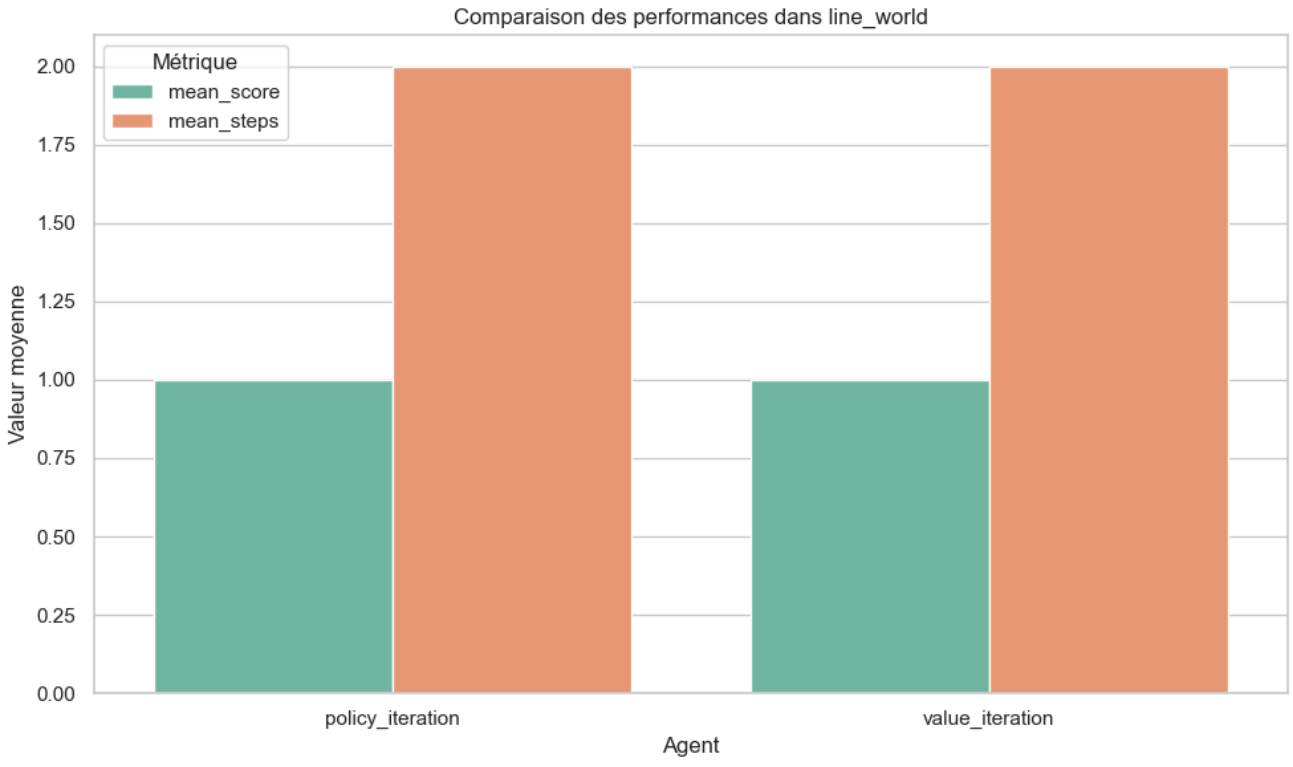
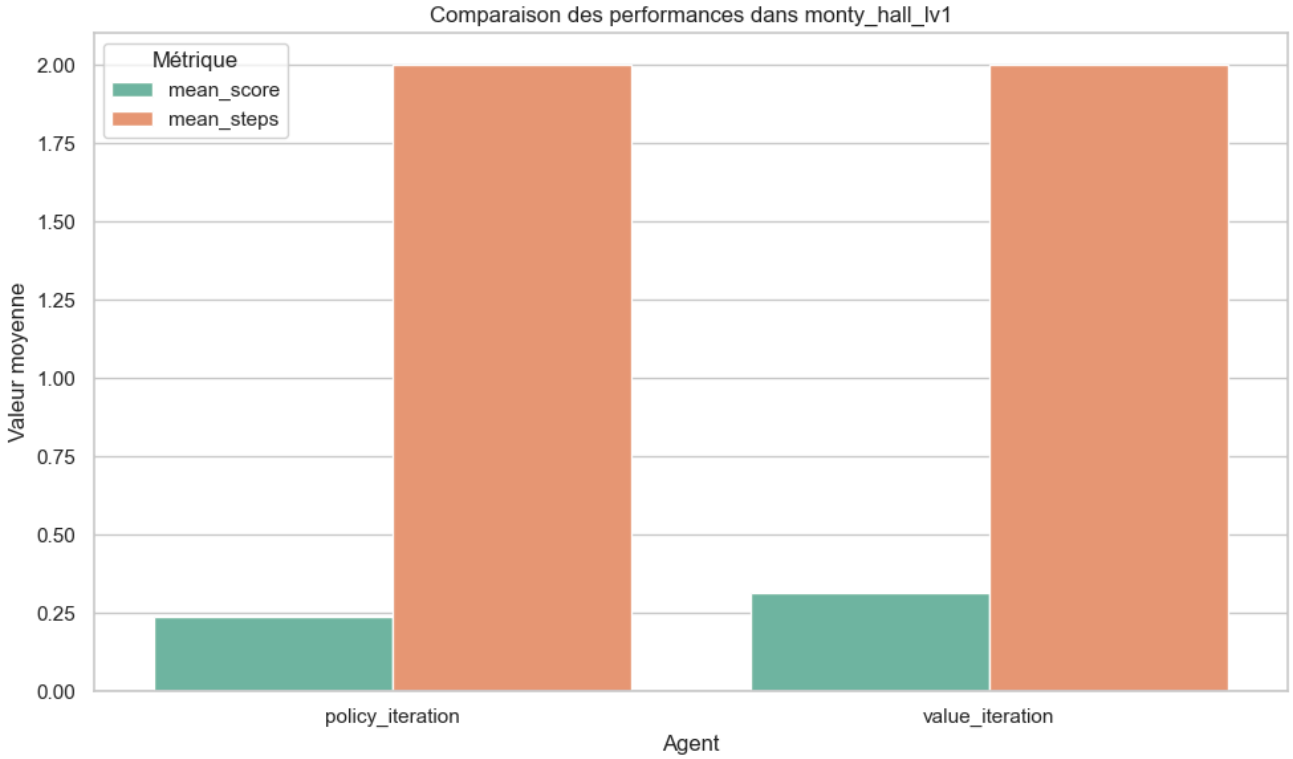
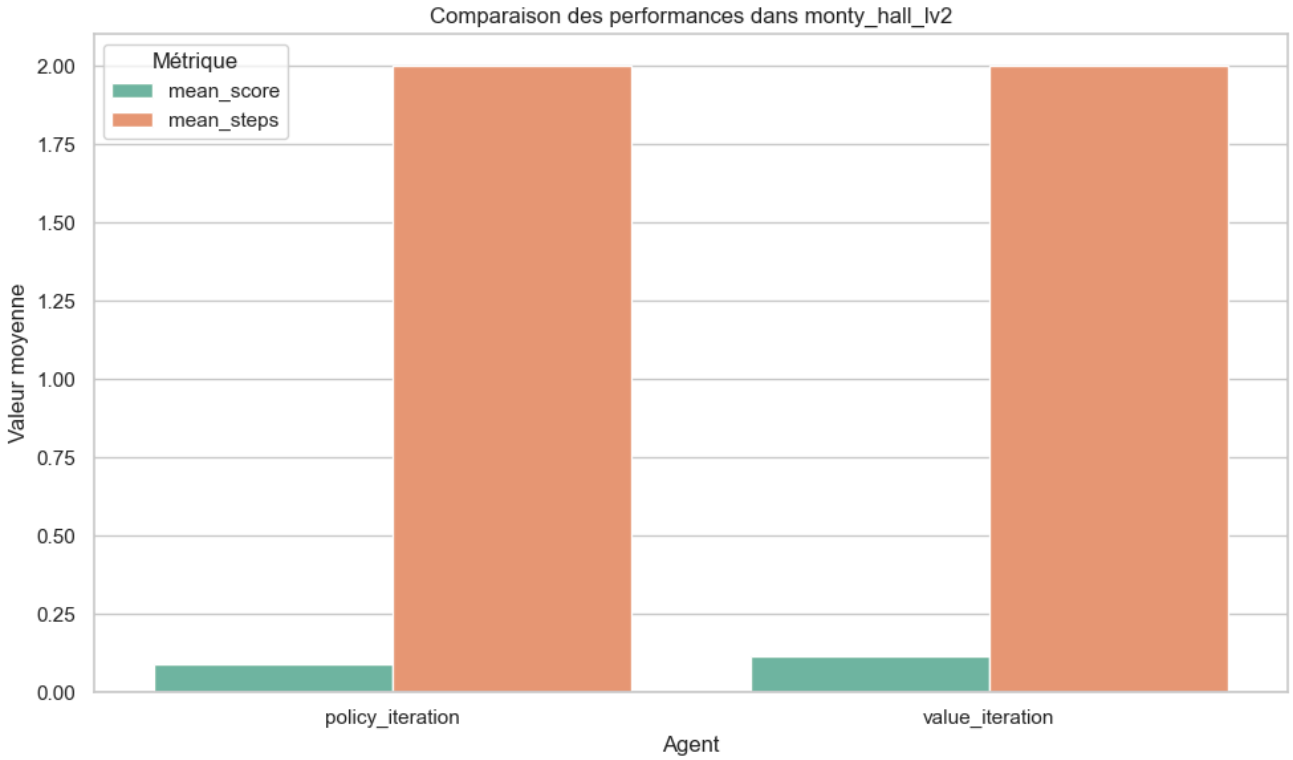
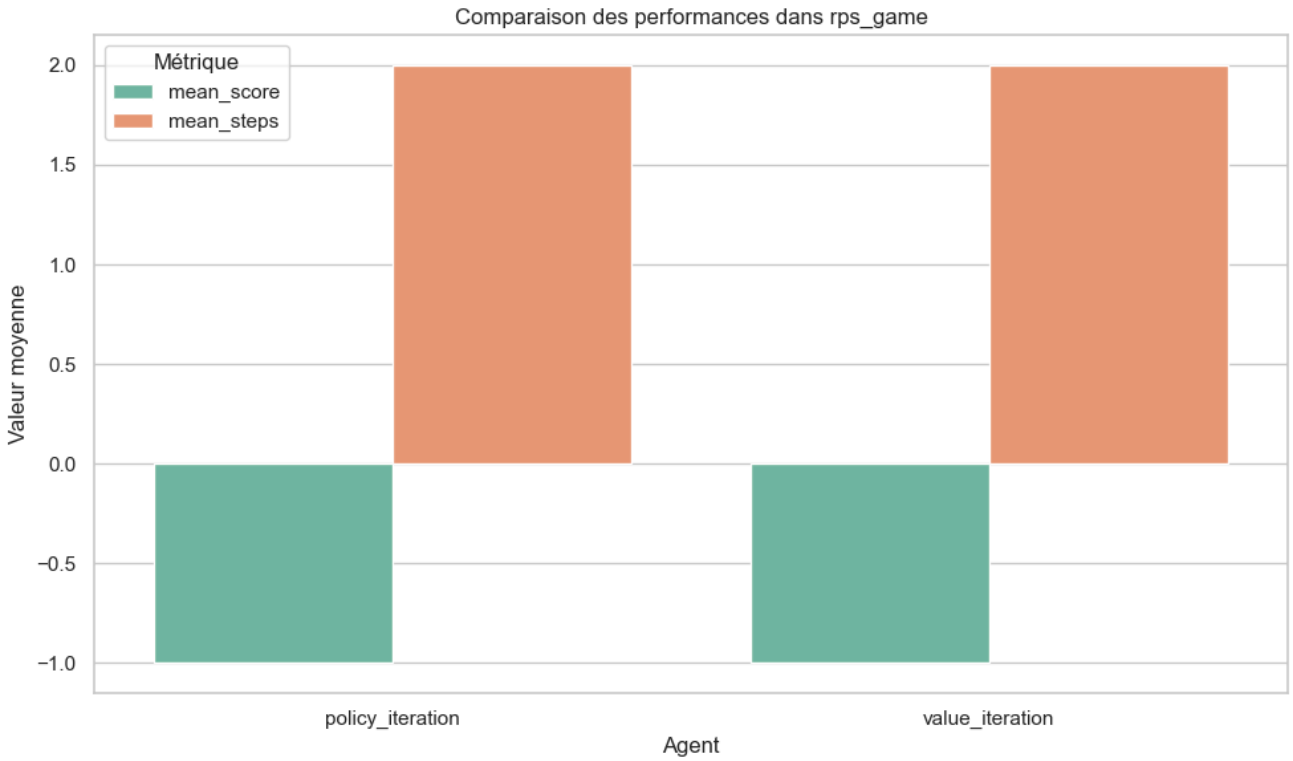
FAMILLE DYNAMIC PROGRAMMING

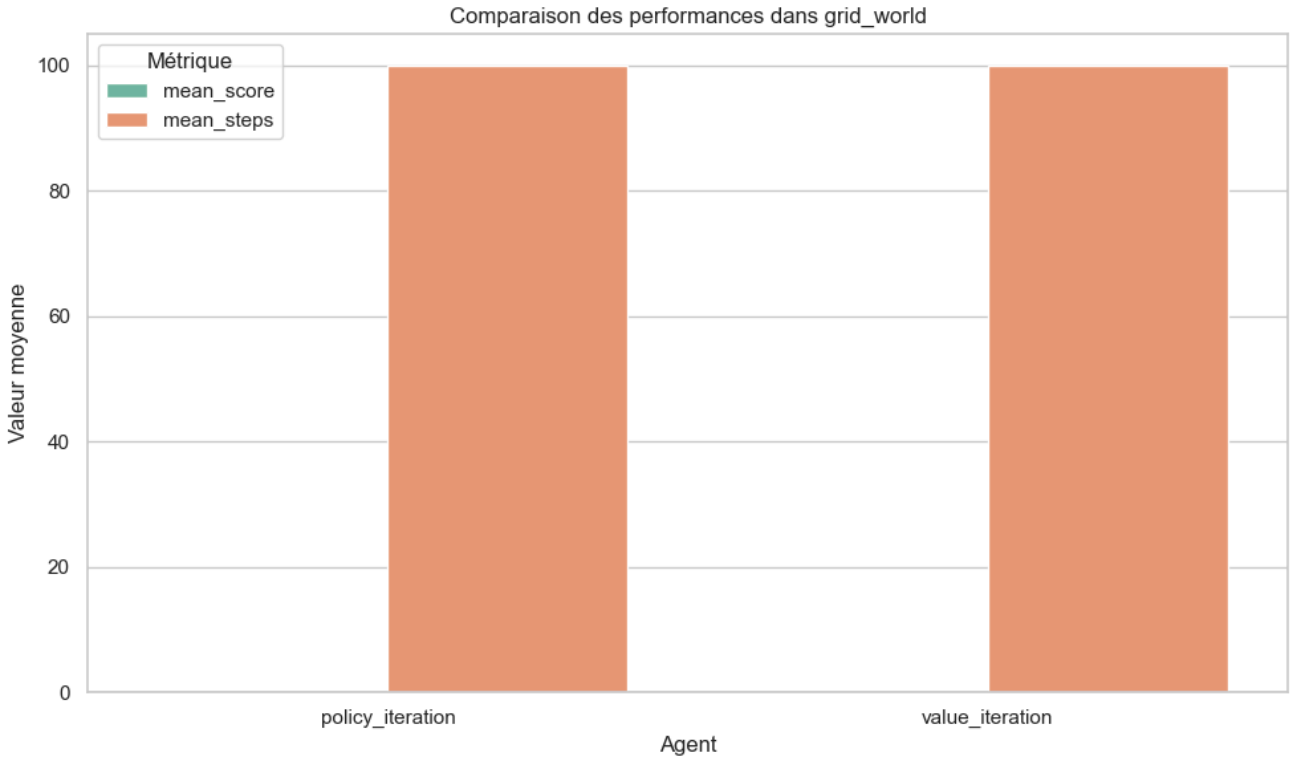
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

file_path = "Reports/dp_agents_comparison.xlsx"
df = pd.read_excel(file_path)

# Configuration d'affichage
sns.set(style="whitegrid")
output_dir = "./dp_graphs"
os.makedirs(output_dir, exist_ok=True)

# Boucle pour chaque environnement
environments = df["env"].unique()
for env in environments:
    plt.figure(figsize=(10, 6))
    subset = df[df["env"] == env]

    # Regrouper pour obtenir les moyennes
    pivot = subset.groupby("agent")[["mean_score", "mean_steps"]].mean().reset_index()
    pivot_melted = pivot.melt(id_vars="agent", value_vars=["mean_score", "mean_steps"],
                              var_name="Métrique", value_name="Valeur")

    sns.barplot(data=pivot_melted, x="agent", y="Valeur", hue="Métrique", palette="Set2")
    plt.title(f"Comparaison des performances dans {env}")
    plt.ylabel("Valeur moyenne")
    plt.xlabel("Agent")
    plt.tight_layout()
    plt.savefig(os.path.join(output_dir, f"{env}_score_steps_comparison.png"))
    plt.show()
```




Comparaison des performances dans monty_hall_lv1
Métrique
mean_score
mean_steps
Valeur moyenne
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
policy_iteration
value_iteration
Agent


Comparaison des performances dans monty_hall_lv2
Métrique
mean_score
mean_steps
Valeur moyenne
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
policy_iteration
value_iteration
Agent





Hyperparameter de Policy et Value Iteration GAMMAS = [0.3, 0.5, 0.7, 0.9] THETAS = [0.0001, 0.00001] Ces deux fontions ont des performance identiques sur gridworld, line world et rps_game mais pour le reste value iteration est plus performant. (oublie pas dutiliser l'image sur la variation de gamma)

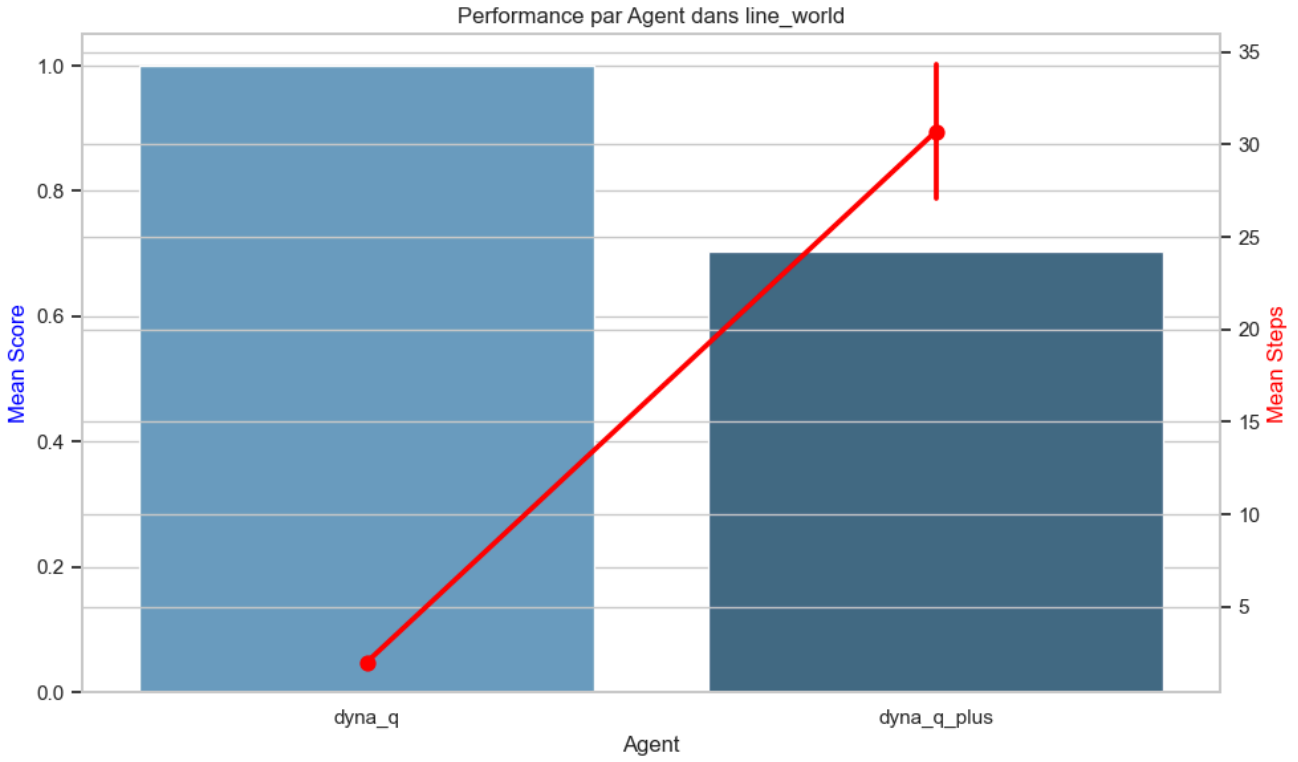
FAMILLE PLANNING METHODS

```
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

# === Chemin vers le fichier Excel ===
file_path = "Reports/planning_agents_comparison.xlsx"
df = pd.read_excel(file_path)
```

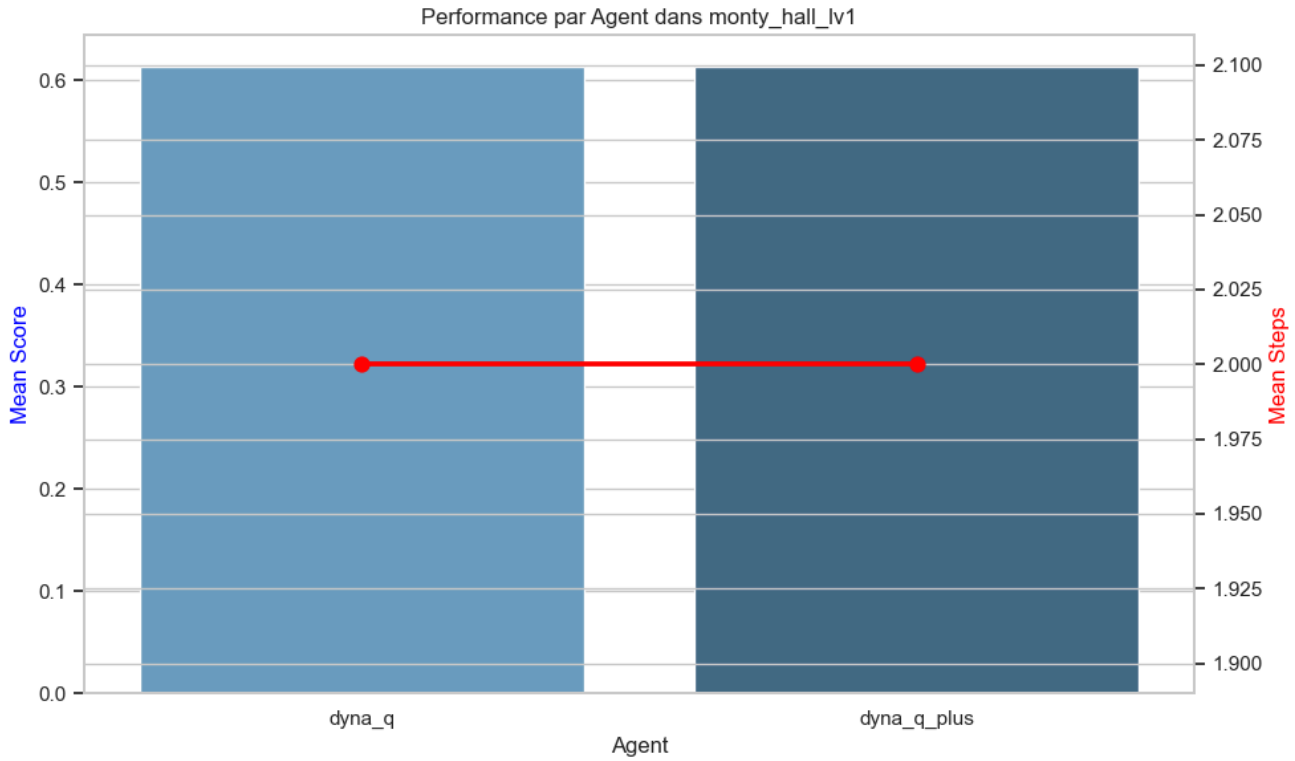
```
envs = df['env'].dropna().unique()

