Layout-Agnostic Wind Farm Control via Transformer Reinforcement Learning

Marcus Binder Nilsen, Julian Quick, Nikolay Dimitrov, Pierre-Elouan , Tuhfe Göçmen

Keywords: Wind farm control, Wake steering, Transformer, Soft Actor-Critic, Layout generalization, Transfer learning

Summary

Wake interactions between turbines cause significant power losses in wind farms, motivating wake steering through coordinated yaw control. Reinforcement learning has shown promise for this task, but existing approaches require retraining for each new farm layout. We propose a transformer-based Soft Actor-Critic architecture that treats turbines as tokens with wind-relative positional encoding, enabling a single policy to generalize across farms with different turbine counts and configurations. Experiments on five layouts (3–6 turbines) show that a generalist trained on four layouts achieves within 0.9% of layout-specific specialists, and attains 66.6% of specialist performance zero-shot on an unseen layout. Fine-tuning reaches expert-level performance with 2.3× fewer site-specific samples.

Contribution(s)

1. We present a transformer-based SAC architecture for wind farm yaw control that achieves layout generalization through per-turbine tokenization and wind-relative positional encoding, handling variable farm sizes via attention masking.
   Context: Prior transformer-based approaches to wind farm control (Kadoche et al., 2025) assume steady-state conditions and do not demonstrate generalization across layouts with different turbine counts. Our work uses a dynamic wake model and directly evaluates zero-shot transfer to unseen configurations.
2. We demonstrate that a generalist agent trained on four layouts (3–4 turbines) achieves within 0.9% of layout-specific specialists on training layouts, and 66.6% of specialist performance zero-shot on a held-out 6-turbine layout never seen during training.
   Context: Results are based on 20 evaluation episodes per layout across 3 agent seeds, using Dynamic Wake Model simulation. The held-out layout contains more turbines than any training layout. The 66.6% zero-shot performance corresponds to approximately 30,000 training steps of a from-scratch specialist.
3. Fine-tuning the pre-trained generalist on the unseen layout reaches 95% of specialist performance in approximately 30,000 steps—a 2.3× reduction in site-specific samples compared to training from scratch.
   Context: This efficiency gain requires resetting optimizer states and entropy coefficient during fine-tuning; retaining these from pre-training yields slower adaptation (81% of specialist). The fine-tuned model slightly exceeds specialist performance (100.8%), suggesting regularization benefits from diverse pre-training.

Layout-Agnostic Wind Farm Control via Transformer Reinforcement Learning

Marcus Binder Nilsen[1], **Julian Quick**[1], **Nikolay Dimitrov**[1], **Pierre-Elouan** [1], **Tuhfe Göçmen**[1]
manils@dtu.dk

[1]**Department of Wind and Energy Systems, Technical University of Denmark**

Abstract

Wake interactions between turbines can reduce wind farm power production by up to 40%. Wake steering—intentionally misaligning turbine yaw angles to redirect wakes away from downstream rotors—offers a promising mitigation strategy, but determining optimal configurations in real-time remains challenging. While reinforcement learning (RL) has shown potential for this task, existing approaches require retraining for each new farm layout, limiting practical deployment. We propose a transformer-based Soft Actor-Critic architecture for layout-agnostic wind farm control. Our approach treats each turbine as an independent token and employs wind-relative positional encoding to capture spatial relationships in a canonical reference frame invariant to absolute wind direction. This design enables a single policy to generalize across farms with different turbine counts and geometric configurations. We evaluate our approach on five layouts ranging from 3 to 6 turbines. A generalist agent trained on four layouts achieves within 0.9% of the power production of layout-specific specialists. When evaluated zero-shot on a held-out 6-turbine layout never seen during training, the generalist achieves 66.6% of specialist performance. Fine-tuning with reset optimizer states reaches expert-level performance with 2.3× fewer site-specific samples than training from scratch. These results demonstrate that transformer-based RL can learn transferable wake physics for practical wind farm control.

