**DQN Project Graphs**

**The hyperparameter sets are as follows:**

**HP\_CONFIGS = {**

**"conservative": {**

**'memory\_size': 50000,**

**'batch\_size': 128,**

**'gamma': 0.95,**

**'epsilon\_decay': 0.98,**

**'epsilon\_min': 0.1,**

**'learning\_rate': 0.001,**

**'tau': 0.1,**

**'network\_layers': [128, 64],**

**'energy\_penalty\_scale': 0.002,**

**'prioritization\_beta': 0.5**

**},**

**"aggressive": {**

**'memory\_size': 200000,**

**'batch\_size': 512,**

**'gamma': 0.99,**

**'epsilon\_decay': 0.99,**

**'epsilon\_min': 0.2,**

**'learning\_rate': 0.0003,**

**'tau': 0.05,**

**'network\_layers': [512, 256],**

**'efficiency\_bonus': 3.0,**

**'task\_reward': 10.0**

**},**

**"balanced": {**

**'memory\_size': 100000,**

**'batch\_size': 256,**

**'gamma': 0.97,**

**'epsilon\_decay': 0.995,**

**'epsilon\_min': 0.05,**

**'learning\_rate': 0.0005,**

**'tau': 0.12,**

**'network\_layers': [256, 128],**

**'energy\_penalty\_scale': 0.0015,**

**'budget\_penalty': 12.0**

**},**

**"hp6": {**

**'memory\_size': 30000,**

**'batch\_size': 64,**

**'gamma': 0.93,**

**'epsilon\_decay': 