# === GRAPHE 1 : Mean Score + Mean Steps pour chaque agent dans chaque environnement ===
for env in envs:
    env_df = df[df['env'] == env]

    fig, ax1 = plt.subplots()

    # Barplot du score moyen
    sns.barplot(
        data=env_df,
        x="agent",
        y="mean_score",
        ci=None,
        ax=ax1,
        palette="Blues_d"
    )
    ax1.set_ylabel("Mean Score", color="blue")
    ax1.set_title(f"Performance par Agent dans {env}")
    ax1.set_xlabel("Agent")

    ax2 = ax1.twinx()
    sns.pointplot(
        data=env_df,
        x="agent",
        y="mean_steps",
        color="red",
        ax=ax2
    )
    ax2.set_ylabel("Mean Steps", color="red")

    plt.tight_layout()
    plt.show()

# === GRAPHE 2 : Impact des hyperparamètres sur le score moyen ===
hyperparams = ["gamma", "alpha", "epsilon", "planning_steps", "kappa"]

for env in envs:
    env_df = df[df["env"] == env]
    for param in hyperparams:
        if param in df.columns and not df[param].isnull().all():
            plt.figure()
            sns.barplot(
                data=env_df,
                x=param,
                y="mean_score",
                hue="agent",
                ci="sd",
                palette="Set2"
            )
            plt.title(f"Impact de {param} sur Score moyen ({env})")
            plt.ylabel("Score moyen")
            plt.xlabel(param.capitalize())
            plt.grid(True, axis="y")
            plt.legend(title="Agent")
            plt.tight_layout()
            plt.show()

