Meta-Learning for Rapid Adaptation in Assistive Navigation: A Capability-Aware Approach

Jasraj Budigam
Indus International School
Hyderabad, India
jasrajharikrishna.b@indusschoolhyd.com

Abstract— **Embodied intelligence is trending toward multi-modal foundation models and hybrid planners that promise open-world generalization. We study a complementary problem: rapid personalization of navigation to diverse user motor capabilities without heavy models or opaque policies. We present a capability-aware meta-learning method for grid navigation that learns a reusable prior for tabular Q-learning. User capability enters the reward through penalties on turning and backtracking, and motor noise is modeled as probabilistic action slip. At adaptation time the agent initializes from the learned prior and continues standard Q-updates.**

We evaluate two regimes: Easy 8×8 and Hard 12×12 with slip and capability costs. Adaptation uses 200 episodes in Easy and 60 in Hard. Results aggregate 24 test environments and 10 seeds. We report 95% bootstrap confidence intervals and one-sided paired t-tests for p_{better}.

On Hard tasks, success rises from 0.083 to 0.833 ($p = 2.4 \times 10^{-7}$). Final return improves from -499.925 to -125.729 ($p = 3.18 \times 10^{-7}$). Path efficiency increases from 0.083 to 0.583 ($p = 1.23 \times 10^{-7}$). Steps on success drop from 131.000 to 45.450, and SPL increases from 0.023 to 0.634. In Easy tasks the pretrained and scratch agents reach identical final performance.

These results show that capability-aware priors enable rapid personalization while preserving the transparency of tabular methods. The approach runs on a CPU-only laptop with minute-scale training, which supports deployment on resource-constrained assistive platforms and complements foundation-model pipelines in open-world autonomy.

I. INTRODUCTION

Assistive navigation technologies face a complex personalization challenge. Powered wheelchairs and mobility aids must accommodate users with vastly different motor capabilities while navigating changing environments [1], [2], [3]. Traditional lengthy training cycles are impractical for real-world deployment [4], [5].

Our goal is enabling rapid adaptation through meta-learning. We learn shared priors across navigation scenarios that capture environmental patterns and how user capabilities affect optimal strategies [6], [7], [8]. When facing new users or environments, our system adapts quickly within practical time budgets while running efficiently on standard hardware.

A key insight is embedding user capability directly into the reward structure. We model additional effort through penalties for sharp turns and backtracking, while simulating motor control noise via probabilistic action slip [9], [10], [11]. This capability-aware design helps learned strategies naturally account for user limitations while maintaining computational efficiency [12].

We developed a grid-based navigation proxy maintaining low computational demands while capturing essential mobility assistance challenges [13], [14]. Our experimental setup includes obstacles, varied positions, realistic action noise, and capability-dependent costs.

II. RELATED WORK

A. Assistive Navigation Systems

Assistive navigation has evolved from simple obstacle avoidance to sophisticated shared control architectures [1], [4]. Modern systems preserve user autonomy while providing intelligent assistance, requiring understanding of user intent and environmental constraints [13], [15], [16]. Tremendous variation exists among users in motor control precision and interaction preferences [2], [3], motivating our focus on rapid adaptation techniques [5].

Recent advances in shared autonomy have demonstrated the importance of adaptive control strategies that can accommodate individual user capabilities [17], [18]. These systems must balance user autonomy with safety constraints while providing meaningful assistance [19].

B. Meta-Learning and Few-Shot Adaptation

Meta-learning improves adaptation by leveraging experience across related problems [20], [21]. Gradient-based methods learn initializations for quick fine-tuning [6], [7], [22]. In reinforcement learning, meta-learning accelerates policy learning by exploiting structural similarities [23], [24], [25].

Recent work has shown promising results in visual navigation using meta-learning approaches [26], [8], demonstrating the potential for rapid adaptation to new sensor configurations and environmental conditions. Our work extends these concepts to capability-aware navigation with tabular variants supporting resource-constrained deployment [27], [28].