0.97,**

**'epsilon\_min': 0.2,**

**'learning\_rate': 0.002,**

**'tau': 0.25,**

**'network\_layers': [64, 32],**

**'energy\_penalty\_scale': 0.001,  # Reduced penalty**

**'reward\_components': {**

**'task': 9.0,**

**'efficiency': 1.5,**

**'budget': 8.0**

**}**

**},**

**"hp5": {**

**'memory\_size': 150000,**

**'batch\_size': 384,**

**'gamma': 0.98,**

**'epsilon\_decay': 0.998,  # Slower exploration decay**

**'epsilon\_min': 0.15,**

**'learning\_rate': 0.0002,**

**'tau': 0.08,**

**'network\_layers': [512, 256, 128],**

**'prioritization\_alpha': 0.7,  # More aggressive prioritization**

**'prioritization\_beta': 0.5**

**},**

**"hp4": {**

**'memory\_size': 75000,**

**'batch\_size': 192,**

**'gamma': 0.96,**

**'epsilon\_decay': 0.985,**

**'epsilon\_min': 0.05,**

**'learning\_rate': 0.0007,**

**'tau': 0.15,**

**'task\_reward': 10.0,  # Higher task completion reward**

**'efficiency\_bonus': 3.0,**

**'network\_layers': [384, 192]**

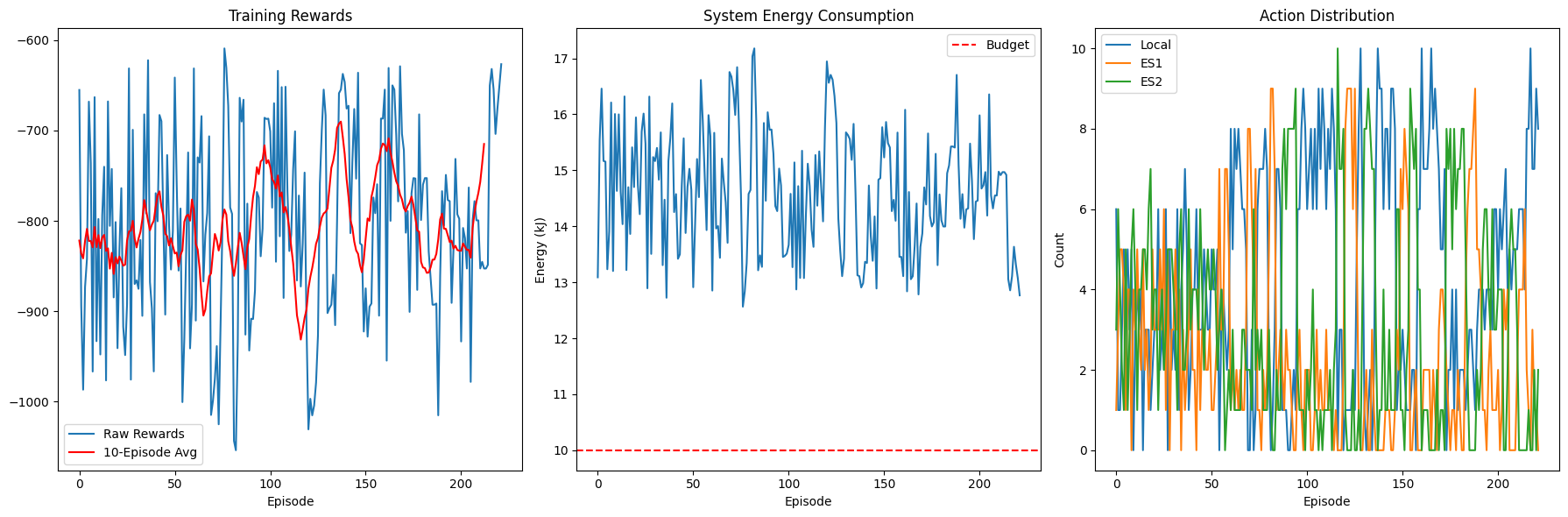
**}**

**}**

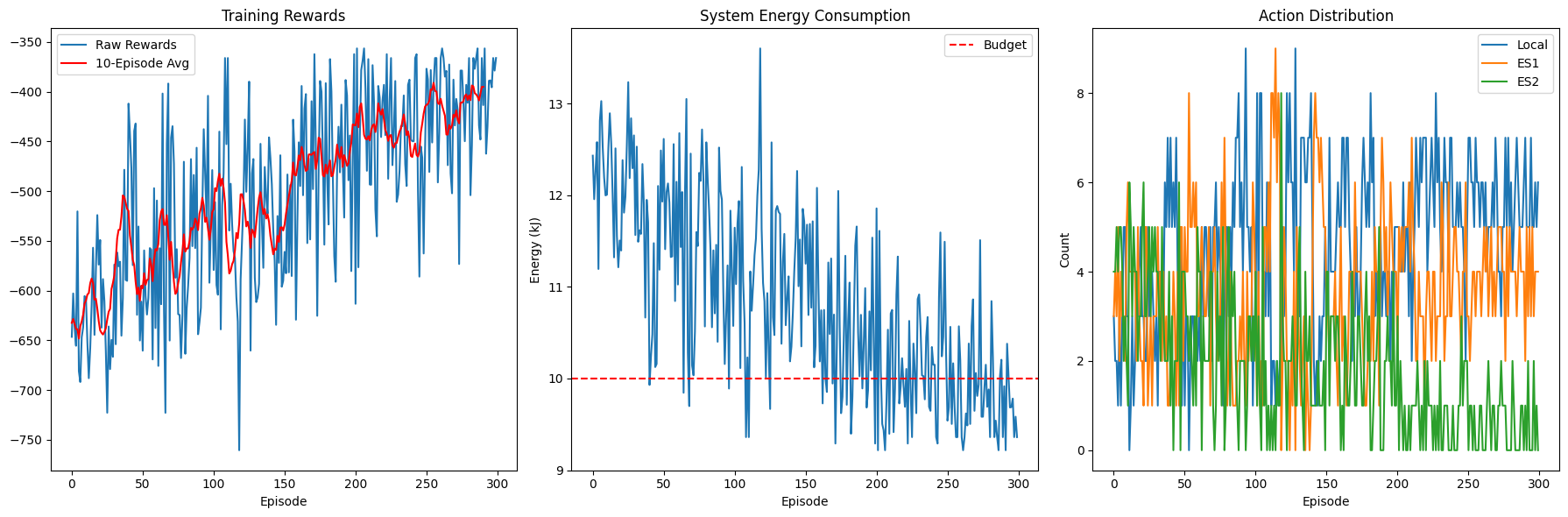
**Hyperparameter Tuning  
  
  
  