# === Résumé des performances globales ===
```

```
summary = df.groupby("agent")[["mean_score", "mean_steps", "time"]].mean().round(2)
print("\nRésumé des performances globales :")
display(summary.sort_values(by="mean_score", ascending=False))  # Pour joli affichage Jupyt
```

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



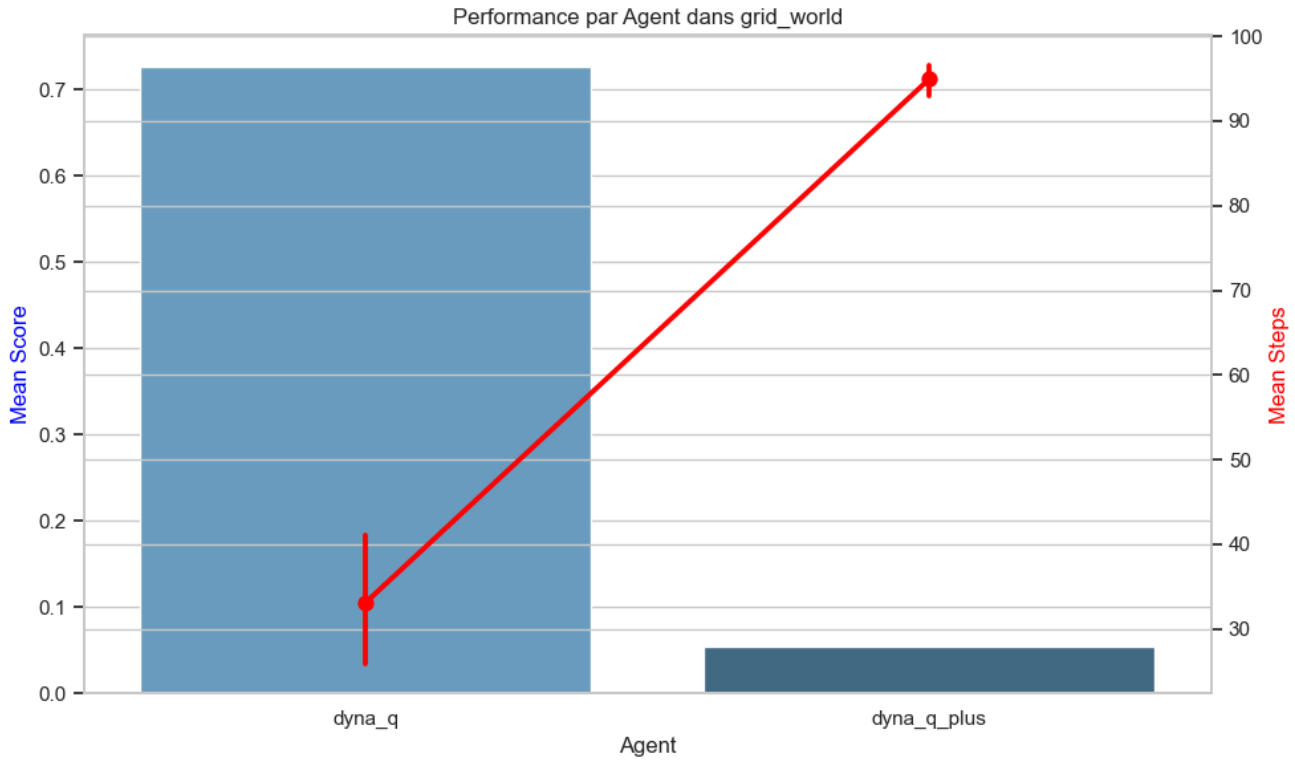

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



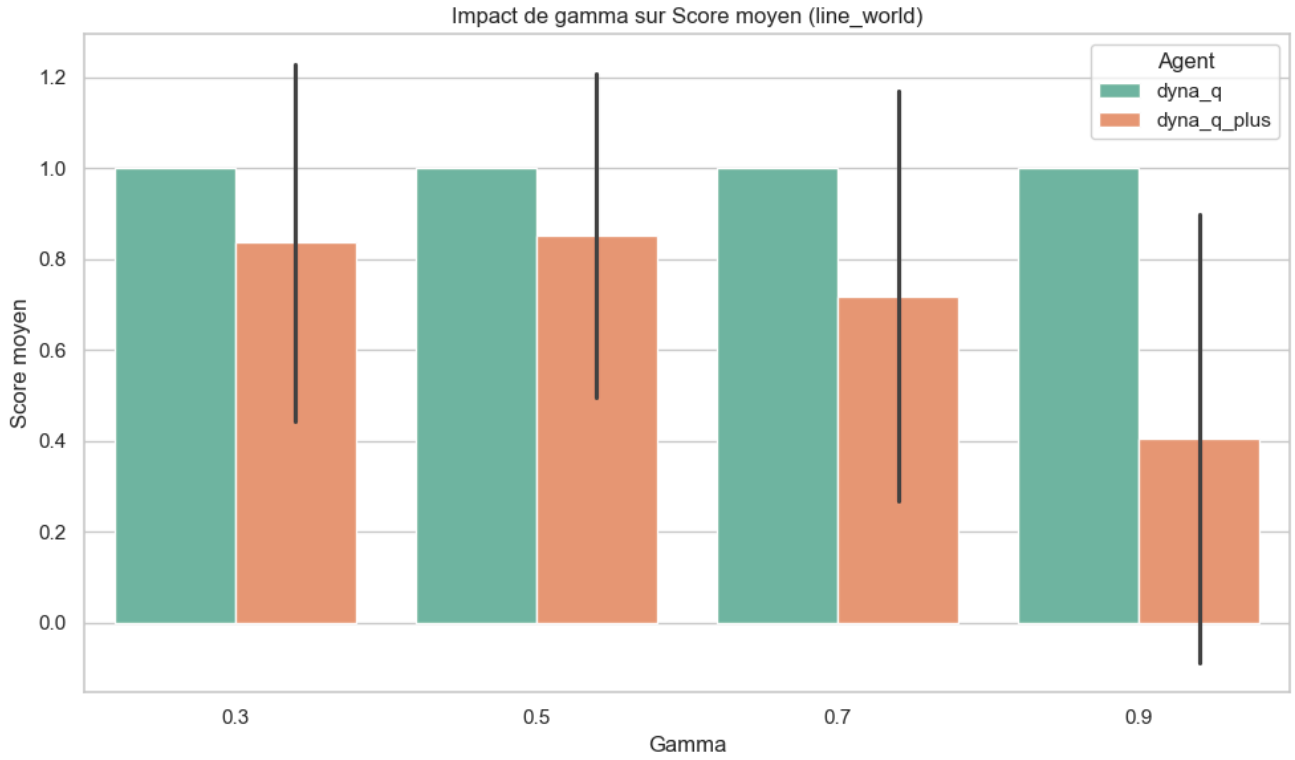

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```

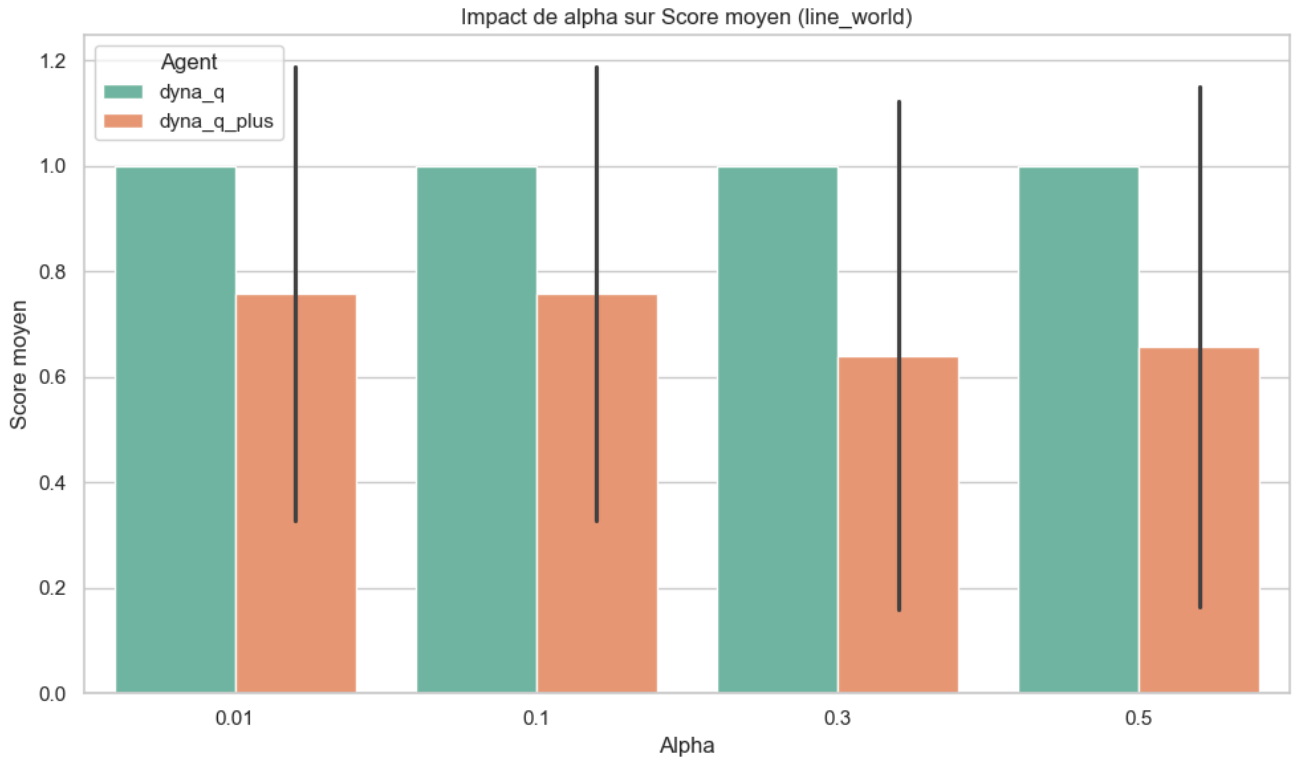

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:22: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



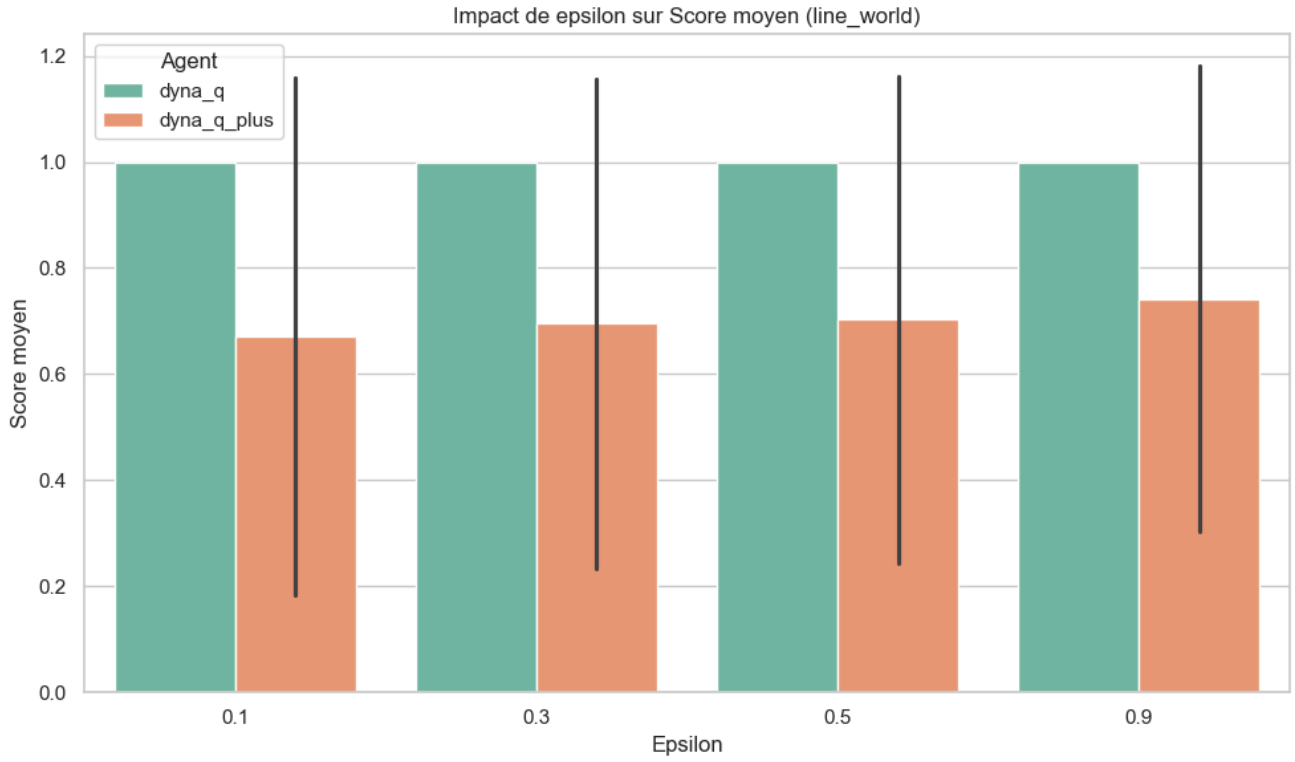

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



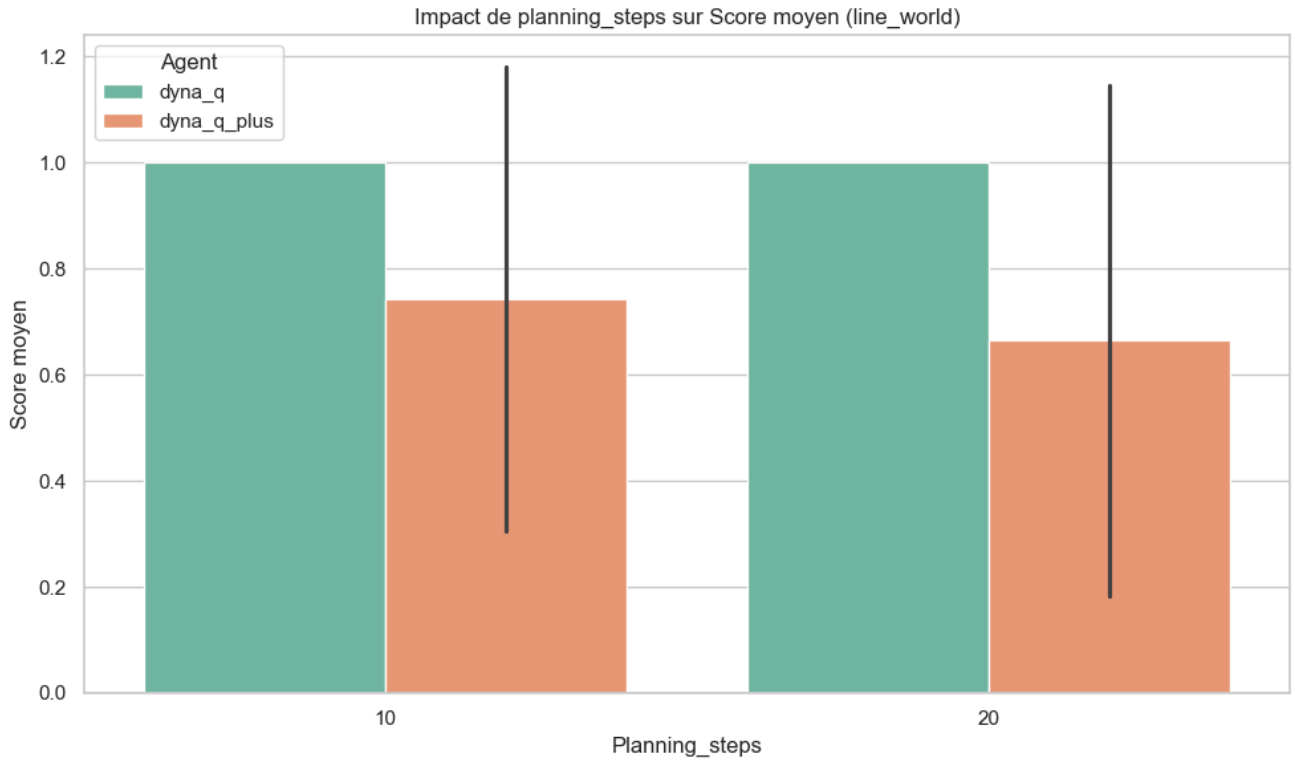

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



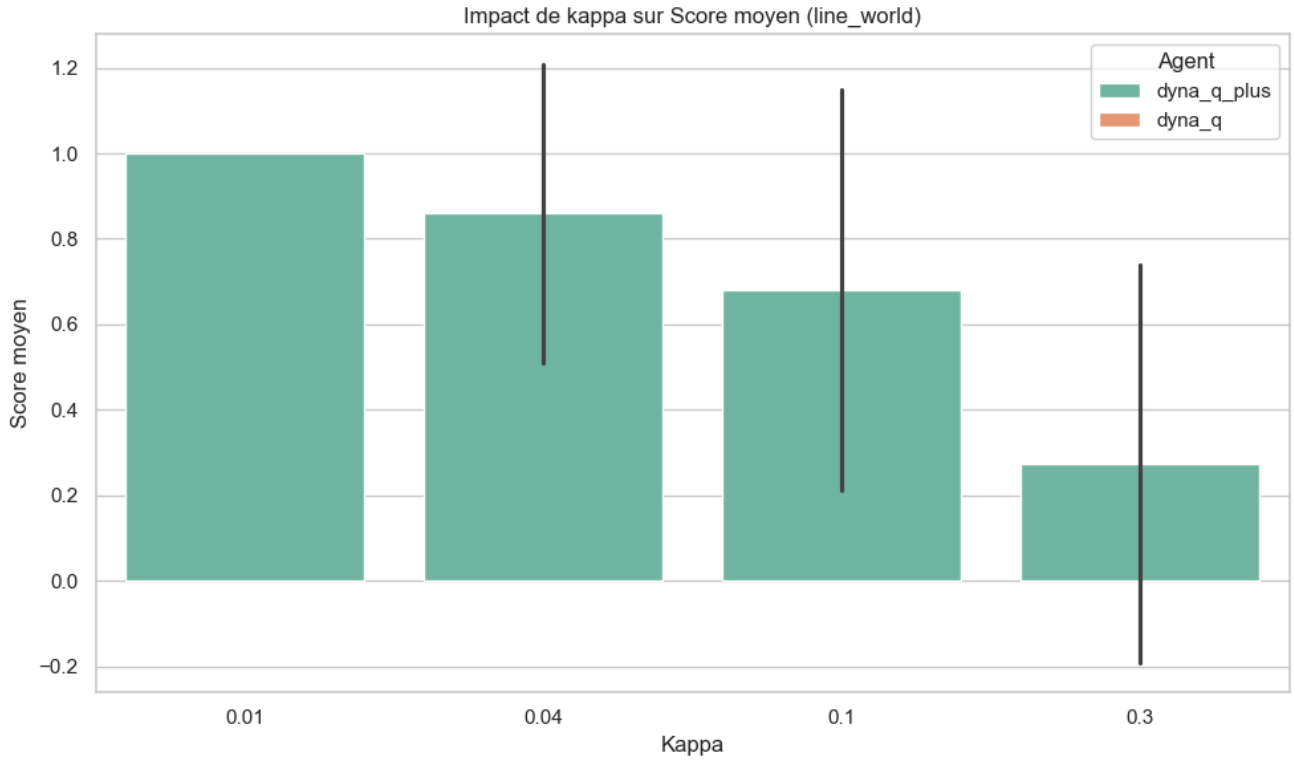

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



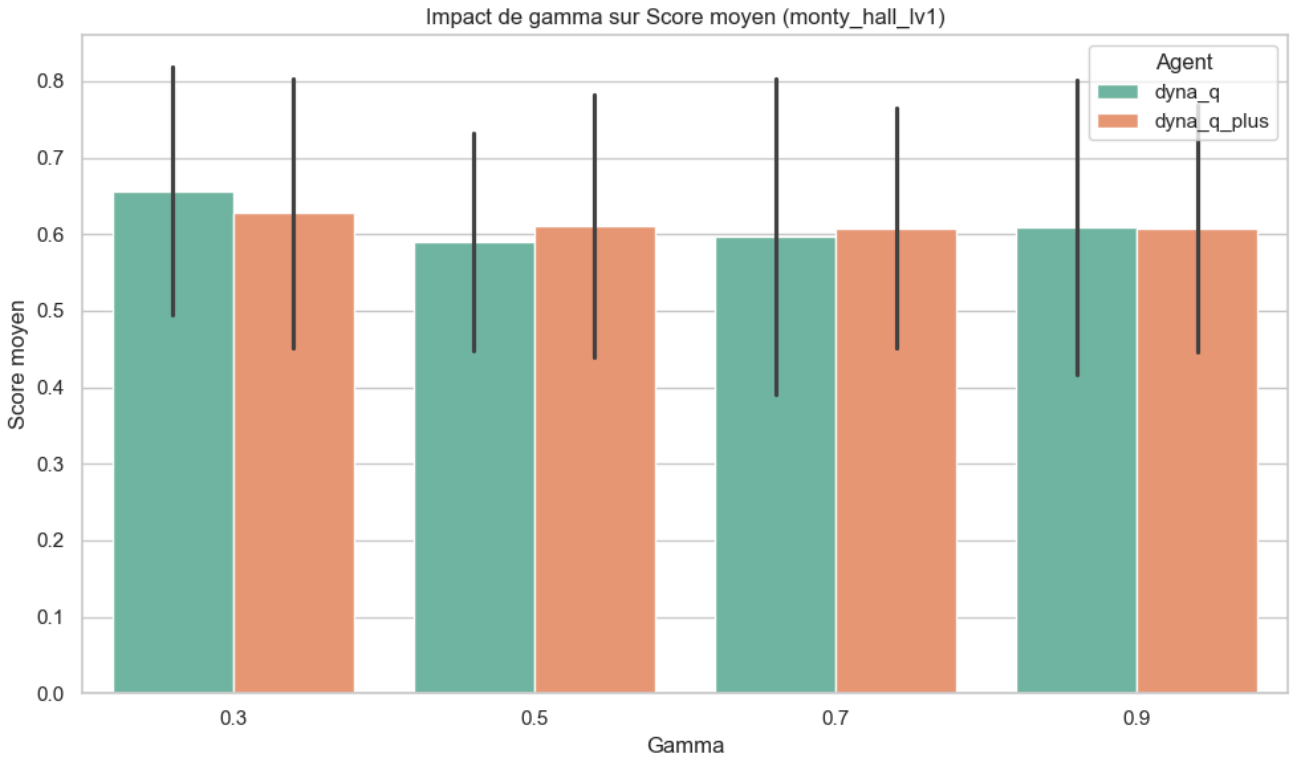

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



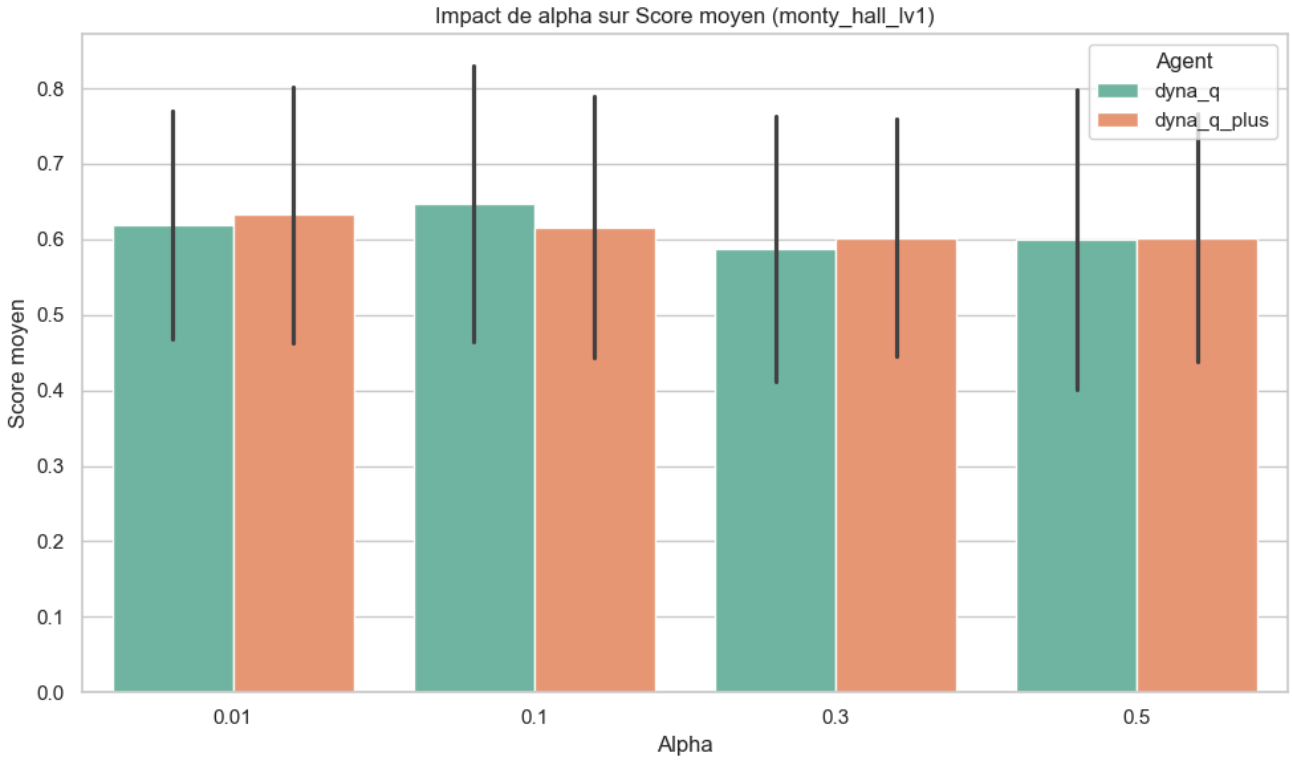

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



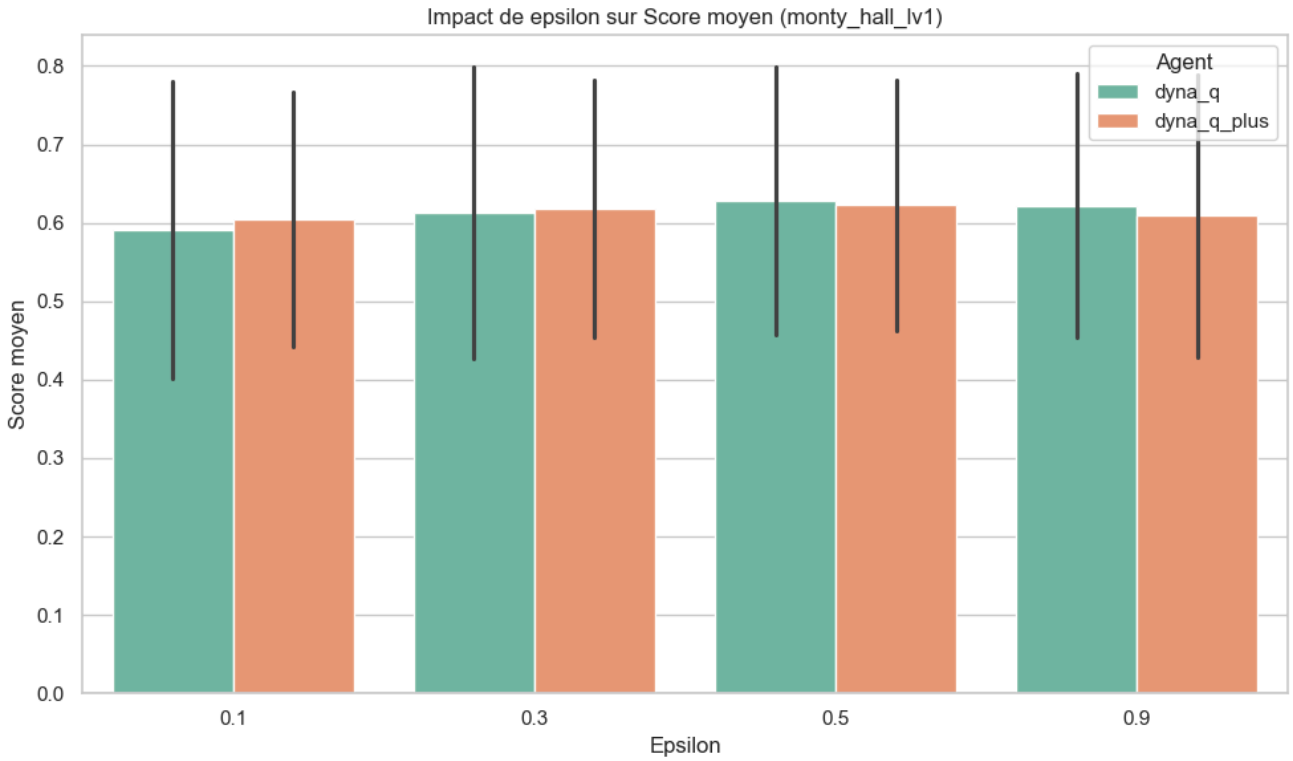

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



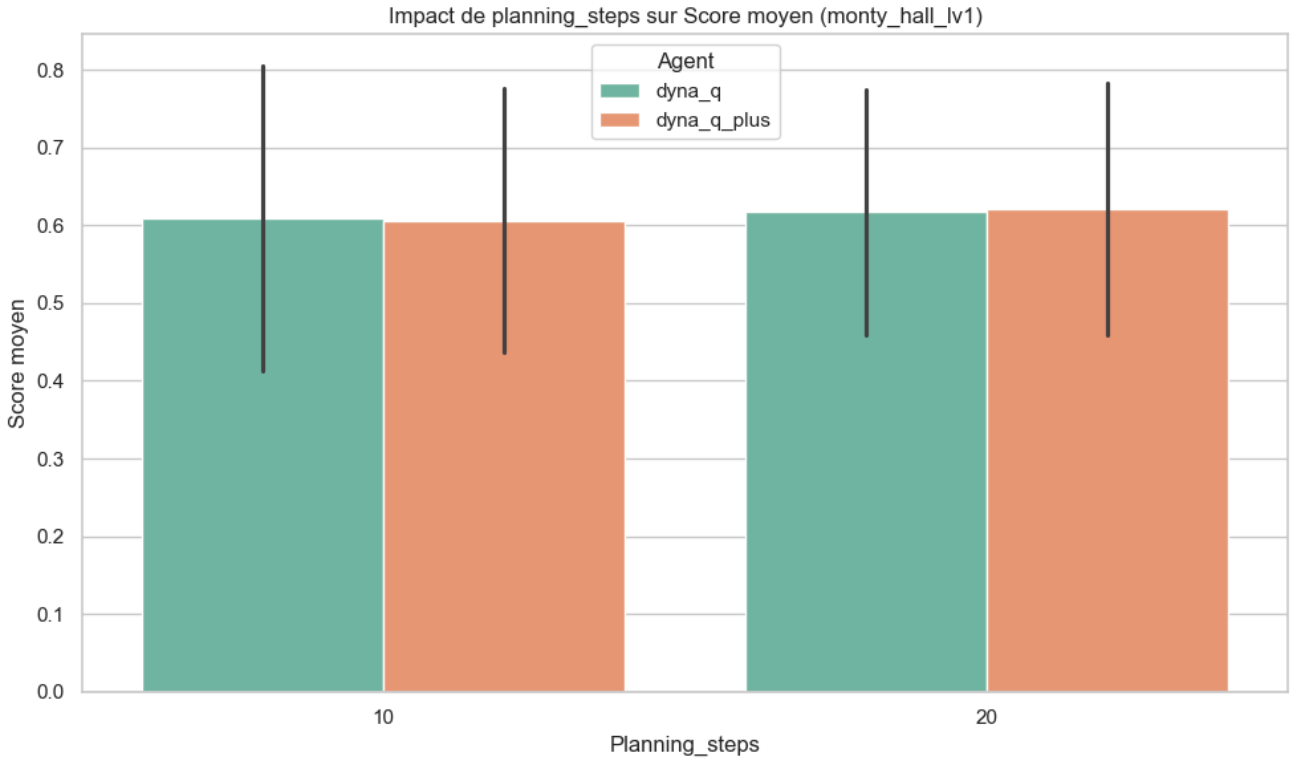

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



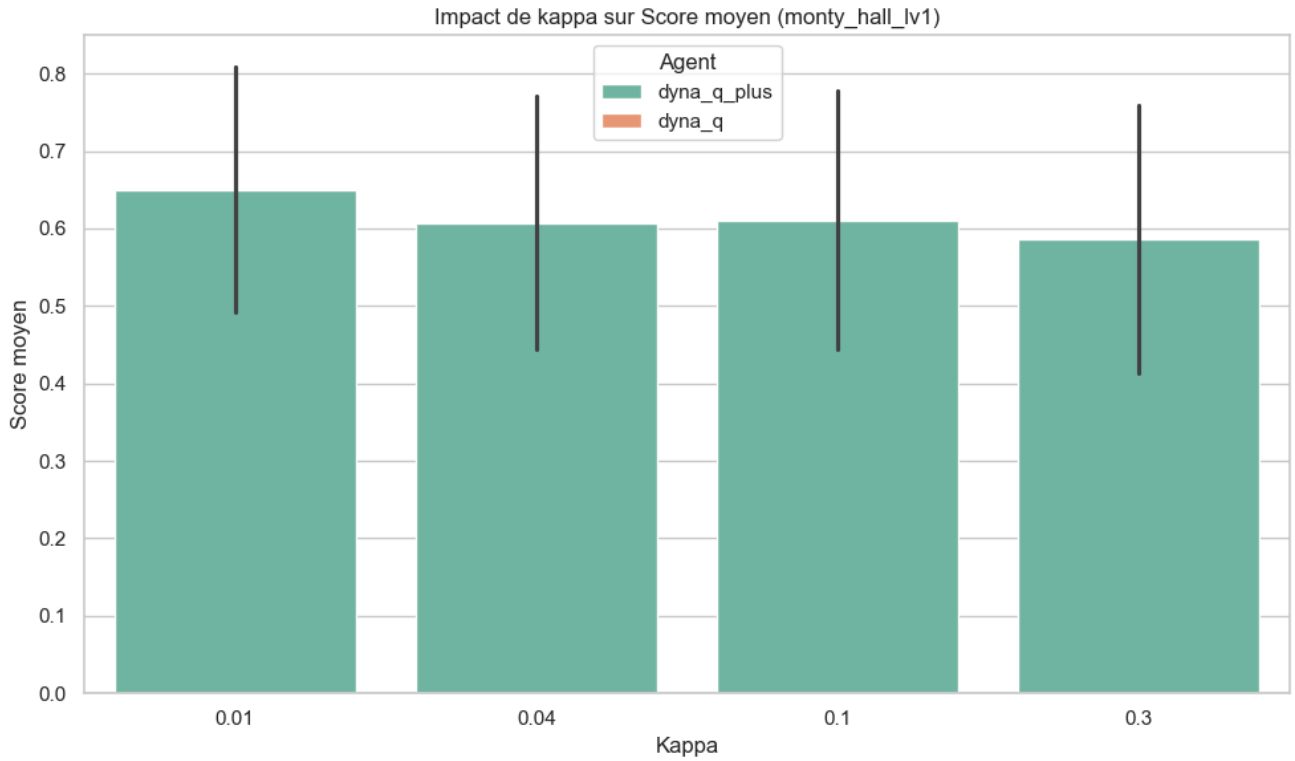

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