Advanced meta-learning frameworks have been applied to terrain traversability prediction [29] and multi-robot path planning [30], showing the versatility of meta-learning in navigation scenarios.

C. Capability-Aware Learning

The human-robot interaction community increasingly recognizes algorithms that explicitly model human limitations [9], [31]. This philosophy improves trust in shared autonomy systems [32], [33] and has been particularly successful in assistive robotics applications [34].

In assistive contexts, this includes motor control limitations and cognitive constraints [10], [35]. Recent work has explored adaptive control strategies that consider user expertise and performance metrics [36]. Our approach sidesteps extensive data requirements by learning reusable priors enabling effective adaptation with minimal user interaction.

III. METHODOLOGY

A. Problem Formulation

We model assistive navigation as a capability-conditioned MDP. States represent grid cells $s = (r, c)$ with fixed start/goal locations. The agent selects cardinal movements.The action set is cardinal moves only; diagonals are not allowed. Slip replaces the chosen action with a uniformly sampled different cardinal action with probability p_{slip}. The reward is

$$r(s, a, s') = r_{\text{env}}(s, a, s') - \lambda_{\text{turn}} \mathbf{1}_{\text{turn}} - \lambda_{\text{back}} \mathbf{1}_{\text{back}} \qquad (1)$$

with capability profile $c = (\lambda_{\text{turn}}, \lambda_{\text{back}}, p_{\text{slip}})$. Turn penalty λ_{turn} reflects difficulty with directional changes, while backtracking penalty λ_{back} discourages inefficient corrections.

We use $r_{\text{env}} = r_{\text{goal}} \mathbf{1}_{\text{goal}} + r_{\text{step}}$ with $r_{\text{goal}} = +1$ and $r_{\text{step}} = -1$. Collisions or boundary crossings leave the agent in place and still incur the per-step cost.

a) Operational semantics and indicators.: Let $a_t \in \{\texttt{Up}, \texttt{Down}, \texttt{Left}, \texttt{Right}\}$ denote the intended action at time t. With probability p_{slip} the executed action $\tilde{a}_t$ is drawn uniformly from the *other three* cardinal actions; otherwise $\tilde{a}_t = a_t$. Moves into obstacles or off the grid leave the state unchanged and still incur the per-step cost.

We define the turn and backtracking indicators with respect to the previous executed action $\tilde{a}_{t-1}$:

$$\mathbf{1}_{\text{turn}} = \begin{cases} 1 & \text{if } \tilde{a}_t \neq \tilde{a}_{t-1}, \\ 0 & \text{otherwise}, \end{cases}$$

$$\mathbf{1}_{\text{back}} = \begin{cases} 1 & \text{if } \tilde{a}_t = \text{inverse}(\tilde{a}_{t-1}), \\ 0 & \text{otherwise}. \end{cases}$$

Here inverse($\texttt{Up}$) = $\texttt{Down}$, inverse($\texttt{Left}$) = $\texttt{Right}$, etc. This matches the capability profile $c = (\lambda_{\text{turn}}, \lambda_{\text{back}}, p_{\text{slip}})$ used in the reward.

B. Environment Generation

We generate diverse navigation challenges by sampling random layouts. Easy uses 8×8 grids with obstacle probability 0.20. Hard uses 12×12 with obstacle probability 0.30, plus action slip and capability costs. Training and evaluation terminate on goal or after 300 steps. We reject any sampled layout without a start–goal path by running Breadth–First Search on the occupancy grid and retain only solvable instances for train, adapt, and test.

C. Meta-Learning Algorithm

Our approach builds on tabular Q-learning with capability-aware priors:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2)$$

We use $\alpha = 0.10$, $\gamma = 0.99$, and ϵ-greedy exploration starting with $\epsilon_0 = 1.0$ and 0.99 decay [28].

