Amélioration de l'apprentissage par renforcement appliquée à la gestion de l'énergie par l'apprentissage des dynamiques indépendantes des actions

Théo Zangato*, Aomar Osmani*, Pegah Alizadeh*

*LIPN - CNRS UMR 7030, Université Sorbonne Paris Nord,
Paris, France

Résumé. Cet article est une traduction française de: "Enhancing Decision- Making through Action-Independent Dynamics Learning" publié à ECAI 2024. L'ajout d'objectifs auxiliaires dans l'apprentissage par renforcement permet aux agents d'acquérir des connaissances supplémentaires, améliorant la recherche de la politique optimale. Cet article présente l'algorithme MP-PPO, qui fusionne les concepts de PPO avec un module de prédiction probabiliste Transformer intégré dans l'architecture de l'Acteur-Critique. Ce modèle tire parti de la dépendance temporelle inhérente aux systèmes de gestion de l'énergie, en prédisant les transitions d'état futurs en apprenant à prédire certaines caractéristiques d'état. Les expériences sur données réelles, démontrent que l'intégration de capacités prédictives pour la prédiction d'états partiels améliore à la fois le rendement échantillonnal et l'efficacité de l'approche originale PPO.

1 Introduction

La transition vers une gestion durable de l'énergie dans les bâtiments est un élément crucial de la transition énergétique mondiale (García Vera et al., 2019; Delmastro, 2022). Les bâtiments représentant une part significative de la consommation énergétique mondiale, l'optimisation de leur usage énergétique grâce à des systèmes de gestion avancés est devenue une priorité (IEA, 2022). Cet article présente une approche innovante des systèmes de gestion de l'énergie des bâtiments (BEMS) en exploitant l'apprentissage par renforcement (RL), via un algorithme Proximal Policy Optimization (PPO) modifié et amélioré par un modèle de prévision. L'objectif de cette recherche est de répondre au défi posé par les environnements qui évoluent indépendamment des actions de l'agent, un scénario fréquent dans les BEMS. En intégrant un module de prévision au mécanisme acteur-critique, nous visons à améliorer la capacité de l'agent à anticiper les changements environnementaux, renforçant ainsi son processus de prise de décision dans la gestion des systèmes de stockage d'énergie.

Des expériences préliminaires ont indiqué que les agents RL traditionnels peuvent avoir des difficultés à discerner l'impact de leurs actions sur la transition de l'environnement, en particulier dans des problèmes d'optimisation multi-objectifs tels que ceux rencontrés dans les BEMS (Pignatelli et al., 2024; Sutton, 1984). Notre travail vise à atténuer ce problème en proposant

un agent capable d'apprendre une partie de la fonction de transition du modèle, permettant ainsi une convergence vers des solutions optimales de manière plus efficace et rapide.
L'innovation principale de notre recherche réside dans la modification de l'algorithme Mask-PPO (Huang et Ontañón, 2020) pour inclure un module de prévision d'état composé d'un réseau Transformer. Ce module se concentre exclusivement sur l'apprentissage des aspects de l'environnement indépendants des actions, permettant à l'agent de prendre des décisions plus éclairées. Nos résultats indiquent que cette approche non seulement facilite l'obtention de meilleures solutions, mais réduit également le nombre d'étapes d'entraînement pour converger.

Notre étude soutient les recherches existantes sur le potentiel du RL dans la gestion de l'énergie et propose une nouvelle approche empirique qui améliore l'efficacité d'échantillonnage et les performances des algorithmes de RL dans des environnements complexes et évolutifs. Les implications de notre travail vont au-delà de la sphère académique, offrant des stratégies prometteuses pour relever les défis réels de la gestion de l'énergie.

2 Méthodologie

2.1 Model-Predictor PPO (MP-PPO)

Notre objectif est de doter l'agent de la capacité à comprendre les dynamiques environnementales qui sont indépendantes de ses actions, mais qui influencent son état futur. Contrairement aux tâches de contrôle où l'état du système est uniquement déterminé par les actions de l'agent, des tâches telles que la gestion de l'énergie sont fortement influencées par des facteurs externes et temporellement dépendants. Notre but est d'augmenter la capacité de l'agent à comprendre ces dynamiques pour une meilleure anticipation et pour distinguer plus efficacement l'impact de ses actions sur les transitions d'état.
On propose une nouvelle version de l'algorithme Mask-PPO (Huang et Ontañón, 2020) en intégrant un module de prédiction qui anticipe les états futurs indépendants des actions de l'agent. Cela implique : 1) d'**adapter la mémoire tampon** (*replay buffer*), 2) d'**intégrer un module de prédiction** et 3) de **fournir aux réseaux Acteur et Critique une représentation latente**, dépendante de la séquence, des informations traitées. [1]

Définition 1 *On considère un MDP* $M =< S, A, P, R, \gamma >$ *où* S, A *sont les ensembles des états et des actions,* P *et* R *les fonctions de transition et de récompense et* γ *le facteur d'actualisation. On définit la composante d'état indépendante des actions* $\tilde{S} \subset S$ *où chaque état* $\tilde{s}_t \in \tilde{S}$ *contient un sous-ensemble d'informations provenant de l'état* $s_t \in S$*. Il représente les aspects de l'environnement qui changent en raison de dynamiques externes plutôt que des interventions de l'agent.*

On note $\tilde{p} : \tilde{S} \times \tilde{S} \to [0, 1]$ *la fonction de probabilité de transition d'un état indépendant des actions* $\tilde{s}_t$ *à* $\tilde{s}_{t+1}$*, définie par* $\tilde{p}(\tilde{s}_{t+1}|\tilde{s}_t)$*, en raison de facteurs externes, indépendants des actions de l'agent. La fonction globale de probabilité de transition* $P : P(s_{t+1}|s_t, a_t)$ *intègre à la fois les dynamiques contrôlables et incontrôlables de la manière suivante :*

$$P(s_{t+1}|s_t, a_t) = \sum_{\tilde{s}_{t+1} \in \tilde{S}} P(s_{t+1}|\tilde{s}_{t+1}, s_t, a_t) \cdot \tilde{p}(\tilde{s}_{t+1}|\tilde{s}_t) \tag{1}$$

1. https ://github.com/zxnga/MP-PPO

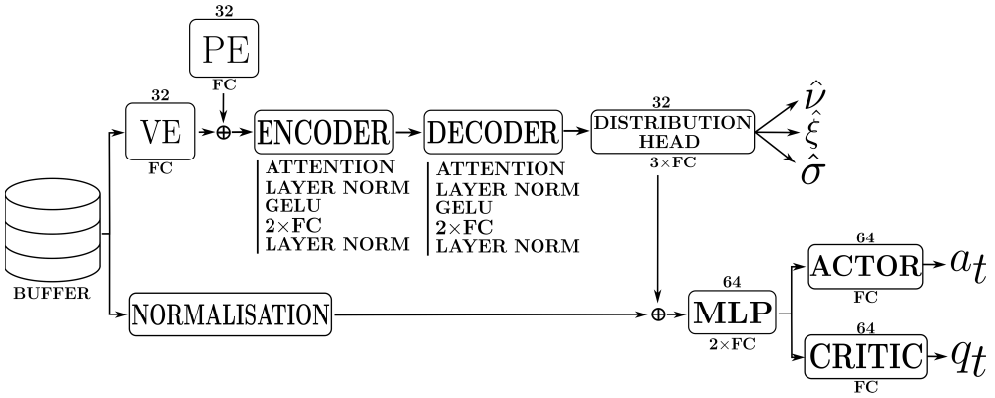


FIG. 1 – *Architecture du modèle.*

Hypothèse 1 *Si l'on considère que la fonction de transition indépendante des actions $\tilde{p}$ est déterministe, on peut simplifier la fonction globale de probabilité de transition comme suit :*

$$P(s_{t+1}|s_t, a_t) = P(s_{t+1}|\tilde{s}_{t+1}, s_t, a_t) \tag{2}$$

Lorsque $\tilde{p}$ devient déterministe, l'incertitude liée aux transitions d'état indépendantes des actions est supprimée. Le modèle se concentre alors sur la prédiction de la manière dont $\tilde{s}_t$ se transforme de manière déterministe en $\tilde{s}_{t+1}$ et sur la manière dont ces transitions, combinées aux actions de l'agent, influencent les transitions d'état globales capturées par P.

2.1.1 Rolling Replay Buffer

Pour permettre l'entraînement du Transformer, on modifie la mémoire (*replay buffer*) (Lin, 1992) ainsi que le mécanisme de collecte des transitions. On introduit le *Rolling Replay Buffer*, qui collecte à la fois les transitions originales et intègre h états futurs indépendants des actions dans chaque transition. Chaque entrée dans la mémoire est représentée sous forme de tuple $\tau_t = (s_t, a_t, r_t, s_{t+1}, \tilde{s}_{t+1:t+h})$. Étant donné que l'agent n'a accès qu'à t et $t+1$ lors de la collecte, la mémoire met à jour toutes les transitions précédentes dans l'horizon prédictif h.

2.1.2 Time Series Prediction Transformer

Le module de prévision traite des séquences d'états indépendants des actions pour prédire leurs valeurs futures. Les données utilisées pour la prédiction présente de fortes dépendances par rapport à la source et au contexte, comme le type de bâtiment, la zone climatique, et des facteurs temporels (ex. jour de la semaine ou saison). Pour gérer divers scénarios avec des différences significatives et traiter des séries de longue durée, on choisit un modèle Transformer (Zeng et al., 2023). Cette architecture, dont la nature probabiliste réduit l'impact des valeurs aberrantes, est entraînée comme un modèle global sur toutes les séries temporelles disponibles, lui permettant d'apprendre des représentations latentes à partir de sources variées . Le modèle intègre dans l'Acteur-Critique les prédictions $\tilde{s}_{t+1:t+h}$ de $t+1$ à $t+h$. Le Transformer est

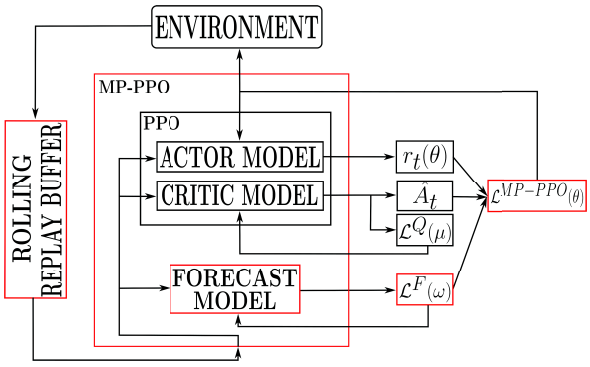


FIG. 2 – *Interactions des composants pendant l'entraînement. En rouge ceux ajoutés/modifiés.*

représenté dans la partie supérieure de l'architecture (Fig. 1) et décrit ci-dessous :

Embedding de valeurs : $\text{VE}(\tilde{s}_t) = W_e \tilde{s}_t$, où W_e est la matrice de poids d'encodage.
Embedding de caractéristiques : Étant donné une liste de i caractéristiques catégorielles $X_1, X_2, \ldots, X_N$, leur cardinalité respective C_i, et une dimension d'encodage correspondante D_i, nous avons N couches d'encodage telles que $\text{FE}_i : \mathbb{R}^{C_i} \to \mathbb{R}^{D_i}$.
Positional Embedding : Conformément à l'implémentation originale, nous utilisons des encodages sinusoïdaux pour suivre la dépendance temporelle dans nos séries. On utilise 6 encodages représentants : la position de la valeur dans la série, la semaine de l'année, le jour de l'année, le jour du mois, le jour de la semaine et l'heure de la journée.
Distribution-based Prediction : La sortie du réseau mappe les caractéristiques d'entrée $\mathbf{X}$ via une fonction $f(\cdot)$ vers un ensemble de paramètres $\mathbf{z} = f(\mathbf{X}; \omega)$, où ω sont les paramètres du réseau. Les paramètres $\mathbf{z}$ sont ensuite transformés en paramètres de distribution de Student via une fonction de mappage de domaine $g(\cdot) : (\hat{\nu}, \hat{\xi}, \hat{\sigma}) = g(\mathbf{z})$ (Zeng et al., 2023). Étant donnés les paramètres prédits, la probabilité conditionnelle des états partiels prédit $\tilde{s}_{t+1:t+h}$ étant donné $\mathbf{X}$ est modélisée comme suit : $p(\tilde{s}_{t+1:t+h}|\mathbf{X}) = \text{StudentT}(\tilde{s}_{t+1:t+h}; \hat{\nu}, \hat{\xi}, \hat{\sigma})$
Loss Function : Le réseau est entraîné en minimisant la *negative log-likelihood (NLL)* des données observées sous la distribution de Student prédite : $\mathcal{L}^F(\omega) = -\log p(\tilde{s}_{t+1:t+h}|\mathbf{X}; \omega)$.

2.1.3 Architecture et Entraînement

Les dynamiques de l'environnement étant fortement dépendantes du temps, elles peuvent être efficacement représentées sous forme de séries temporelles avec des motifs temporels distincts. En divisant ces séries temporelles de manière égale, leurs propriétés circulaires peuvent être exploitées. Pour préserver la séquentialité, les trajectoires dans la mémoire ne sont pas mélangées. L'échantillonnage utilise une stratégie circulaire, garantissant un *batch size* M constant et réduisant le sur-apprentissage sur des sous-ensembles de données spécifiques. Une position de départ aléatoire p est sélectionnée, et les séquences sont extraites de manière circulaire, offrant ainsi des échantillons diversifiés et représentatifs à travers les époques : $\text{Sample} = \{(s_p, s_{p+1}, \ldots, s_{(p+M) \mod |\mathcal{B}|})\}$, où $|\mathcal{B}|$ et le nombre d'éléments dans le *buffer*.

Le module Transformer est entraîné en utilisant les états indépendants des actions $\tilde{s}_{t+1+h}$ échantillonnés à partir des trajectoires de la mémoire tampon. L'architecture globale intègre les informations du module de prévision avec les observations de l'environnement pour apprendre la politique. L'observation est concaténée avec la sortie du Décodeur, et le vecteur résultant est traité par un Perceptron Multi-Couche (MLP) avant d'être transmis aux réseaux Acteur et Critique (Fig. 1). Cette approche fournit une représentation significative et augmente la dimension de l'entrée, améliorant ainsi les performances et l'efficacité d'échantillonnage de notre algorithme RL, comme démontré par (Ota et al., 2020).

Le mécanisme de mise à jour pour l'ensemble du réseau implique la rétropropagation de la *loss* du réseau de prédiction $\mathcal{L}^F$ pendant l'optimisation simultanée de l'Acteur-Critique (Fig. 2). L'optimisation de ce dernier repose sur le *Clipped Surrogate Objective* pour $\pi_\theta(a_t|s_t)$ définit par $L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)A_t^{\hat{\pi}_\theta}, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t^{\hat{\pi}_\theta})\right]$, un terme d'entropie $S[\pi_\theta](s_t)$ et la *loss* associée à la fonction de valeur $\mathcal{L}^Q(\mu) = \mathbb{E}\left[\left(\delta_t Q_\mu^{\pi_\theta}\right)^2\right]$, avec δ la *TD-error* (Schulman et al., 2017).