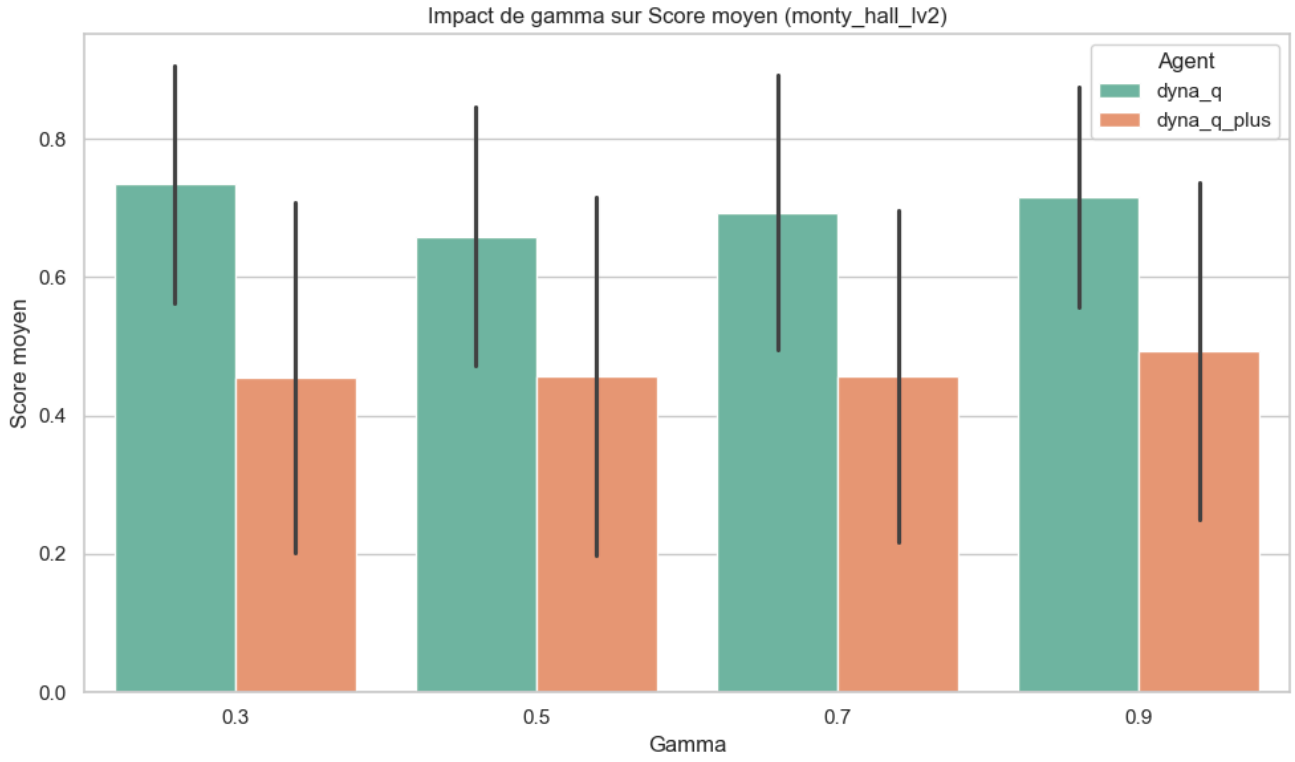

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



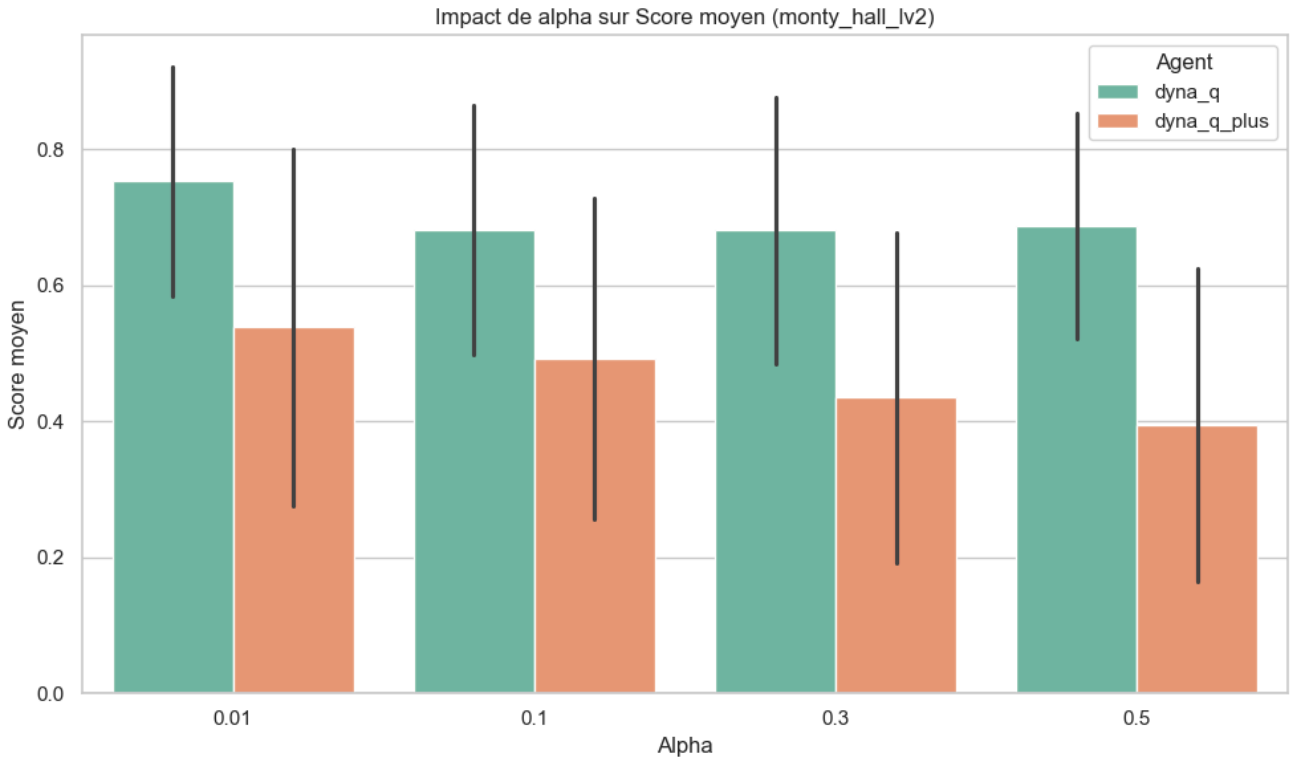

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



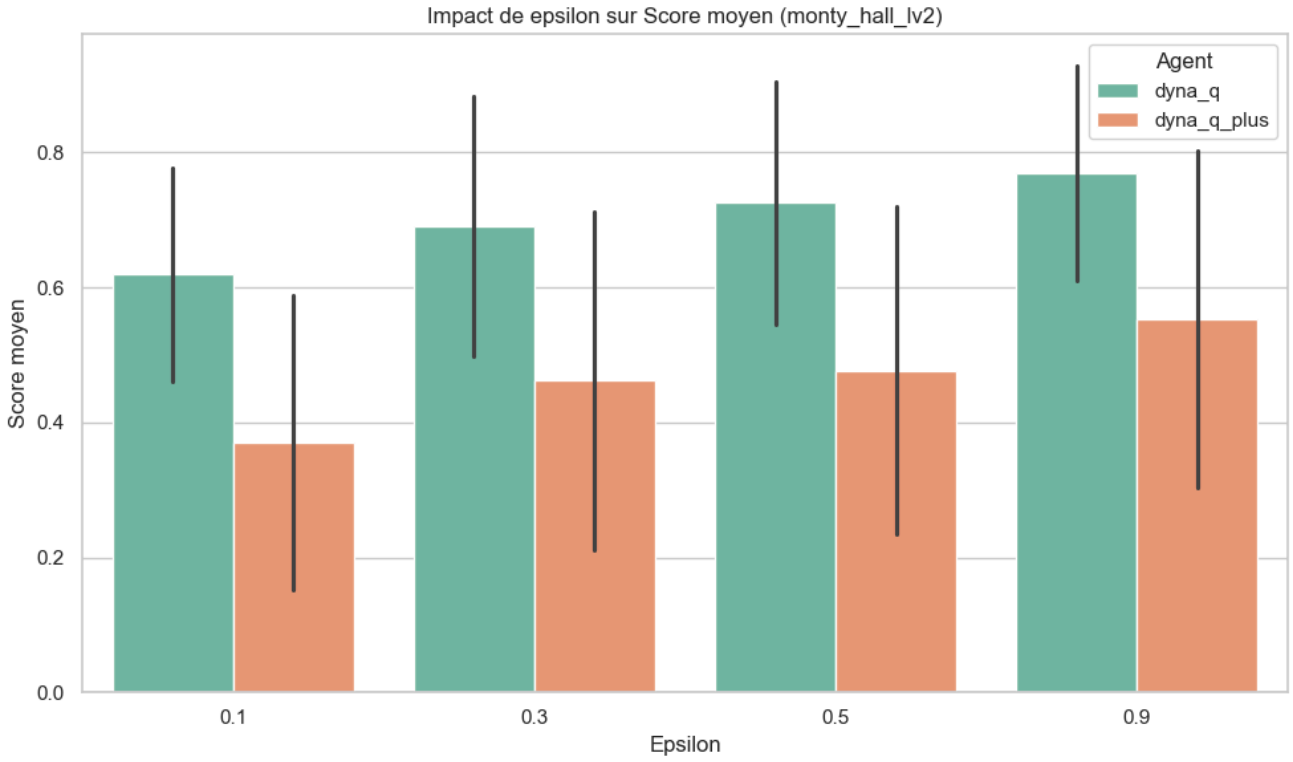

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



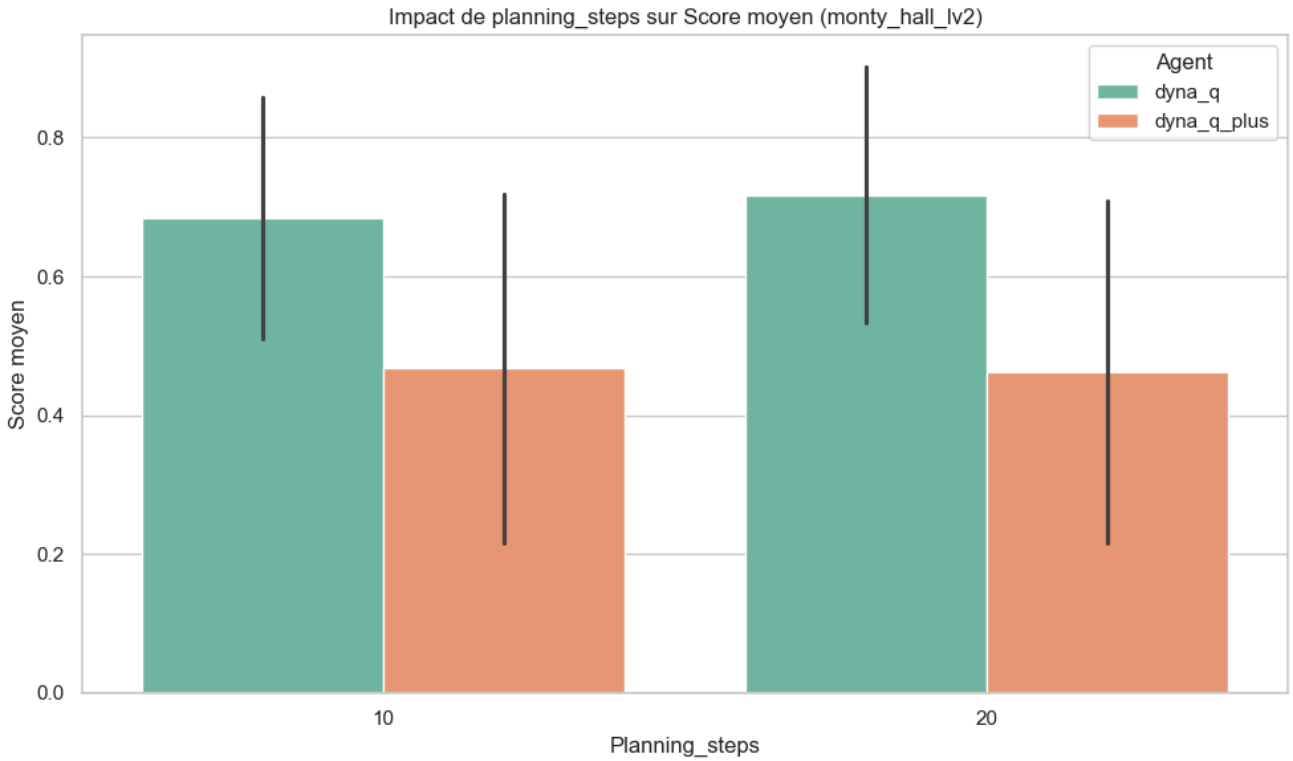

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



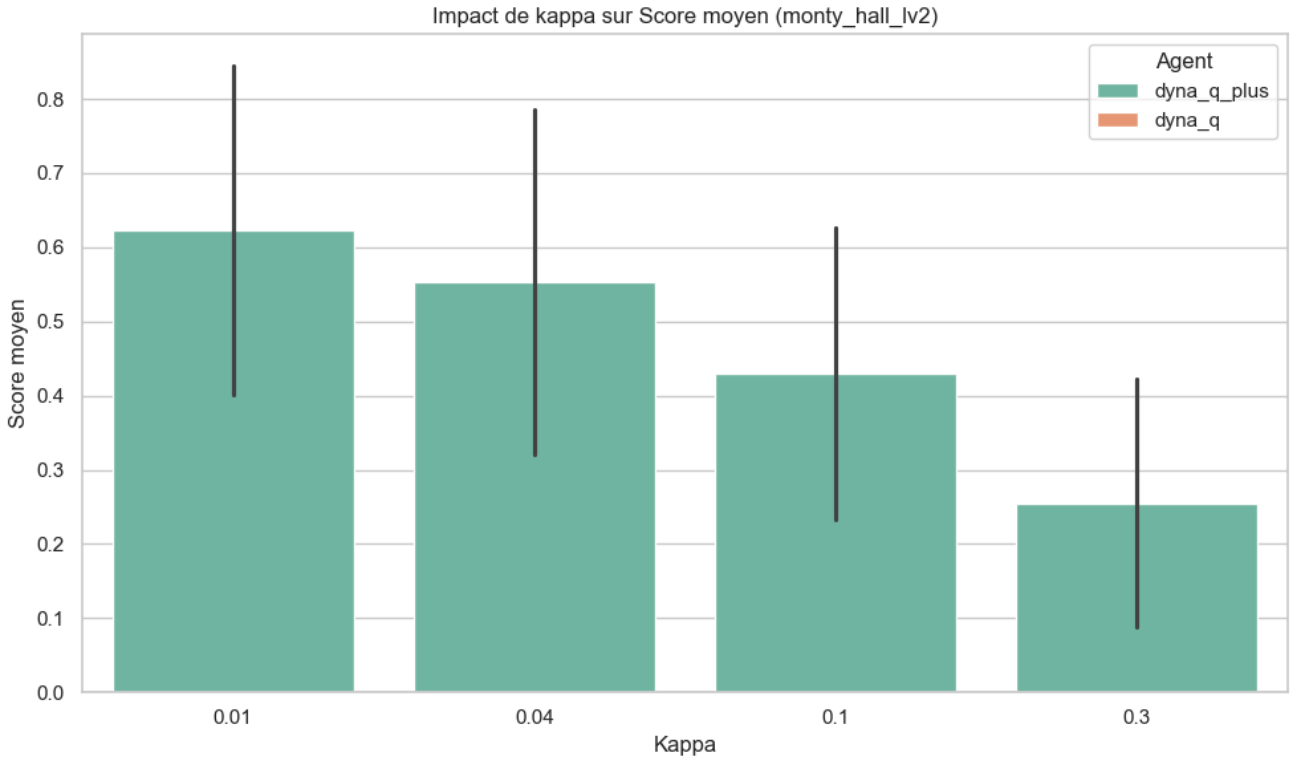

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



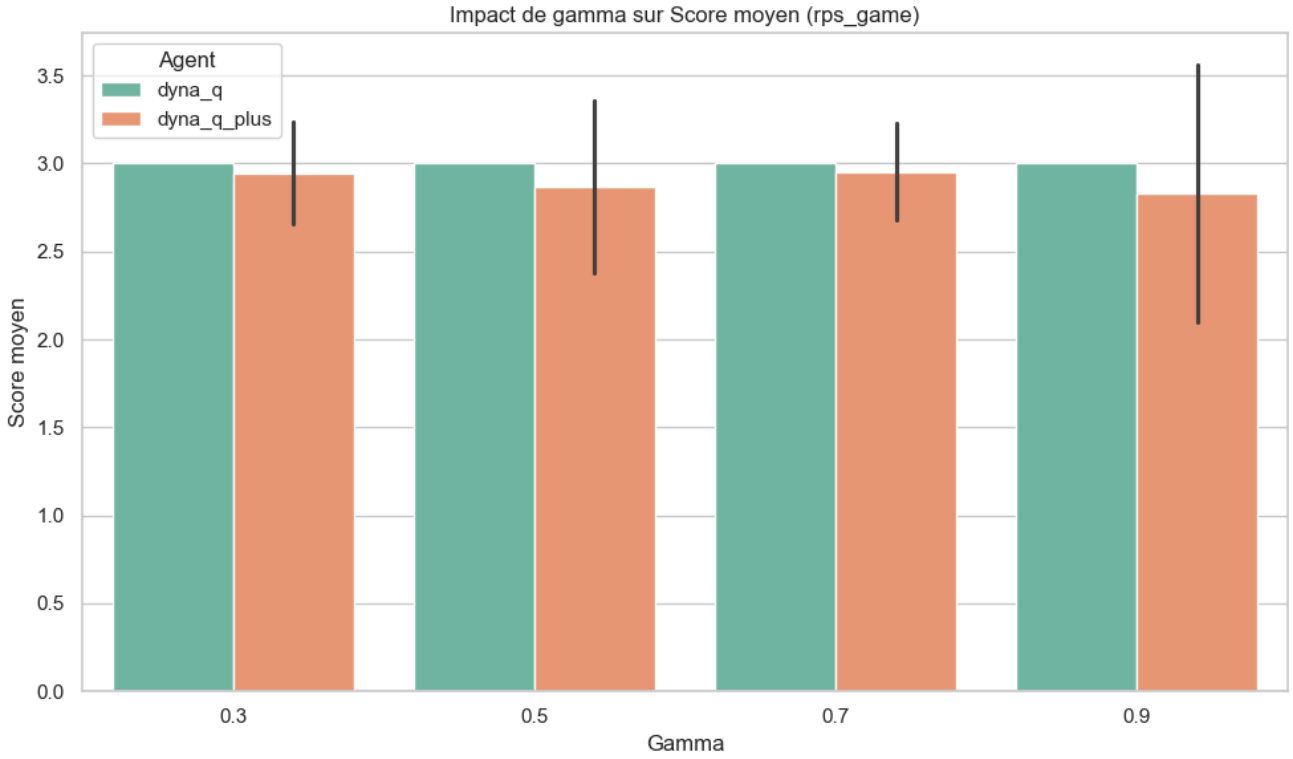

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



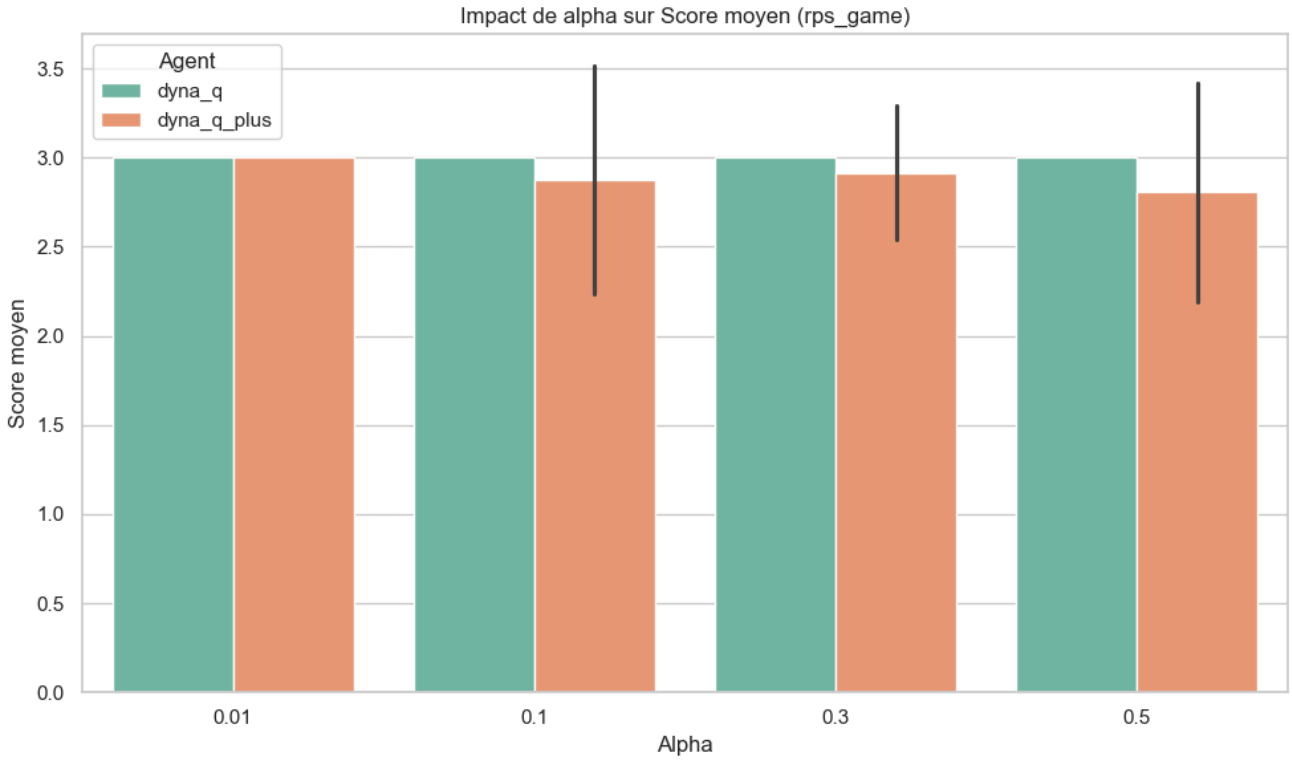

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



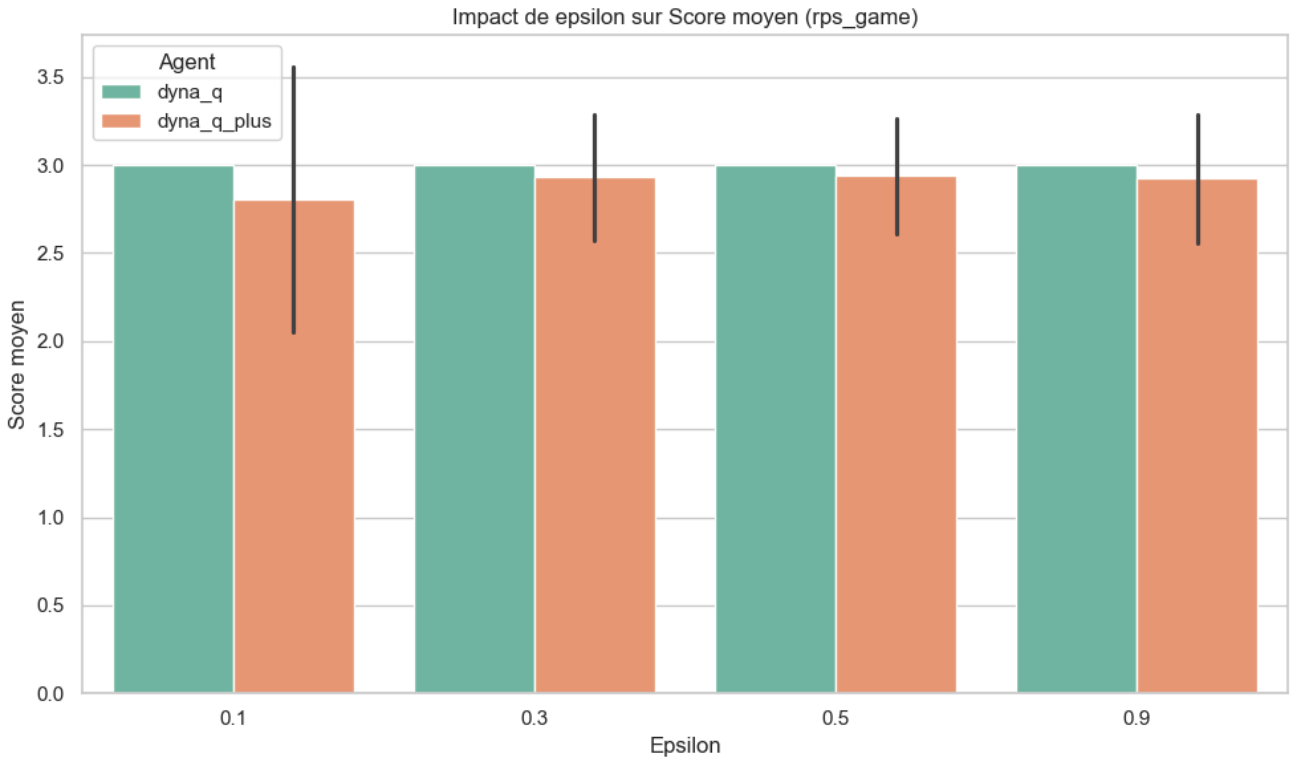

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



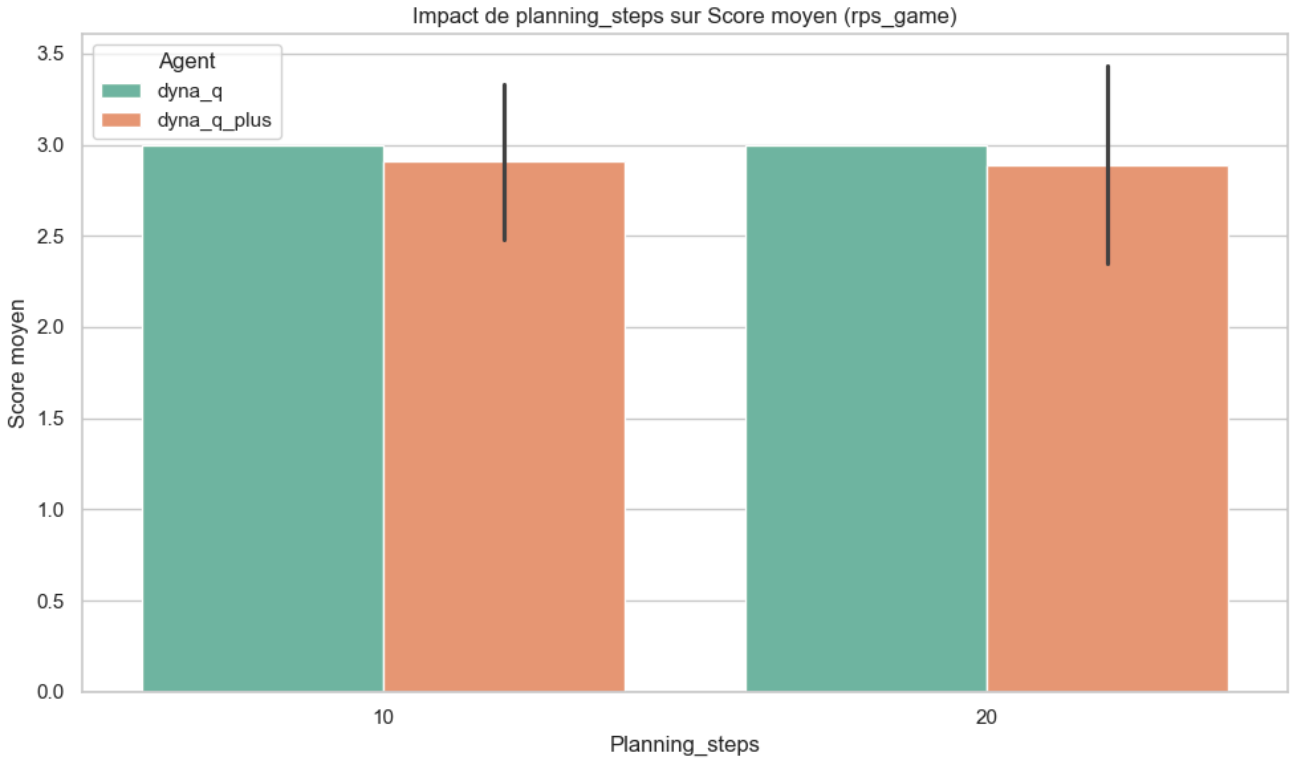

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



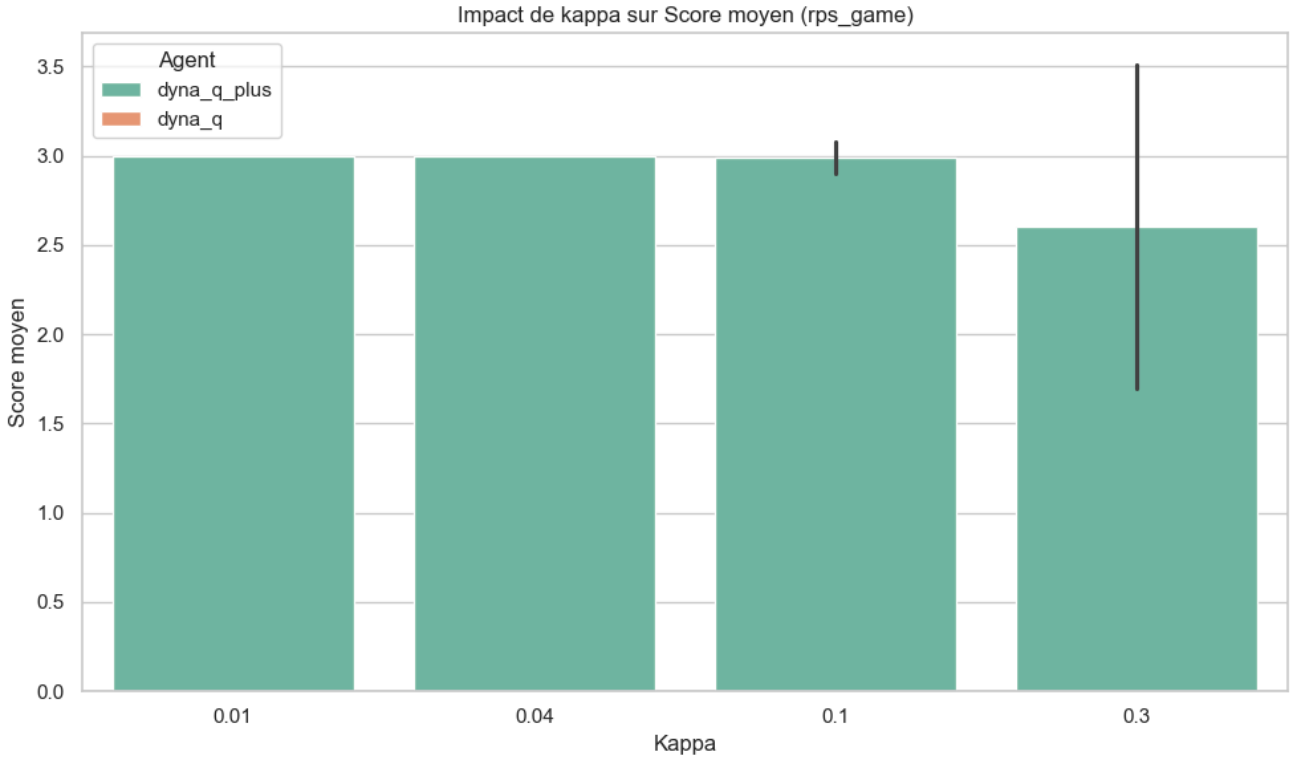

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



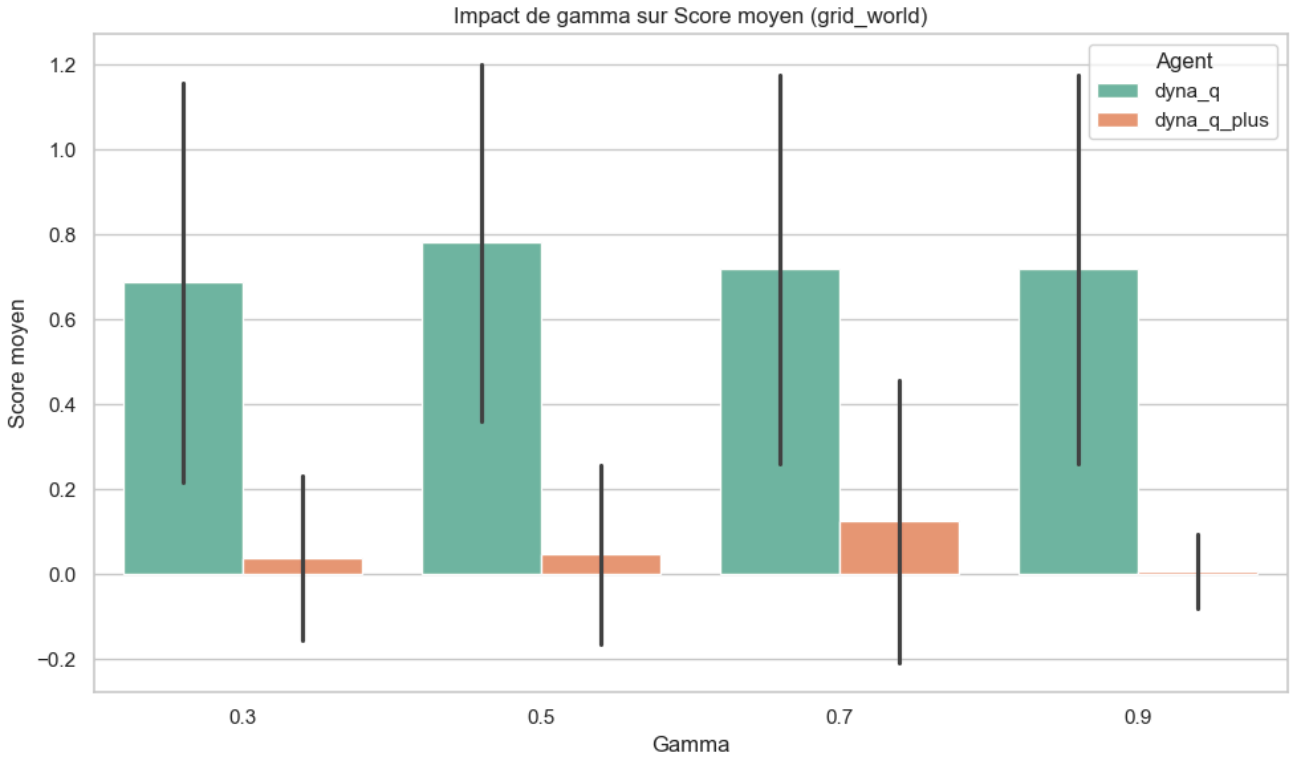

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



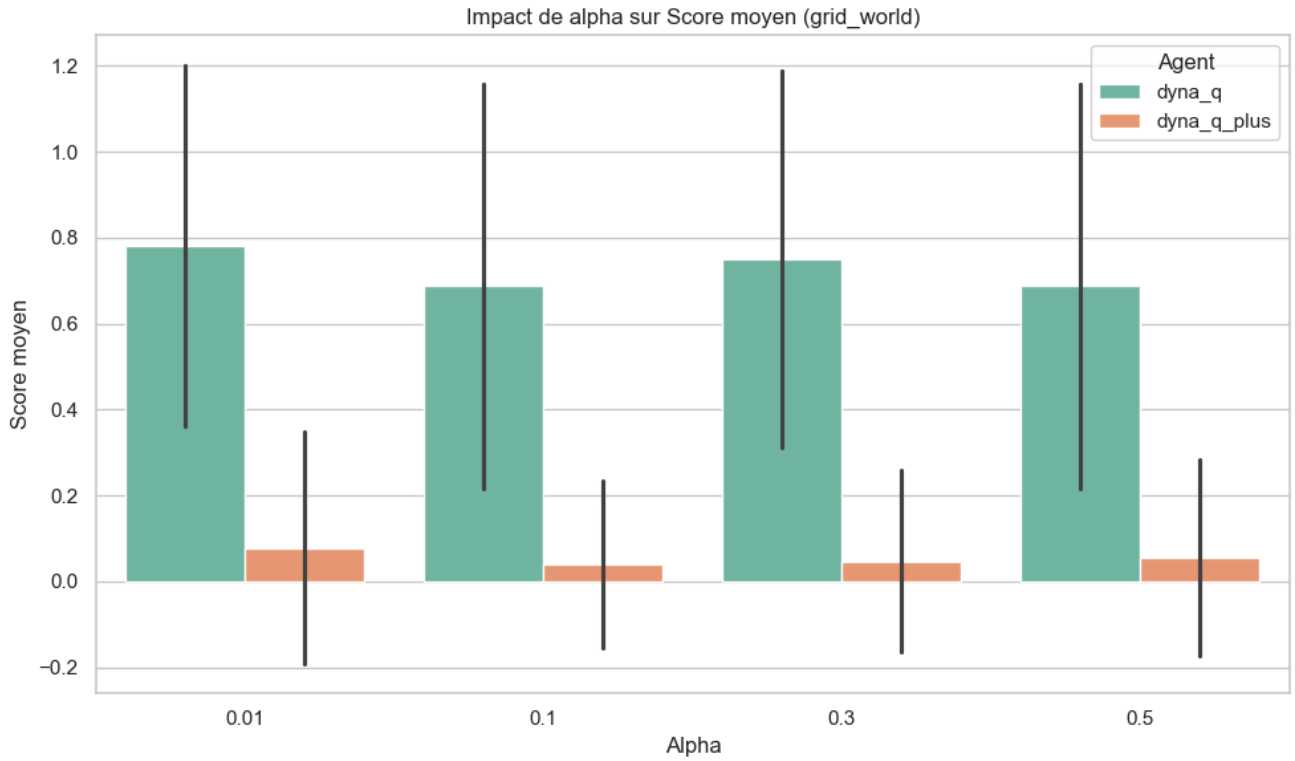

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



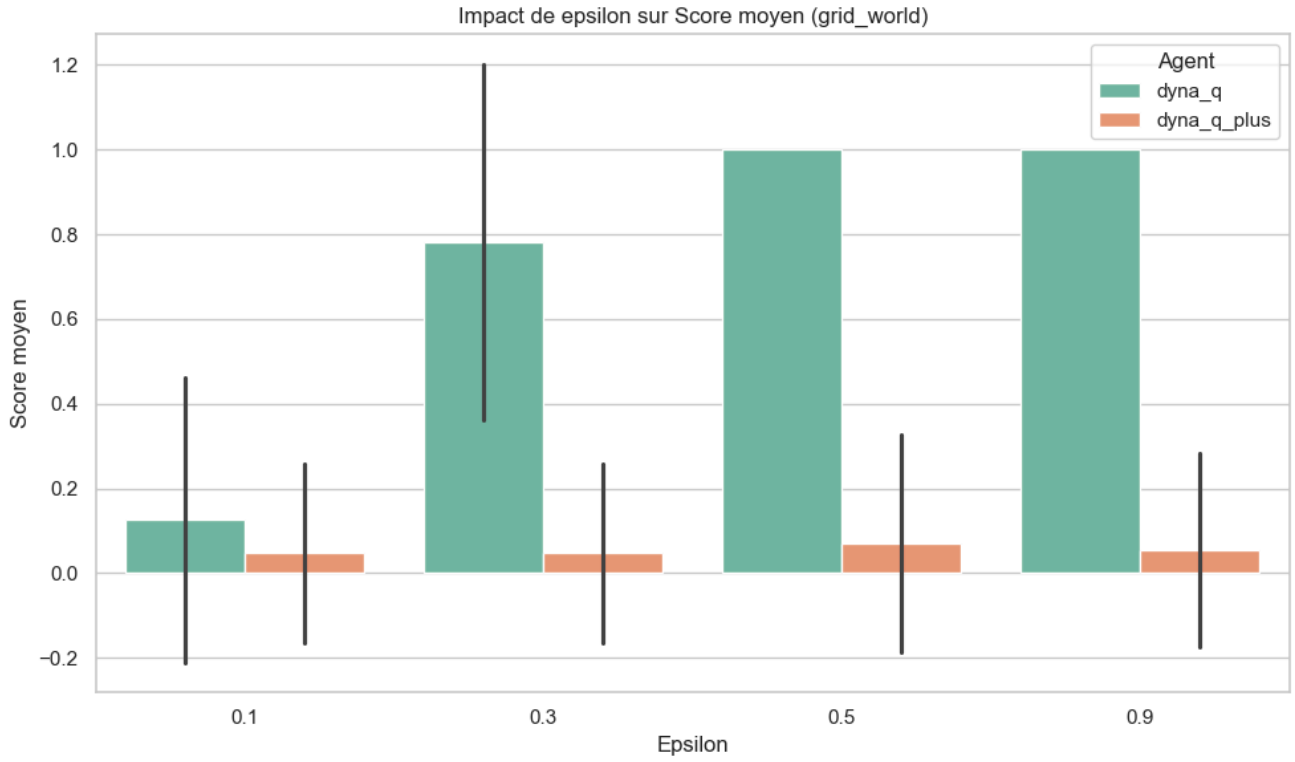

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



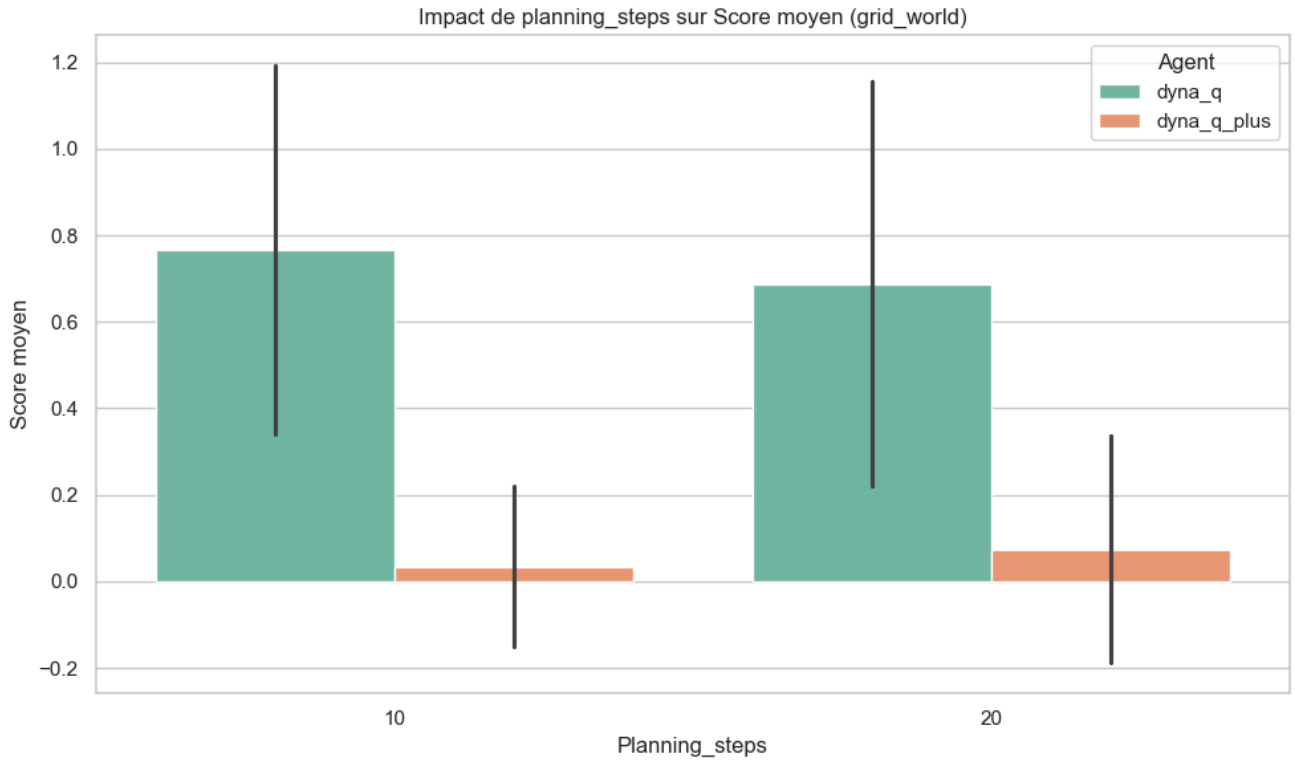

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



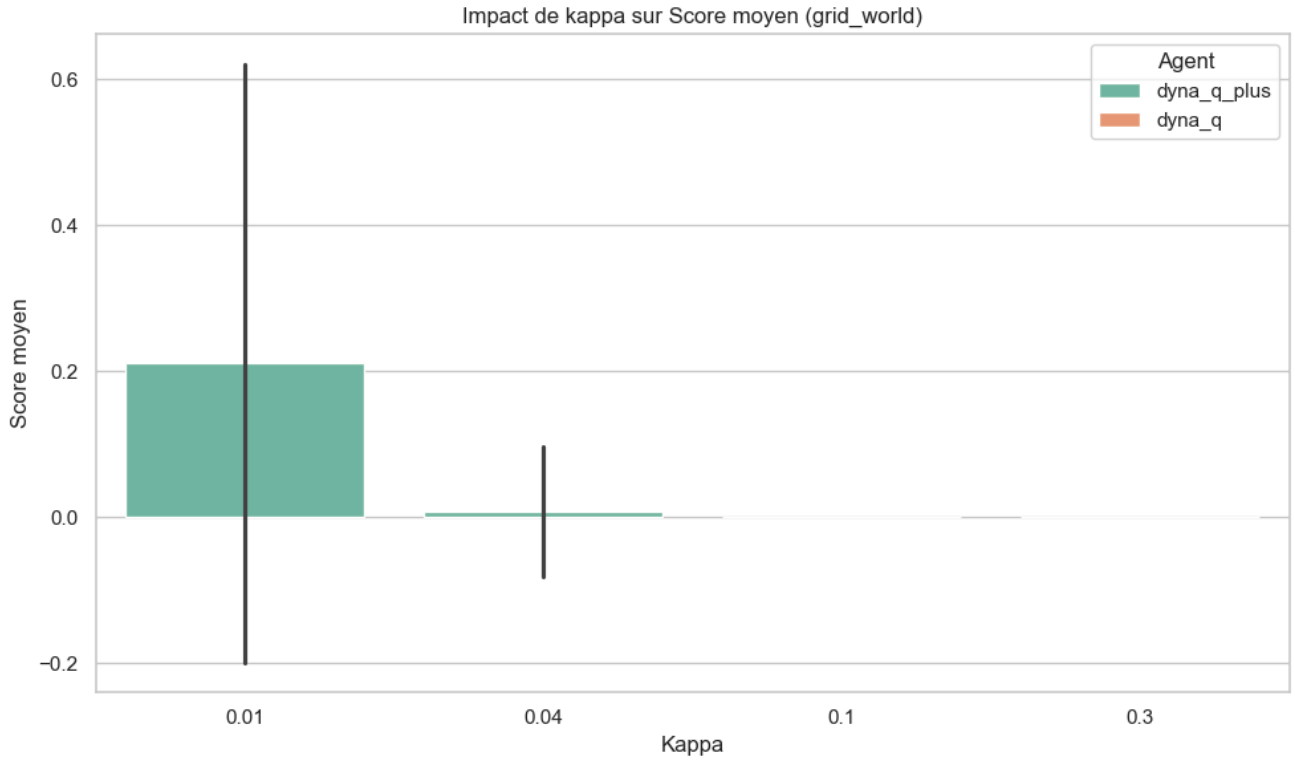

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```