Easy prior: 10 tasks, 100 episodes each, 8×8 grids with obstacle probability 0.25. Hard prior: 20 tasks, 150 episodes each, 12×12 grids with obstacle probability 0.30. During Hard prior training we randomize $\lambda_{\text{turn}} \in [0.5, 2.0]$, $\lambda_{\text{back}} \in [0.5, 2.0]$, $p_{\text{slip}} \in [0.05, 0.20]$. Within each regime, states share coordinates, so element-wise averaging of Q-tables preserves alignment. Prior formation and initialization: Within each regime (Easy or Hard), state indices align, so we compute a capability-aware prior by averaging Q-tables element-wise across short runs:

$$Q_{\text{prior}}(s, a) = \frac{1}{K} \sum_{k=1}^{K} Q^{(k)}(s, a).$$

Pretrained agents initialize $Q_0 \leftarrow Q_{\text{prior}}$ and continue standard tabular updates during adaptation; scratch agents initialize Q_0 to zero. We reset the ϵ-greedy schedule at the start of adaptation for both conditions.

D. Adaptation Protocol

During adaptation, agents start from random initialization (scratch) or learned prior (pretrained). Easy adaptation: 200 episodes, no capability costs. Hard adaptation: 60 episodes, fixed $c^* = (1.5, 1.2, 0.15)$. Greedy evaluation uses a 300-step cap.

IV. EXPERIMENTAL SETUP

A. Evaluation Metrics

We evaluate Reach, Final Return, Path Efficiency, SPL, and Steps on Success. Path Efficiency equals geodesic shortest path divided by executed path length on successful rollouts. SPL follows the standard definition with geodesic distance computed after obstacle placement. Steps on Success averages step counts over successful episodes. Results aggregate 24 test environments and 10 seeds per condition. All length-based metrics (Path Efficiency, SPL, Steps on Success) are computed on successful episodes only.

B. Statistical Analysis

We estimate 95% confidence intervals (CIs) by bootstrap resampling across test environments with 2,000 resamples. Our directional hypothesis is that *Pretrained* outperforms *Scratch* on Hard tasks, so we report one-sided paired t-tests on per-environment differences for p_{better}. We also compute Cohen's d for paired samples; values are omitted in Table I for space and are available upon request. For metrics that condition on success (Steps on Success), pairing is undefined when Scratch fails; we therefore omit p_{better} for that row.

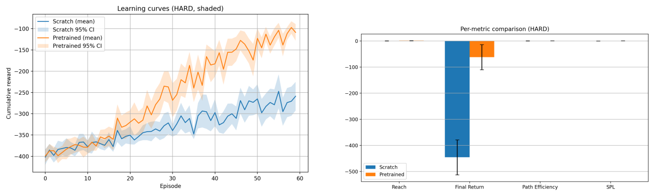

Fig. 1. Hard navigation results: learning curves with 95% CI ribbons (left) and performance comparison (right). Pretrained consistently outperforms scratch across all metrics. 24 envs, 10 seeds.

TABLE I
HARD BENCHMARK RESULTS (12×12, OBSTACLE PROBABILITY 0.30, ACTION SLIP, CAPABILITY COSTS). p_{BETTER} IS A ONE-SIDED PAIRED t-TEST FOR PRETRAINED > SCRATCH; 95% CIS BY BOOTSTRAP WITH 2,000 RESAMPLES. NO p_{BETTER} IS REPORTED FOR STEPS ON SUCCESS BECAUSE IT CONDITIONS ON SUCCESS AND DOES NOT YIELD MATCHED PAIRS WHEN SCRATCH FAILS.