1 Introduction

Wind farm efficiency is fundamentally constrained by aerodynamic coupling between turbines. Each turbine creates a downstream wake—a region of reduced velocity and increased turbulence—that can account for up to 40% of potential power loss (Howland et al., 2019). Wake steering, which intentionally misaligns turbine yaw angles to redirect wakes away from downstream rotors, has emerged as a promising mitigation strategy (Boersma et al., 2017). However, determining optimal yaw configurations in real-time remains challenging due to the turbulent, nonlinear dynamics of wind farms.

Reinforcement learning (RL) offers a compelling approach to this control problem (Göçmen et al., 2024; Abkar et al., 2023). Rather than relying on idealized wake models, RL agents learn control policies directly from operational data. Despite this potential, current RL policies are typically trained on specific farm layouts and cannot generalize to new configurations without expensive retraining—hindering practical deployment across diverse installations.

Recent work has begun exploring transformer-based RL for layout-agnostic control (Kadoche et al., 2025). However, existing approaches often assume steady-state conditions, and true layout generalization remains largely unsolved. We address these limitations with a Transformer-based Soft Actor-

Critic (SAC) architecture featuring wind-relative positional encoding. Our key innovation is treating each turbine as an independent token that attends to peers through learned spatial interactions, enabling a single policy to handle variable turbine counts and arbitrary geometric configurations.

We evaluate our approach across two training regimes: single-layout specialists and a multi-layout generalist trained on diverse topologies. Our results show that:

- Multi-layout training achieves comparable performance to layout-specific specialists (within 0.9% on average)
- The generalist achieves 66.6% of specialist performance on an unseen 6-turbine layout without any layout-specific training
- Fine-tuning the pre-trained generalist reaches expert-level performance 2.3× faster than training from scratch

2 Problem Formulation

We formulate wind farm yaw control as a Markov Decision Process and evaluate our approach using a dynamic wake simulation environment.

2.1 Simulation Environment

We use WindGym (DTU, 2025), a reinforcement learning environment built on the Dynamic Wake Meandering (DWM) model implemented in Dynamiks (DTU, 2024). The DWM model captures wake propagation and meandering dynamics, balancing computational efficiency with physical fidelity suitable for RL experiments. All experiments use the DTU 10MW reference turbine (Bak et al., 2013).

The simulation operates with an internal timestep of $\Delta t_{\text{sim}} = 5$ s, while the control agent acts at $\Delta t_{\text{env}} = 10$ s intervals. Each episode spans 20 flow passthroughs (approximately 30–60 minutes simulated time). Wind conditions are fixed at $U_\infty = 10$ m/s and turbulence intensity $I = 7\%$, with wind direction sampled uniformly as $\theta_\infty \sim \mathcal{U}[260°, 280°]$.

2.2 Wind Farm Layouts

To investigate layout generalization, we use five configurations shown in Figure 1. Training layouts A–D range from 3 to 4 turbines: inline arrays (A: 3×1, C: 4×1), a square grid (B: 2×2), and a right-triangle arrangement (D). Layout E (3×2 grid, 6 turbines) is held out for zero-shot transfer evaluation. All layouts use uniform $5D$ spacing between adjacent turbines. In all layouts, the turbines are aligned so that the wake effects will be strongest when the flow comes from the east-west direction. We denote the position of turbine i as $\mathbf{p}_i = (x_i, y_i) \in \mathbb{R}^2$, expressed in meters relative to a fixed farm origin.

2.3 Markov Decision Process. Formulation

We define the Markov Decision Process. $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ with discount factor $\gamma = 0.99$. The transition dynamics P are defined implicitly by the DWM wake model.

2.3.1 State Space

The state s_t comprises per-turbine observations $\mathbf{o}_i \in \mathbb{R}^{60}$ for each turbine $i \in \{1, \ldots, N\}$. Each observation contains four sensor channels—wind speed, wind direction deviation from farm mean, yaw angle, and power output—with $H = 15$ timesteps of history. All values are normalized to $[-1, 1]$. Notably, wind direction is encoded as a *deviation* from the farm mean rather than an absolute angle; the global direction is captured through positional encoding (Section 3).