```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```




```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\4175606820.py:56: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



Résumé des performances globales :

agent	mean_score	mean_steps	time
dyna_q	1.21	8.23	3.04
dyna_q_plus	0.95	26.34	2.07

Hyperparamètres de dyna q et dyna q plus GAMMAS = [0.3, 0.5, 0.7, 0.9] ALPHAS = [0.01, 0.1, 0.3 , 0.5] EPSILONS = [0.1, 0.3, 0.5, 0.9] PLANNING_STEPS = [10, 20] KAPPAS = [0.01, 0.04, 0.1, 0.3] Dans monty hall 2 et line world dyna q atteint un reward plus élevé mais en plus d'étape tandis que dyna q plus atteint un reward légèrement moins élevé mais en moins d'étape. les performance entre les deux sont sensiblement identique sur le reste des environnement. LINE WORLD si on accorde trop d'importance au recompense future (gamme élevé)ou un apprentissage trop rapide on obtiens un score moins élévé. E n'affecte pas beaucoup le score plus le bonus d'exploration kappa est grand moins dyna q plus est performant

MONTY HALL LV1 ici la variation des hyperparamètres à une légère influence sur le score des agents

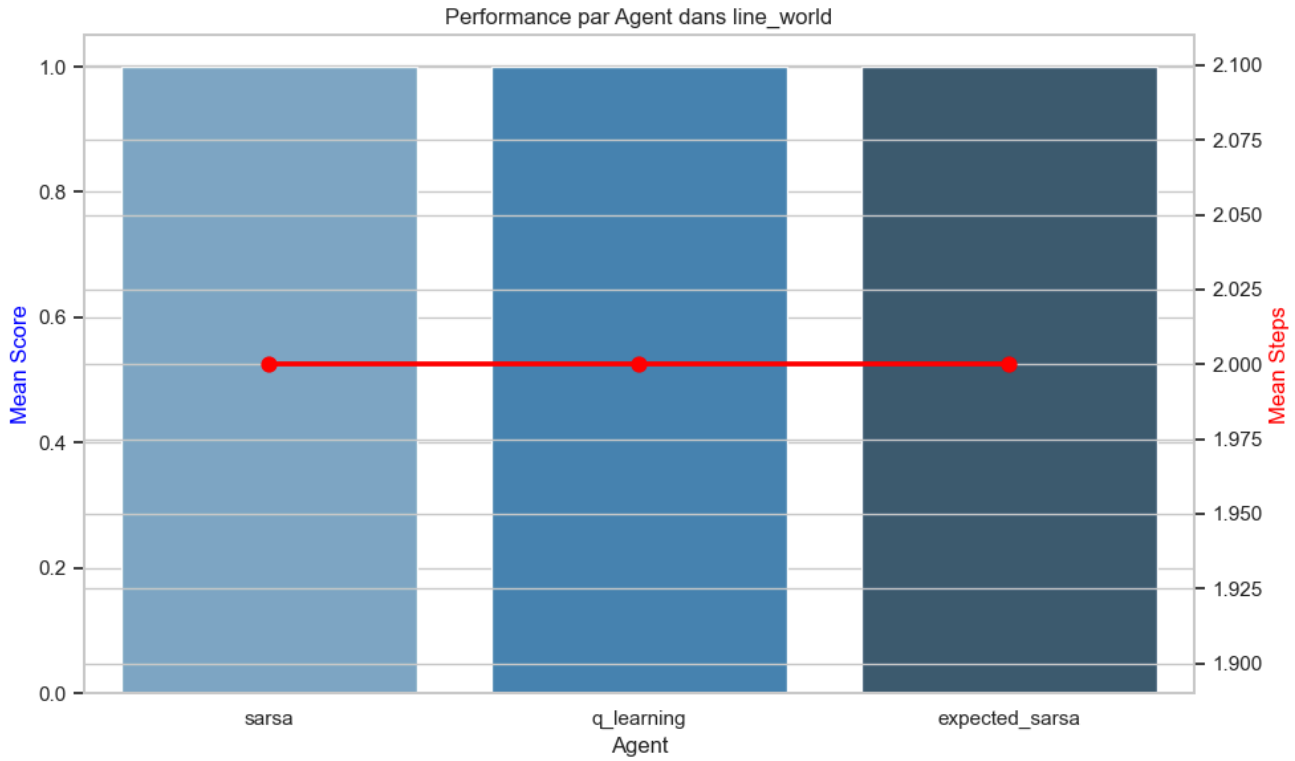
meilleur paramètre des PM dans line world {y : 0.5, a : 0.01, € : 0.9, planning_steps : 10, k : 0.01}

FAMILLE TEMPORAL DIFFERENCE METHODS

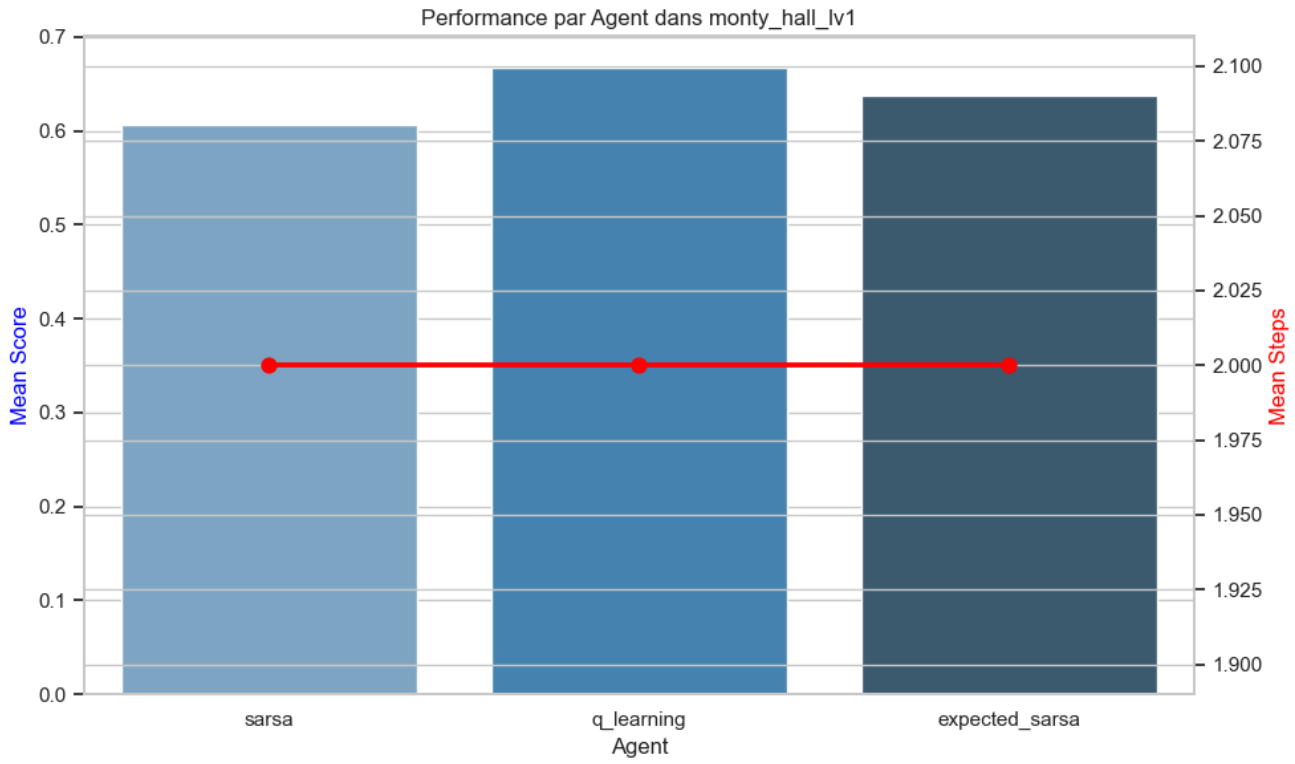
```
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