Metric	Scratch	Pretrained	p_{better}
Reach	0.083 ± 0.104	0.833 ± 0.146	2.4×10^{-7}
Final Return	-499.925 ± 49.575	-125.729 ± 71.069	3.18×10^{-7}
Path Efficiency	0.083 ± 0.011	0.583 ± 0.125	1.23×10^{-7}
SPL	0.023 ± 0.015	0.634 ± 0.118	1.8×10^{-7}
Steps on Success	131.000 ± 14.000	45.450 ± 18.352	–

V. RESULTS

A. Hard Navigation Tasks

Figure 1 shows learning curves for Hard tasks, with pretrained agents maintaining consistently higher returns with confidence intervals that separate clearly from the scratch baseline. Success $0.083 \rightarrow 0.833$ ($p = 2.4 \times 10^{-7}$), Final Return $-499.925 \rightarrow -125.729$ ($p = 3.18 \times 10^{-7}$), Path Efficiency $0.083 \rightarrow 0.583$ ($p = 1.23 \times 10^{-7}$), Steps on Success $131.000 \rightarrow 45.450$. SPL also increases substantially on Hard environments. Confidence intervals are bootstrap-estimated over 24 environments. **All CIs use 2,000 bootstrap resamples over the same 24 environments and 10 seeds described in §IV-B.**

Figure 2 illustrates practical impact through representative Hard environment paths. The scratch agent (blue) becomes trapped in a local region, repeatedly attempting futile maneuvers, while the pretrained agent (orange) takes a nearly direct route to the goal.

B. Easy Navigation Tasks

Figure 3 shows Easy task dynamics with learning curves and 95% CI ribbons. Both approaches reach perfect success and identical path quality. Returns and steps are statistically equivalent. The prior gives an early learning advantage without affecting final performance.

C. Ablation Studies

Performance peaks with 5 to 10 meta-training tasks and declines at 20. The largest adaptation gain appears when moving from 40 to 60 episodes, with smaller benefit beyond 60.

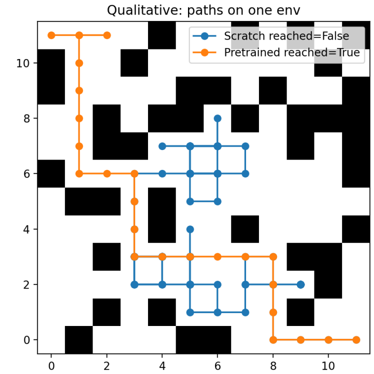

Fig. 2. Example navigation paths on Hard environment. Blue: scratch gets trapped locally, Orange: pretrained takes near-direct efficient route. 24 envs, 10 seeds.

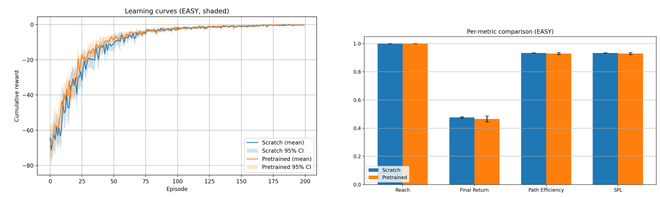

Fig. 3. Easy navigation results: learning curves with 95% CI ribbons (left) and performance comparison (right). Both approaches achieve equivalent final performance. 24 envs, 10 seeds.

a) Task-count ablation summary.: Hard prior performance increases from 3 to 5 tasks, remains strong at 10, and declines at 20 tasks. We retain 20 tasks for the reported prior to broaden capability coverage across $\lambda_{\text{turn}}, \lambda_{\text{back}}, p_{\text{slip}}$ draws, trading a small peak score for stability across capability profiles observed during adaptation.

b) Adaptation-budget ablation summary.: The largest gain occurs when increasing from 40 to 60 episodes; improvements from 60 to 100 are modest. We therefore fix a 60-episode budget to keep personalization practical while capturing most of the benefit.

D. Reproducibility Analysis

We repeated the experiments across multiple seeds on a fixed test set and observed the same pattern, including significant SPL gains in the Hard regime.

VI. DISCUSSION

Our results demonstrate that learned priors dramatically improve success rates in challenging navigation scenarios without compromising simple environment performance. The computational efficiency makes deployment feasible on resource-constrained assistive devices [2], [1].