  
(i) Set A(Conservative)**

****

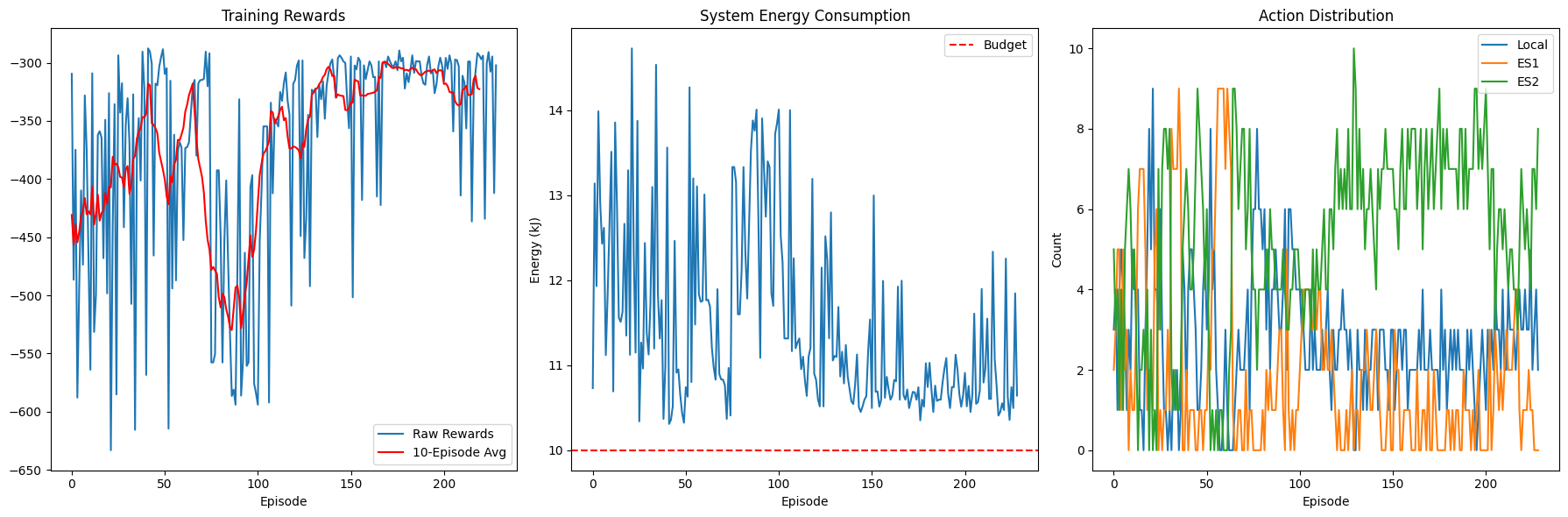
**(ii)Set B(Aggressive)**

****

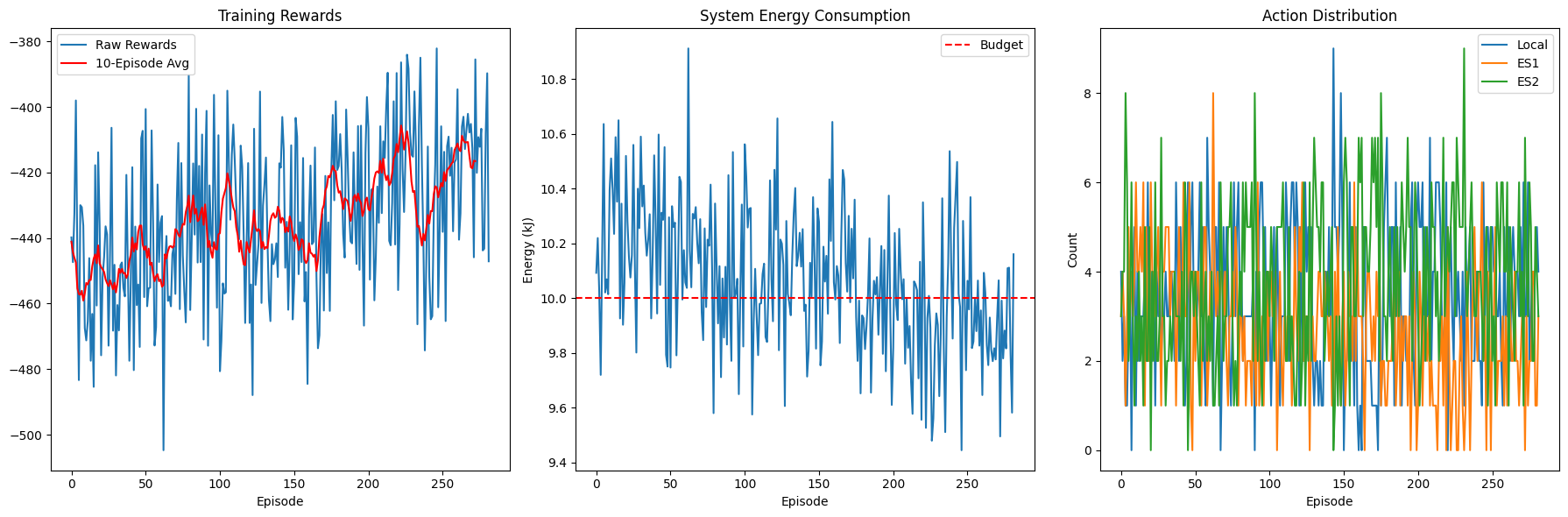
**(iii)Set c(balanced)**

****

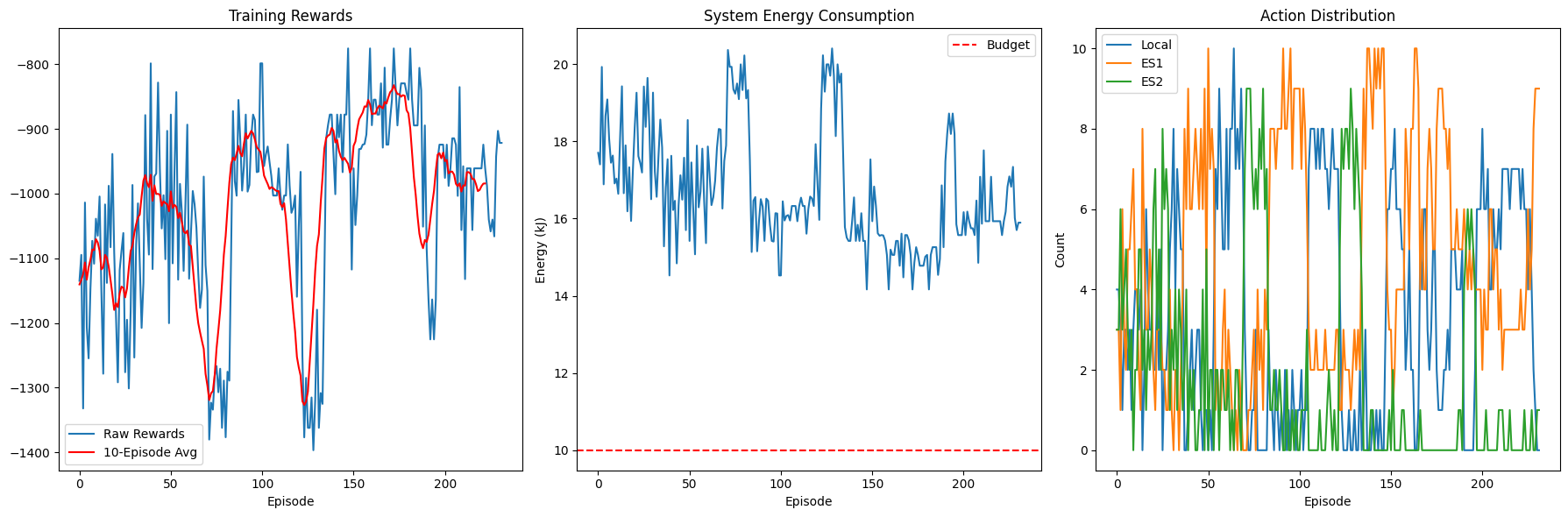
**(iv)Hp 6**

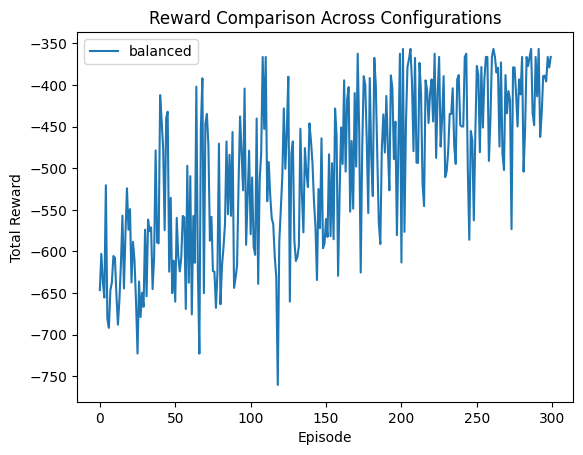