file_path = "Reports/td_agents_comparison.xlsx"
df = pd.read_excel(file_path)

# === GRAPHE 1 : Mean Score + Mean Steps par agent et environnement ===
envs = df['env'].unique()

for env in envs:
    env_df = df[df['env'] == env]

    fig, ax1 = plt.subplots()

    # Score moyen
    sns.barplot(
        data=env_df,
        x="agent",
        y="mean_score",
        ci=None,
        ax=ax1,
        palette="Blues_d"
    )
    ax1.set_ylabel("Mean Score", color="blue")
    ax1.set_title(f"Performance par Agent dans {env}")
    ax1.set_xlabel("Agent")

    # Mean Steps sur axe secondaire
    ax2 = ax1.twinx()
    sns.pointplot(
        data=env_df,
        x="agent",
        y="mean_steps",
        color="red",
        ax=ax2
    )
    ax2.set_ylabel("Mean Steps", color="red")

    plt.tight_layout()
    plt.show()
```

```python
# === GRAPHE 2 : Impact des hyperparamètres sur le score moyen ===
hyperparams = ["gamma", "alpha", "epsilon"]

for env in envs:
    env_df = df[df["env"] == env]
    for param in hyperparams:
        if param in df.columns:
            plt.figure()
            sns.barplot(
                data=env_df,
                x=param,
                y="mean_score",
                hue="agent",
                ci="sd",
                palette="Set2"
            )
            plt.title(f"Score moyen des agents sur l'environnement {env} selon {param}")
            plt.ylabel("Score moyen")
            plt.xlabel(param.capitalize())
            plt.grid(True, axis="y")
            plt.legend(title="Agent")
            plt.tight_layout()
            plt.show()

# === Résumé des performances par agent ===
summary = df.groupby("agent")[["mean_score", "mean_steps", "time"]].mean().round(2)
summary = summary.sort_values(by="mean_score", ascending=False)

from IPython.display import display
print("\nRésumé des performances moyennes :")
display(summary)
```

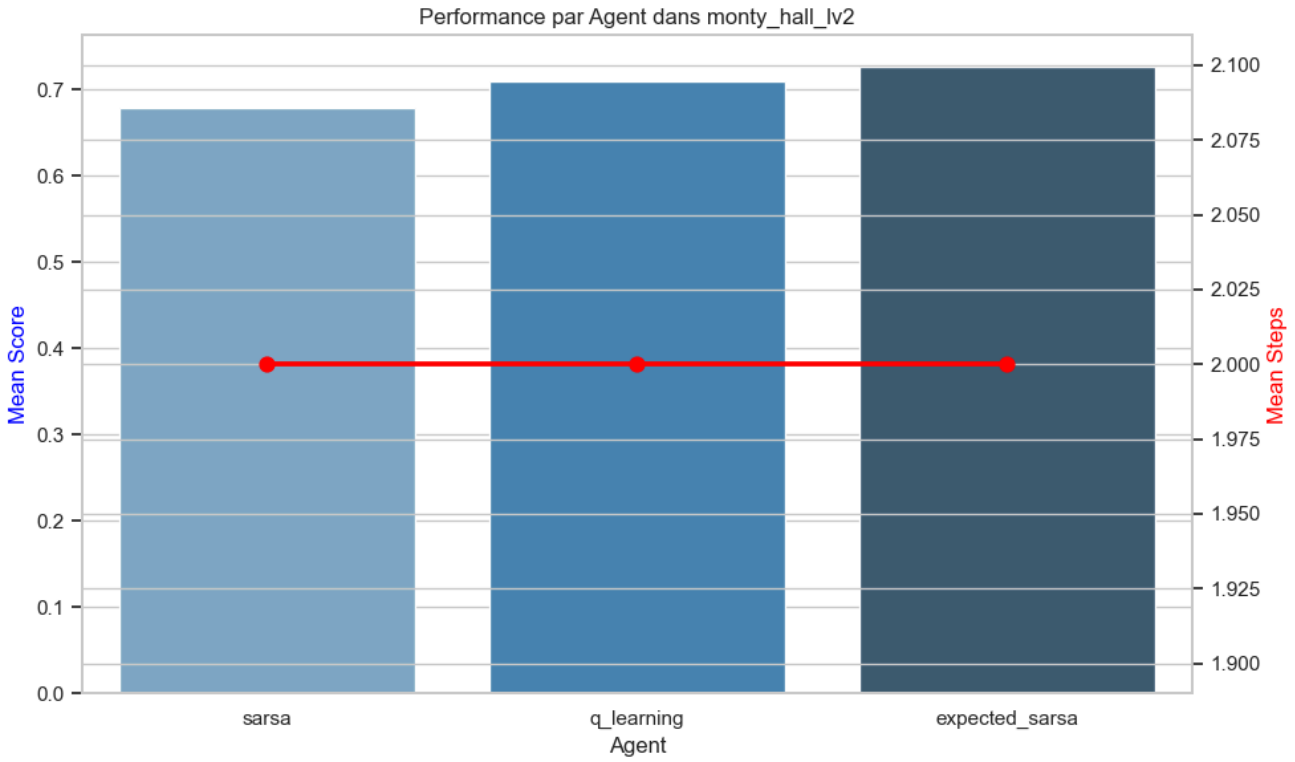
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



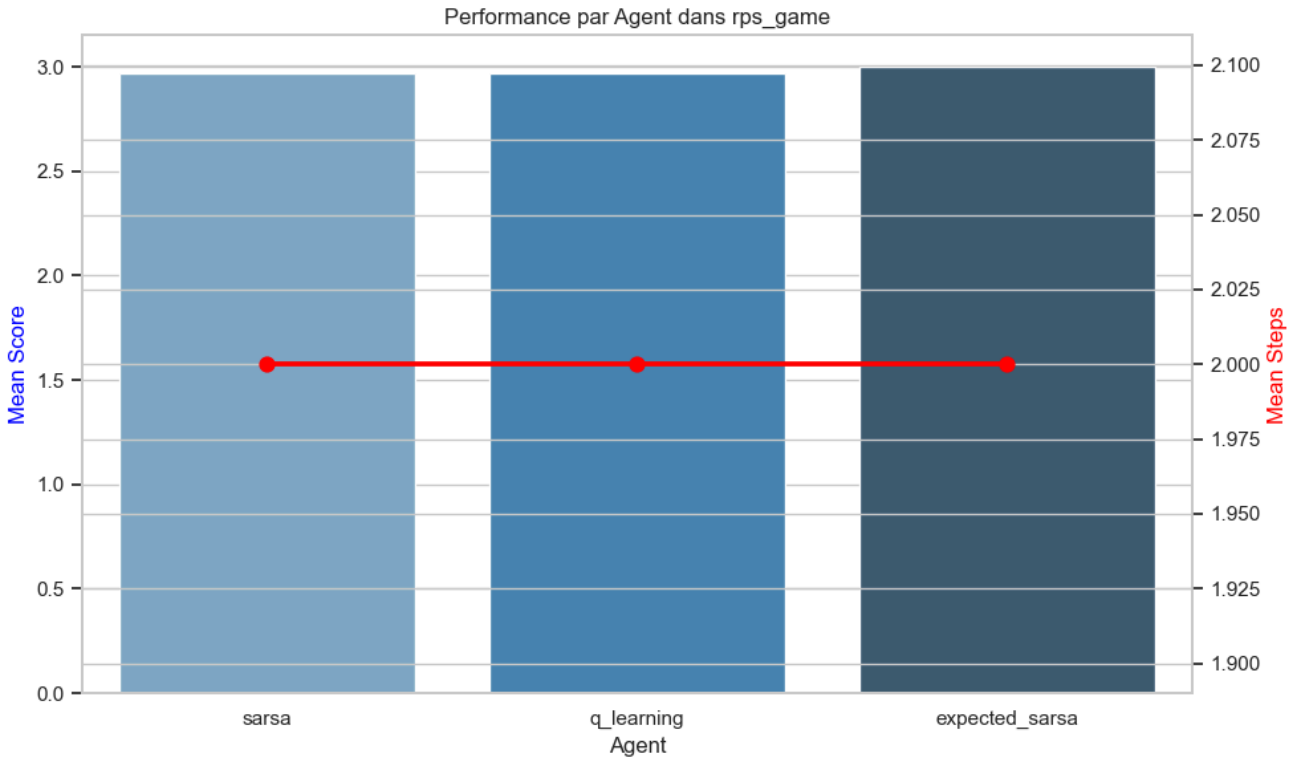

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



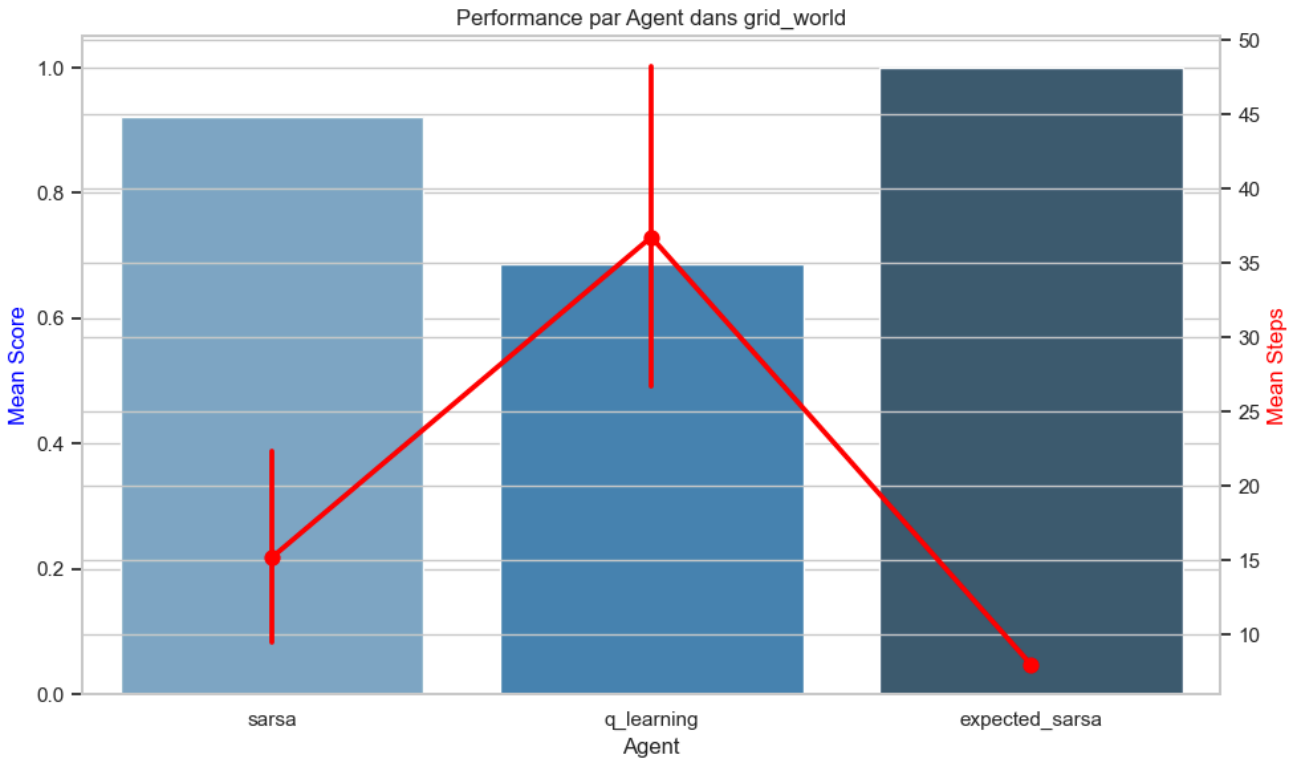
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



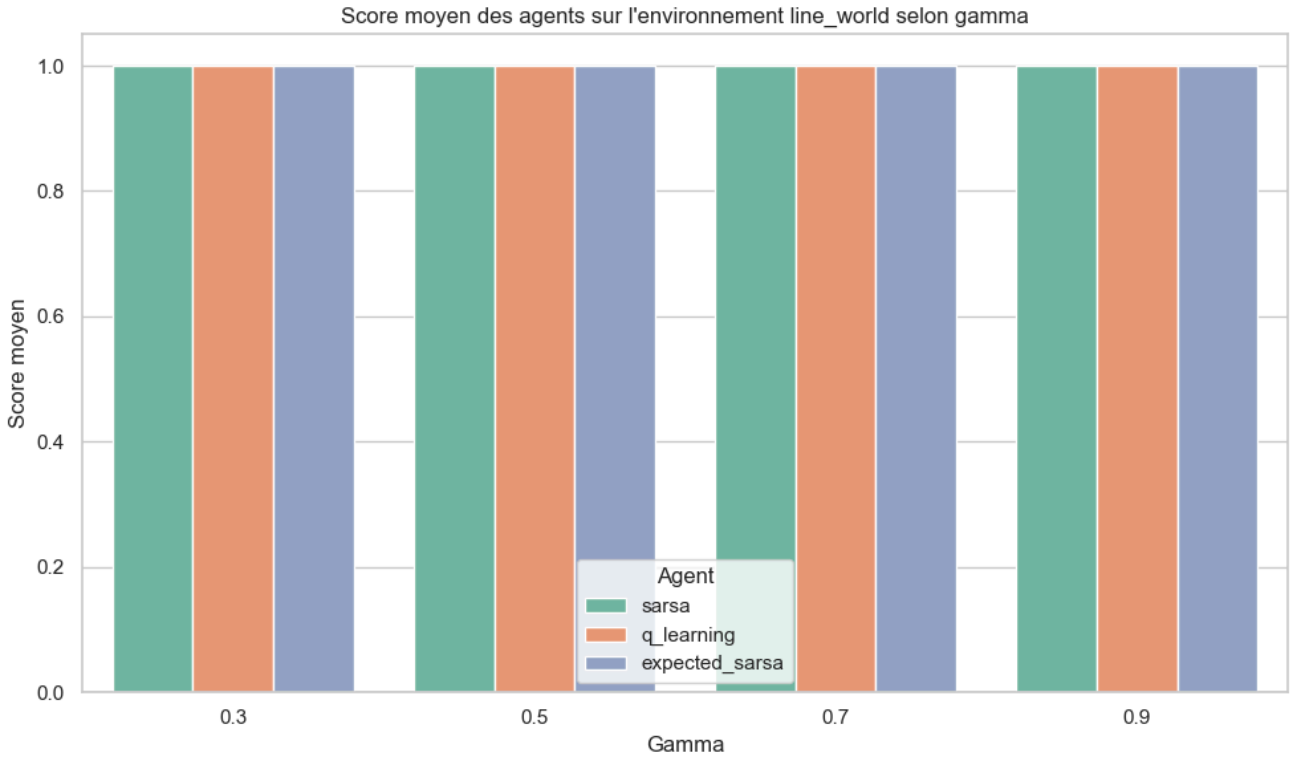

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



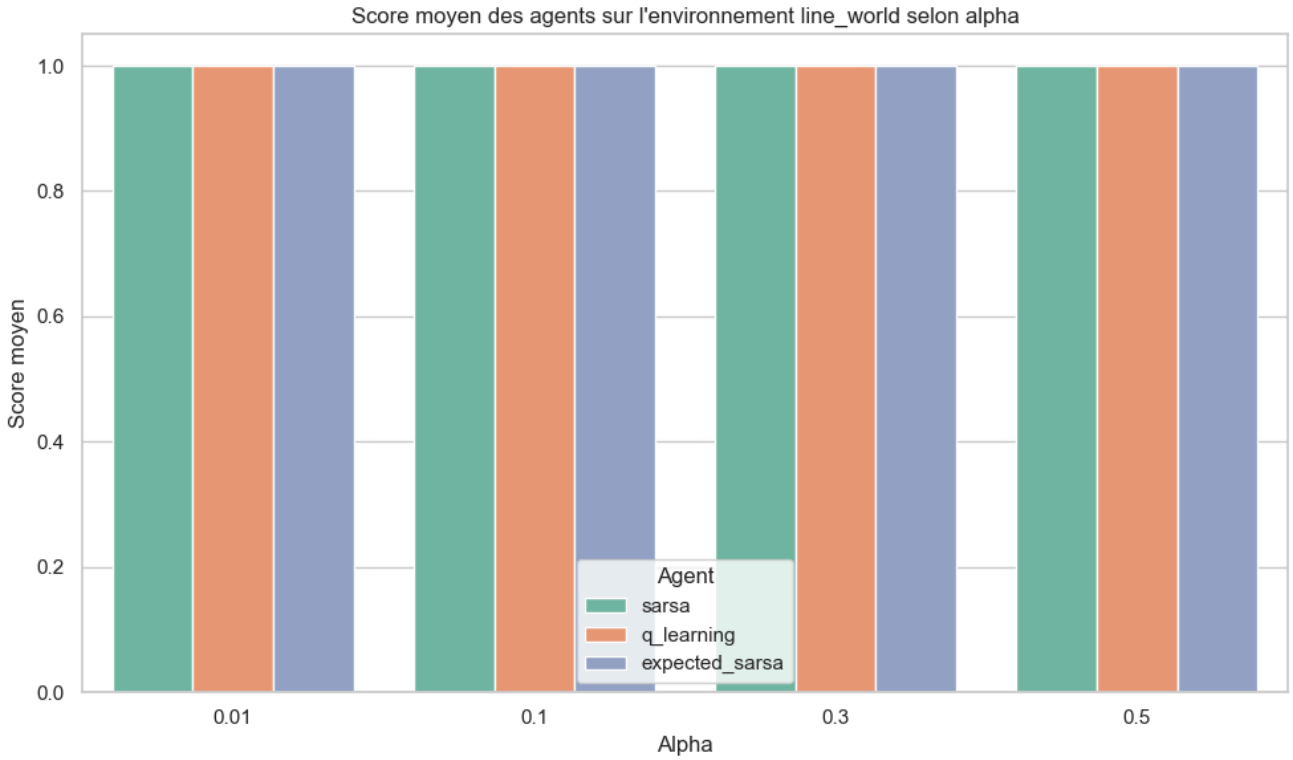

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.barplot(
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:16: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. As
sign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(
```



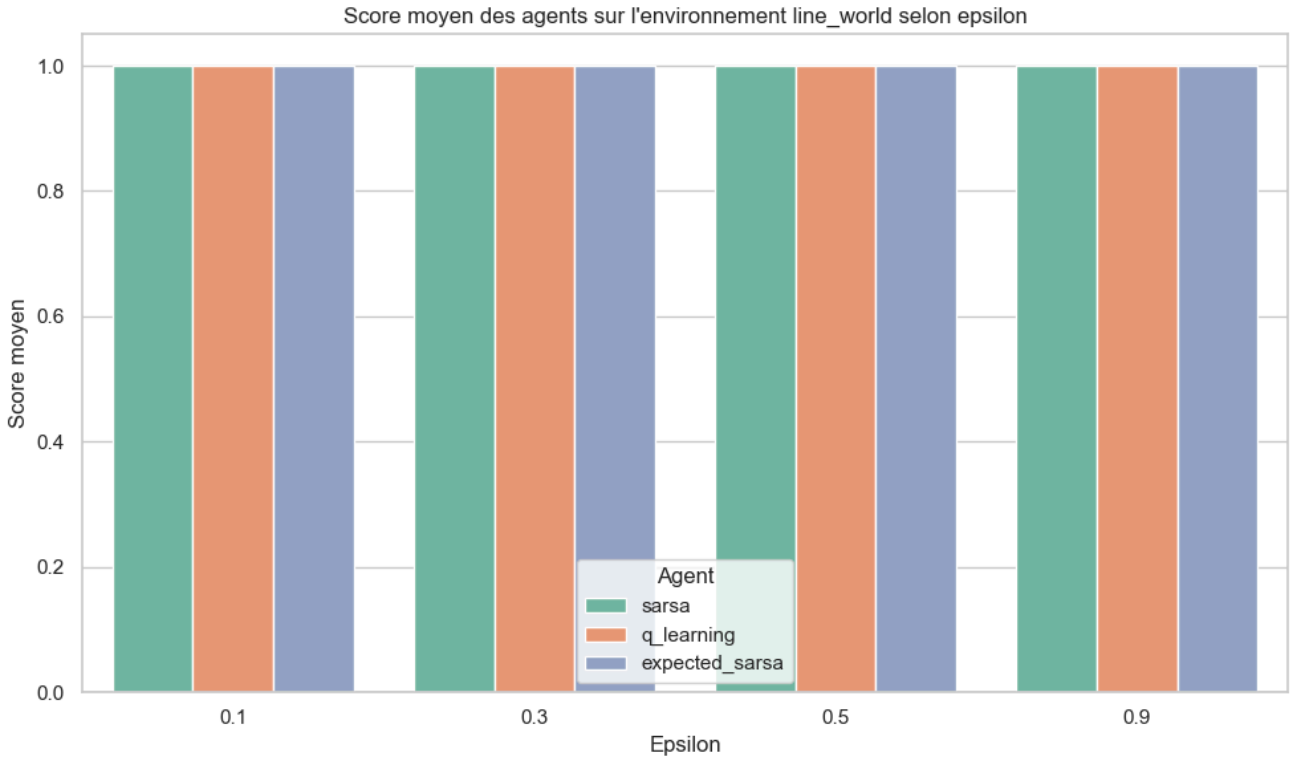
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