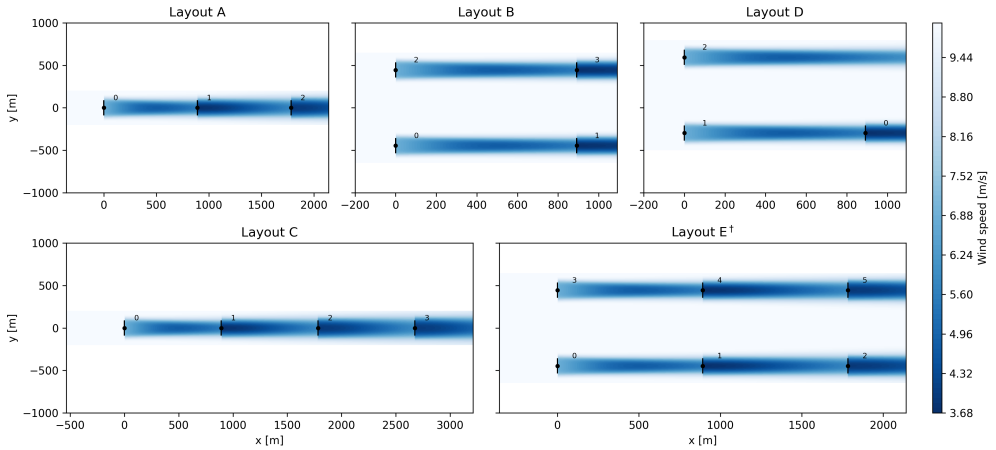


Figure 1: Wind farm layouts used in this study. Training layouts A–D contain 3–4 turbines; Layout E (6 turbines) is held out for zero-shot evaluation. All layouts use $5D$ spacing. Turbine markers are colored by local wind speed to illustrate wake-induced velocity deficits ($\theta_w = 270$).

For the generalist agent, observations are zero-padded to $N_{\max} = 6$ turbines with attention masks indicating valid positions.

2.3.2 Action Space

Each turbine receives a continuous yaw rate command $a_i \in [-1, 1]$, mapped to $\dot{\gamma}_i \in [-5, +5]$ per environment step. Yaw angles are constrained to $\gamma_i \in [-30, +30]$.

2.3.3 Reward Function

We define a baseline-relative reward measuring improvement over greedy control:

$$r_t = \frac{P_t^{\text{agent}}}{P_t^{\text{baseline}}} - 1 \tag{1}$$

where P_t^{agent} and P_t^{baseline} are farm power outputs for the learning agent and a greedy baseline (zero yaw misalignment from the mean inflow direction), respectively. Positive rewards indicate coordination benefits beyond the baseline heuristic.

3 Algorithm

This section presents the transformer-based architecture for wind farm control. The key innovation is treating turbines as tokens with wind-relative positional encoding, enabling layout-agnostic control through learned spatial attention. Turbine-specific observations are encoded as tokens and the turbine locations are encoded as a bias term.

3.1 Soft Actor-Critic Foundation

We employ the Soft Actor-Critic (SAC) (Haarnoja et al., 2018) for its sample efficiency and training stability. SAC maximizes a maximum entropy objective with automatic temperature tuning. For variable-size farms, we adapt the target entropy to scale with actual turbine count: $\bar{\mathcal{H}} = -N \cdot d_a$, where N is the number of real (non-padded) turbines and $d_a = 1$ is the action dimension per turbine.