Notre *loss* modifiée résultante pour la mise à jour conjointe Transformer-Acteur-Critique peut être décrite, avec les coefficients $c1$ et $c2$, comme suit :

$$\mathcal{L}^{\text{MP-PPO}} = \mathbb{E}\left[\mathcal{L}^{\text{CLIP}}(\theta) - c_1\mathcal{L}^Q(\mu) + \mathcal{L}^F(\omega) + c_2 S[\pi_\theta](s_t)\right] \tag{3}$$

3 Expérimentations

3.1 Cadre expérimental

Dans les systèmes de gestion de l'énergie des bâtiments (BEMS), l'objectif est de gérer et de coordonner efficacement les systèmes énergétiques, y compris le stockage et la production d'énergie renouvelable, afin d'optimiser les coûts et la consommation d'énergie. On utilise l'environnement Citylearn (Vázquez-Canteli et al., 2019), qui exploite des données réelles provenant de 60 bâtiments différents, résidentiels ou commerciaux, situés dans diverses zones climatiques aux USA. Chacun est équipé d'un contrôleur qui régule les flux d'énergie vers et depuis son unité de stockage d'énergie (ESU). Ce dernier influence dynamiquement la consommation $E_t^r > 0$ provenant du réseau : à la hausse lors du stockage et à le baisse lors de l'utilisation d'énergie stockée. On note : L_t l'énergie dont on ne peut pas reporter la consommation, E_t^{th} la consommation d'énergie thermique, E_t^{ESU} les transferts d'énergie dans l'ESU et E_t^{pv} l'énergie produite par les sources renouvelables, tous en kWh. H_t est l'état de charge (SOC) normalisé de l'ESU par rapport à sa capacité v et C_t^g le coût d'un kWh. La consommation totale d'énergie du bâtiment est alors donnée par : $E_t = L_t + E_t^{th} + E_t^{ESU} + E_t^{pv}$.

État : $s_t = \left\{B, C_t^{\text{g}}, \{H_t\}, E_t^{\text{pv}}, L_t, \{K_t\}\right\}$, avec B identifiant le type de bâtiment par une méthode de clustering décrite par (Zangato et al., 2024).

État indépendant des actions : $\hat{L}_t = L_t - E_t^{pv}$, Dans ce contexte, la tâche principale du Transformer est de prédire les valeurs futures de $\hat{L}_t$ en tant que série temporelle univariée.

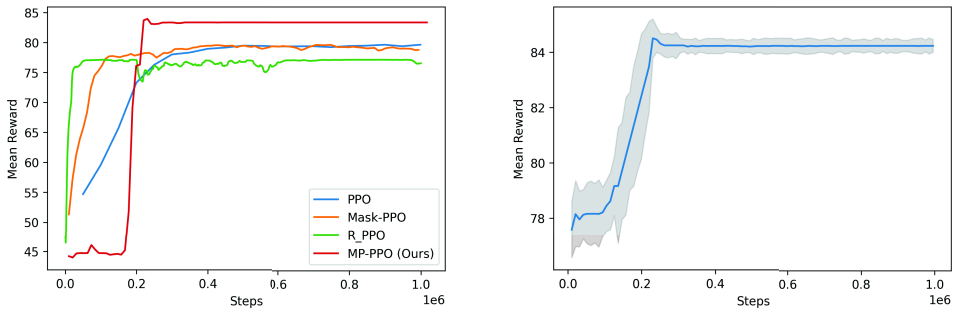
Actions : L'agent gère l'ESU d de chaque bâtiment via un espace d'actions discrétisé $S = \{x \in \mathbb{R} \mid x = -1 + 0.1n, n \in \{0, 1, 2, \ldots, 20\}\}$.

Récompense : La fonction de récompense est une combinaison linéaire de deux fonctions qui suivent la somme des coûts financiers et environnementaux C. La première partie encou-

rage l'agent à stocker la production d'énergie excédentaire, en utilisant un signal de récompense adversaire représenté par l'indice b : $R^1(t) = C_t^b - C_t = C_t^g \left(E_t^{r,b} - E_t^r\right)$. La deuxième partie se concentre sur la gestion des cycles de charge et de décharge, avec ζ un hyperparamètre, fixé empiriquement à $0,2$, introduit pour ajuster le coût de l'utilisation de l'énergie stockée lors des différentes phases : $R^2(t) = (1-\zeta)C_t^d\left(\max(0, H_{t-1} - H_t)(1-\alpha)\right) + \zeta C_t^g \sum_{e \in E^{th}} E_t^r + e$, avec α un coefficient de dégradation.