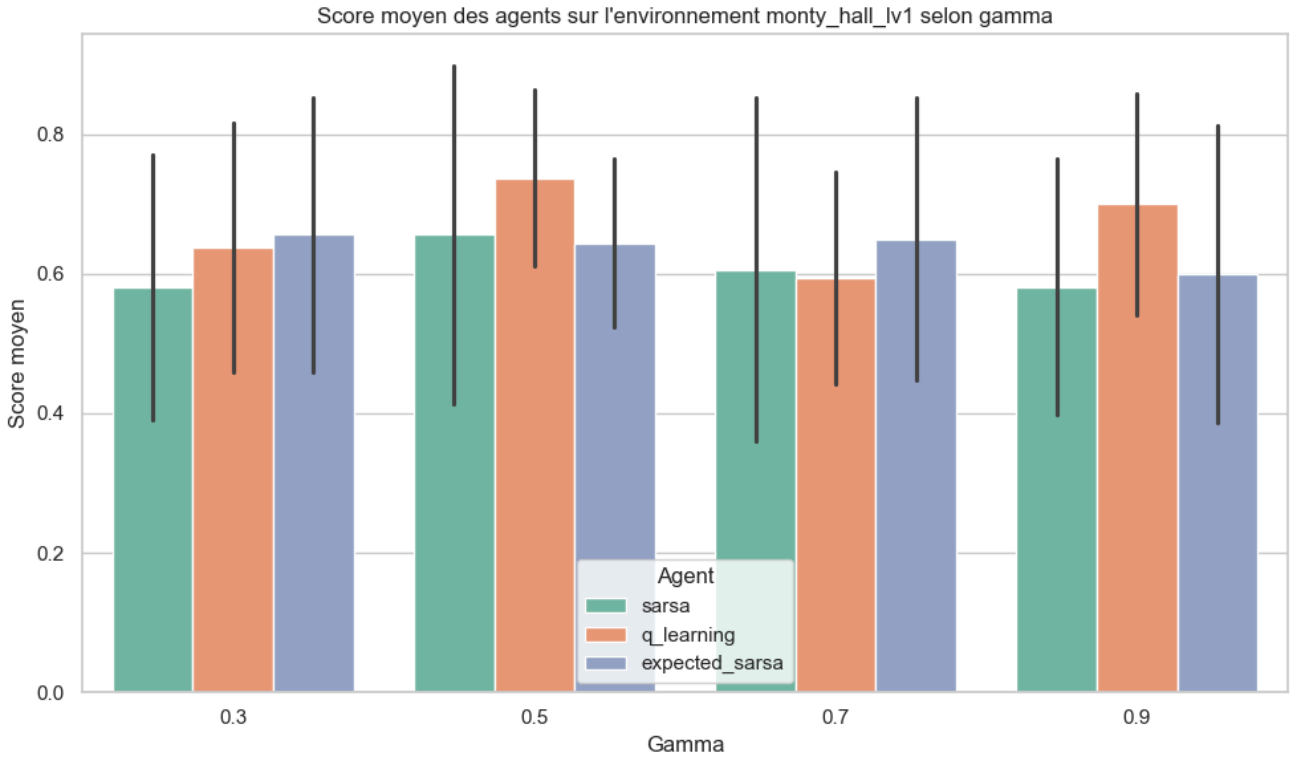

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



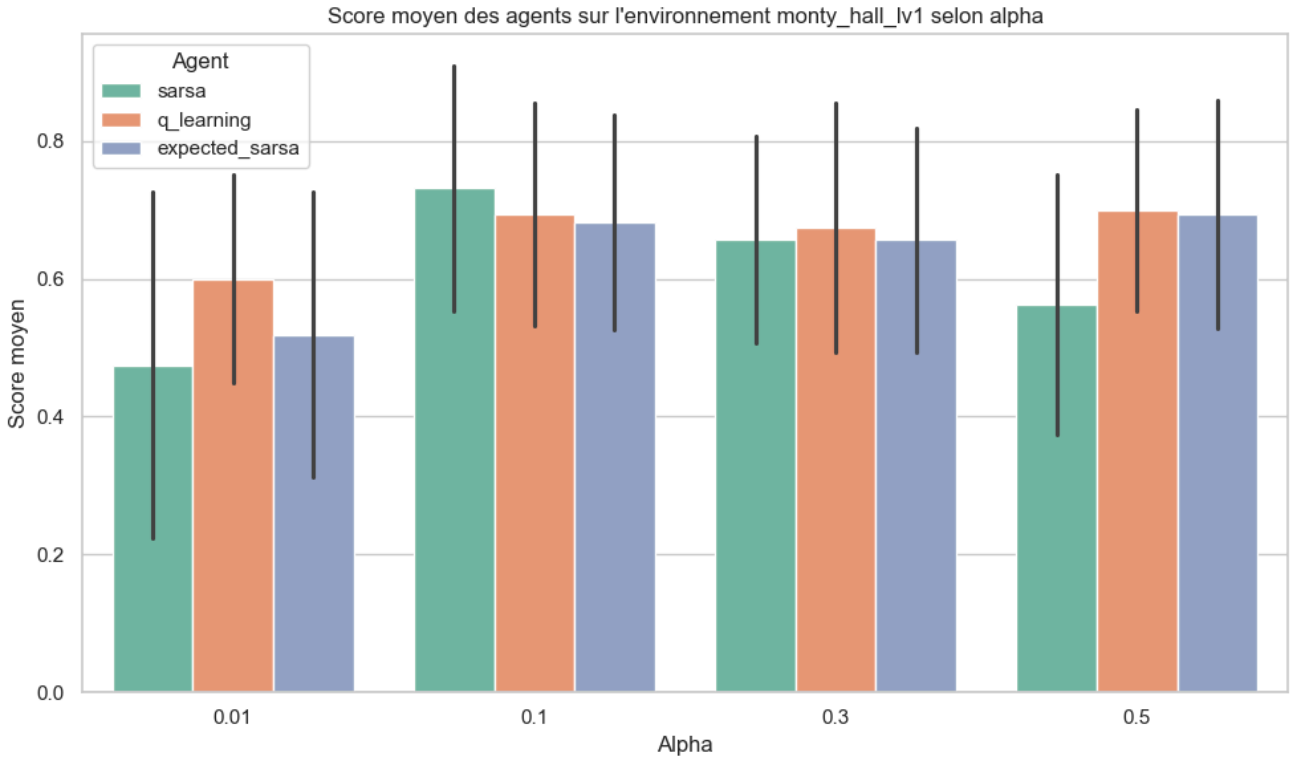
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



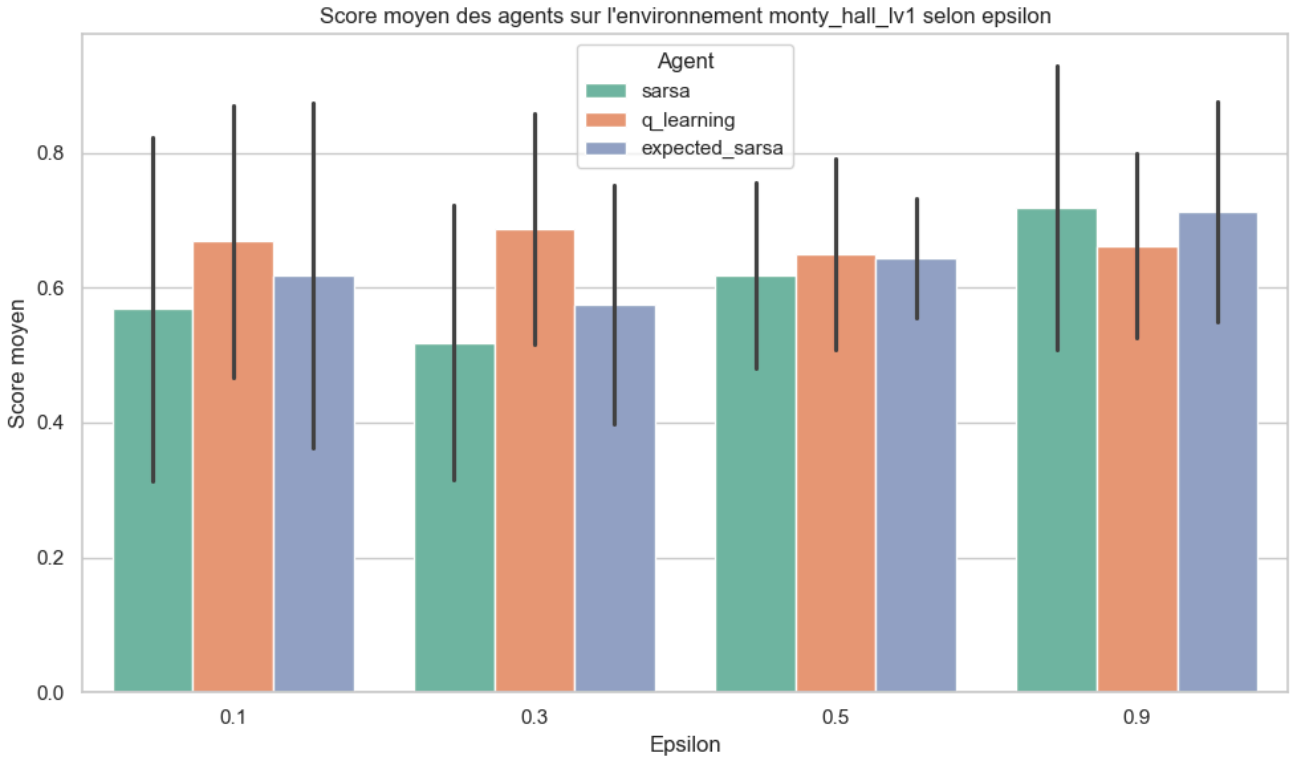

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



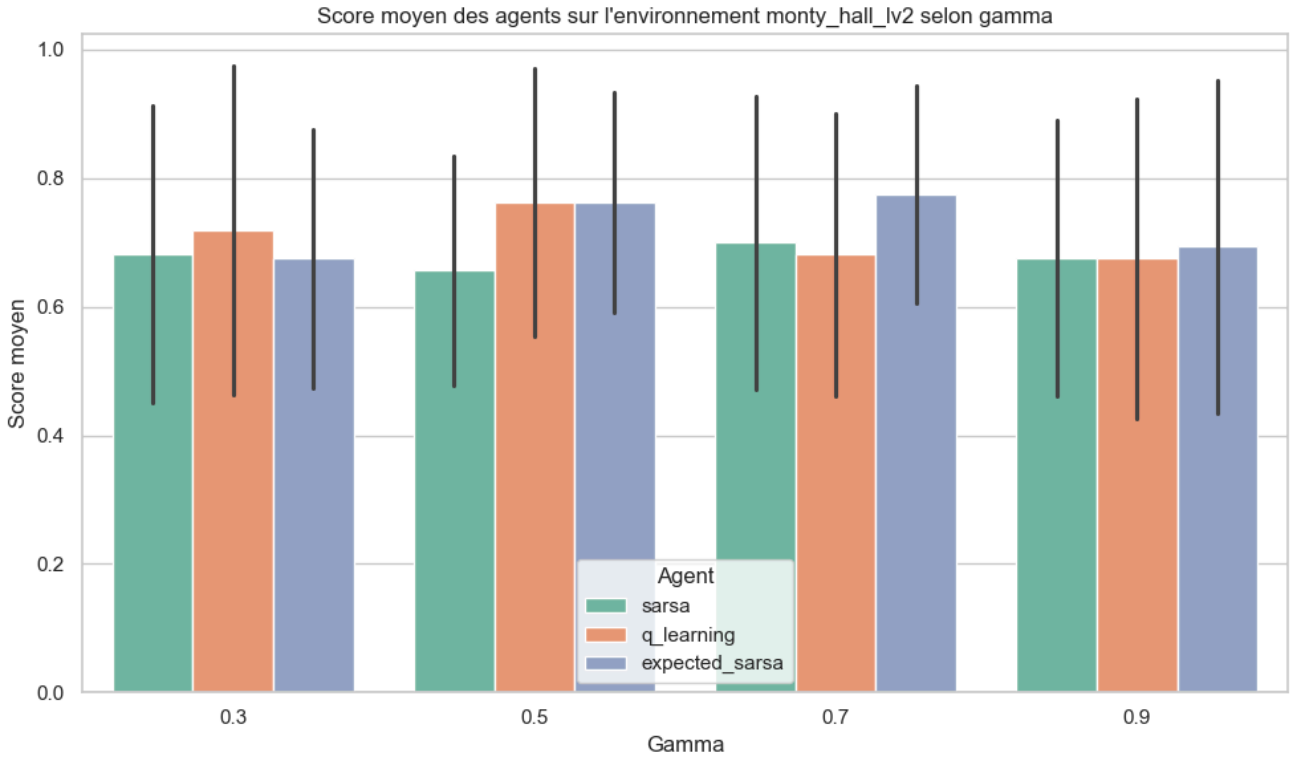
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



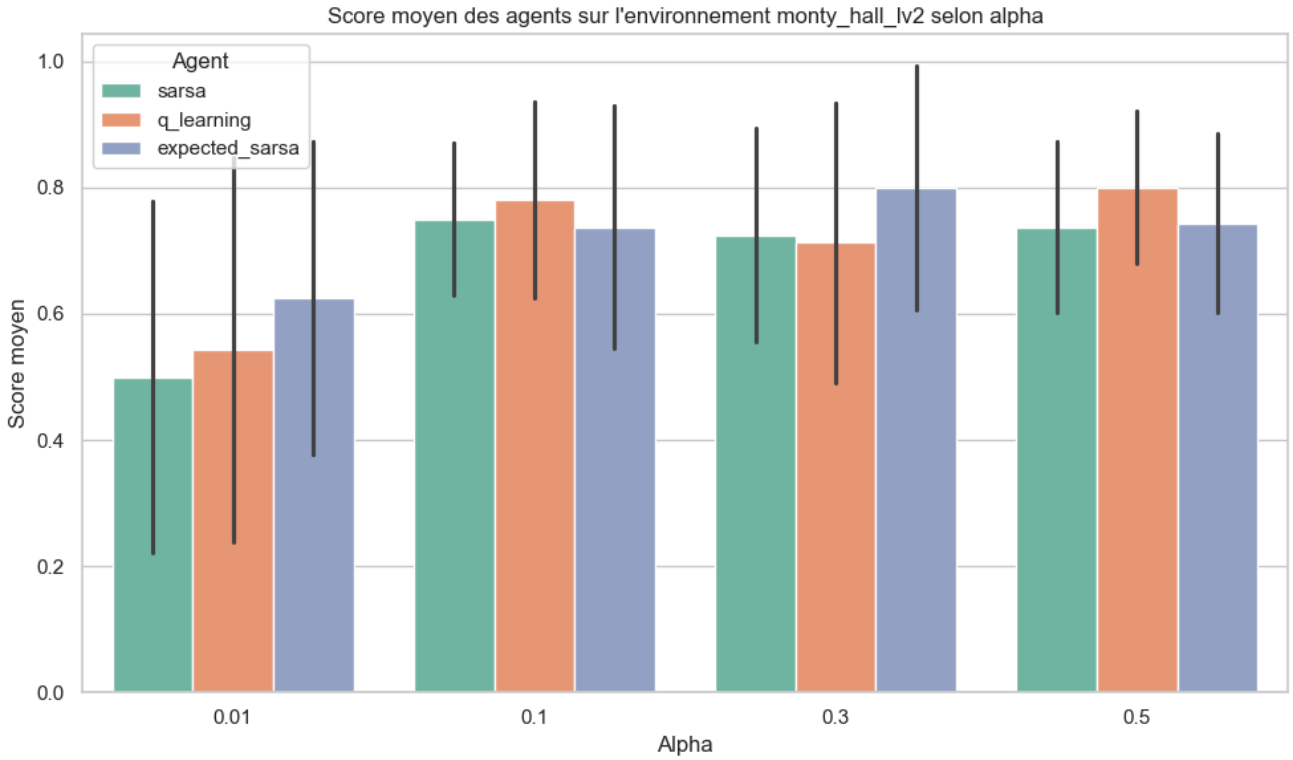

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



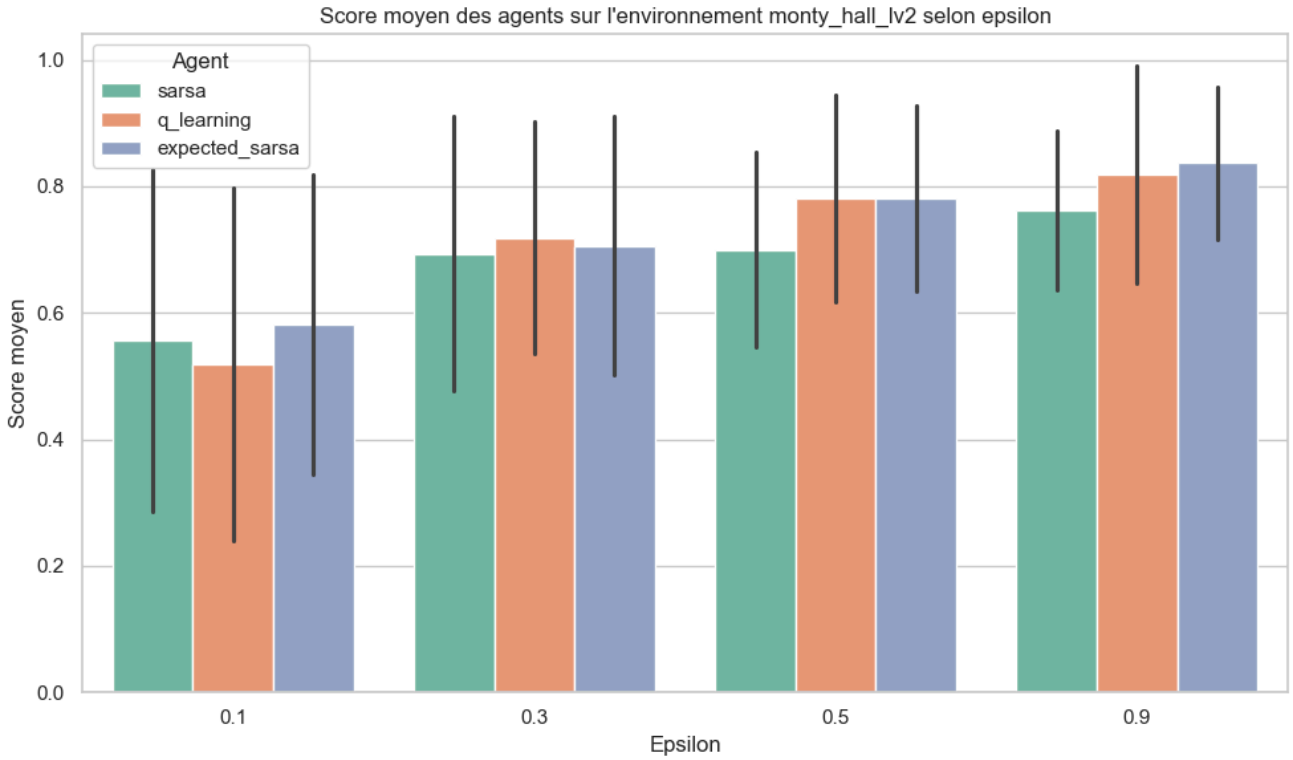
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



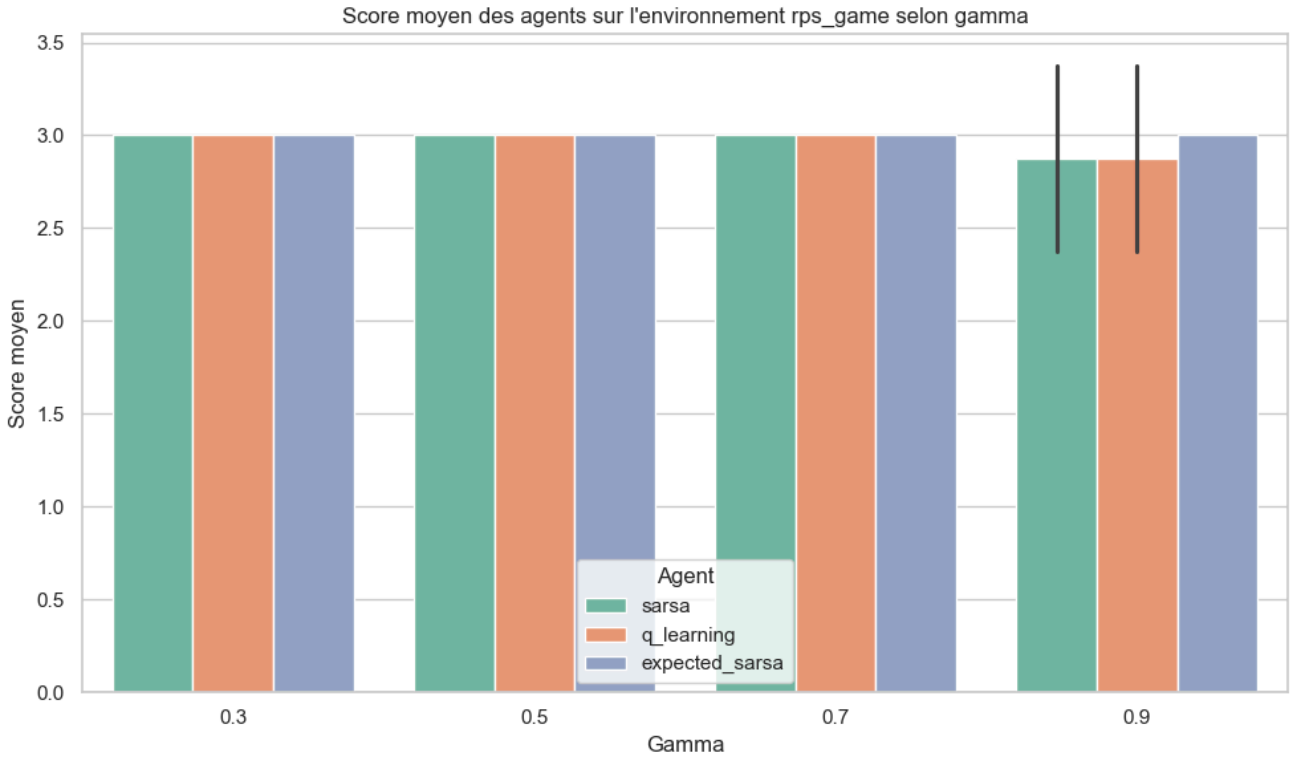

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



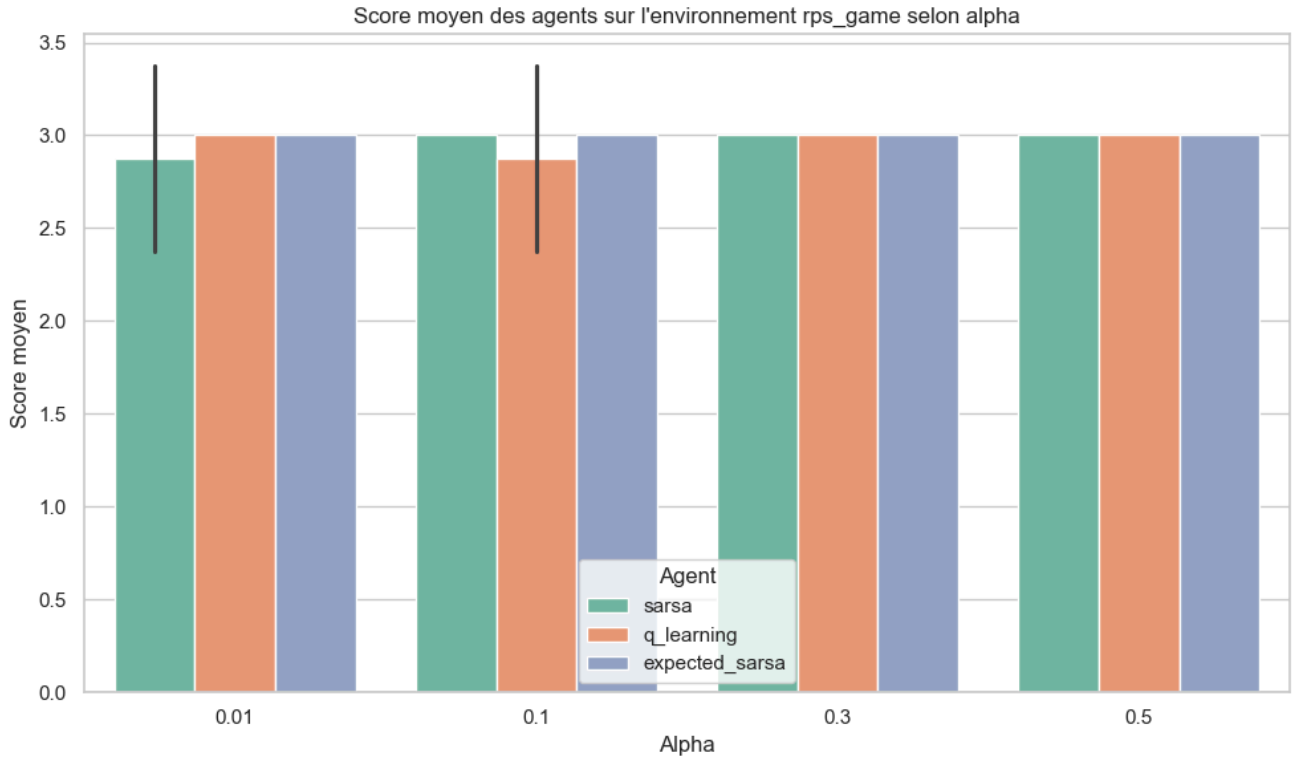
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