Algorithm 1 Transformer-SAC for Multi-Layout Wind Farm Control

Input: Layout set $\mathcal{L}$, max turbines $N_{\max}$, rotor diameter D

Initialize actor π_θ, critics Q_{ϕ_1}, Q_{ϕ_2}, target networks, buffer $\mathcal{D}$
for each episode **do**
  Sample layout $L \sim \mathcal{L}$ with N turbines, wind direction θ_w
  Create attention mask $M_i = \mathbb{I}[i > N]$ for $i = 1, \ldots, N_{\max}$
  for each step t **do**
    Transform positions: $\mathbf{p}_i^{\text{rel}} = \mathbf{R}(\theta_w - 270) \cdot \mathbf{p}_i / D$ for $i = 1, \ldots, N$
    Sample $\mathbf{a} \sim \pi_\theta(\cdot \mid \mathbf{o}_{1:N}, \mathbf{p}_{1:N}^{\text{rel}}, \mathbf{M})$
    Execute $\mathbf{a}$, observe reward r and next state
    Store $(\mathbf{o}_{1:N}, \mathbf{a}, r, \mathbf{o}'_{1:N}, \theta_w, N)$ in $\mathcal{D}$
    Sample minibatch; recompute $\mathbf{p}^{\text{rel}}$ from stored θ_w
    Update $Q_{\phi_1}, Q_{\phi_2}, \pi_\theta, \alpha$ via SAC with target entropy $\bar{\mathcal{H}} = -N \cdot d_a$
  end for
end for

3.2 Transformer Architecture

The architecture processes the wind farm as a set of turbine tokens, learning spatial wake interactions through self-attention. Figure 2 illustrates the actor and critic networks, and Algorithm 1 summarizes the training procedure.

3.2.1 Per-Turbine Tokenization

In this framework, each turbine is represented as an individual token. Each turbine token representation is computed by encoding the 60-dimensional observation vector into a 128-dimensional embedding via a single multi-layer perceptron (MLP) that is used for each turbine-token embedding. This embedding is the token representation that enters the transformer. Note that turbine positions are handled separately through positional encoding (as explained in Section 3.2.2), rather than being part of the token itself.

3.2.2 Wind-Relative Positional Encoding

Standard positional encodings assume a fixed reference frame, but wake physics are directional: turbines affect those *downwind*. We rotate coordinates so wind arrives from a canonical direction (270):

$$\mathbf{p}_i^{\text{rel}} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \mathbf{p}_i / D \tag{2}$$

where $\phi = \theta_w - 270$ is the rotation angle and D is rotor diameter. This provides rotation invariance: identical farms under different wind directions produce identical positional features.

We implement relative positional bias by computing pairwise displacements $\mathbf{r}_{ij} = \mathbf{p}_j^{\text{rel}} - \mathbf{p}_i^{\text{rel}}$ and learning a bias function $b(\mathbf{r}_{ij})$ added to attention logits:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + B\right) V \tag{3}$$

where Q, K, and V are learned linear projections of layer-specific token representations. The bias, $B_{ij} = b(\mathbf{r}_{ij})$, is implemented as an MLP with 64 neurons per layer (as shown within Figure 2). This directly encodes spatial relationships (e.g., "turbine j is $5D$ upwind of turbine i"). The computed bias is shared across all attention layers.

3.2.3 Transformer Encoder

Turbine tokens pass through L transformer layers, each containing H attention heads, with pre-layer normalization (Xiong et al., 2020) for training stability:

$$\tilde{\mathbf{h}}^{(\ell)} = \mathbf{h}^{(\ell-1)} + \text{MHSA}\left(\text{LN}(\mathbf{h}^{(\ell-1)})\right) \tag{4}$$

$$\mathbf{h}^{(\ell)} = \tilde{\mathbf{h}}^{(\ell)} + \text{FFN}\left(\text{LN}(\tilde{\mathbf{h}}^{(\ell)})\right) \tag{5}$$

To handle variable farm sizes, we pad observations to $N_{\max}$ and apply attention masking, excluding padded positions from attention computation.

3.2.4 Actor and Critic Networks

The actor produces per-turbine actions through shared linear heads applied to each token's final representation:

$$\mu_i, \log \sigma_i = W_\mu \mathbf{h}_i^{(L)},\ W_\sigma \mathbf{h}_i^{(L)} \tag{6}$$

Because the same weights are applied independently to each token, the network naturally handles variable turbine counts. Weight sharing guarantees permutation equivariance: reordering turbine inputs produces correspondingly reordered outputs, forcing the model to learn position-aware control through attention rather than memorizing index-specific policies.