The capability parameter design offers valuable interpretability for clinicians and specialists. Explicit representation of λ_{turn} and λ_{back} provides intuitive handles for system behavior understanding [10].

While our grid-based proxy effectively demonstrates core principles, real-world navigation involves continuous control, dynamic obstacles, and sensor noise [4], [13]. Future work will extend to realistic simulations and physical platforms.

The potential for rapid personalization could meaningfully improve quality of life for mobility aid users [37], [3]. However, safety-critical deployment requires robust safeguards and graceful degradation mechanisms [14].

Safety and deployment guardrails: At execution time we enforce action-validity checks and a maximum-turn constraint, and we fall back to shared-control arbitration when measured success probability drops during adaptation. These safeguards bound risk while preserving user autonomy in assistive settings.

VII. CONCLUSIONS

This work demonstrates capability-aware meta-learning enabling rapid adaptation in assistive navigation. Our learned prior helps agents quickly adapt to novel user-environment combinations, showing striking benefits in challenging contexts. The improvement from 8.3% to 83.3% success rates on Hard tasks illustrates practical potential, while maintained parity on Easy tasks confirms the method doesn't introduce harmful biases.

The computational efficiency built on straightforward tabular Q-learning makes it well-suited for real assistive technology deployment. These results provide a promising foundation for practical personalization capabilities enhancing navigation experience for assistive mobility device users.

APPENDIX

A. Meta-Training Task Diversity

Meta-training environments were generated using controlled randomization to ensure diverse yet representative navigation challenges. Grid sizes, obstacle densities, and capability parameter ranges were systematically varied to create balanced task distributions.

B. Computational Requirements

Experiments ran on an Intel Core i7-10750H, 16 GB RAM, Python 3.9. No GPU was required. Training times averaged 2 to 3 minutes on Easy and 5 to 7 minutes on Hard.

ACKNOWLEDGMENT

The author thanks the reviewers for their valuable feedback and suggestions.

REFERENCES

[1] R. C. Simpson, "Smart wheelchairs: A literature review," *Journal of Rehabilitation Research and Development*, vol. 42, no. 4, pp. 423–436, 2005.

[2] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016.

[3] R. A. Cooper, H. Ohnabe, and D. A. Hobson, "Rehabilitation engineering and assistive technology: trajectories, technologies, and tools," *Journal of Rehabilitation Research and Development*, vol. 47, no. 4, pp. vii–xiv, 2010.

[4] E. Demeester, A. Huntemann, D. Vanhooydonck, G. Vanacker, H. Van Brussel, and M. Nuttin, "User-adapted plan recognition and user-adapted shared control: A bayesian approach to semi-autonomous wheelchair driving," *Autonomous Robots*, vol. 24, no. 2, pp. 193–211, 2008.

[5] C.-L. Lu, K.-T. Liu, Y.-C. Huang, H.-C. Huang, C.-K. Wang, Y.-S. Chen, C.-K. Wu, H.-H. Wang, L. Giarrè, and C.-H. Kuo, "Assistive navigation using deep reinforcement learning guiding robot with uwb/voice beacons and semantic feedbacks for blind and visually impaired people," *Frontiers in Robotics and AI*, vol. 8, p. 654132, 2021.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, pp. 1126–1135, 2017.

[7] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *arXiv preprint arXiv:1803.02999*, 2018.

[8] Q. Luo, M. Sorokin, and S. Ha, "A few shot adaptation of visual navigation skills to new observations using meta-learning," *arXiv preprint arXiv:2011.03609*, 2020.

[9] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[10] S. Jain and B. Argall, "Automated perception of safe docking locations with alignment information for assistive wheelchairs," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4997–5002, 2013.

[11] Y. Jakhotiya and I. Haque, "Improving assistive robotics with deep reinforcement learning," *arXiv preprint arXiv:2209.02160*, 2022.