****

**(v)Hp 5**

****

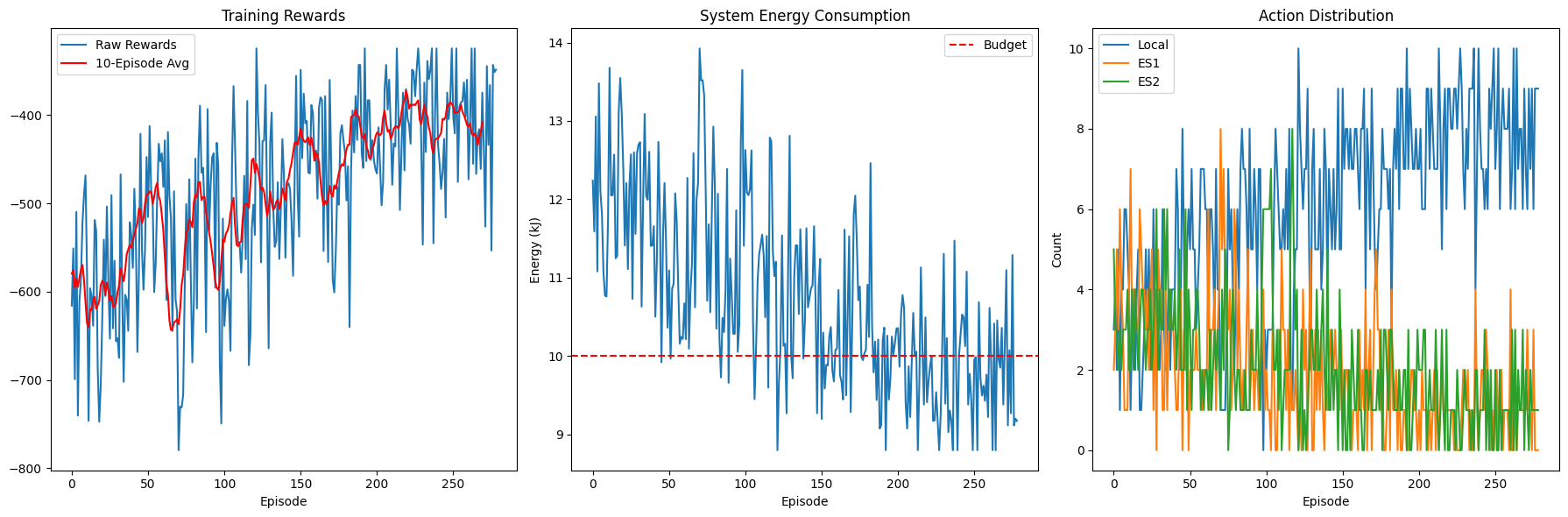
**(vi)Hp 6**

**  
  
  
The best results were seen in the balanced hyperparameter set.**

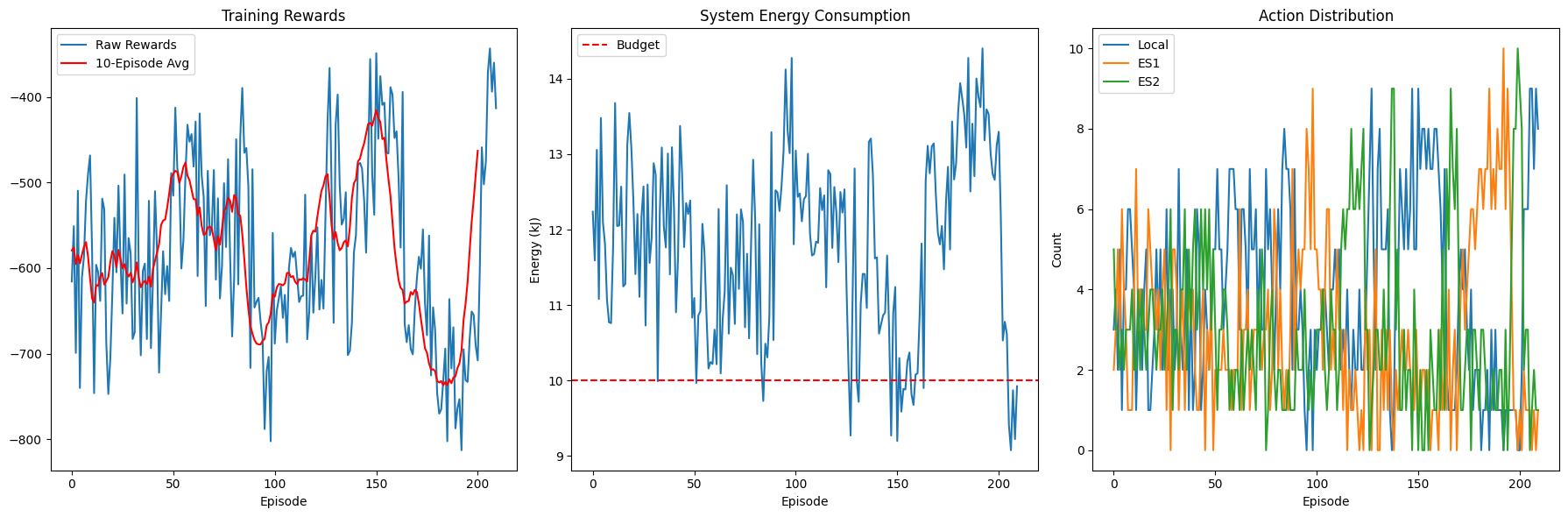
****

**New Model with few files(10)**

**(i)Learning rate =0.0005**

