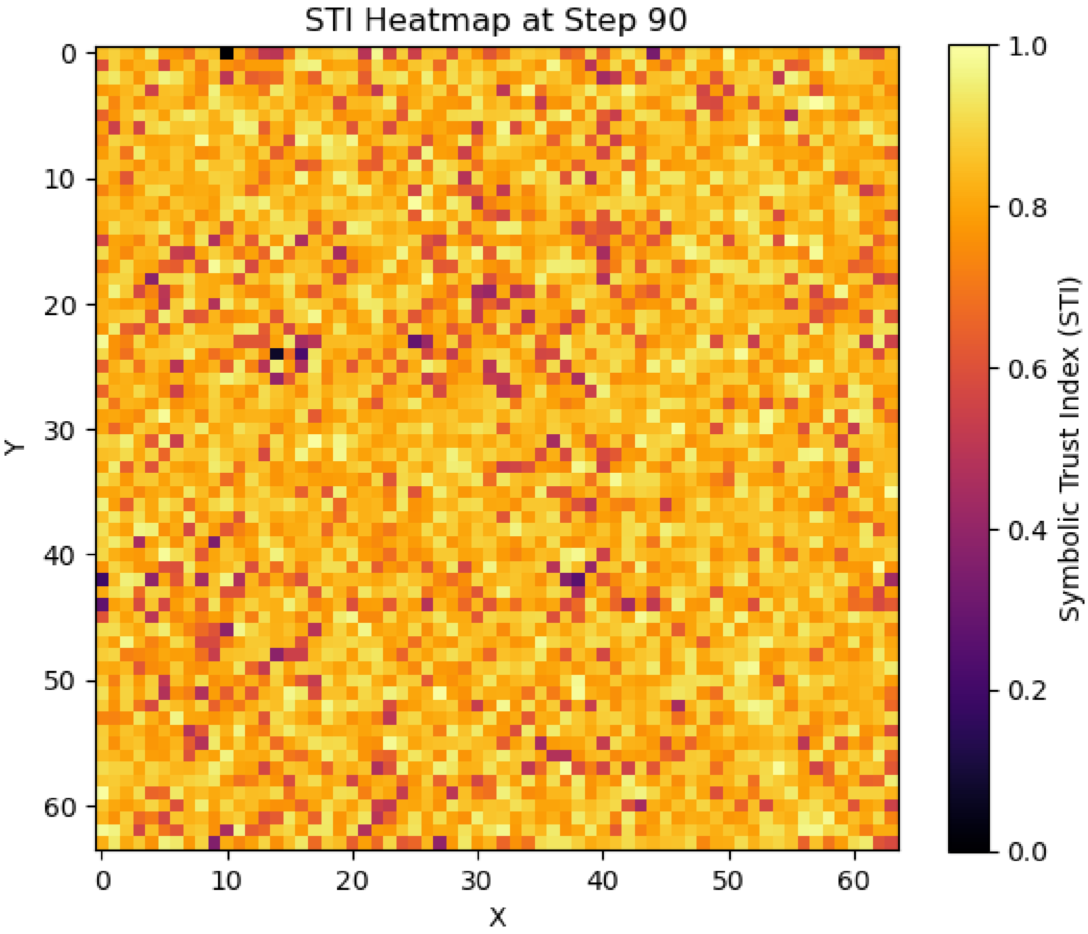
```
Step 80: Avg STI = 0.8097
```



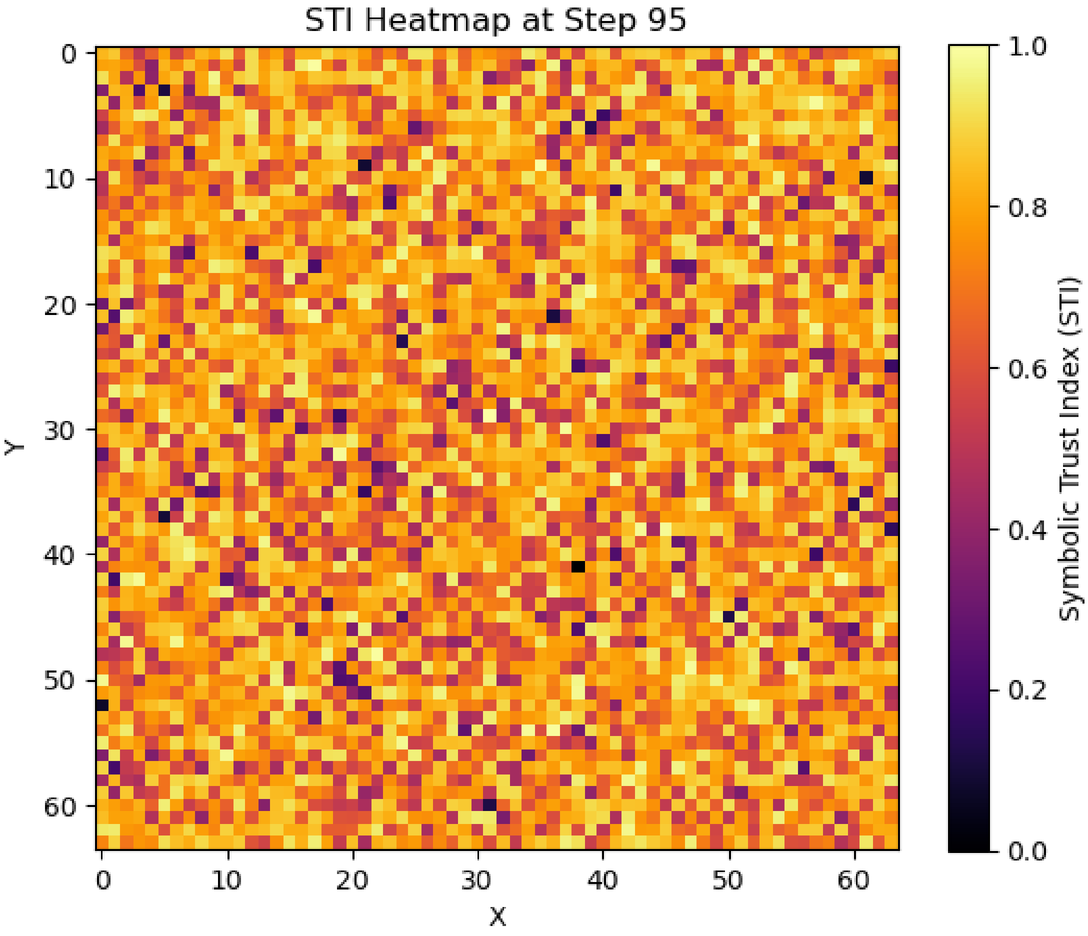
```
Step 85: Avg STI = 0.7345
```



```
Step 90: Avg STI = 0.8111
```



```
Step 95: Avg STI = 0.7343
```


STI Heatmap at Step 95
0
10
20
30
40
50
60
Y
0
10
20
30
40
50
60
X
1.0
0.8
0.6
0.4
0.2
0.0
Symbolic Trust Index (STI)

[ ]: