用于住宅负荷快速时间尺度需求响应的多智能体强化学习

Vincent Mai 机器人与具身人工智能实验室，Mila 蒙特利尔大学，加拿大
vincent.mai@umontreal.ca

Philippe Maisonneuve GERAD 与 Mila 蒙特利尔理工学院，加拿大
philippe.maisonneuve@polymtl.ca

Tianyu Zhang Mila 蒙特利尔大学，加拿大
tianyu.zhang@mila.quebec

Hadi Nekoei
Mila 蒙特利尔大学，加拿大
nekoeihe@mila.quebec

Liam Paull 机器人与具身人工智能实验室，Mila，蒙特利尔大学，加拿大
liam.paull@umontreal.ca

Antoine Lesage-Landry GERAD 与 Mila，蒙特利尔理工学院，加拿大
antoine.lesage-landry@polymtl.ca

arXiv:2301.02593v1 [cs.MA] 6 Jan 2023

摘要

为整合大量可再生能源资源，电力电网必须能够应对发电功率在快速时间尺度上的大幅波动。通过需求响应进行频率调节，有望协调空调等在时间上具有灵活性的负荷，以抵消这些波动。现有采用离散控制并考虑动态约束的方法，在数百个智能体参与的快速时间尺度动作选择任务中，难以取得令人满意的效果。我们提出一种去中心化智能体，采用多智能体近端策略优化并通过局部通信进行训练。我们探索了两种通信框架：人工设计的通信框架，以及通过定向多智能体通信学习得到的框架。由此得到的策略在频率调节方面表现出色且稳健，并能无缝扩展至任意数量的住宅，同时保持处理时间恒定。

关键词

多智能体强化学习，需求响应，电力系统，可再生能源并网，通信，协调

1 引言

为实现联合国提出的将全球变暖限制在 +1.5摄氏度以内的气候变化目标，全球发电必须从化石燃料转向风力涡轮机和太阳能电池板等可再生能源。根据国际能源署的数据，2019年，电力和热力生产占全球排放量的 40% [3]，，其中 64% 来自燃烧化石燃料发电 [1]。因此，电力行业必须从传统的燃料燃烧模式转向基于可再生自然现象的发电方式，例如风力涡轮机和太阳能光伏。可再生能源发电会受到短期、高幅度波动的影响，这种现象称为间歇性。例如，云层经过时，太阳能发电量会骤降；天空重新放晴后，发电量又会急剧上升。这些变化可能在几秒内发生，给电网运营商带来重大挑战：为了确保电网稳定，电力需求与发电量之间必须近乎完美地保持平衡 [29]。换句话说，发电量与消耗必须始终相等。因此，用间歇且不确定的发电取代稳定且确定的发电，会加剧对电力平衡的需求。在秒级时间尺度上，这种平衡任务称为频率调节 [9, 51]。

在发电侧，诸如将过剩能源储存于电池中，或依靠化石燃料电厂提供支持等方案，都需要大量投资，且后者并非可再生能源。另一种做法是引入需求响应计划 [49]，以缓解可再生能源的间歇性 [51]。需求响应旨在通过在时间上协调负荷，调整电力需求以匹配供给。这些负荷必须具备灵活性，即能够在实现自身功能的同时调节消耗。例如，电脑显示器就不具备这种灵活性，因为使用时必须保持全功率运行。相比之下，供暖设备、空调或热水器等恒温负荷是理想的候选对象：只要空气或水的温度处于用户偏好的范围内[12]，它们就不必始终开启。这类设备也十分普及，占全球功耗的很大一部分 [2, 40]。频率调节的目标不同于削峰；削峰旨在将负荷转移至其他时段，例如一天中的其他时段。而频率调节则是利用负荷的灵活性来平衡发电功率的高频变化。

本文聚焦于利用居民空调实现快速时间尺度需求响应以进行频率调节。这带来了若干物理和算法方面的约束：(1) 空调**以离散方式**供电，即只能开启或关闭，这限制了控制的灵活性；(2) 空调受到硬件**动态约束**的限制，例如锁定：关闭后，必须等待一段时间才能重新开启，以保护压缩机；(3) 由于应用场景是住宅，隐私十分重要，因此**应尽量减少通信**；(4) 为电网提供足够的功率灵活性，必须考虑大规模负荷聚合，因此该方法必须具备**可扩展性**；(5) 为便于实施，控制还应**去中心化**，并采用**局部通信**；(6) 决策必须在**数秒的时间尺度**内作出；最后，(7) 控制算法应能够**应对不确定性**，以适应未来的调节信号。

这些约束阻碍了经典方法的部署。贪心算法采用集中式架构，难以考虑长期动态约束 [35]。标准模型预测控制同样采用集中式架构，即使是去中心化版本，也需要求解一个多时段组合优化问题，且无法随智能体数量的增加而扩展 [26**?**]。我们提出使用多智能体强化学习（MARL）来解决这一问题，

学习适用于离散且受约束控制（1、2）以及有限局部通信（3、5）的去中心化且可扩展的策略（4）。训练完成后，这些策略便可根据不确定性下的期望值（7），实时做出最优决策（6）。由于该问题融合了当前多智能体强化学习面临的几项重要挑战，即通信、长期信用分配、协调与可扩展性 [21], , 因此也可作为多智能体强化学习算法的一个有趣基准。我们采用多智能体近端策略优化（MA-PPO） [60] 训练智能体，并使用集中式训练、去中心化执行（CTDE）。[28]。我们测试了两种局部通信框架——人工设计的框架和学习得到的框架——二者的表现均优于基线方法。我们的主要贡献有三点：

- 一个开源的多智能体环境[1]，用于模拟秒级时间尺度下通过需求响应进行频率调节的现实问题。该仿真器兼容 OpenAI Gym [11] 框架。
- 两个去中心化、响应迅速的智能体[1] 由多智能体近端策略优化训练。第一个智能体采用人工设计的通信策略，第二个则通过定向多智能体通信（TarMAC）[17]学习要共享哪些数据。两者在为期两天的仿真中均优于基线方法。
- 对训练后智能体的动态特性、通信、可扩展性和鲁棒性进行了深入分析。

下一节将介绍需求响应与多智能体强化学习领域的既有研究。第 3 节将描述环境并建立问题模型。第 4 节将介绍经典方法和基于学习的方法。最后，第 5 节将展示实验结果，并分析智能体的性能、动态特性、鲁棒性和可扩展性。

2 相关工作

通过需求响应进行频率调节，通常采用模型预测控制（MPC） [26, 33, 39, 40, 43, 56],，即基于给定时域内的轨迹预测来选择最佳动作，有时还会结合机器学习 [4, 19, 32]。除 [37], 这些研究外，其他研究均未考虑锁定等短期动态约束。MPC 方法依赖混合整数规划，随着智能体数量增加，其可扩展性难以维持，因此无法在快速时间尺度下进行控制。此外，这些研究通常要求集中式实体访问住宅数据，从而引发保密问题。文献 [13]提出了一种基于乘子法的分布式模型预测控制替代方案。该方法未考虑锁定约束，也不适用于快速时间尺度下的决策，因为它在每个时间步都需要进行多轮集中式通信，还要解决多个优化问题并将连续设定值转换为二元动作。

为解决这些问题，人们采用在线优化（OO）方法 [34,62] , 因为这类方法具有较高的计算效率和可扩展性。具体而言， [35] 将 OO 用于频率调节，并采用适用于空调等设备的二元控制设置。然而，这些方法依赖贪心优化，缺乏前瞻性，因此在面对动态约束时性能有限。为解决峰值削减等较长时间尺度的功率平衡问题，研究人员开发了强化学习（RL）方法 , 例如通过需求响应 [6]或协调负荷与发电机 [47, 58]。CityLearn 环境 [52] 提出了一个用于需求响应的多智能体强化学习（MARL）标准环境，并在此基础上开发了多种方法，例如 [45]利用智能逆变器和智能储能管理来调节配电网中的电压幅值，以及 [53] 用于电网交互型互联建筑的负荷塑形。随后，AlphaBuilding ResCommunity 环境 [54]实现了详细的热模型。然而，CityLearn 和 AlphaBuilding ResCommunity 都关注较长时间尺度的控制，因此不适用于高频调节，也未考虑空调的锁定和二元约束。PowerGridworld 环境 [10] 是 CityLearn 的一种更灵活的替代方案，可进行快速时间尺度仿真，但未提供负荷的详细热模型、锁定或二元控制选项，也未提供用于比较的经典基线方法。已有研究使用 MARL 解决高频调节问题，但仅涉及发电侧 [57]。据我们所知，文献中尚无使用 MARL 实现频率调节与需求响应的案例，同时具备秒级控制和灵活的二元负荷（如受锁定等硬件动态约束的空调）。

更广泛地说，多智能体强化学习已被用于促进虚拟环境中的协作，例如 Dota 2 [44], Hide and Seek [8] 或 Hanabi [20], , 也用于现实环境，例如交通信号灯控制 [55], 单户能源管理 [5] 或网约车 [46]。多智能体强化学习问题为强化学习带来了一些额外挑战 [21], , 例如环境的非平稳性、学习协调与通信的需求，以及训练和部署的扩展问题。对已知强化学习算法进行多智能体改造，例如在线PPO [48, 60], 或离线DDPG [36, 38] 和 DQN [42], , 已在许多问题中取得了良好表现。然而，一些特定问题（例如需要与大量智能体通信的问题）需要专门的算法 [25]。例如，TarMAC [17],使用注意力机制，根据消息的重要性对其进行汇总。

3 问题表述

3.1 环境

该环境是一个包含 N栋房屋的聚合仿真，每栋房屋都配备一台空调（AC）设备。假设每栋房屋的室外温度$T_{o,\ t}$都相同，即这些房屋位于同一地理区域内，并将室外温度仿真为周期为一天的正弦变化。除非另有说明，最高温度为34摄氏度，于下午6点达到；最低温度为28摄氏度，于上午6点达到。$T_{o,\ t}$ 因此始终高于目标室内温度 T_T 20摄氏度，使每个家庭都能向电网提供灵活性。环境模型每4秒更新一次。将恒温负荷建模为多区域设备，或为其配备多台空调 [7] , 是未来的研究方向。有关环境的更多详情见附录C。附录A提供了符号表。

3.1.1 住宅热模型。每栋住宅 $i = 1,\ 2,\ \ldots\ldots,\ N$均采用基于 Gridlab-D Residential 模块用户指南 [24]的二阶模型进行仿真。在时间 t，室内空气温度$T^i_{h,\ t}$和

[1]代码托管在 https://github.com/ALLabMTL/MARL_，用于_快速_时间尺度_需求响应。

热质量温度$T^i_{m,\,t}$根据住宅特征 θ^i_T（墙体热导 U^i_h、热质量 C^i_m、空气热容C^i_h 和热容表面热导 H^i_m）、室外温度$T_{o,\,t}$以及空调移除的热量$Q^i_{a,\,t}$进行更新。默认情况下，每栋住宅的热特性相同，均对应一栋面积为 100 平方米、1 层且具有标准隔热性能的住宅。在训练和部署期间，初始热质量温度和空气温度通过在目标温度上添加正向随机噪声来设定。虽然默认情况下不使用太阳得热，但也可以将其加入仿真中，参见附录 C.1.1。

3.1.2 空调。 再次依据 Gridlab-D 指南 [24]，对空调 i 的制冷能力$Q^i_{a,\,t}$和功耗$P^i_{a,\,t}$进行模拟，模拟基于空调特性 θ^i_a，包括制冷量 K^i_a、性能系数COP^i_a和潜冷比例 L^i_a。模型和参数也在附录 C.2 中进行了说明。此外，为保护压缩机，设置了一项严格的动态约束：压缩机关闭后，必须等待规定时间才能再次启动 [61]。这项约束称为锁定。默认情况下，锁定时长 l^i最大值 设为 40 秒。

3.1.3 调节信号。 电力系统运营商向聚合商发送一个信号 ρ_t，该信号涵盖完整的聚合负荷消耗：我们无法控制的系统负荷，例如计算机、洗衣机或照明设备，以及灵活用电，在本例中即空调。令$\rho_t = D_{o,\,t}+s_t$ 其中$D_{o,\,t}$表示不可控负荷的电力需求，s_t 表示目标聚合空调用电，即灵活负荷。我们将$D_{a,\,t}$ 定义为空调满足热目标所需的功率，也就是说，使温度保持在目标值附近所需的功率。为了聚焦于发电功率的高频变化，我们假设s_t在低频下表现平稳，即其在 5 分钟尺度上的均值为$D_{a,\,t}$。我们添加一个均值为 0 的高频变化项$\delta_{s,\,t}$，以表示聚合商希望缓解的可再生能源间歇性。我们将调节信号建模为$s_t = D_{a,\,t} + \delta_{s,\,t}$。

聚合灵活用电是所有空调消耗的总和：$P_t = \sum_i^N P^i_{a,\,t}$。目标是协调聚合中的空调，使 P_t 跟踪 s_t。

基准信号。 为了计算所需的平均功率 $D_{a,\,t}$，我们创建了一个数据集，其中包含在相关参数的离散集合的所有组合下，由无需锁定的 Bang-Bang控制器（对于温度而言是最优的）在 5 分钟时段内所需的平均功率。每个时间步，我们从该数据集中对每台空调的平均电力需求进行插值，并将其相加以计算 $D_{a,\,t}$。实际应用中，基准信号将根据历史数据进行估算或获取。随后，聚合商在承诺跟踪信号 s_t 时会考虑其值。这样可以确保所需的功率调节足以让房屋保持在可接受的温度范围内，同时为电网提供灵活性。

对高频变化进行建模。高频变化 $\delta_{s,\,t}$ 采用一维柏林噪声 [31], 建模，这是一种平滑、通过程序生成的零均值噪声。柏林噪声产生$\delta_{p,\,t} \in [-1, 1]$，并且我们有 $\delta_{s,\,t} = D_{a,\,t}\beta_p\delta_{p,\,t}$ 其中 β_p 是一个设为 0.9 的振幅参数。我们的柏林噪声由 5 个倍频程构成，每个周期为 400 秒，且每个倍频程包含 5 个步长；因此它是以下噪声之和

周期分别为 80、40、20、10 和 5 秒。更多详情见附录 C.3.2。

3.1.4 智能体之间的通信。 为了实现智能体之间的协调，它们必须能够相互通信。为了使智能体的实现具有去中心化、灵活和隐私保护的特点，我们采用有限的局部通信。例如，这使设备能够通过简单的射频发射器进行通信，而无需任何额外的基础设施。因此，我们将通信范围限制在 N_c 个邻居之内。这与低部署投资的论点一致，即使用需求响应进行频率调节。

3.2 去中心化部分可观测马尔可夫决策过程

在本节中，我们将上述环境形式化为一个去中心化的部分可观测马尔可夫决策过程（Dec-POMDP），其特征由元组 S，A，O，P，R γ 表示。令 S 为全局状态，$\mathrm{A} = \prod_{i=1}^N \mathrm{A}^i$ 为联合动作空间，$\mathrm{O} = \prod_{i=1}^N \mathrm{O}^i$ 为联合观测空间。O^i 对 S 进行部分观测。P 描述环境的转移概率，R 表示各智能体的奖励函数，γ 则为折扣参数。

3.2.1 状态、转移概率和动作。智能体并不知道环境状态 $X \in \mathrm{S}$ 及其转移概率 P。它们由第 3.1 节所述的环境动力学进行模拟。每个智能体 i 的动作 $a^i_t \in \mathrm{A}^i$ 是一个用于控制空调状态的二元决策。如果剩余锁定时间 l^i_t 大于零，空调将忽略开启动作。在实际应用中，空调内置的备用控制器会阻止开启决策的执行。

3.2.2 观测与通信。默认情况下，智能体 i 在时间步 t接收观测 $o^i_t = \{T^i_{h,t}, T^i_{m,t}, T^i_T, \omega^i_t, l^i_t, s_t/N, P_t/N\}$，其中 $T^i_{h,t}$, $T^i_{m,t}$和 T^i_T分别表示室内空气温度、热质量温度和目标温度，ω^i_t表示空调的开关状态，l^i_t表示其剩余锁定时间，s_t/N表示每个智能体的调节信号，P_t/N表示聚合体中每个智能体的总能耗。

每个智能体 i 与其 N_c 邻居通信。消息的大小没有严格限制，但应尽量小，且其内容不受限制。我们将智能体 i的所有 N_c邻居构成的集合定义为 M^i。默认情况下，我们将智能体组织成一维结构：

$$M^i = \{i-\lfloor N_c/2 \rfloor, \ldots, i-1, i, i+1, \ldots, i+\lfloor N_c/2 \rfloor\}。$$

3.2.3 奖励。对于每个智能体 i，奖励 r^i_t 是对两项惩罚加权求和后计算得出的：一项是其空气温度与目标值之间的偏差，该目标值因智能体而异；另一项是对信号的跟踪误差，该误差对所有智能体都相同。因此，这是一个具有个体约束的合作场景。我们使用 $\alpha_{\text{温度}} = 1$ 和 $\alpha_{\text{信号}} = 3 \times 10^{-7}$对奖励进行归一化：0.5 摄氏度的误差与每个智能体 912 瓦的误差受到同等惩罚（每个智能体的耗电量为 6000 瓦）。

$$r^i_t = -\left(\alpha_{\text{temp}}\left(T^i_{h,t} - T^i_{T,t}\right)^2 + \alpha_{\text{sig}}\left(\frac{P_t - s_t}{N}\right)^2\right)$$

4 经典与基于学习的算法

4.1 经典基线

据我们所知，目前没有一种经典基线方法能在第 1 节列出的所有约束下都表现良好。不过，简单的算法可以优化特定目标，因此我们将其作为多智能体强化学习智能体结果的基线方法。

*4.1.1 Bang-Bang*控制器。 Bang-Bang控制器（BBC）会在空气温度$T^i_{h,\ t}$高于目标值 T^i_T时开启空调，低于目标值时关闭空调。这是一种不考虑需求响应、但能近乎最优地控制温度的去中心化算法。当锁定时长 l^imax 为 0 时，BBC 能最优地控制温度，但不考虑信号。由于基准信号 $s_{0,\ t}$ 的计算旨在实现最优温度控制，BBC 的信号跟踪误差主要源于信号的高频变化。

4.1.2 贪心短视。 贪心控制器是一种集中式算法，用于求解背包问题 [16]，其中集合的容量为调节信号，每台空调的重量为其消耗 $P^i_{h,\ t}$，其值为温度差 $T^i_{h,\ t} - T^i_T$。在每个时间步，根据 $(T^i_{h,\ t} - T^i_T)/P^i_{h,\ t}$计算出的价值优先级选择空调，直到聚合消耗 P_t高于调节信号 s_t。由于不对未来进行规划，贪心短视方法会很快耗尽可用空调，因为大多数空调都处于锁定状态。然而，当锁定时长l^imax为 0 时，该方法能够最优地跟踪信号 s_t，并将温度控制作为次要目标。我们采用贪心短视方法，因为与第 2 节所述的 OO 方法相比，它更适合这些设置。事实上，OO 仅使用过去的状态信息，必须以严格的在线方式实现。两种框架都具有短视性，在处理锁定约束时面临类似的困难。

4.1.3 模型预测控制。 模型预测控制（MPC）在其标准形式下是一种集中式算法，它对环境进行建模，并确定在 H 个时间步的时域内能够带来最高奖励总和的动作。由于信号具有随机性，MPC 假定时域 H 内未来信号保持恒定，并在考虑锁定的情况下求解最优轨迹。然而，由于这是一个大规模组合优化问题，其可扩展性会随着智能体数量 N 和时域 H 的增大而变差。最理想情况下，复杂度为多项式级，但最坏情况下则呈指数级。因此，我们无法在超过 10 个智能体的情况下运行 MPC $H = 60$ 秒，并且不得不将每个动作之间的时间步长增至 12 秒。更多详情见附录 D.1。

4.2 基于学习的方法

我们部署了两种基于深度强化学习的算法，即 MA-DQN 和 MA-PPO，二者均采用 CT-DE 范式。MA-DQN 仅使用手工设计的通信，而 MA-PPO 则实现了两种通信范式：手工设计的通信和学习得到的通信。有关架构和超参数的详细信息见附录 D.2。

4.2.1 集中式训练、分散式执行。 CT-DE 范式 [28] 假设在智能体训练期间共享信息，而智能体仅根据各自的分散式观测执行动作。这可以降低环境的非平稳性 [21] 并稳定训练。在我们的研究中，所有智能体都是同质的，因此可以使用参数共享 [22]。因此，所有空调均由同一策略的相同实例控制，该策略基于所有智能体的共享经验进行训练。

4.2.2 MA-DQN。 多智能体深度Q网络（MA-DQN）是 DQN 在 CT-DE 中的改编版本。DQN 是一种用于离散动作空间的离策略算法。DQN 智能体主要由一个 Q-网络构成，用于预测每个可能 a^i_t 对应的动作-观测对（$a^i_t, \tilde{o}^i_t$）的 Q 值。在训练过程中，每个智能体在时间步 t 的转移 $\Theta^i_t = \{\tilde{o}^i_t, a^i_t, r^i_t, \tilde{o}^i_{t+1}\}$ 都会记录在一个共享的经验回放缓冲区中。从该缓冲区采样数据，以训练 Q-网络预测 $Q(a^i_t, \tilde{o}^i_t)$，并根据贝尔曼最优性方程，以目标值 $T(a^i_t, \tilde{o}^i_t)$ 进行监督：

$$T(a^i_t, \tilde{o}^i_t) = r^i_t + \gamma \max_a Q(a, \tilde{o}^i_{t+1}).$$

给定输入 $\tilde{o}^i_t$，选择预测 Q-值最大的动作 a^i_t。训练期间加入 ϵ-贪心探索。

4.2.3 MA-PPO。 多智能体近端策略优化（MA- PPO） [60] 是裁剪 PPO [48], 这一在策略的策略梯度算法在 CT-DE 中的适配版本。智能体联合学习策略$\pi_\theta(a^i_t|\tilde{o}^i_t)$（也称为演员）和价值函数 $V_\phi(\tilde{o}^i_t)$（也称为评论家）。在每个训练轮次中，策略保持固定，所有智能体的转移 $\Theta^i_t = \{\tilde{o}^i_t,\ a^i_t,\ \pi_{\theta_t}(a^i_t|\tilde{o}^i_t)\ r^i_t\}$会在一个或多个长度为 H 的回合 Θ^i_t 中一并记录。对于每个 $G^i_t = \sum_{\tau=0}^{H-t} \gamma^\tau r^i_{t+\tau}$，根据未来经验计算回报。随后，在存储的记忆上训练新的策略参数 θ_{t+1}，以优化裁剪PPO目标函数 $L(\tilde{o}^i_t,\ a^i_t,\ \theta_{t+1},\ \theta_t)$，在先前策略附近的接近性约束下最大化优势 $A^{\pi_{\theta_t}}(\tilde{o}^i_t,\ a^i_t) = G^i_t - V_\phi(\tilde{o}^i_t)$。然后训练评论家参数 ϕ，使得 $V_\phi(\tilde{o}^i_t)$ 预测回报 G^i_t。擦除记忆并开始新的训练轮次。

探索由策略固有的随机性实现。在 CT-DE 设置中，V_ϕ 仅在训练期间使用，并获得有关其他智能体状态的额外信息。

4.2.4 通信。

手工设计的通信。 对于 MA-DQN 和采用手工设计通信的 MA-PPO，消息根据每个智能体的状态进行设计，从而有效提高了对整体状态的可观测性。智能体 j 的消息$m_{j,\ t}$包含其空气温度与目标温度之间的当前差值 $T^j_{h,\ t} - T^j_T$、剩余锁定时间 l^j_t 以及当前状态 ω^j_t。来自智能体 $j \in M^i$ 的消息 $\{m^i_{j,\ t}\}_{\forall j \in M_i}$与观测 o^i_t 拼接，作为神经网络的输入 $\tilde{o}^i_t$。来自智能体j、发送给智能体 i的消息$m^i_{j,\ t}$在 $\tilde{o}^i_t$向量中的位置固定，由其相对位置 $i - j$ 决定。采用手工设计通信的 MA-PPO 称为 MA-PPO-HE。

定向多智能体通信。 为了让智能体学会通信，我们在 MA-PPO 中实现了 TarMAC [17]。TarMAC

是一种基于注意力的定向通信算法，每个智能体都会输出一个键、一条消息和一个查询。键会与消息一同发送给其他智能体，其他智能体再将其与自己的查询相乘，以计算分配给该消息的注意力权重。随后，所有消息都会以注意力权重为权重进行聚合。键、消息和查询这三个模块均经过训练。TarMAC 为智能体提供了更大的灵活性：它不限制通信内容，也允许智能体与数量不同于训练期间的房屋进行通信。更多详情请参阅附录 D.2.1。我们将此版本称为 TarMAC-PPO。

不进行通信。 也可以在不进行通信的情况下训练智能体。在这种情况下，它只能观察到 o_t^i。该智能体称为 MA-PPO-NC。

4.2.5 智能体训练。 学习型智能体在具有 $N_{\text{tr}} = \{10, 20, 50\}$ 栋房屋、并与 $N_{\text{c}_{\text{tr}}} = \{9, 19, 49\}$ 个其他智能体通信的环境中接受训练。我们使用 16 个不同的随机种子训练每个智能体：其中 4 个用于环境，4 个用于网络初始化。训练共进行了 3286800 个时间步，相当于 152 天，并分为 200 个回合。每个回合开始时，每栋房屋的温度都高于目标值；该温度取自均值为 0、标准差为 $\sigma = 5$ 摄氏度的高斯分布的绝对值。我们通过网格搜索调整超参数，具体见附录 D。本文旨在证明，基于学习的方法能够在高频调节问题上取得优异性能。因此，我们不汇总训练后智能体的统计数据；相反，对于每种情况，我们根据测试回报从各随机种子中选出两个表现最好的智能体，并报告这两个智能体在基准环境中的最佳得分。

5 结果与分析

5.1 性能指标

我们在一个基准环境中部署智能体，在包含 N_{de} 房屋的轨迹上运行 43200 步，即完整的两天。我们使用调节信号 s_t 与聚合功率消耗 P_t 之间的每个智能体均方根误差（RMSE）来评估其性能。我们还测量温度均方根误差——一个针对所有智能体的温度均方根误差，以及一个针对聚合最大温度误差的温度均方根误差——以确保热控制。每栋住宅的温度初始化均不相同，因此我们在温度得到控制后，即运行 5000 步后，开始计算均方根误差。作为参考，单台空调开启时的功耗为 6000 瓦。由于模型预测控制的计算时间，其性能的评估方式有所不同，具体说明见附录 D.1。除非另有说明，结果均为 10 个环境随机种子上的均值和标准差。

5.2 智能体的性能

表1展示了在有锁定和无锁定的环境中，不同智能体在包含 N_{de} 10、50、250 和 1000 栋房屋时的性能。

每个智能体的信号均方根误差通常会随着 N_{de} 的增加而降低。这是因为相对离散化误差较小，同时智能体数量越多，误差彼此抵消的机会也越大，附录 E 中对此有解释。与预期一致，BBC 能很好地控制温度，但无法跟踪信号。没有锁定时，贪心短视方法能够实现接近最优的信号跟踪，其中误差是由离散化造成的。它也能很好地控制温度。然而，启用锁定后，它会因可用智能体耗尽而失败。对于 10 个智能体，模型预测控制效果良好，但其性能受限于较低的 12 秒控制频率。由于计算时间方面的原因，无法在 $N_{\text{de}} = 50$ 上运行。DQN 能很好地控制温度，但在信号跟踪方面仅略优于 BBC。两种 PPO智能体的表现都显著更好，并且在 N_{de} 较高时，TarMAC-PPO 优于 MA-PPO-HE。关于无通信时的结果将在第 5.5 节讨论。

图 1 展示了 50 栋房屋中各智能体两天内的表现。曲线上的每个点均为 10 分钟内的平均值。平均偏移通过对差值取平均来反映误差的偏差，使正负值相互抵消；平均误差则是绝对差值的平均值。由于初始状态的影响，信号和消耗曲线起初都很高，随后呈现出与室外温度相同的正弦变化模式。未启用锁定时，BBC 的温度偏移和信号偏移较小，但信号误差显著，因为它无法跟踪信号的高频变化。启用锁定后，如第 4.1.1 节所述，它的消耗不足，导致温度出现正偏移，而基准信号则上升以进行补偿。由于信号变化的幅度较大，这对误差的影响并不明显。

DQN 智能体的信号偏移和误差较小，尤其是在信号变化幅度较低的夜间。白天的信号误差仍然显著。另一方面，两个 MA-PPO 智能体的信号和温度偏移都接近 0。它们的信号误差也显著低于其他智能体，因为它们能够跟踪高频变化。

5.3 随智能体数量增加的可扩展性

如表1所示，近端策略优化智能体，尤其是 TarMAC-PPO，能够随着智能体数量的增加平稳扩展。图 2 展示了智能体部署在 $N_d = 50$ 个住宅和 1000 个住宅中时，800 秒内的消耗和信号。对于 $N_d = 50$，智能体无法完美匹配信号。不过，同一智能体在 1000 个住宅中表现更好。事实上，由于环境同质，局部策略通过平均误差能够平稳扩展。TarMAC-PPO 中表现最佳的智能体是在包含 $N_{\text{tr}} = 10$ 个住宅的环境中训练的。对于 MA-PPO-HE，在 $N_{\text{tr}} = 20$ 上训练的智能体往往表现最佳。如图 3 所示，使用 $N_{\text{tr}} = 50$ 进行训练可能会使信用分配更加困难。

5.4 PPO智能体的动态特性

如图4所示，MA-PPO-HE 和 TarMAC-PPO 的策略都会让空调保持在锁定状态或开启状态，从不关闭。这种做法有利于温度控制：需要关闭设备以便在锁定期后升温的智能体，无法在锁定期间完成升温，因此之前就会开启过久。智能体一旦可以开启便立即开启，但会根据环境和其他智能体发送的消息来控制关闭时机。

学习到的策略有一个引人入胜的特点：MA-PPO-HE智能体通过循环行为进行协调。如图4所示，空调根据其在聚合中的位置依次开启，并呈现重复模式。我们训练的每个 MA-PPO-HE智能体都出现了这种现象，不过模式周期或移动方向各不相同。这些模式使得

表1：不同智能体的性能，在 10 个环境随机种子下计算。

每个智能体 RMSE		N_{de} = 10 信号 (W)	T. (°C)	最高温度 (°C)	N_{de} = 50 信号 (W)	T. (°C)	最高温度 (°C)	N_{de} = 250 信号 (W)	T. (°C)	最高温度 (°C)	N_{de} = 1000 信号 (W)	T. (°C)	最高温度 (°C)
无 l.o	贪心	194 ± 1	0.04	0.06	70 ± 1	0.03	0.05	63 ± 1	0.03	0.052	63 ± 1	0.03	0.05
	BBC	806 ± 147	0.02	0.03	392 ± 50	0.02	0.04	310 ± 11	0.02	0.03	272 ± 12	0.02	0.03
40秒 l.o	贪心	2668 ± 14	0.87	0.93	3166 ± 12	1.09	1.15	3313 ± 12	1.16	1.22	3369 ± 15	1.18	1.24
	BBC	830 ± 207	0.05	0.09	426 ± 63	0.05	0.10	318 ± 7	0.05	0.10	296 ± 4	0.05	0.10
	MPC	344 ± 96	0.07	0.12	-	-	-	-	-	-	-	-	-
	MA-DQN	541 ± 86	0.05	0.09	321 ± 24	0.05	0.10	246 ± 8	0.05	0.11	234 ± 4	0.05	0.12
	MA-PPO-HE	253 ± 1	0.04	0.08	161 ± 8	0.04	0.08	127 ± 2	0.04	0.11	122 ± 3	0.05	0.13
	TarMAC-PPO	**247 ± 3**	**0.04**	**0.07**	**158 ± 2**	**0.04**	**0.09**	**115 ± 1**	**0.05**	**0.13**	**101 ± 2**	**0.05**	**0.14**
	MA-PPO-NC	434 ± 2	0.06	0.08	215 ± 1	0.06	0.14	132 ± 1	0.06	0.16	107 ± 1	0.06	0.17

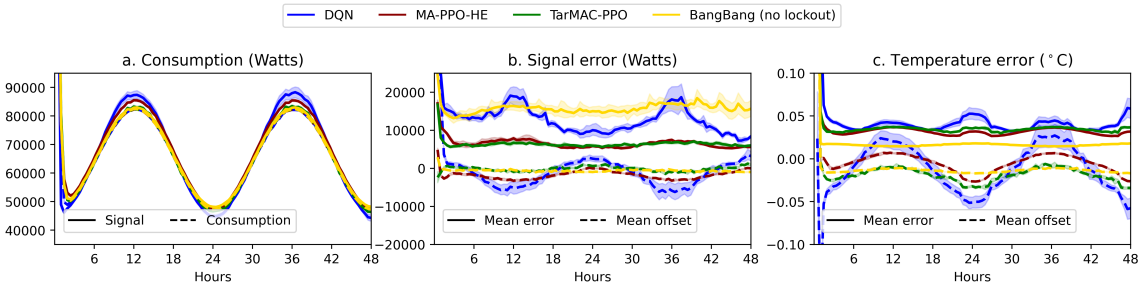


图 1：在 2 天内，MA-PPO-HE 和 TarMAC-PPO 在信号和温度控制方面优于 DQN 和 BBC，使用了 N_{de} = 50个智能体。

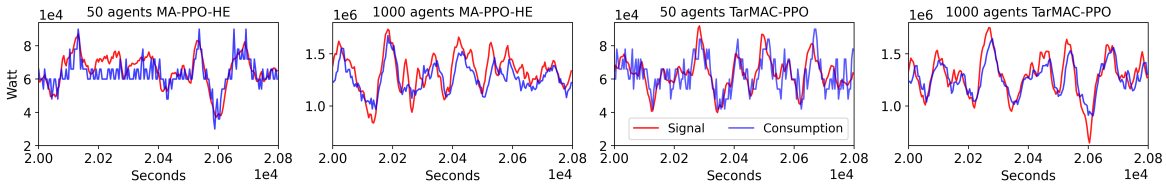


图 2：两种 MA-PPO 策略　　随着智能体数量增加，能够无缝扩展： N_{de} = 50 **an 在 800 秒时的信号和消耗　　d** 1000.

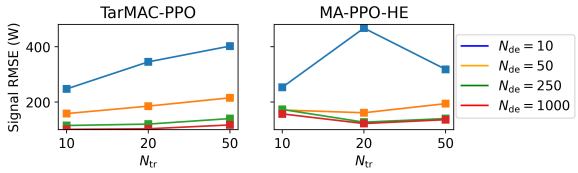


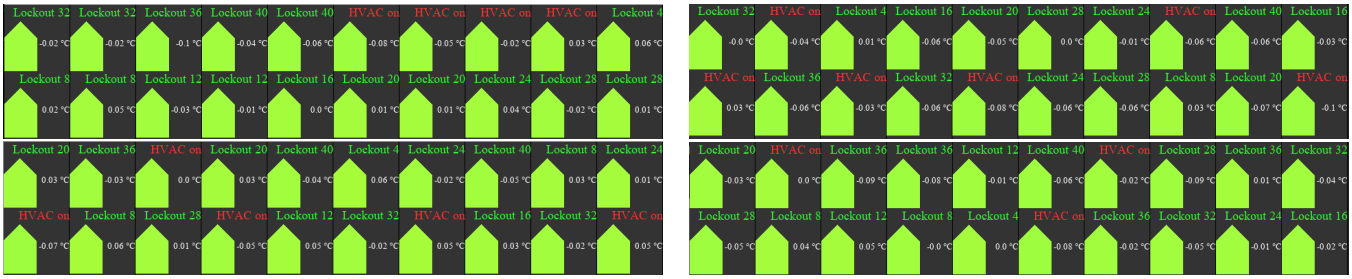
图 3：增加智能体进行训练 N_{tr} 并不会带来更好的性能，即使部署在大型 N_{de} 上也是如此。

智能体能够协调，这得益于稳定的消息结构，即智能体 j 的消息相对于智能体 i 的消息在 o_t^i 向量中的固定位置。另一方面，TarMAC-PPO智能体的集体行为并不遵循某种模式。事实上，聚合消息不包含有关邻居结构的信息。它们通过灵活的消息内容实现协调。

5.5 通信

智能体需要进行通信以协调并取得最佳效果。直观来看，通信对象越多，表现就越好因为环境的可观测性得到了提升。但实际上情况并非总是如此，如图5所示。对于 TarMAC-PPO，与 9 个邻居通信往往能取得最佳表现。较高的 $N_{c_{de}}$ 值可能会导致聚合时重要消息的权重降低。对于 MA-PPO-HE，与 19 个智能体通信比与 49 个通信效果更好。实际上，在 MA-PPO-HE 中，智能体必须具有 $N_{c_{de}} = N_{c_{tr}}$。在训练期间，与更多智能体通信会因输入规模中增加了不可控元素而加大信用分配的难度。图 Figure 5 也清楚地表明，经过通信训练的智能体在不进行通信时表现不佳。图6展示了一个 TarMAC 智能体的表现，该智能体在具有 N_{de} = 50 个智能体的环境中，使用 N_{tr} = 10 和 $N_{c_{tr}}$ = 9 进行训练；图中改变了它能够通信的邻居数量 $N_{c_{de}}$。通信量较低时表现较差，但在约 7 或 8 个智能体时趋于稳定。

然而，正如表1中 MA-PPO-NC 的表现所示，即使没有通信，也可以训练智能体，使其表现优于 Bang-Bang控制。在不与其他智能体协调的情况下，



(a) MA-PPO-HE　　(b) TarMAC-PPO

图4：由两种不同的PPO智能体控制的20栋房屋的状态。右上角的数字表示剩余锁定时间。（左）MA-PPO-HE的两个不同智能体（$N_{c_{de}} = 19$）呈现出"20栋房屋"（上）和"3栋房屋"（下）的模式。（右）两个不同的TarMAC-PPO智能体未呈现这种模式。

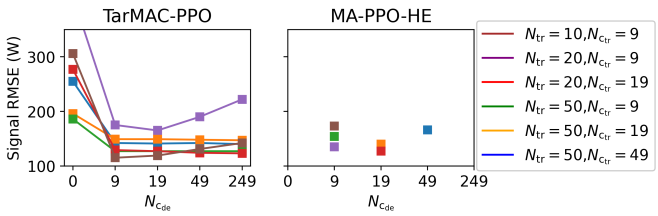


图5：TarMAC-PPO的性能不会随着$N_{c_{de}} = 9$增加，而MA-PPO-HE在$N_{c_{de}} = 19$时表现更好，适用于$N_{de} = 250$个智能体。

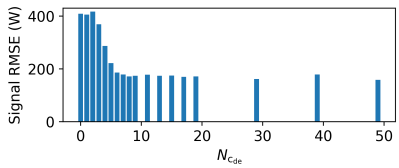


图6：只要TarMAC-PPO智能体与至少$N_{c_{de}} = 7$个智能体通信，其在 $N_{de} = 50$上就能表现良好。

智能体可以学会采取平均而言表现良好的行动，以最大限度地减少信号误差。当智能体数量较少时，例如 $N_{de} = 10$或 50，效果并不理想。然而，随着 N_{de} 增加，性能差距会缩小：良好的平均策略应用于大量智能体时会表现出色。另一种理解方式是，当 N_{de} 很大时，每个智能体对最终结果的重要性都变得微不足道。因此，可以将整个群体视为一个平均智能体，并将问题表述为一个平均场博弈 [50, 59]。

有趣的是，在 N_{de} 较高时，MA-PPO-HE在通信缺陷下表现更好。这可能是因为MA-PPO-HE的协调会导致局部偏差的策略，而这些策略无法从平均效应中获益；平均效应能在 N_{de} 增加时降低相对误差。

5.6 鲁棒性

所有呈现的结果都是在某些假设条件下得出的，例如住宅和空调同质、室外温度和信号曲线一致，以及通信无故障。如果这些智能体部署到现实世界后，将会面临这些条件不再成立的情况。本节评估训练好的智能体在部署条件受到不同扰动时的鲁棒性。

5.6.1 通信故障。 如前所示，通信对于智能体取得良好表现至关重要。在这项鲁棒性测试中，我们模拟了通信故障。每个时间步，每条消息 m_j^i 以概率 p_d 发生故障。对于 TarMAC-PPO，这会导致消息无法被接收。对于 MA-PPO-HE，消息的每个元素都会被设为 0。我们测试了在 $N_{de} = 10$ 下，针对 50、250 和 1000 栋住宅的最佳智能体，故障概率分别为 $p_d = 0.1$ 和 0.5，结果见表 2。MA-PPO-HE 智能体的协调依赖于其稳定的通信结构。因此，它们难以应对通信故障。有趣的是，当 N_{de} 较高时，故障的影响会减小，甚至在 $N_{de} = 1000$ 时带来更好的表现。这可能是因为由此产生的策略无法进行局部协调，且局部偏差较小。TarMAC-PPO 能完美应对暂时的通信故障，因为它会聚合消息。即使在 $p_d = 0.5$ 的情况下，以及智能体仅与 $N_{c_{tr}} = 9$ 个邻居通信时，情况也是如此。

5.6.2 异质住宅与空调。在现实中，不同住宅具有不同的热特性。空调的额定功率或锁定时长也并非总是相同。我们将训练效果最佳的 MA-PPO-HE 和 TarMAC-PPO 智能体部署到不符合这些假设的 50 户环境中，以评估它们对单独扰动的鲁棒性。我们还在具有这些条件的环境中训练了新的智能体，使其能够学会应对异质性。两个智能体都将相关特征作为 o_t^i 的一部分进行观测；在 MA-PPO-HE 中，这些特征也包含在消息 m_j^i 中。这些智能体以 -T 后缀标识。我们通过为每栋住宅的 θ_h^i 中的每个元素添加高斯噪声来模拟热特性的异质性，标准差为原始值的 50%（最终值不能为负）。对于空调的制冷能力 K_a^i，则从 10、12.5、15、17.5 和 20 千瓦中为每栋住宅均匀随机选择一个值。最后，通过在 32、36、40、44 和 48 秒之间均匀采样，测试了锁定时长 l_{max} 的异质性。

结果如表3所示。与 MA-PPO-HE 相比，TarMAC-PPO 对智能体异质性的鲁棒性强得多。这是因为

表 2：故障通信条件下的性能（5 个随机种子）

		$N_{\text{的}} = 10$			$N_{\text{的}} = 50$			$N_{\text{的}} = 250$			$N_{\text{的}} = 1000$		
每个智能体 RMSE		信号 (W)	T. (°C)	最高温度 (°C)	信号 (W)	T. (°C)	最高温度 (°C)	信号 (W)	T. (°C)	最高温度 (°C)	信号 (W)	T. (°C)	最高温度 (°C)
MA-PPO-HE	$p_d = 0$	253 ± 1	0.04	0.08	161 ± 8	0.04	0.08	127 ± 2	0.04	0.11	122 ± 3	0.05	0.13
	$p_d = 0.1$	504 ± 2	0.07	0.14	207 ± 1	0.04	0.11	138 ± 2	0.05	0.13	118 ± 1	0.05	0.14
	$p_d = 0.5$	597 ± 2	0.10	0.19	274 ± 1	0.06	0.15	148 ± 1	0.06	0.151	115 ± 2	0.06	0.17
TarMAC-PPO	$p_d = 0$	247 ± 3	0.04	0.07	158 ± 2	0.04	0.09	115 ± 1	0.05	0.13	101 ± 2	0.05	0.14
	$p_d = 0.1$	246 ± 2	0.04	0.07	158 ± 2	0.04	0.09	115 ± 2	0.05	0.12	101 ± 1	0.05	0.14
	$p_d = 0.5$	248 ± 2	0.04	0.07	159 ± 3	0.04	0.09	115 ± 2	0.05	0.13	101 ± 1	0.05	0.14

