Explanation of Spawning Logic (spawn_child_models)

This section explains the function used to spawn child models when the parent model's performance is insufficient or the data distribution is different.

```
def spawn_child_models(parent_model, new_loader, new_data_sample, dataset_name, data_type, node_id, tree):
    kl_threshold = 0.5
    performance_threshold = 70.0
    parent_accuracy = evaluate_model(parent_model, new_loader, data_type)
    kl_div = compute_kl_divergence(tree.data_distribution["parent"]["sample"], new_data_sample)
    print(f"Parent Accuracy on {dataset_name}: {parent_accuracy:.2f}%, KL Divergence: {kl_div:.4f}")
    if parent_accuracy < performance_threshold and kl_div < kl_threshold:
        print(f"Spawning Child Models for {dataset_name}...")
        parent_features = parent_model.get_features(tree.data_distribution["parent"]["sample"].to(device)).mean(dim=0)
        if data_type == "sequence":
            child_model = ChildRNN(input_size=len(vocab), hidden_size=128, output_size=10, parent_features=parent_features).to(device)
        else:
            child_model = ChildCNN(parent_features=parent_features).to(device)

        child_optimizer = optim.Adam(child_model.parameters(), lr=0.001)
        for epoch in range(2):
            child_model.train()
            for data, target in new_loader:
                data, target = data.to(device), target.to(device)
                child_optimizer.zero_grad()
                output = child_model(data)
                loss = criterion(output, target)
                loss.backward()
                child_optimizer.step()

            print(f"Child Model {node_id} Epoch {epoch+1}, Loss: {loss.item():.4f}")
            tree.add_node(node_id, child_model, dataset_name, {"loader": new_loader, "sample": new_data_sample, "type": data_type})
            tree.add_edge("parent", node_id)
            return True
        return False
```

Line-by-line explanation:

- Sets KL divergence and performance thresholds.
- Evaluates parent model's accuracy on new data and computes KL divergence between parent and new data samples.
- If accuracy is low and KL divergence is small, spawns a child model:
  - Extracts parent features.
  - Chooses ChildRNN or ChildCNN based on data type.
  - Trains the child model for 2 epochs on the new data.
  - Adds the child model to the tree and links it to the parent.
- Returns True if a child model was spawned, otherwise False.

Purpose:

- This function enables knowledge transfer and adaptation to new data by creating specialized child models when needed.