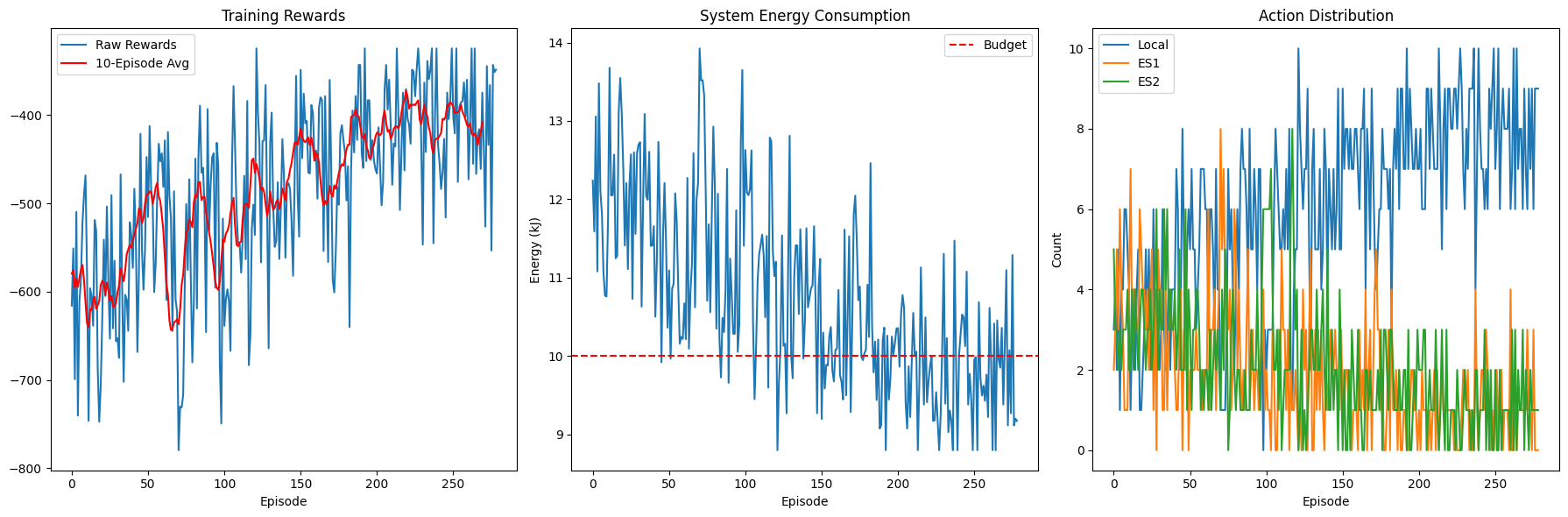
****

**(ii)Learning rate =0.0001**

****

**New model in different Environments**

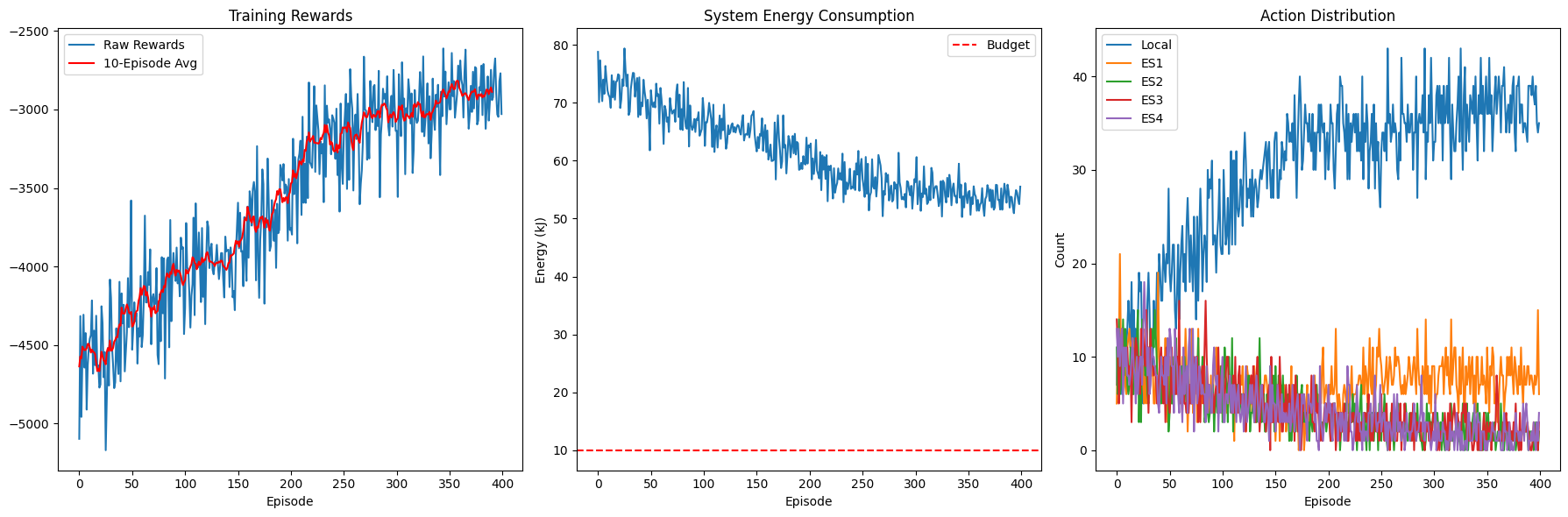
**(i)md=3,es=2,num\_tasks=10**

****

**(ii)md=4,es=1,num\_tasks=20**

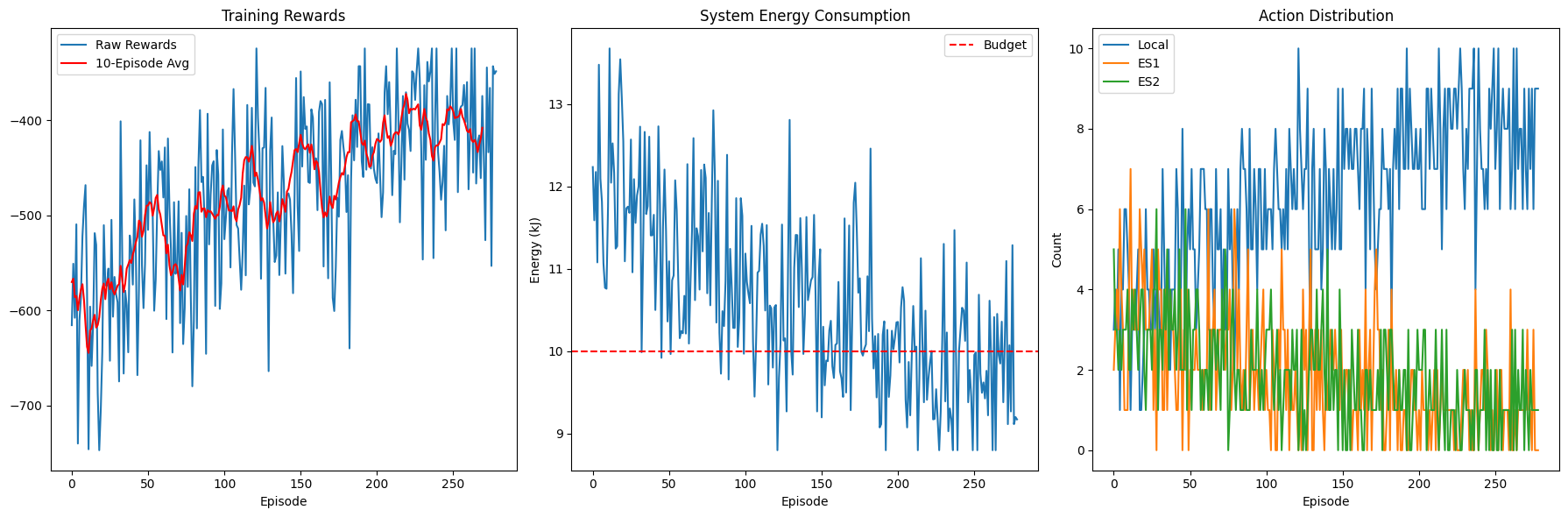
