CARBONOSPHERE-AI:
A Planetary Digital Twin for Carbon Sink Intelligence and Climate Regulation
Explained as a Conversation Between Researchers

Aniket Agarwal
Bachelor of Technology, Computer Sciences
Amity University, Punjab, India

Conversation Between Two Researchers

Researcher 1 (R1): I am excited to share our new project, CARBONOSPHERE-AI, which is a self-evolving digital twin of the Earth focused on carbon sinks and greenhouse gas regulation.

Researcher 2 (R2): Interesting! But first, why do we need such a project? Aren't current climate models sufficient?

R1: Good question. Current climate models mainly focus on emissions reduction and predicting temperature changes. But carbon sinks—the forests, oceans, and soils that absorb CO_2—are not dynamically optimized. They are complex, slow-responding, and prone to sudden tipping points. CARBONOSPHERE-AI fills this gap by actively modeling and governing these sinks.

R2: So, you are saying it focuses on carbon absorption rather than just emissions? How does that help the environment?

R1: Exactly. By understanding and managing carbon sinks, we can stabilize the carbon cycle. This prevents excess CO_2 from accumulating, reduces climate risk, and promotes ecosystem restoration. Essentially, it targets the root cause of atmospheric carbon imbalance rather than just its source.

R2: I see. Can you explain the core idea of the project in simple terms?

R1: Think of Earth as a living network where regions like forests, oceans, and soils interact. Each region is a node in a graph, and the interactions—like carbon flux—are edges. We then simulate how interventions in one area affect others globally. We use Graph Neural Networks (GNNs) to learn these interactions.

R2: How does a Graph Neural Network help here?

R1: GNNs capture complex spatial dependencies. For example, cutting a forest in one region might increase carbon in the atmosphere, which impacts oceanic absorption elsewhere. GNNs learn these patterns automatically. Here's a minimal example:

```
from torch_geometric.nn import GCNConv

class EarthGNN(torch.nn.Module):
    def __init__(self):
        super().__init__()
        self.conv1 = GCNConv(3, 16)
        self.conv2 = GCNConv(16, 1)

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index).relu()
        return self.conv2(x, edge_index)
```

R2: That's clear. But carbon sinks follow physical laws. How do you make sure the AI predictions are realistic?

R1: Great point. We use Physics-Informed Neural Networks (PINNs) to embed constraints like ocean solubility limits and phytoplankton growth dynamics. This ensures that the AI never predicts physically impossible outcomes.

R2: Interesting. What about long-term trends and delayed effects?

R1: Carbon sinks respond slowly. To capture this, we integrate neuromorphic memory using spiking neural networks. These networks remember decade-scale patterns and detect early warning signs of tipping points.

R2: And you mentioned it actively controls carbon sinks. How does that work?

R1: We use reinforcement learning, specifically Proximal Policy Optimization (PPO). The AI experiments with interventions like reforesting or enhancing oceanic sinks, observes their impact, and updates its strategy to maximize carbon absorption.

```
action = policy(state)
next_state, reward = environment.step(action)
policy.update(state, action, reward)
```

R2: What about software and implementation? How is this actually running?

R1: We use a mix of Python libraries:

- **PyTorch + PyTorch Geometric** for GNNs and learning spatial interactions.
- **PINNs** implemented in TensorFlow or PyTorch for physics constraints.
- **Gym + PPO** for reinforcement learning of intervention policies.
- **Plotly Dash** for interactive 3D visualization of carbon flux and sink states.
- **NetCDF + ERA5 data** for real-world ocean and atmospheric inputs.

R2: Can you show an example of how visualization works?

R1: Sure. We create interactive 3D graphs where node color indicates carbon flux:

```
import plotly.graph_objects as go

fig = go.Figure(data=[go.Scatter3d(
    x=positions_x, y=positions_y, z=positions_z,
    mode='markers',
    marker=dict(size=5, color=carbon_flux, colorscale='Viridis')
)])
fig.show()
```

R2: Wow, that's interactive and intuitive. How does this project differ from other carbon models?

R1: Most models are static, predictive, and isolated. CARBONOSPHERE-AI is adaptive, integrates AI with physics, economics, and policy, and interacts with the system dynamically. It can propose optimal interventions instead of just predicting scenarios.

R2: Could this help in real-world climate policy?

R1: Absolutely. Policymakers could use our simulations to test interventions, prioritize areas for restoration, and assess trade-offs between economic cost and environmental benefit.

R2: How can this project be further developed in the future?

R1: Future directions include:

- Higher-resolution regional twins
- Integration of biodiversity and water cycles
- Real-time satellite data assimilation
- Global climate negotiation and policy simulations

R2: And overall, how will it impact the environment?

R1: By focusing on carbon sinks, it stabilizes atmospheric CO_2, restores ecosystems, informs better climate strategies, and helps prevent catastrophic tipping points.

R2: That's revolutionary. Finally, what is the main takeaway?

R1: CARBONOSPHERE-AI transforms climate modeling from static prediction to adaptive, proactive, sink-centric intelligence. It bridges science, policy, and technology to actively safeguard the planet's carbon balance.

R2: I'm convinced. This is indeed a paradigm shift in climate intelligence.