The critic concatenates observations with actions before encoding, then aggregates turbine representations via masked mean pooling:

$$\mathbf{h}_{\text{pool}} = \frac{1}{N}\sum_{i=1}^{N}(1 - M_i) \cdot \mathbf{h}_i^{(L)} \tag{7}$$

where $M_i = 1$ indicates padding. The pooled representation passes through an MLP to produce a scalar Q-value for the farm.

3.3 Implementation Details

We use $L = 2$ transformer layers with embedding dimension $d = 128$, $H = 4$ attention heads, and feed-forward hidden dimension 256. Training uses Adam Kingma & Ba (2017) with learning rate 3×10^{-4}, batch size 256, replay buffer capacity 10^6, and Polyak averaging ($\tau = 0.005$). Gradient clipping (max norm 1.0) improves stability. Full hyperparameters are provided in the supplementary material.

4 Application

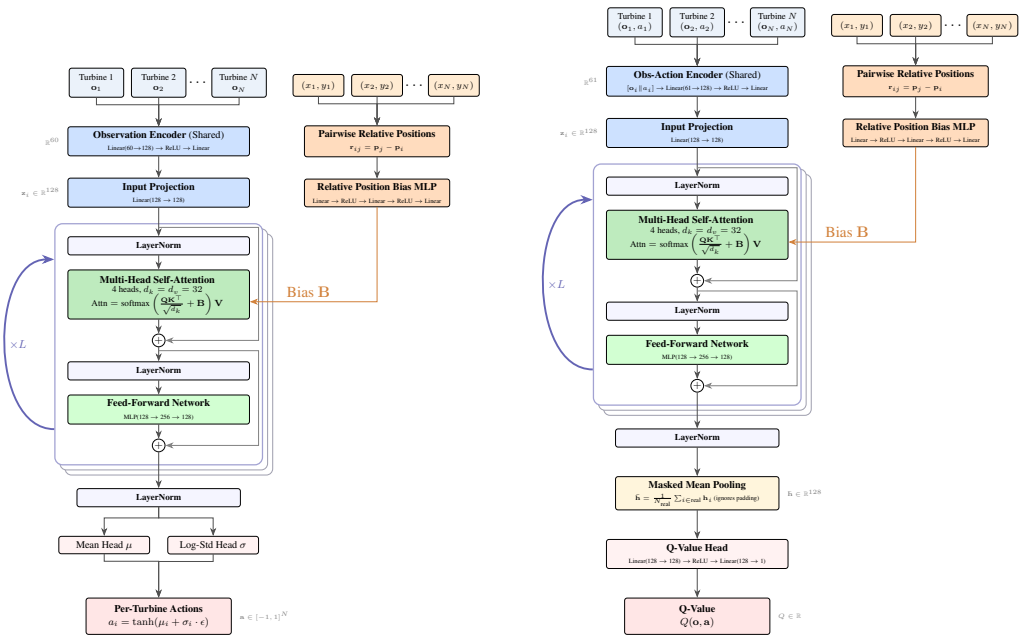
4.1 Training Protocol

For multi-layout training, we implement layout-randomized sampling: at each episode reset, we uniformly sample a layout from $\{A, B, C, D\}$, initialize yaw angles with small random perturbations ($|\gamma_i| \leq 15$), and sample wind direction from $\mathcal{U}[260, 280]$. The replay buffer accumulates transitions from all layouts, and minibatches contain mixed-layout samples. Variable farm sizes are handled through padding and attention masking.

For single-layout baselines, we train specialists exclusively on individual layouts using identical hyperparameters, providing a comparison point for assessing generalization benefits.

4.2 Evaluation Protocol

All agents are evaluated over 20 episodes (250 steps each, approximately 40 minutes simulated time) with wind directions sampled from $\mathcal{U}[260, 280]$. We report mean episode return and power



(a) Actor network architecture.

(b) Critic network architecture.

Figure 2: Transformer-based actor-critic architecture. (a) The actor encodes per-turbine observations, applies wind-relative positional bias to attention, and produces yaw rate commands via shared action heads. (b) The critic processes observation-action pairs similarly, then aggregates via masked mean pooling to output a single Q-value. Both networks use identical transformer encoders with relative positional bias.

improvement over the greedy baseline. Each configuration uses 3 random seeds; shaded regions in figures indicate ± 1 standard deviation.