****

(iii)md=10,es=4,num\_tasks=50

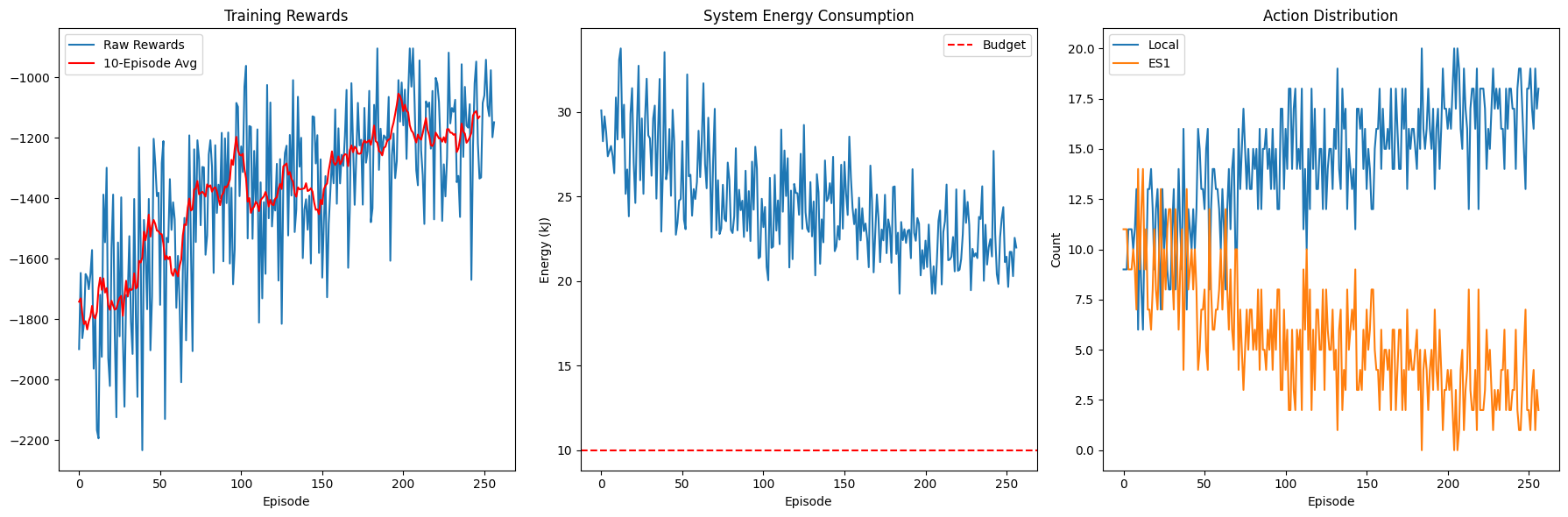


**Traditional Model**

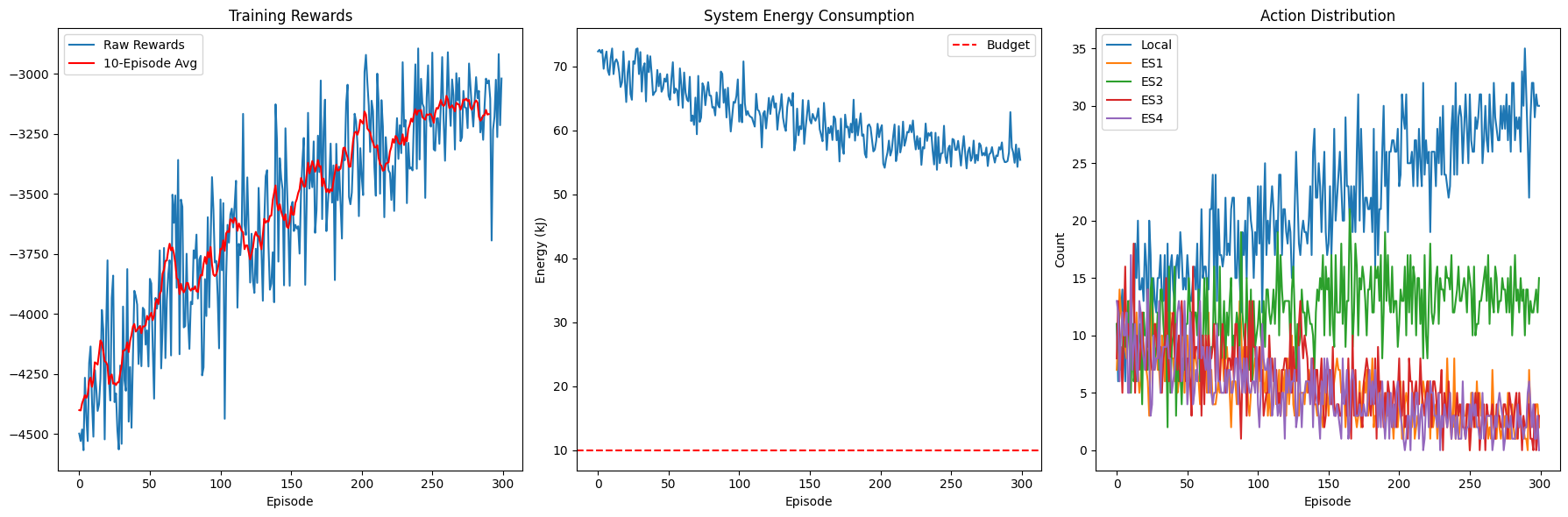
(i)md=3,es=2,num\_tasks=10



(ii)md=4,es=1,num\_tasks=20



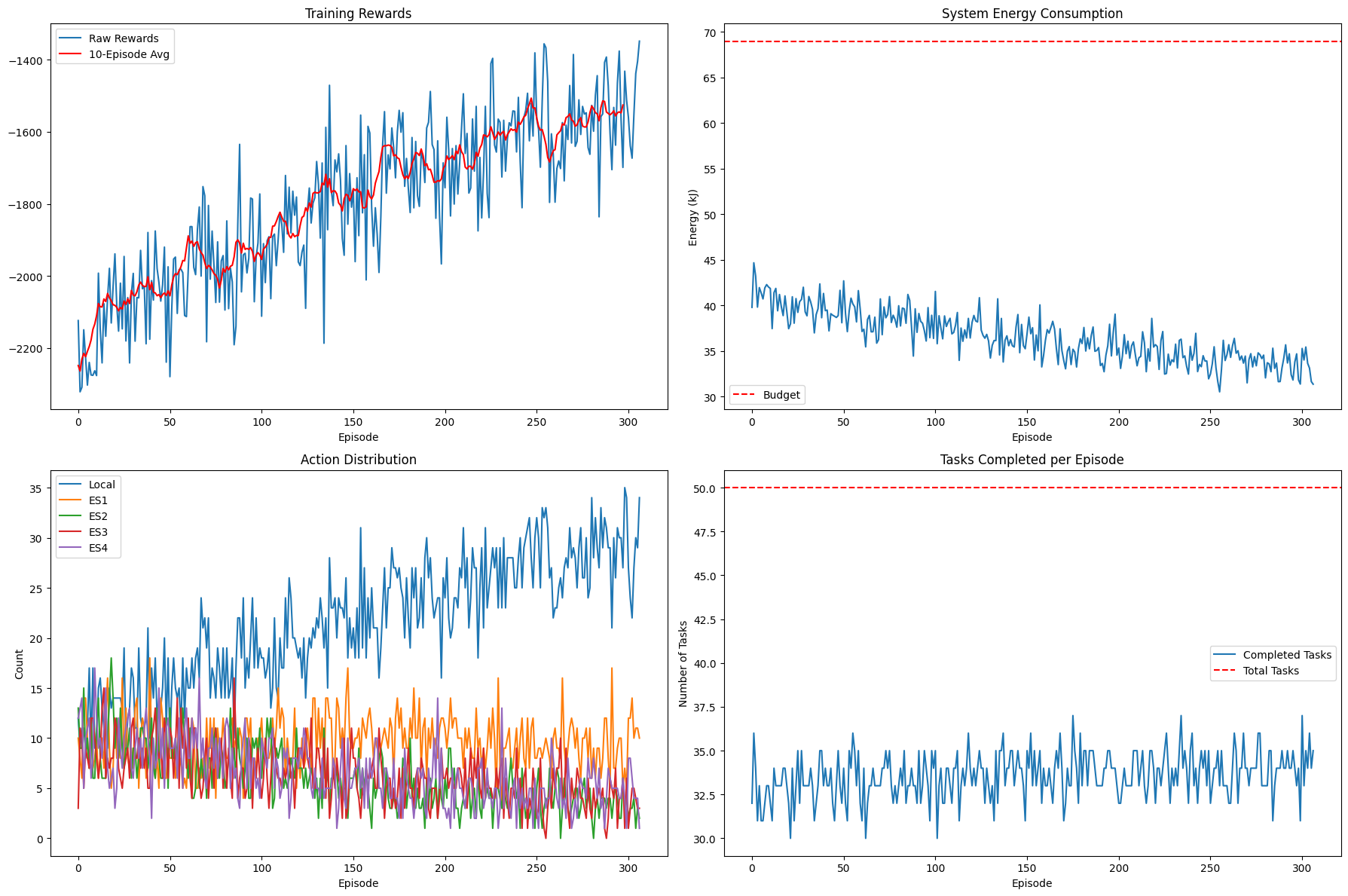
(iii)md=10,es=4,num\_tasks=50



In the third case, the training stops due to early stopping due to decrease in mean rewards over episode.

The traditional model and the new model provide similar rewards for most cases. But

The new model works better for 50 tasks as shown in the graph.  
  
  
  
  
  
Traditional Model Stress





--- Final Average Results (last 50 episodes) ---

Traditional DQN:

Avg Reward: 24.48

Avg Energy (kJ): 17.42

Avg Completion: 100.00%

Advanced DQN:

Avg Reward: 25.11

Avg Energy (kJ): 17.24

Avg Completion: 100.00%  
  