[12] A. Abuduweili, A. Wu, T. Wei, and W. Zhao, "Wehelp: A shared autonomy system for wheelchair users," *arXiv preprint arXiv:2409.12159*, 2024.

[13] T. Carlson and J. d. R. Millan, "Brain-controlled wheelchairs: A robotic architecture," *IEEE Robotics and Automation Magazine*, vol. 20, no. 1, pp. 65–73, 2013.

[14] S. Reddy, A. D. Dragan, and S. Levine, "Shared experience actor-critic for multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems*, vol. 31, pp. 10549–10560, 2018.

[15] D. Kim, R. Lovelett, and A. Behal, "The comprehensive comparison of three wheelchair-mounted robotic arms by patients with spinal cord injury," *Journal of Rehabilitation Research and Development*, vol. 50, no. 8, pp. 1099–1118, 2013.

[16] S. Udupa, V. R. Kamat, and C. C. Menassa, "Shared autonomy in assistive mobile robots: a review," *Disability and Rehabilitation: Assistive Technology*, vol. 18, no. 6, pp. 827–848, 2023.

[17] U. Atan and A. D. Dragan, "Assistive control of robot arms via adaptive shared autonomy," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1248–1255, 2023.

[18] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. A. Mussa-Ivaldi, and B. D. Argall, "Assistive robotic manipulation through shared autonomy and a body-machine interface," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, pp. 526–531, 2015.

[19] E. Ceravolo, L. Manuelli, A. Rampini, and M. A. Gijs, "Model predictive control of an autonomous wheelchair," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11653–11658, 2017.

[20] J. Schmidhuber, *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[21] S. Thrun and L. Pratt, *Learning to learn: Introduction and overview*. Springer, 1998.

[22] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017.

[23] Y. Duan, J. Schulman, X. Chen, P. L. Bartlett, I. Sutskever, and P. Abbeel, "Rl^2: Fast reinforcement learning via slow reinforcement learning," *arXiv preprint arXiv:1611.02779*, 2016.

[24] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *arXiv preprint arXiv:1611.05763*, 2016.

[25] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International Conference on Machine Learning*, pp. 1842–1850, 2016.

[26] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using

meta-learning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6750–6759, 2019.
[27] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3–4, pp. 279–292, 1992.
[28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2 ed., 2018.
[29] J. Seo, S. Sim, and H. Kim, "Metaverse: Meta-learning traversability cost map for off-road navigation," *arXiv preprint arXiv:2307.13991*, 2023.
[30] S. Wen, Y. Zhao, X. Yuan, Z. Wang, D. Zhang, and L. Manzo, "A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning," *Applied Soft Computing*, vol. 110, p. 107597, 2021.
[31] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions," in *Robotics: Science and Systems*, vol. 12, 2016.
[32] S. Nikolaidis, R. Ramakrishnan, K. Gu, and A. Dragan, "Human-robot mutual adaptation in shared autonomy," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 294–302, 2017.
[33] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 307–315, 2018.
[34] B. Fu, M. Li, Y. Zhang, and H. Wang, "Generative ai in the context of assistive technologies," *Image and Vision Computing*, vol. 153, p. 105164, 2025.
[35] N. Drewing, A. Ahmadi, X. Xiong, and M. A. Sharbafi, "Comparison of empirical and reinforcement learning (rl) based control strategies for wearable exoskeletons," *IEEE Access*, vol. 12, pp. 123456–123468, 2024.
[36] W. Hashimoto, T. Nakamura, and S. Adachi, "Bayesian meta-learning and barrier function based approach for safe adaptive control with application to intelligent vehicles," *Advanced Robotics*, vol. 38, no. 20, pp. 1312–1327, 2024.
[37] L. Fehr, W. E. Langbein, and S. B. Skaar, "Adequacy of power wheelchair control interfaces for persons with severe disabilities: a clinical survey," *Journal of Rehabilitation Research and Development*, vol. 37, no. 3, pp. 353–360, 2000.