4.3 Baselines

We compare against:

- **Greedy control**: each turbine aligns with local wind direction (zero yaw).
- **MLP-SAC**: standard SAC with fully-connected networks, receiving flattened observations.
- **Layout-specific specialists**: transformer-SAC trained on single layouts.

5 Results

We present results evaluating our transformer-based controller across three dimensions: (1) comparison with MLP baselines on single layouts, (2) multi-layout training effectiveness, and (3) zero-shot generalization and fine-tuning efficiency on unseen layouts.

5.1 Transformer vs. MLP on Single Layouts

Figure 3 compares MLP-SAC and Transformer-SAC on Layouts B and E. On the smaller 4-turbine Layout B, both architectures achieve similar performance. However, on the 6-turbine Layout E, a significant gap emerges: the transformer converges to mean returns more than twice those of MLP-SAC. This suggests that attention-based spatial reasoning becomes increasingly important as farm complexity grows. Both approaches consistently outperform the greedy baseline (zero return), demonstrating clear benefits of learned coordination.

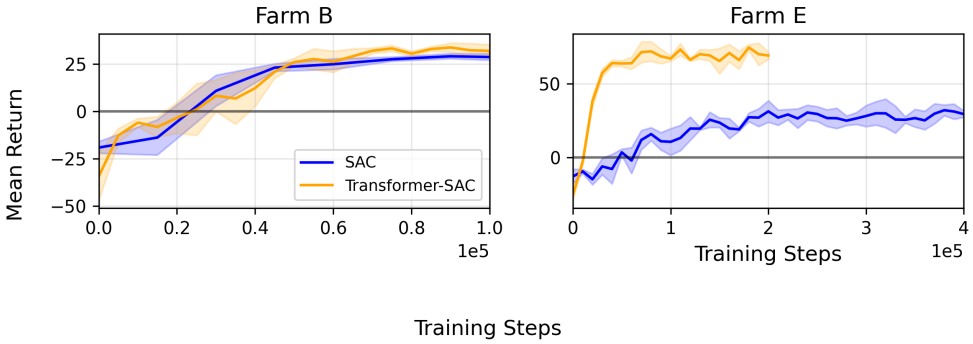


Figure 3: Learning curves comparing MLP-SAC and Transformer-SAC on Layouts B (4 turbines) and E (6 turbines). The transformer's advantage increases with farm size, suggesting attention mechanisms better capture complex wake interactions.

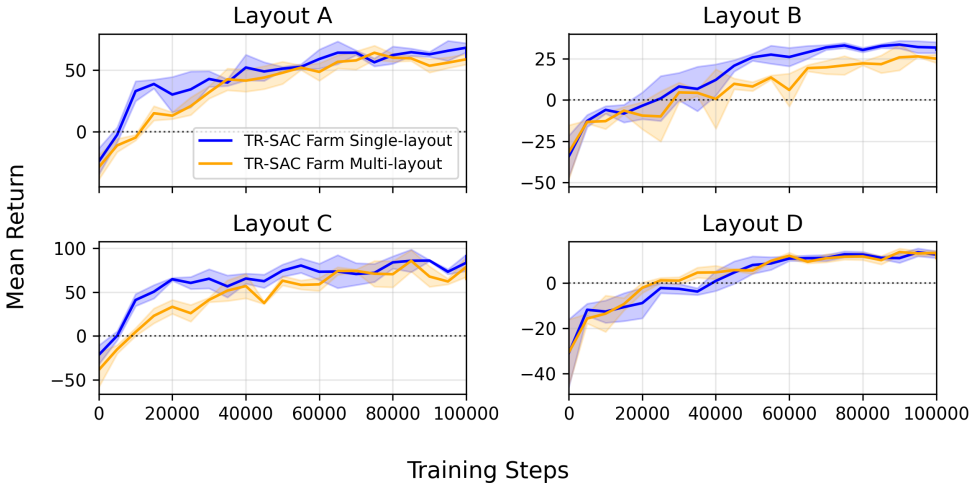


Figure 4: Learning curves for single-layout specialists (blue) versus the multi-layout generalist (orange) on training layouts A–D. Shaded regions show ± 1 standard deviation across 3 seeds.