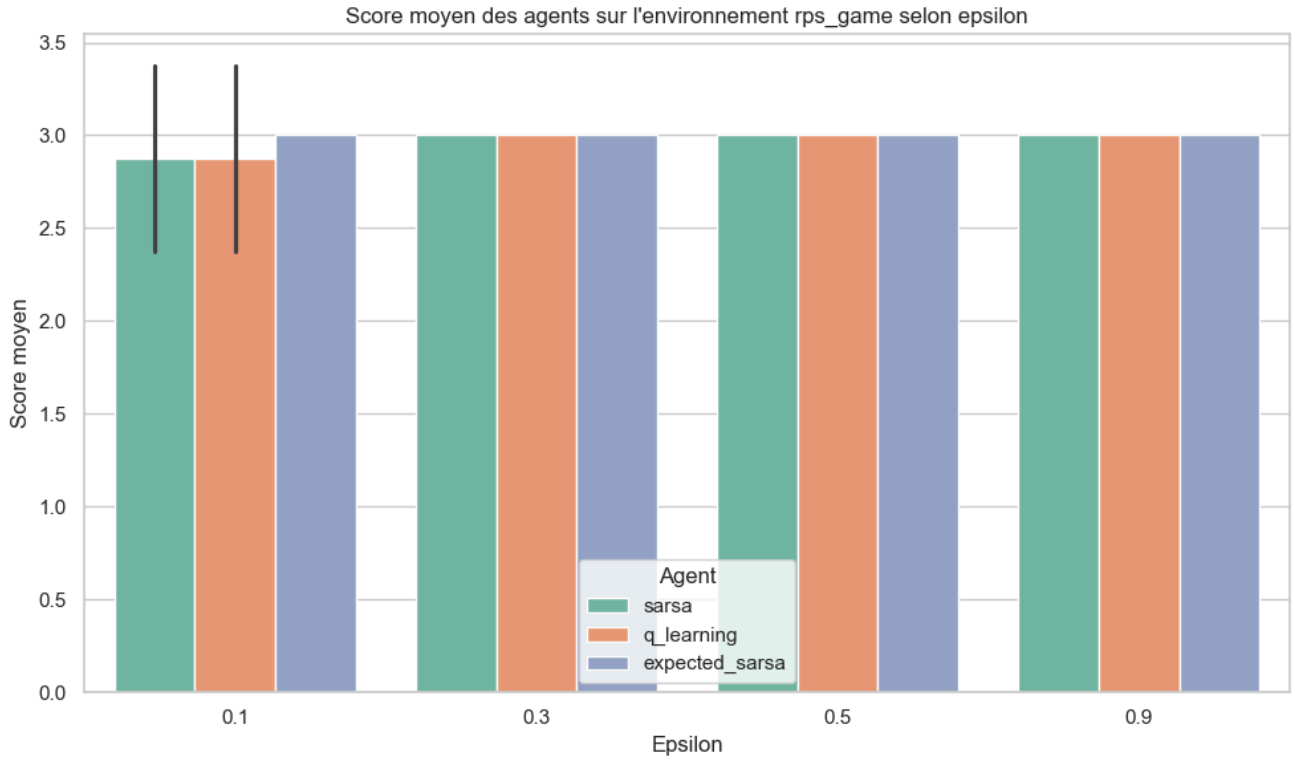

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



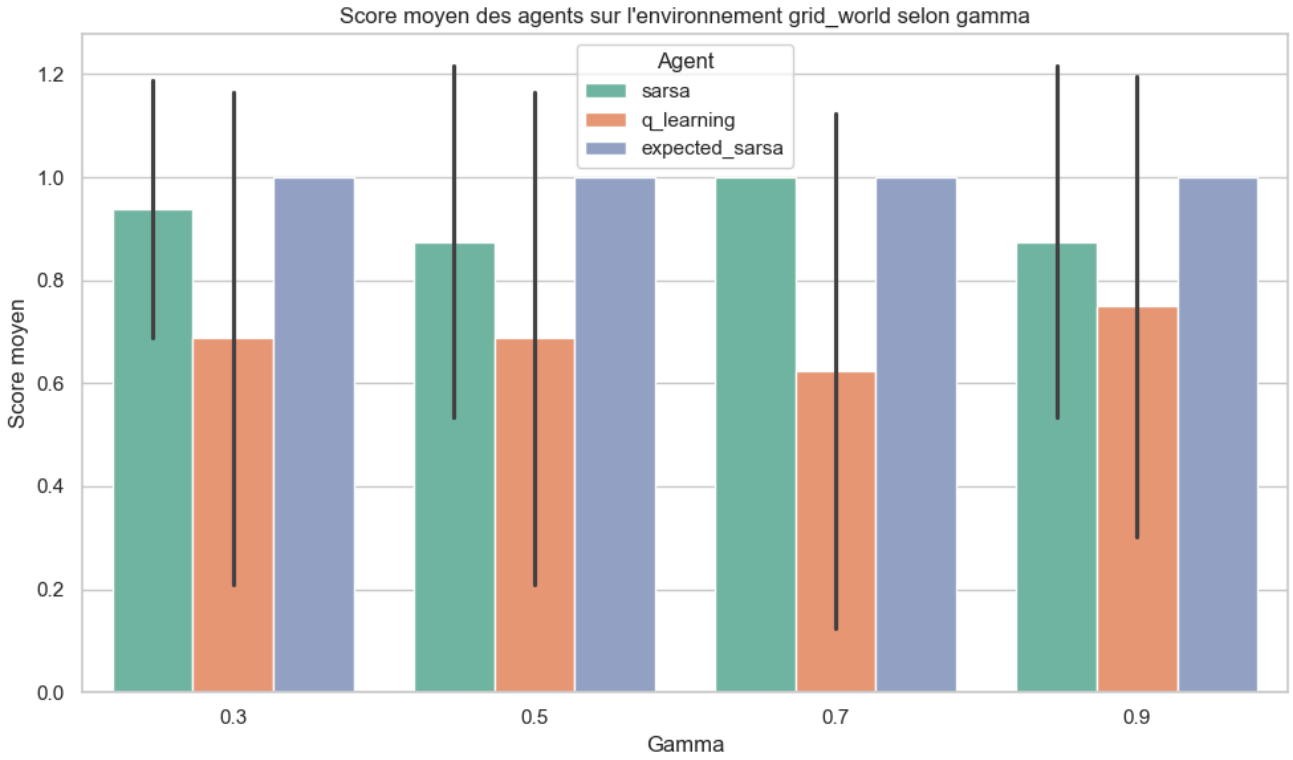
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



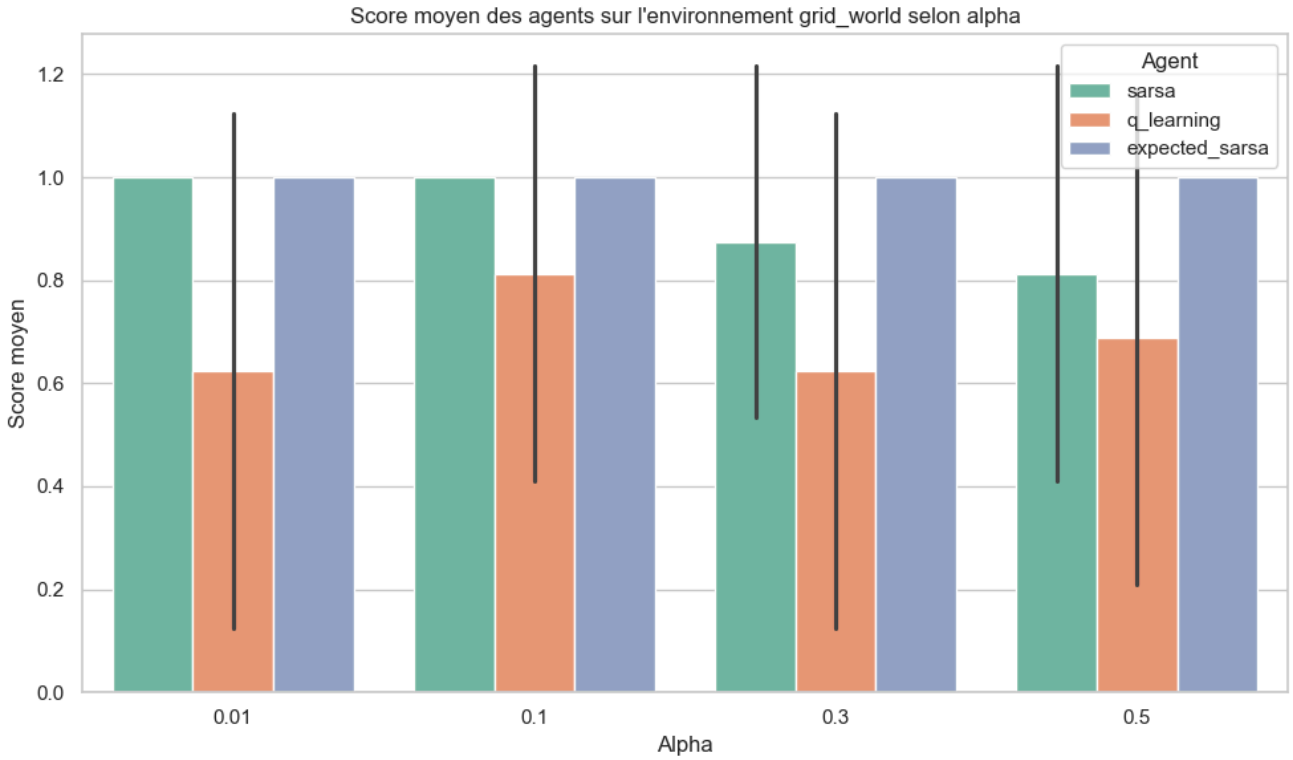

```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



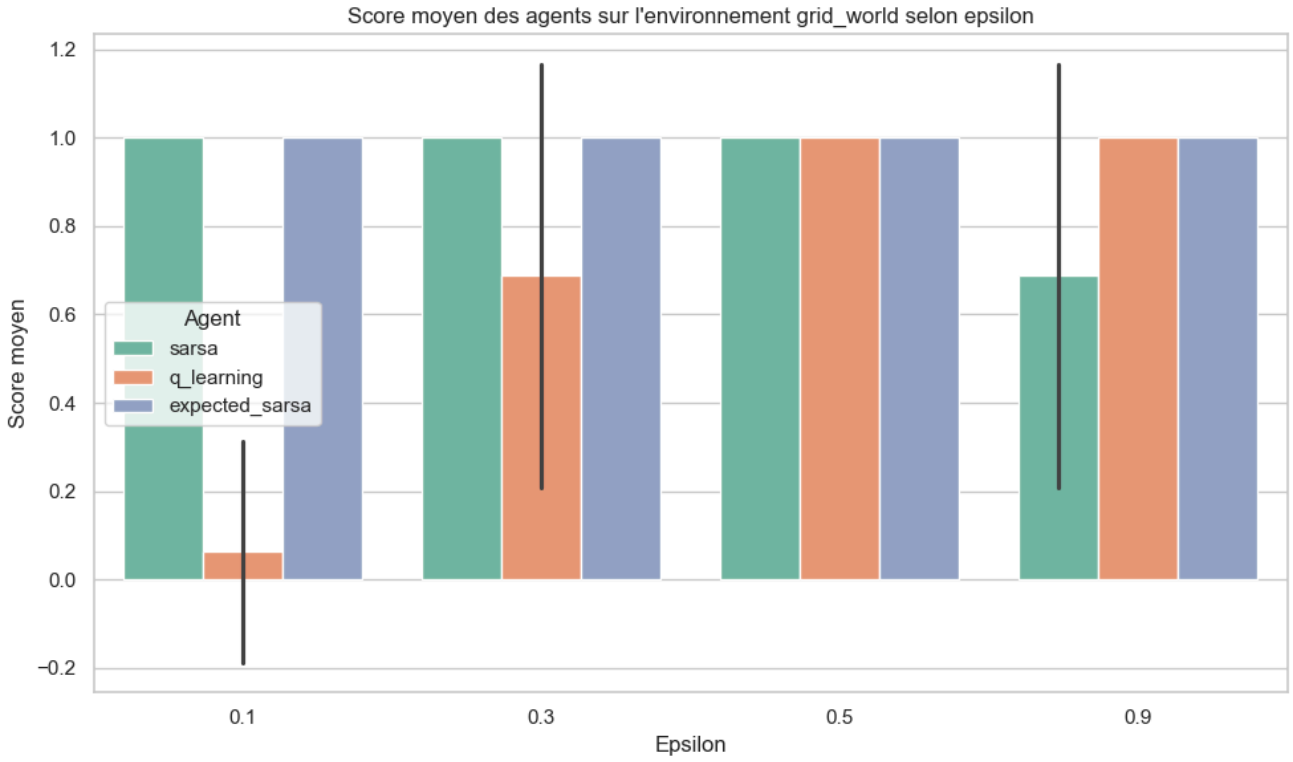
```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```




```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



```
C:\Users\kianm\AppData\Local\Temp\ipykernel_11612\1808295642.py:50: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar='sd'` for the same effect.

  sns.barplot(
```



Résumé des performances moyennes :

agent	mean_score	mean_steps	time
expected_sarsa	1.27	3.20	0.39
sarsa	1.23	4.64	0.24
q_learning	1.21	8.95	0.59

HYPERPARAMETRES de sarsa, expected sarsa et q_learning GAMMAS = [0.3, 0.5, 0.7, 0.9] ALPHAS = [0.01, 0.1, 0.3, 0.5] EPSILONS = [0.1, 0.3, 0.5, 0.9] Dans le line world il ont le meme résultat, dans MH1 q_learning est en tete, expected sarsa légèrement en tête dans MH2 et rps_game et largement dans gridworld, globalement expected Sarsa est notre meilleure agents

LINE WORLD: les hyperparamètres n'ont pas vraiment d'incidence sur le score des agents:

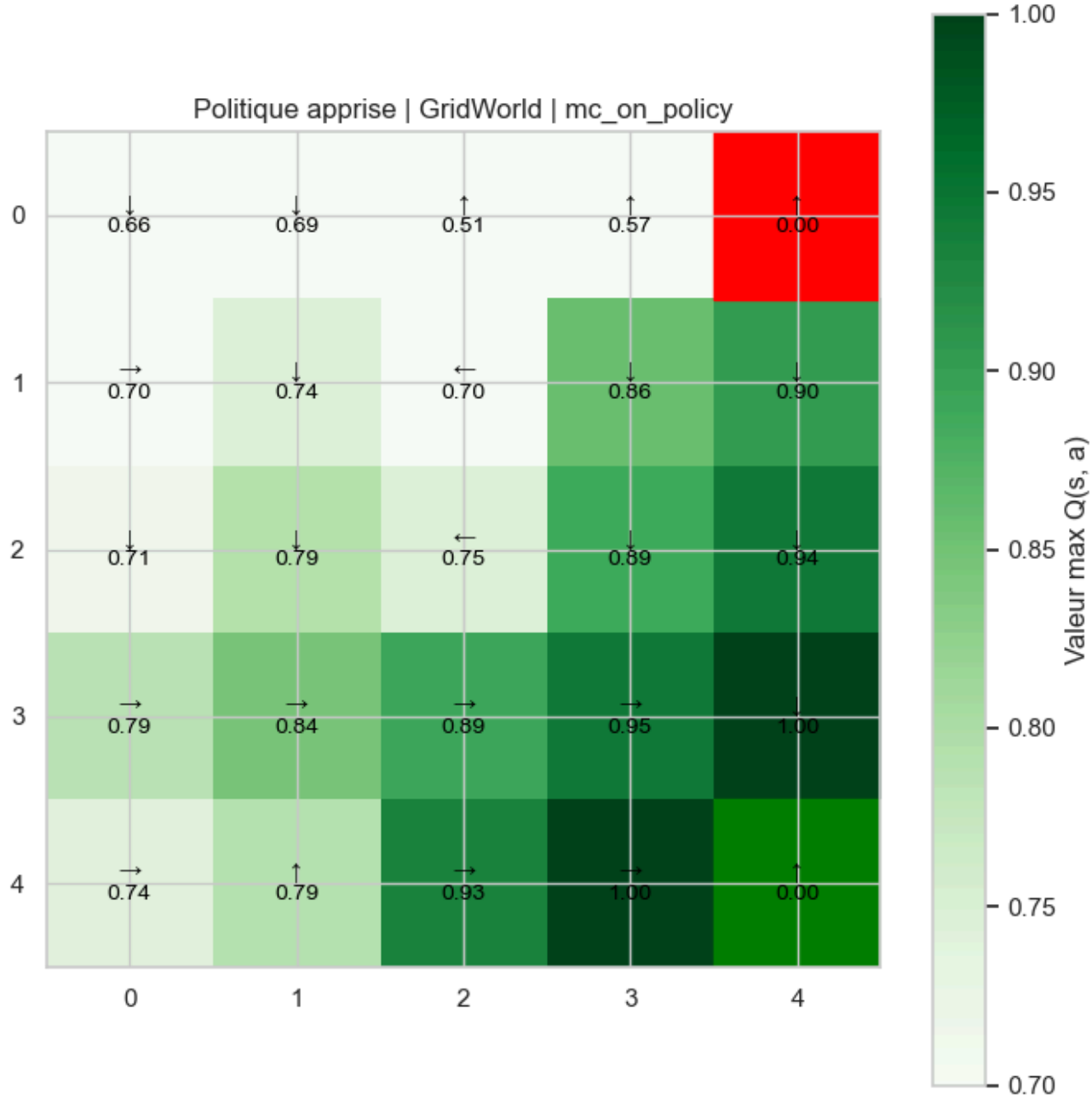
MH1 : un gamma trop élevé ou bas (> ou < à 0.5) réduit le score des agents, plus on accorde de l'importance au recompense future plus on obtiens de meilleures score

meilleur paramètre des TD dans MH1 {y : 0.5, a : 0.1, € : 0.9}

FAMILLE MONTE CARLO METHODS

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Chargement et nettoyage de la Q-table ===
q_table = pd.read_csv("Reports/MC_Results/q_tables/Q_GridWorld_mc_on_policy.csv", index_col

# Si plus de 4 lignes (actions), on garde que les 4 premières
if q_table.shape[0] > 4:
    print("⚠️ Q-table contient plus de 4 actions. On garde uniquement les 4 premières.")
    q_table = q_table[:4]

# Vérification
assert q_table.shape[0] == 4, f"Q-table incorrecte : elle doit avoir 4 lignes (actions), ma

# === Paramètres ===
rows, cols = 5, 5
actions = ["↑", "↓", "←", "→"]
best_actions = np.argmax(q_table, axis=0).reshape((rows, cols))
q_max = np.max(q_table, axis=0).reshape((rows, cols))

# === Affichage ===
plt.figure(figsize=(7, 7))
ax = plt.gca()

# Affichage de la valeur max et des flèches
for r in range(rows):
    for c in range(cols):
        action = actions[best_actions[r, c]]
        value = q_max[r, c]
        ax.text(c, r, f"{action}\n{value:.2f}", ha='center', va='center', fontsize=10, colo

# Affichage de la heatmap avec cmap inversée (plus foncé = meilleur)
im = ax.imshow(q_max, cmap="Greens", origin="upper", vmin=0.7, vmax=1.0)
cbar = plt.colorbar(im, ax=ax)
cbar.set_label("Valeur max Q(s, a)")

# Ajouter but (vert) et piège (rouge)
ax.add_patch(plt.Rectangle((4 - 0.5, 0 - 0.5), 1, 1, color='red', zorder=0))    # piège
ax.add_patch(plt.Rectangle((4 - 0.5, 4 - 0.5), 1, 1, color='green', zorder=0))  # but

# Style
ax.set_xticks(np.arange(cols))
ax.set_yticks(np.arange(rows))
ax.set_xticklabels(np.arange(cols))
ax.set_yticklabels(np.arange(rows))
ax.set_title("Politique apprise | GridWorld | mc_on_policy")
ax.set_aspect('equal')
plt.tight_layout()
plt.show()
```



ε = 0.1 introduit de l'exploration constante, expliquant les choix de trajectoires moins directes dans la zone haute gauche.

Les Q-values plus faibles en haut sont dues aux trajectoires aléatoires imposées par l'ε-greedy, même après convergence.