表3：住宅和空调异质性下的性能

	MA-PPO-HE		MA-PPO-HE-T	
每个智能体 RMSE	信号 (W)	最高温度 (°C)	信号 (W)	最高温度 (°C)
同质	161 ± 8	0.08	-	-
房屋热特性	285 ± 8	0.17	222 ± 7	0.11
空调制冷	292 ± 3	0.15	181 ± 3	0.14
锁定时长	324 ± 9	0.15	246 ± 4	0.09
	TarMAC-PPO		TarMAC-PPO-T	
同质	158 ± 2	0.09	-	-
房屋热特性	184 ± 2	0.12	174 ± 2	0.11
空调制冷	187 ± 2	0.16	185 ± 9	0.16
锁定时长	192 ± 3	0.09	251 ± 4	0.08

表4：环境变化下的鲁棒性（5 个随机种子）

	MA-PPO-HE		TarMAC-PPO	
每个智能体 RMSE	信号 (W)	最高温度 (°C)	信号 (W)	最高温度 (°C)
与训练时相同	161 ± 8	0.08	158 ± 2	0.09
太阳得热	190 ± 6	0.09	174 ± 2	0.10
室外温度 + 4摄氏度	203 ± 4	0.11	198 ± 2	0.11
室外温度 - 4摄氏度	170 ± 1	0.09	184 ± 2	0.12
信号平均值 + 30%	401 ± 2	0.11	302 ± 2	0.14
信号平均值 - 30%	337 ± 4	0.10	317 ± 1	0.11
信号噪声幅度 + 30%	188 ± 5	0.08	179 ± 3	0.09
信号噪声频率 + 100%	200 ± 4	0.08	198 ± 5	0.09

MA-PPO-HE 的协调方案基于智能体邻居的稳定动力学，尤其是锁定时长。相比之下，TarMAC-PPO 对不同的动态特性更加灵活。对于这两种智能体，都可以通过在此类环境中训练智能体，并让它们观测这些特征，来减轻异质性的影响。对于锁定时长的异质性，情况则有所不同：TarMAC-PPO 似乎无法在此类条件下得到令人满意的训练。一个有趣的观察是，TarMAC-PPO 在与 $N_{\text{c}_{\text{tr}}} = 49$ 个智能体通信时取得了最佳结果。智能体存在异质性时，需要更多邻居才能获得具有代表性的输入。

5.6.3 其他环境。我们还在不同于训练环境的环境中测试了智能体，这些环境具有不同的室外温度 T_o、太阳得热 Q_s、过低或过高的平均信号 D_a，以及更高或变化更快的信号变化 δ_s。如表4所示，两个智能体都能很好地适应这些变化，其中 TarMAC-PPO 通常表现更好。当信号异常时，也就是说，信号过低或过高，无法正确控制温度时，信号与温度目标之间存在权衡。MA-PPO-HE 更优先考虑温度，因此信号均方根误差更高。

5.7 处理时间

表5报告了基线和训练后智能体的动作选择处理时间。结果显示的是25个时间步（100秒仿真）的数据，模型预测控制除外；模型预测控制仿真了100秒，共10个时间步。计算所用设备为笔记本电脑上的12核、2.2 GHz英特尔 i7-8750H CPU。

表5：仿真100秒时动作选择的计算时间（秒）。对于去中心化系统，我们报告每个智能体的耗时；对于其他系统，则报告整个系统的耗时。

智能体	去中心化	$N_{\text{去}} = 10$	$N_{\text{去}} = 1000$
TarMAC-PPO	Yes	0.002	0.001
MA-PPO-HE	Yes	0.006	0.006
DQN	Yes	0.003	0.002
BBC	Yes	0.00001	0.00001
贪心短视	No	0.1	3.7
模型预测控制 - $H = 40$秒	No	92.6680	-

由于去中心化的学习型智能体只需在规模相对较小的神经网络中进行一次前向传播，因此使用 4 秒时间步时，动作选择所需的时间足够短，可以满足控制要求。贪心短视等集中式方法在智能体数量较多时扩展性较差。模型预测控制已进行简化，采用 12 秒而非 4 秒的时间步，时域也缩短至 40 秒；但当智能体超过 10 个时，所需时间仍长得无法接受。

6 结论

本文针对高频调节需求响应问题，采用由 MA-PPO 训练的去中心化实时智能体，对配备空调、具有离散且动态约束的住宅负荷进行控制。我们测试了两种本地通信框架——固定的人工设计消息和学习得到的定向通信。使用少量智能体训练的策略显著优于基线方法，可无缝扩展至大量房屋，并且对大多数扰动具有鲁棒性。我们的结果表明，多智能体强化学习可用于成功解决电力电网中因整合可再生能源而产生的一些复杂多智能体问题。未来要将此类算法应用于真实电力系统，可开展仿真到现实迁移、整合更复杂的灵活负荷，以及研究电网安全问题。

参考文献

[1] 国际能源署。 [无日期]。能源统计数据浏览器——数据工具。网址：https://www.iea.org/data-and-statistics/data-tools/energy-statistics-data-browser（访问日期：2022年9月15日）。[2] 国际能源署。2018年。《制冷的未来》。https://www.iea.org/reports/the-future-of-cooling [3] 国际能源署。2021年。《能源相关温室气体排放：概述》。[4] Roya Ahmadiahangar、Tobias Häring、Argo Rosin、Tarmo Korõtko 和 João Martins。2019年。使用基于机器学习的回归模型进行住宅负荷预测以预测灵活性。载于《2019年IEEE环境与电气工程国际会议暨2019年IEEE欧洲工业与商业电力系统会议》（EEEIC / I&CPS Europe）。1–4。https: //doi.org/ 10 1109 2019 8783634 。/EEEIC。。Mehdi Ahrarinouri、[5] Mohammad Rastegar 和 Ali Reza Seifi。2021年。住宅建筑能源管理的多智能体强化学习。IEEE工业信息学汇刊 17, 1（2021年1月），659–666。https://doi.org/10.1109/TII.2020.2977104[6] Sally Aladdin、Samah El-Tantawy、Mostafa M. Fouda 和 Adly S. Tag Eldien。2020年。MARLA-SG：用于智能电网高效需求响应的多智能体强化学习算法。IEEE Access 8（2020），210626–210639。https: //doi.org/ 10 1109 2020 3038863 。/ACCESS。。Uzma Amin、MJ Hossain 和 [7] E Fernandez。2020年。用于多区域办公楼需求响应的暖通空调系统最优电价控制。清洁生产杂志 270（2020），122059。[8] Bowen Baker、Ingmar Kanitscheider、Todor Markov、Yi Wu、Glenn Powell、Bob McGrew 和 Igor Mordatch。2020年。多智能体自动课程学习中涌现的工具使用。（2020年2月）。arXiv:1909.07528 [cs, stat]。[9] Hassan Bevrani、Arindam Ghosh 和 Gerard Ledwich。2010年。可再生能源与频率调节：综述及新视角。IET可再生能源发电 4, 5（2010），438–457。[10] David Biagioni、Xiangyu Zhang、Dylan Wald、Deepthi Vaidhynathan、Rohit Chintala、Jennifer King 和 Ahmed S. Zamzam。2021年。PowerGridworld：用于电力系统多智能体强化学习的框架。https: //doi.org/10.48550/ARXIV.2111.05969[11] Greg Brockman、Vicki Cheung、Ludwig Pettersson、Jonas Schneider、John Schulman、Jie Tang 和 Wojciech Zaremba。2016年。OpenAI Gym。arXiv:arXiv:1606.01540[12] Duncan S Callaway。2009年。利用电力负荷的储能潜力提供负荷跟踪和调节，并应用于风能。能源转换与管理 50, 5（2009），1389–1400。[13] Bingqing Chen、Jonathan Francis、Marco Pritoni、Soummya Kar 和 Mario Bergés。2020年。Cohort：协调异构恒温控制负荷以实现需求灵活性。载于第七届ACM能源高效建筑、城市与交通系统国际会议论文集。31–40。[14] Bingqing Chen、Jonathan Francis、Marco Pritoni、Soummya Kar 和 Mario Bergés。2020年。COHORT：协调异构恒温控制负荷以实现需求灵活性。载于第七届 ACM能源高效建筑、城市与交通系统国际会议论文集。31–40。https://doi.org/10.1145/3408308.3427980 arXiv:2010.03659 [cs, eess]。[15] CIBSE。2015年。《指南A：环境设计》（第8版）。特许建筑设备工程师学会。[16] George B. Dantzig。1957年。离散变量极值问题。运筹学 5, 2（1957年4月），266–288。https://doi.org/10.1287/opre.5.2.266[17] Abhishek Das、Théophile Gervet、Joshua Romoff、Dhruv Batra、Devi Parikh、Mike Rabbat 和 Joelle Pineau。2019年。TarMAC：定向多智能体通信。载于第36届国际机器学习会议论文集。PMLR，1538–1546。https://proceedings.mlr.press/v97/das19a.html [18] Steven Diamond 和 Stephen Boyd。2016年。CVXPY：用于凸优化的嵌入Python的建模语言。机器学习研究杂志（2016）。https://stanford.edu/~boyd/papers/pdf/cvxpy_paper.pdf 即将发表。[19] Ivana Dusparic、Colin Harris、Andrei Marinescu、Vinny Cahill 和 Siobhán Clarke。2013年。基于负荷预测的多智能体住宅需求响应。载于2013年第一届IEEE可持续发展技术会议（SusTech）。90–96。https://doi.org/ 10 1109 2013 6617303 。/SusTech。。Andrew Fuchs、[20] Michael Walton、Theresa Chadwick 和 Doug Lange。2021年。用于深度强化学习的心智理论：以 Hanabi 为例。（2021年1月）。http://arxiv.org/abs/2101.09328 arXiv:2101.09328 [cs]。[21] Sven Gronauer 和 Klaus Diepold。2021年。多智能体深度强化学习：综述。人工智能评论（2021年4月）。https://doi.org/10.1007/s10462-021-09996-w[22] Jayesh K. Gupta、Maxim Egorov 和 Mykel Kochenderfer。2017年。使用深度强化学习进行协作式多智能体控制。计算机科学讲义，第10642卷。Springer国际出版公司，Cham，66–83。https: //doi.org/ 10 1007 978 3 319 71682 4_5 。/ - - - - Gurobi Optimization, LLC。。[23] Gurobi Optimizer参考手册。2022 https: //www.gurobi.com

[24] Betelle纪念研究所。 [无日期]。GridLAB-D Wiki。http://gridlab-d.shoutwiki.com/wiki/Main_Page http://gridlab-d.shoutwiki.com/wiki/Main_Page（访问日期：2022年9月15日）。[25] Jiechuan Jiang 和 Zongqing Lu。2018年。学习用于多智能体协作的注意力通信。载于《神经信息处理系统进展》第31卷。Curran Associates, Inc. https://proceedings.neurips.cc/paper/2018/hash/6a8018b3a00b69c008601b8becae392b-Abstract.html[26] Jin、Mohammed Olama、Teja Kuruganti、James Nutaro、Christopher Winstead、Yaosuo Xue 和 Alexander Melin。2018年。采用模型预测控制调节建筑物暖通空调系统的开关，以补偿太阳能发电波动。载于2018年第9届IEEE分布式发电系统电力电子国际研讨会（PEDG）。 1 5 10 1109 2018 8447840 – 。[27] 2017 https://doi.org/ 。/PEDG。 。Diederik P. Kingma 和 Jimmy Ba。 。Adam：一种随机优化方法。arXiv:1412.6980 [cs]（2017年1月）。http://arxiv.org/abs/1412.6980 arXiv:1412.6980。[28] Landon Kraemer 和 Bikramjit Banerjee。2016年。将多智能体强化学习作为去中心化规划的演练。Neurocomputing 190（2016年5月），82 94 10 1016 2016 01 031 – 。https://doi.org/ 。/j.neucom。 。[29] 2007 Prabha Kundur。 。电力系统稳定性。《电力系统稳定性与控制》（2007年），7–1。[30] Alexandre Lacoste、Alexandra Luccioni、Victor Schmidt 和 Thomas Dandres。2019年。量化机器学习的碳排放。arXiv:1910.09700（2019年11月）。https://doi.org/10.48550/arXiv.1910.09700 arXiv:1910.09700 [cs]。[31] A. Lagae、S. Lefebvre、R. Cook、T. DeRose、G. Drettakis、D.S. Ebert、J.P. Lewis、K. Perlin 和 M. Zwicker。2010年。程序化噪声函数综述。Computer Graphics Forum 29, 8（2010年12月），2579–2600。https://doi.org/10.1111/j.1467- 8659.2010.01827.x[32] Fiorella Lauro、Fabio Moretti、Alfonso Capozzoli 和 Stefano Panzieri。2015年。用于建筑物主动需求响应系统的模型预测控制。Energy Procedia 83（2015年），494–503。https://doi.org/10.1016/j.egypro.2015.12.169 能源与建筑可持续性：第七届国际会议SEB-15论文集。[33] Young M Lee、Raya Horesh 和 Leo Liberti。2015年。结合现场储能与发电系统的需求响应最优暖通空调控制。Energy Procedia 78（2015年），2106–2111。[34] Antoine Lesage-Landry 和 Joshua A Taylor。2018年。部分观测负荷下的设定点跟踪。IEEE电力系统汇刊 33, 5（2018年），5615–5627。[35] Antoine Lesage-Landry、Joshua A Taylor 和 Duncan S Callaway。2021年。带二元约束的在线凸优化。IEEE自动控制汇刊（2021年）。[36] Timothy P. Lillicrap、Jonathan J. Hunt、Alexander Pritzel、Nicolas Heess、Tom Erez、Yuval Tassa、David Silver 和 Daan Wierstra。2019年。利用深度强化学习进行连续控制。（2019年7月）。arXiv:1509.02971 [cs, stat]。[37] M Liu 和 Y Shi。2015年。针对辅助服务的聚合异质二阶恒温控制负荷模型预测控制。IEEE电力系统汇刊 31, 3（2015年），1963–1971。[38] Ryan Lowe、Yi Wu、Aviv Tamar、Jean Harb、Pieter Abbeel 和 Igor Mordatch。2020年。用于混合合作与竞争环境的多智能体演员-评论家。（2020年3月）。http://arxiv.org/abs/1706.02275 arXiv:1706.02275 [cs]。[39] Mehdi Maasoumy、Borhan M Sanandaji、Alberto Sangiovanni-Vincentelli 和 Kameshwar Poolla。2014年。商业建筑向智能电网提供调节服务的模型预测控制。载于2014年美国控制会议。IEEE，2226–2233。[40] J L Mathieu、S Koch 和 D S Callaway。2012年。通过电力负荷状态估计与控制管理实时能源不平衡。IEEE电力系统汇刊 28, 1（2012年），430–440。[41] Volodymyr Mnih、Adrià Puigdomènech Badia、Mehdi Mirza、Alex Graves、Timothy P. Lillicrap、Tim Harley、David Silver 和 Koray Kavukcuoglu。2016年。深度强化学习的异步方法。arXiv:1602.01783 [cs]（2016年6月）。http://arxiv.org/abs/1602.01783 arXiv: 1602.01783。[42] Volodymyr Mnih、Koray Kavukcuoglu、David Silver、Andrei A. Rusu、Joel Veness、Marc G. Bellemare、Alex Graves、Martin Riedmiller、Andreas K. Fidjeland、Georg Ostrovski、Stig Petersen、Charles Beattie、Amir Sadik、Ioannis Antonoglou、Helen King、Dharshan Kumaran、Daan Wierstra、Shane Legg 和 Demis Hassabis。2015年。通过深度强化学习实现人类级别的控制。Nature 518, 75407540（2月 2015 529 533 10 1038 14236 ），– 。https://doi.org/ 。/[43] nature Mohammed M. Olama、Teja Kuruganti、James Nutaro 和 Jin Dong。2018年。协调和控制建筑物暖通空调系统，为电网提供频率调节。Energies 11, 7 2018 10 3390 11071852 （）。[44] https://doi.org/ 。/en OpenAI、Christopher Berner、Greg Brockman、Brooke Chan、Vicki Cheung、Przemysław Dębiak、Christy Dennison、David Farhi、Quirin Fischer、Shariq Hashme、Chris Hesse、Rafal Józefowicz、Scott Gray、Catherine Olsson、Jakub Pachocki、Michael Petrov、Henrique P. d O. Pinto、Jonathan Raiman、Tim Salimans、Jeremy Schlatter、Jonas Schneider、Szymon Sidor、Ilya Sutskever、Jie Tang、Filip Wolski 和 Susan Zhang。2019年。利用大规模深度强化学习玩Dota 2。（2019年12月）。http://arxiv.org/abs/1912.06680 arXiv: 1912.06680。

[45] Aisling Pigott、Constance Crozier、Kyri Baker 和 Zoltan Nagy。2021年。GridLearn：面向电网感知型建筑能源管理的多智能体强化学习。https://doi.org/10 48550 2110 06396 。/ARXIV。 。Zhiwei Qin、Hongtu Zhu 和 Jieping Ye。 。拼车强化学习：扩展综述。（2022年8月）。http://arxiv.org/abs/2105.01099 arXiv:2105.01099 [cs]。[47] Martin Roesch、Christian Linder、Roland Zimmermann、Andreas Rudolf、Andrea Hohmann 和 Gunther Reinhart。2020年。利用多智能体强化学习服务于工业的智能电网。《应用科学》10, 19（2020年）。https://doi.org/ 10 3390 10196900 。/app John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017年。近端策略优化算法。（2017年）。https://doi.org/10.1007/s00038-010-0125-8[49] Pierluigi Siano。2014年。需求响应与智能电网——综述。《可再生与可持续能源评论》30（2014年），461–478。[50] Jayakumar Subramanian、Raihan Seraj 和 Aditya Mahajan。2018年。平均场团队的强化学习。载于国际自主智能体与多智能体系统会议自适应与学习智能体研讨会。[51] Josh A Taylor、Sairaj V Dhople 和 Duncan S Callaway。2016年。无燃料电力系统。《可再生与可持续能源评论》57（2016年），1322–1336。[52] Jose R. Vazquez-Canteli、Sourav Dey、Gregor Henze 和 Zoltan Nagy。2020年。CityLearn：规范化多智能体强化学习在需求响应与城市能源管理领域的研究。（2020年12月）。https://doi.org/10.48550/arXiv.2012.10504 arXiv:2012.10504 [cs]。[53] Jose R. Vazquez-Canteli、Gregor Henze 和 Zoltan Nagy。2020年。MARLISA：通过迭代顺序动作选择，利用多智能体强化学习为电网交互型互联建筑进行负荷塑形。载于第七届*ACM*能源高效建筑、城市与交通系统国际会议论文集（日本，线上活动）*(BuildSys '20)*。计算机协会，纽约，纽约州，美国，170–179。https://doi.org/10.1145/ 3408308.3427604[54] Zhe Wang、Bingqing Chen、Han Li 和 Tianzhen Hong。2021年。AlphaBuilding ResCommunity：用于社区级负荷协调的多智能体虚拟测试平台。《应用能源进展》4（2021年），100061。[55] Hua Wei、Nan Xu、Huichu Zhang、Guanjie Zheng、Xinshi Zang、Chacha Chen、Weinan Zhang、Yanmin Zhu、Kai Xu 和 Zhenhui Li。2019年。CoLight：学习用于交通信号控制的网络级协作。载于第*28*届 *ACM* 国际信息与知识管理会议论文集（*CIKM '19*）。计算机协会，纽约，纽约州，美国，1913–1922。https://doi.org/ 10 1145 3357384 3357902 。/。Xiaoyu Wu、Jinghan He、Yin Xu、Jian Lu、Ning Lu 和 Xiaojun Wang。2018年。住宅暖通空调设备的分层控制，用于一次调频。*IEEE*智能电网汇刊 9，4（2018年），3844–3856。https: //doi.org/10.1109/TSG.2017.2766880[57] Lei Xi、Jianfeng Chen、Yuehua Huang、Yanchun Xu、Lang Liu、Yimin Zhou 和 Yudan Li。2018年。基于时间隧道理念的多智能体强化学习智能发电控制。能源 153（2018年），977–987。https: //doi.org/ 10 1016 2018 04 042 。/j.energy. . . Yaodong Yang、Jianye Hao、Yan Zheng、Xiaotian Hao 和 Bofeng Fu。2019年。使用基于熵的集体多智能体强化学习框架进行大规模家庭能源管理。载于第*18*届国际自主智能体与多智能体系统会议论文集（*AAMAS '19*）。国际自主智能体与多智能体系统基金会，里奇兰，南卡罗来纳州，2285–2287。[59] Yaodong Yang、Rui Luo、Minne Li、Ming Zhou、Weinan Zhang 和 Jun Wang。2018年。平均场多智能体强化学习。（2018年2月）。https: //doi.org/10 48550 1802 05438 。/arXiv. . Chao Yu、Akash Velu、Eugene Vinitsky、Yu Wang、Alexandre Bayen 和 Yi Wu。2021年。近端策略优化在合作式多智能体博弈中的惊人效果。（2021年7月）。arXiv:2103.01955 [cs]。[61] Wei Zhang、Jianming Lian、Chin-Yao Chang 和 Karanjit Kalsi。2013年。用于需求响应的空调负荷聚合建模与控制。*IEEE*电力系统汇刊 28，4（2013年），4655–4664。[62] X Zhou、E Dall'Anese 和 L Chen。2019年。联网分布式能源资源的在线随机优化。*IEEE*自动控制汇刊 65，6（2019年），2387–2401。

A 符号说明

表6列出了本文使用的不同符号。

B 研究项目的碳排放

由于本研究的模型训练和运行消耗了大量电力，我们公布其估算的碳足迹。

实验使用私有基础设施进行，其碳效率为 0.049 $kgCo_2eq/kWh$。累计进行了 10895 天（即 261480 小时）的计算，主要使用 Intel Xeon Processor E5-2683 v4 型中央处理器（热设计功耗为 120 瓦）。我们假设中央处理器的平均功耗为其热设计功耗的一半。

估计总排放量为 628 $kgCO_2eq$，其中 0% 得到了直接抵消。这相当于一辆普通汽车行驶 2550 千米，或燃烧 314 千克煤所产生的排放。

这些估算使用 MachineLearning Impact 计算器 [30] 得出。

C 环境详情

C.1 房屋热特性详细模型

每栋住宅的空气温度分别变化，其变化取决于住宅的热特性 θ_h、当前状态、室外条件（例如室外温度和太阳得热），以及住宅内空调的运行状态。该二阶模型基于 Gridlab-D 的 Residential 模块用户指南 [24]。

使用 Gridlab-D 模块，我们对一栋 8×12.5 米、单层的矩形房屋进行建模，其天花板高度为 2.5 米，配有 4 扇 1.8 $米^2$、双层铝制窗户和 2 扇 2 $米^2$ 木门，由此得到表 7 中所列的以下数值。

为了模拟住宅的空气温度$T_{h,\ t}$及其热质量温度$T_{m,\ t}$的变化，我们假设温度分布均匀，且不考虑住宅内的热传播。我们定义以下变量：

$$
\begin{aligned}
a &= C_m C_h / H_m \\
b &= C_m (U_h + H_m)/H_m + C_h \\
c &= U_h \\
d &= Q_{a,t} + Q_{s,t} + U_h T_{o,t} \\
dT_{A0}/dT &= \left(H_m T_{m,t} - (U_h + H_m) T_{h,t} \right. \\
&\quad \left. + U_h T_{o,t} + Q_{h,t} + Q_{s,t}\right)/C_h .
\end{aligned}
$$

随后计算以下系数：

$$
\begin{aligned}
r_1 &= (-b + \sqrt{b^2 - 4ac})/2a \\
r_2 &= (-b - \sqrt{b^2 - 4ac})/2a \\
A_1 &= (r_2 T_{a,t} - dT_{A0}/dT - r_2 d/c)/(r2 - r1) \\
A_2 &= T_{h,t} - d/c - A_1 \\
A_3 &= (r_1 C_h + U_h + H_m)/H_m \\
A_4 &= (r_2 C_h + U_h + H_m)/H_m .
\end{aligned}
$$

这些系数最终应用于以下动态方程：

$$
\begin{aligned}
T_{a,t+1} &= A_1 e^{r_1 \delta t} + A_2 e^{r_2 \delta t} + d/c \\
T_{m,t+1} &= A_1 A_3 e^{r_t \delta t} + A_2 A_4 e^{r_2 \delta_t} + d/c .
\end{aligned}
$$

C.1.1 太阳得热。可以向仿真器中添加太阳得热。其计算依据为 CIBSE《环境设计指南》[15]。

住宅的照明特性 θ_S，包括窗户面积和 0.67 的遮阳系数，是对太阳得热进行建模所必需的，$Q_{s,\ t}$。

然后，作出以下假设：

- 纬度为 30。
- 在该纬度，上午 7:30 之前和下午 5:30 之后的太阳得热可以忽略不计。
- 窗户均匀分布在建筑物的 4 个朝向上。
- 所有窗户均为垂直窗户。

这使我们能够根据一天中的时间和一年中的日期，计算四次二元多项式的系数，以对住宅的太阳得热进行建模。

C.2 空调详细模型

同样，我们依据 Gridlab-D Residential 模块用户指南[24], 对空调开启时的功耗$P_{a,\ t}$ 以及从空气中获取的热量$Q_{a,\ t}$进行建模，建模依据是空调的特征 θ_H，例如制冷量 K_a、性能系数 COP_a和潜冷比例 L_a。

COP_a和 L_a 被视为常数，并采用指南中的默认值：$COP_a = 2.5$ 和 $L_a = 0.35$。我们有：

$$
\begin{aligned}
Q_{a,t} &= -\frac{K_a}{1 + L_a} \\
P_{a,t} &= \frac{K_a}{COP_a} .
\end{aligned}
$$

我们将 K_a 设为 15 千瓦，即 50 000 BTU/hr，以便即使室外温度较高，也能控制空气温度。这个功率高于大多数住宅空调的功率，但即使在室外温度较高时，也能提供足够的灵活性（室外温度为 38 摄氏度时，一台 5 千瓦的空调必须始终开启，才能将温度维持在 20 摄氏度）。这一选择不会显著影响我们的结果：室外温度较低时，该问题等同于使用较低的空调功率。

C.3 调节信号

C.3.1 基准信号的插值。 如第 3.1.3 节所述，我们通过插值估算$D_{a,t}$。Bang-Bang控制器连续运行 5 分钟，不设置锁定，并计算消耗的平均功率。由此可近似估算特定情况下所需的功率。

通过估算单个住宅在以下参数超过 400 万种组合下的$D_{a,t}$，建立了一个数据库：房屋热特性 θ_h、室内空气温度与热质量温度之间的差值 $T_{a,t}$和 $T_{m,t}$、目标温度 T_T、室外温度 $T_{o,t}$，以及空调的制冷量 K_a。如果在仿真中加入太阳得热，还会考虑一天中的时刻和一年中的日期。

仿真环境时，每隔 5 分钟，$D_{a,t}$ 通过对集群中每栋住宅插值得到的必要消耗求和来计算。除 θ_h 的 4 个元素和 K_a外，大多数参数的插值过程都是线性的，而这两者

表6：符号表

智能体数量	N	集群中的房屋数量（一般）
	N_{tr}	训练环境中的房屋数量
	N_{de}	测试环境中的房屋数量
	$N_{\text{c}_{\text{tr}}}$	训练期间用于通信的智能体数量
	$N_{\text{c}_{\text{de}}}$	部署时用于通信的智能体数量
温度	T_h	室内空气温度
	T_m	室内热惰性体温度
	T_o	室外温度
	T_T	目标室内温度
信号与功率	s_0	基准信号
	ρ	电力系统运营商信号
	s	调节信号
	D_a	空调所需的平均功率
	D_o	不可调负载所需功率
	δ_s	信号变化
	δ_p	柏林噪声
	β_p	变化幅度参数
	P	集群总功耗
空调状态	ω	状态（开启或关闭）
	l	锁定剩余时间
	P_a	功耗
	Q_a	空调移除的热量
住宅热模型	θ_h	房屋热特性
	U_h	外墙热导
	C_m	房屋热容
	C_h	空气热容
	H_m	热容表面热导
	θ_s	房屋照明特性
	Q_s	太阳得热
空调模型	θ_a	空调特性
	K_a	制冷量
	COP_a	性能系数
	L_a	潜冷比例
	$l_{\max}$	锁定时长
部分可观测马尔可夫决策过程	$a, \mathcal{A}$	动作，动作空间
	$o, \mathcal{O}$	观测，观测空间
	$S, \mathcal{S}$	状态，状态空间
	$r, \mathcal{R}$	奖励，奖励函数
	$\mathcal{P}$	转移概率
	M	通信智能体集合
	m_j^i	来自 j 至 i的消息
	$\tilde{o}$	拼接后的观测与消息
	γ	折扣因子
	$\alpha_{\text{温度}}$ $\alpha_{\text{信号}}$	奖励函数中的权重
算法	H	时域
	Θ	转移
	$Q(a, o)$ $T(a, o)$	Q-值预测（使用 Q 或目标网络）
	π_θ	由 θ参数化的策略
	V_ϕ	评论家参数化为 ϕ
	G	回报
	A^π	策略 π的优势

而是使用最近邻来降低该操作的复杂度。

C.3.2 柏林噪声。使用一维柏林噪声来计算 $\delta_{\Pi,\ t}$，即发电的高频分量。专为该领域设计

表 7：默认房屋热参数 θ_h

U_h	2.18×10^2 瓦/K
C_m	3.45×10^6 焦耳/K
C_h	9.08×10^5 焦耳/K
H_m	2.84×10^3 瓦/K

对于计算机图像生成，这种噪声在我们的应用场景中具有几个有趣的特性。

柏林噪声通常由多个称为倍频程的子噪声叠加生成。可以限制这些子噪声取值的范围。因此，可以在考虑多种频率的非规则噪声的环境中测试智能体，同时将噪声值限制在符合现实的范围内。此外，噪声的平均值可以轻松设定且不会发生偏移，从而确保在足够长的时间范围内，噪声的平均值为 0。

每个倍频程由两个参数决定：振幅和频率比。频率表示两个随机偏差之间的距离，振幅表示变化幅度。通常，频率越高，振幅越低。这样，振幅较大的噪声分布在较宽的区间内，而振幅较小的噪声出现得更频繁且分布更集中。

在我们的案例中，我们使用 5 个倍频程，每个倍频程之间的振幅比为 0.9，频率则与倍频程的编号成正比。

D 算法详情

D.1 模型预测控制

我们的模型预测控制基于集中式模型。在每个时间步，利用有关智能体状态的信息，找出未来 H 个时间步内使奖励函数最小化的控制量。随后将最优的即时动作传达给智能体。在每个时间步，算法都会计算 H时间步时域内的理想控制组合。

由于信号和温度的代价函数均以均方根误差为最小化目标，因此该问题被建模为二次混合整数规划。用于求解模型预测控制的求解器是商业求解器 Gurobi [23] 与 CVXPY [18]。Gurobi 是一款需许可的求解器，其确切的内部行为我们并不清楚，因此它对我们的模型预测控制而言是一个黑箱。不过，我们知道它使用分支定界算法求解凸整数问题。求解速度主要取决于求解器启发式算法的质量。

模型预测控制每个步骤所需的计算时间会随着智能体数量和/或 H大幅增加。为了能够用足够多的智能体和滚动时域测试这一方法，并获得合理的性能，必须将智能体做出决策的时间步长增加到 12 秒（其他智能体为 4 秒）。

无法在合理的时间内让模型预测控制智能体运行 48 小时的实验。为此，我们并行启动了 200 个智能体，并让它们从随机的仿真时间点开始运行。为了使环境尽快达到稳定状态，温度噪声被降低至 0.05 摄氏度。随后，我们测量了每个智能体在仿真开始后的前 2 小时内的平均均方根误差。

尽管如此，要测试超过 10 个智能体的模型预测控制，同时将计算时间控制在足以用于实时运行的范围内，仍然是不可能的。也就是说，计算时间必须短于两个时间步之间的间隔。

在每个时间步，模型预测控制都会求解以下优化问题：

$$\min_{a \in \{0,1\}^{N \times H}} \sum_{t \in H} \alpha_{\text{sig}} \Big(\sum_{i \in N} P_{i,t} - s_0 \Big)^2 + \alpha_{\text{temp}} \sum_{i \in N} (T_{h,t,i} - T_{t,t,i})^2,$$

使其满足环境的以下物理约束：

$$T_{h,t,i}, T_{m,t,i} = F_1(a_{i,t}, T_{h,t-1,i}, T_{m,t-1,i}) \; \forall \; t \in H, i \in N$$

$$P_{h,t,i} = a_{i,t} F_2(\theta_a^i) \; \forall \; t \in H, i \in N,$$

以及锁定约束：

$$l_{max}(a_{i,t} - \omega_{i,t-1}) - \sum_{k=0}^{l_{max}} (1 - \omega_{i,t-k}) \leq 0 \; \forall \; t \in H, i \in N,$$

其中，F_1 和 F_2 是凸函数，可根据 C 节中给出的物理方程推导得出。

D.2 基于学习的方法

D.2.1 TarMAC 和 MA-PPO。TarMAC 的原始实现 [17] 基于异步优势演员-评论家（A3C）算法 [41]构建。其训练所用的环境回合非常短，因此智能体可以将全部记忆作为一个小批量进行在线训练。

这在我们的环境中无法实现，因为训练回合持续约 16000 个时间步。因此，我们基于现有的 MA-PPO 实现构建了 TarMAC。训练演员和评论家时使用了相同的损失函数。

评论家以所有智能体的观测作为输入。

演员的架构如图8所示。智能体 i的观测经过第一个多层感知机（MLP），输出一个隐藏状态 x。 x 随后通过三个多层感知机生成一个键、一个值和一个查询。键和值会发送给其他智能体，而智能体 i 则接收其他智能体的键和值。其他智能体的键与智能体 i的查询进行点积运算，再经过 Softmax 生成注意力。在此处应用掩码，以施加局部通信约束，并确保智能体 i 只监听其邻居。随后将注意力作为值的权重，并将加权后的值求和，生成智能体i的通信向量。对于多轮通信，将通信向量与 x 拼接后输入另一个多层感知机，以生成新的 x；然后按通信跳数重复通信过程。完成后，再次将最终的 x 与通信向量拼接，并输入最后一个多层感知机，即演员，以生成动作概率。

我们采用集中式训练方法，在训练过程中将智能体之间的通信连接到计算图中。训练完成后，智能体可以去中心化部署。

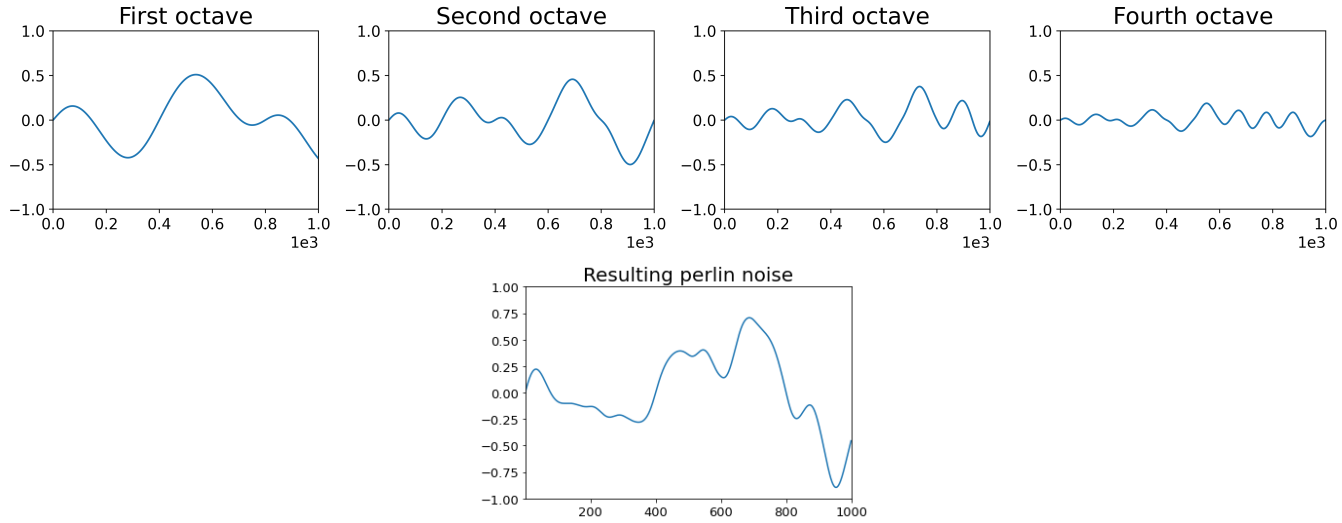


图7：多个倍频程叠加形成柏林噪声的示意图。倍频程的频率越高，其振幅越小。

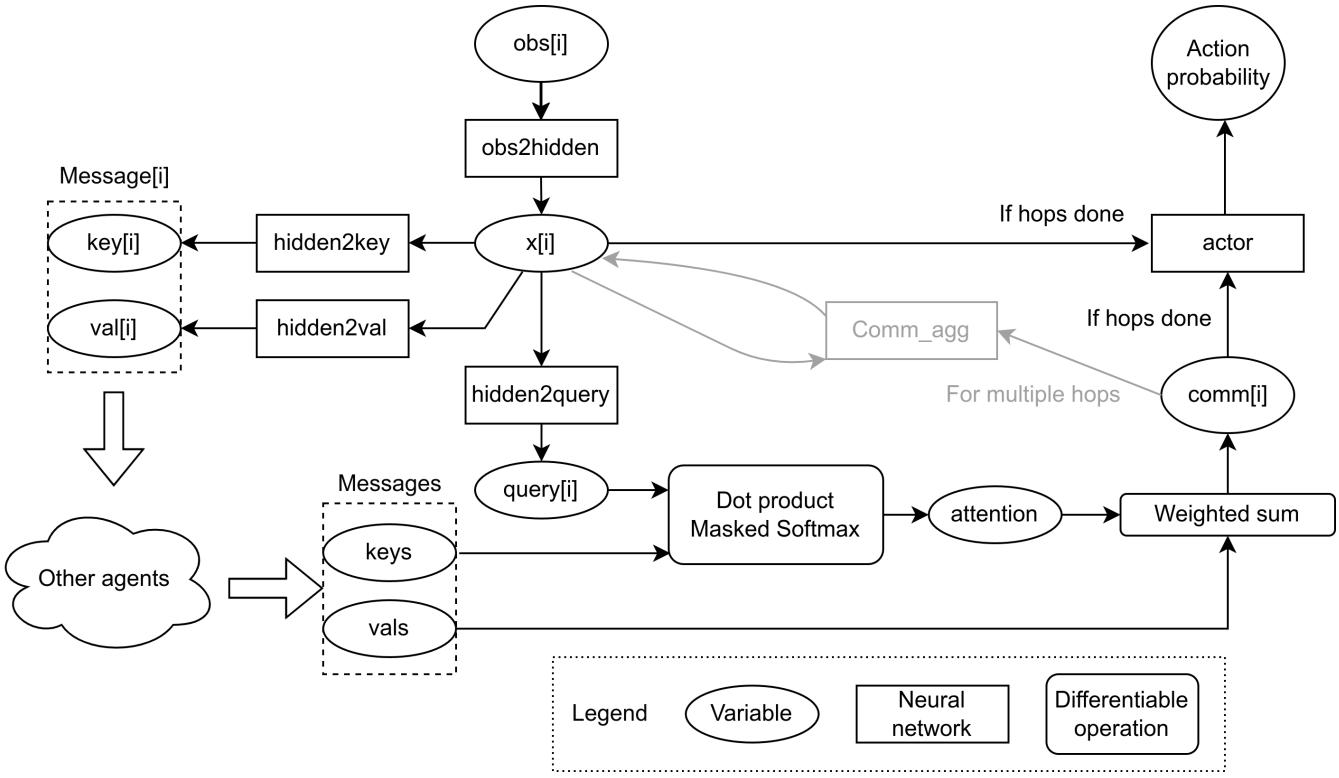


图8：TarMAC-PPO 空调的架构　　tor

D.2.2 神经网络架构与优化。对于 MA-DQN 和 MA-PPO-HE，每个神经网络的结构都相同，只有输入和输出的数量不同。网络由两个包含 100 个神经元的隐藏层组成，使用 ReLU 激活，并通过 Adam [27] 进行训练。

对于 TarMAC-PPO，演员的*obs2hidden*、*hidden2key*、*hidden2val*、*hidden2query* 和*actor* 多层感知机（如图8所示）均有一个大小为 32 的隐藏层。*obs2hidden* 和*actor* 使用 ReLU 激活，而三个通信多层感知机则使用双曲正切激活。隐藏状态 x 的大小也为 32。

集中式评论家是一个多层感知机，包含两个大小为 128、使用 ReLU 激活的隐藏层。输入规模为智能体数量

乘以每个智能体的观测规模，输出规模为智能体数量。

所有网络的输入均通过常数进行归一化，以便于训练。网络使用 Adam 进行优化。

D.2.3 超参数。 我们通过网格搜索仔细调优了超参数。表 8 展示了本文所述智能体所选用的超参数。

表 8：训练超参数

超参数	TarMAC-PPO	MA-PPO	DQN
学习率	0.001	0.001	0.0001
小批量大小	256	512	256
裁剪参数	0.2	0.2	-
最大梯度范数	0.5	0.5	-
训练轮数	200	200	-
更新次数	10	10	-
回合数	200	200	-
折扣因子 γ	0.99	0.99	0.99
键向量大小	4 或 8	-	-
通信向量大小	8	-	-
通信轮数	1	-	-
缓冲区容量	-	-	65536
ϵ 衰减	-	-	0.995
ϵ 最小值	-	-	0.01

E N_{de} 与每个智能体的均方根误差

本节讨论由 N 个同质智能体组成的聚合的每个智能体的信号均方根误差与 N 乘以整数 $k \in \mathbb{N}$ 之间的关系。

我们将规模为 kN 的聚合视为由 k 个同质群组 g_j 构成的聚合，每个群组包含 N 个智能体，消耗的功率为 $P^j_{g,\,t} = \sum_i^N P^i_{a,\,t}$。我们有：$P_t = \sum_i^{kN} P^i_{a,\,t} = \sum_j^k P^j_{g,\,t}$。

我们假设每个群组跟踪信号的相等份额 $s^j_t = s_t/k$。我们假设跟踪误差 $P^j_{g,\,t} - s^j_t$ 服从均值为 0、标准差为 σ_g 的高斯分布。该高斯误差与其他群组的噪声不相关。

根据高斯随机变量的性质，对于所有满足 $k \in \mathbb{N}$ 的 $k \geq 1$，聚合信号误差 $P_t - s_t$ 服从均值为 $\mu_k = 0$、标准差为 $\sigma_k = \sqrt{k}\sigma_g$ 的高斯分布。

因此，kN 个智能体组成的群体的信号均方根误差（这是对 σ_k 的测量估计）约为 $\sqrt{k}$ 倍于由 N 个智能体组成、用于估计 σ_g 的群体的均方根误差。最后，每个智能体的均方根误差通过将群体的均方根误差除以智能体数量计算得出。因此，kN 个智能体的每个智能体均方根误差约为 $\sqrt{k}/k = 1/\sqrt{k}$ 倍于 N 个智能体的均方根误差。

这一讨论直观地解释了为什么智能体数量增加时，相对均方根误差会降低。不过，这一解释基于这样的假设：每个群体的误差均无偏；而对于我们的智能体而言，这一假设未必成立。这也解释了为什么从 N个 $= 10$ 增加到 N个$= 1000$时，均方根误差并没有降低10倍。