3.2 Expérimentations et résultats

On compare notre modèle à des variantes de PPO comme points de référence afin d'évaluer l'impact global de notre approche et d'identifier les caractéristiques qui contribuent à l'amélioration des performances. On le compare donc à la version originale de PPO (Schulman et al., 2017), sa version discrète en utilisant un vecteur de masquage sur l'espace des actions (Huang et Ontañón, 2020) et sa version récurrente utilisant un LSTM (Raffin et al., 2021).



(a) Récompenses moyennes.

(b) Impact du pré-entraînement des poids du Transformer.

FIG. 3 – *Résultats expérimentaux pendant l'entrainement des modèles.*

Bien que les versions standard et masquée de PPO atteignent des récompenses finales similaires, cette dernière converge plus rapidement (Fig 3 (a)), car l'agent contourne la courbe d'apprentissage liée aux contraintes dynamiques sur l'espace des actions, lui permettant de se concentrer sur l'optimisation des coûts énergétiques. De plus, le masquage réduit l'espace des actions, diminuant ainsi le nombre d'étapes d'exploration nécessaires pour couvrir efficacement cet espace. MP-PPO rencontre des difficultés d'apprentissage au début du processus en raison des taux d'erreur élevés du module de prédiction, qui fournissent des informations erronées à l'agent. Comme le réseau est entraîné conjointement, le prédicteur reçoit un nombre limité de mises à jour — 10 par cycle de mémoire tampon, soit tous les 8 000 pas — ce qui ralentit sa convergence. Cependant, à mesure que le module de prédiction s'améliore et que les erreurs diminuent, la collecte de récompenses de l'agent s'accélère, menant finalement à une meilleure solution globale par rapport aux références. Les résultats finaux pour une année de gestion énergétique à travers les bâtiments de test, basés sur les fonctions objectifs calculant les coûts financiers et environnementaux, sont présentés dans le Tableau 1.

TAB. 1 – *Coût financier et environnemental annuel. Les résultats sont normalisés en utilisant uniquement l'électricité provenant du réseau, 0.9 indique une réduction de 10%.*

Groupe de bâtiment	**Évaluation par rapport aux algorithmes références.**			
	Mask-PPO	***MP-PPO (Ours)***	***R-PPO***	***PPO***
1	1.0[a], 0.937[b]	**0.98**,**0.918**	1.0, 1.0	1.018, 0.951
2	1.013, 0.858	1.0, **0.847**	1.0,1.0	1.029, 0.876
3	1.02, 0.887	0.99, **0.88**	1.0, 1.0	1.018, 0.935

[a]Émissions de CO_2 annuelles [b]Coût financier annuel

3.2.1 Pré-entrainement du Transformer

En pré-entraînant les couches Transformer de notre algorithme durant une période de chauffe, où seuls les poids du prédicteur sont mis à jour, nous visons à réduire la stagnation initiale de la politique. Les mises à jour du Transformer se produisant en parallèle avec celles de l'Acteur-Critique, les inexactitudes de prédiction initiales peuvent ralentir le processus d'apprentissage de l'agent. Cette approche vise à faciliter un apprentissage plus efficace, permettant une convergence plus rapide vers des solutions optimales en démarrant avec un modèle plus précis des dynamiques de l'environnement. Notre investigation explore comment le pré-entraînement peut atténuer ces erreurs et améliorer l'efficacité globale du processus d'apprentissage. Le pré-entraînement, effectué sur 15 000 pas et réparti sur 50 à 200 époques, équilibre fiabilité des prédictions et capacité de généralisation en évitant le sur-aprentissage.

Cette approche améliore significativement les performances initiales de l'agent (Fig.3b) . En commençant avec une compréhension plus affinée de l'environnement, l'agent obtient des récompenses initiales plus élevées par rapport aux scénarios précédents. Cela élimine le comportement indésirable des premières étapes qui pénalise l'acquisition des récompenses élevées, permettant ainsi une convergence plus rapide avec moins d'échantillons.

