📌 HFCTM-II API Usage Guide 🚀

This guide explains **how to use the HFCTM-II API** with **example requests**, expected responses, and usage instructions.

📌1️⃣Base API URL

If running locally, your API is available at:

bash
CopyEdit
`http://localhost:8000`

If hosted on a server, replace `localhost` with your **server IP or domain**:

bash
CopyEdit
`http://YOUR_SERVER_IP:8000`

📌2️⃣API Documentation

Use **interactive documentation** to test API endpoints:

- **Swagger UI:** `http://localhost:8000/docs`
- **Redoc UI:** `http://localhost:8000/redoc`

📌3️⃣API Endpoints

📌 Run Recursive Inference

🔹 **Purpose:** Runs the HFCTM-II model for a given number of inference cycles.

🔹 **Endpoint:**

http
CopyEdit
```
POST /inference
```

🔹 **Request Body (JSON):**

json
CopyEdit
```
{
  "iterations": 10
}
```

🔹 **Example cURL Request:**

bash
CopyEdit
```
curl -X POST "http://localhost:8000/inference" -H "Content-Type:
application/json" -d '{"iterations": 10}'
```

🔹 **Response (JSON):**

json
CopyEdit
```
{
  "knowledge_state": [0.34, -0.21, 0.88, -0.42, 0.19, 0.55, -0.67,
0.72],
  "trust_matrix": [
    [1.0, 0.92, 0.88, 0.85, 0.80, 0.79, 0.78, 0.76],
    [0.92, 1.0, 0.91, 0.87, 0.82, 0.81, 0.79, 0.77],
    ...
  ]
}
```

✅ **Returns the updated recursive knowledge state and trust network**.

📌 Get Knowledge State

- 🔹 **Purpose:** Retrieves the **current recursive embeddings** from the model.
- 🔹 **Endpoint:**

http
CopyEdit
```
GET /state
```

- 🔹 **Example Request:**

bash
CopyEdit
```
curl -X GET "http://localhost:8000/state"
```

- 🔹 **Response (JSON):**

json
CopyEdit
```
{
  "knowledge_state": [0.35, -0.18, 0.91, -0.38, 0.21, 0.60, -0.62,
0.75]
}
```

✅ **Returns the latest recursive knowledge embeddings.**

📌 Get Trust Matrix

- 🔹 **Purpose:** Retrieves the **current AI trust dynamics**.
- 🔹 **Endpoint:**

http
CopyEdit
```
GET /trust
```

- 🔹 **Example Request:**

bash
CopyEdit

```
curl -X GET "http://localhost:8000/trust"
```

🔹 **Response (JSON):**

json
CopyEdit

```
{
  "trust_matrix": [
    [1.0, 0.92, 0.88, 0.85, 0.80, 0.79, 0.78, 0.76],
    [0.92, 1.0, 0.91, 0.87, 0.82, 0.81, 0.79, 0.77],
    ...
  ]
}
```

✅ **Returns the recursive trust matrix for AI cognition.**

📌 Detect Egregore Activity

🔹 **Purpose:** Runs **wavelet-based anomaly detection** to identify adversarial distortions.
🔹 **Endpoint:**

http
CopyEdit

```
GET /egregore-detection
```

🔹 **Example Request:**

bash
CopyEdit

```
curl -X GET "http://localhost:8000/egregore-detection"
```

🔹 **Response (JSON):**

json
CopyEdit

```
{
  "anomaly_score": 0.073,
  "egregore_detected": false
```

```
}
```

✅ **Returns an anomaly score and whether an egregore distortion is present.**

📌 Apply Chiral Inversion

- 🔹 **Purpose:** Manually **flip** the inference matrix RRR to counteract adversarial attractors.
- 🔹 **Endpoint:**

http
CopyEdit
```
POST /chiral-inversion
```

- 🔹 **Example Request:**

bash
CopyEdit
```
curl -X POST "http://localhost:8000/chiral-inversion"
```

- 🔹 **Response (JSON):**

json
CopyEdit
```
{
  "status": "Chiral inversion applied",
  "new_inference_matrix": [
    [-0.05, 0.11, -0.03, 0.08, -0.12, 0.09, -0.07, 0.10],
    [0.11, -0.04, 0.09, -0.07, 0.05, -0.03, 0.08, -0.02],
    ...
  ]
}
```

✅ **Reverses adversarial perturbations in AI cognition.**

📌 Reset the Model

- **Purpose:** Resets **all recursive embeddings, trust networks, and inference matrices**.
- **Endpoint:**

http
CopyEdit
```
POST /reset
```

- **Example Request:**

bash
CopyEdit
```
curl -X POST "http://localhost:8000/reset"
```

- **Response (JSON):**

json
CopyEdit
```
{
  "status": "HFCTM-II reset successfully",
  "knowledge_state": [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
}
```

✅ **Resets AI cognition while preserving framework integrity.**

📌4️⃣ Testing API with Python

Using `requests` Library

You can also test the API using Python:

python
CopyEdit
```
import requests

BASE_URL = "http://localhost:8000"

# Run Inference
```

```
response = requests.post(f"{BASE_URL}/inference", json={"iterations":
5})
print(response.json())

# Get Knowledge State
response = requests.get(f"{BASE_URL}/state")
print(response.json())

# Detect Egregore
response = requests.get(f"{BASE_URL}/egregore-detection")
print(response.json())

# Apply Chiral Inversion
response = requests.post(f"{BASE_URL}/chiral-inversion")
print(response.json())

# Reset Model
response = requests.post(f"{BASE_URL}/reset")
print(response.json())
```

✅ **This automates API testing with Python!**

📌5️⃣ Summary of API Endpoints

Command	Method	Description
Run Inference	`POST /inference`	Runs recursive inference cycles and updates knowledge embeddings.
Get Knowledge State	`GET /state`	Retrieves the latest recursive knowledge embeddings.
Get Trust Matrix	`GET /trust`	Fetches the fractal trust network used for AI alignment.

Detect Egregore	`GET /egregore-detection`	Runs wavelet transform scan for adversarial distortions.
Apply Chiral Inversion	`POST /chiral-inversion`	Manually neutralizes ideological attractors via inversion.
Reset Model	`POST /reset`	Resets HFCTM-II to its initial E8-projected state.