4 Conclusion

Ce travail présente MP-PPO, un nouvel algorithme de RL qui intègre un composant de prévision pour apprendre les aspects indépendants des actions dans les transitions d'état. Un modèle Transformer est utilisé pour comprendre les dynamiques de l'environnement qui ne sont pas affectées par les actions de l'agent. Cette distinction permet à l'agent d'isoler les effets de ses actions sur les états ultérieurs tout en facilitant la compréhension des phénomènes à long terme. Plus précisément, en apprenant les transitions indépendantes des actions, le modèle acquiert la capacité de faire des prévisions sur de longues périodes (jusqu'à 24 heures). Ces prévisions sont essentielles dans des tâches avec des motifs cycliques, tels que la gestion de l'énergie, permettant à l'agent de tirer efficacement parti des tendances récurrentes. MP-PPO s'appuie sur une variante masquée de l'algorithme PPO, avec un potentiel d'application pour tous les algorithmes *model-free*. Une évaluation empirique, utilisant des données réelles de l'environnement Citylearn démontre que notre algorithme améliore les performances, surpassant les références établies comme l'implémentation conventionnelle de PPO et permettant une réduction jusqu'à 15% des coûts financiers et environnementaux associés aux bâtiments.

Références

Delmastro, C. (2022). Buildings sectorial overview. Technical report, International Energy Agency (IEA). Accessed : October 11, 2024.

García Vera, Y. E., R. Dufo-López, et J. L. Bernal-Agustín (2019). Energy management in microgrids with renewable energy sources : A literature review. *Applied Sciences*.

Huang, S. et S. Ontañón (2020). A closer look at invalid action masking in policy gradient algorithms. *CoRR abs/2006.14171*.

IEA (2022). Approximately 100 million households rely on rooftop solar pv by 2030. Technical report, International Energy Agency (IEA). Accessed : October 11, 2024.

Lin, L.-J. (1992). Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning 8*, 293–321.

Ota, K. et al. (2020). Can increasing input dimensionality improve deep reinforcement learning ? In *37th International Conference on Machine Learning*, pp. 7424–7433.

Pignatelli, E. et al. (2024). A survey of temporal credit assignment in deep reinforcement learning. *Transactions on Machine Learning Research*.

Raffin, A. et al. (2021). Stable-baselines3 : Reliable reinforcement learning implementations. *Journal of Machine Learning Research 22*(268), 1–8.

Schulman, J. et al. (2017). Proximal policy optimization algorithms. *CoRR abs/1707.06347*.

Sutton, R. S. (1984). Temporal credit assignment in reinforcement learning. *PhD Thesis*.

Vázquez-Canteli, J. R. et al. (2019). Citylearn v1.0 : An openai gym environment for demand response with deep reinforcement learning. In *6th ACM for Energy-Efficient Buildings, Cities, and Transportation*, pp. 356–357.

Zangato, T., A. Osmani, et P. Alizadeh (2024). Optimisation de la gestion de l'Énergie par l'apprentissage par renforcement et le clustering de séries temporelles pour la génération de politiques individualisées. *Extraction et Gestion des Connaissances, RNTI-E-40*, 11–22.

Zeng, A. et al. (2023). Are transformers effective for time series forecasting ? In *Proceedings of the AAAI conference on artificial intelligence*, Volume 37, pp. 11121–11128.

Summary

Incorporating auxiliary objectives into Reinforcement Learning allows agents to acquire additional knowledge, thereby increasing their search for the optimal policy. This article presents the Model-Predictor Proximal Policy Optimization (MP-PPO) algorithm, which merges the concepts of various PPO variants with a Transformer probabilistic prediction module. This model capitalizes on the time dependence inherent in energy management systems, predicting future state transitions by learning to predict certain state characteristics. Notably, our algorithm seamlessly integrates this predictive capability into the Actor-Critic architecture, avoiding the need for an external model. Through experiments on real data, we demonstrate that integrating predictive capabilities for partial state prediction improves both the sample effectiveness and efficiency of the original PPO.