5.2 Multi-Layout Training

Figure 4 compares single-layout specialists against the multi-layout generalist on training layouts A–D. Despite receiving only 25% of its training samples from each layout, the generalist achieves comparable asymptotic performance to layout-specific specialists across all four configurations.

Table 1 quantifies this comparison. The *cost of generalization*—the relative performance reduction compared to specialists—averages only 0.9%. Layout B exhibits the highest cost (2.8%), likely because the 2×2 grid requires coordinated yaw offsets in both streamwise and lateral directions. For Layout D, the generalist slightly *outperforms* the specialist (-0.2% cost), suggesting exposure to diverse configurations provides beneficial regularization.

5.3 Zero-Shot Transfer

To test whether the generalist learns transferable wake physics rather than memorizing layout-specific policies, we evaluate on Layout E—a 3×2 grid with 6 turbines never seen during training. This layout presents several challenges: more turbines than any training layout, a novel geometry, and complex wake interactions with multiple upstream influences per turbine.

Table 1: Cost of generalization on training layouts. The generalist achieves within 0.9% of specialists on average.

Layout	**Specialist (MW)**	**Generalist (MW)**	**Cost (%)**
A (3×1)	15.60 ± 0.09	15.49 ± 0.28	$+0.7$
B (2×2)	22.22 ± 0.14	21.60 ± 0.34	$+2.8$
C (4×1)	20.06 ± 0.37	19.99 ± 0.16	$+0.4$
D (triangle)	17.89 ± 0.14	17.93 ± 0.06	-0.2
Average	–	–	$+0.9$

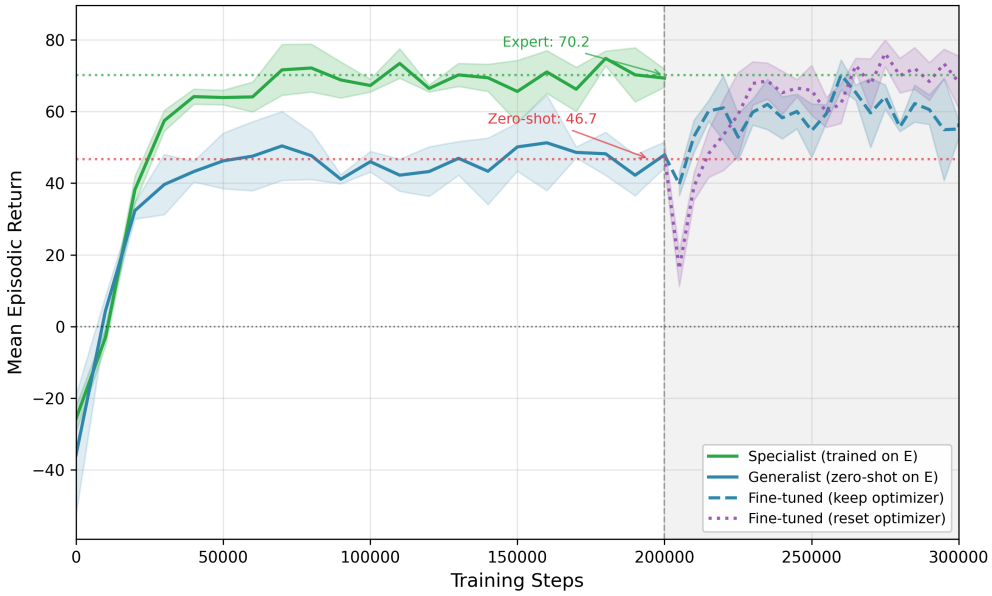


Figure 5: Zero-shot transfer and fine-tuning on unseen Layout E. The generalist (trained on A–D) achieves 66.6% of specialist performance without Layout E exposure. Fine-tuning with reset optimizer reaches expert-level performance; retaining optimizer state yields slower adaptation.

Figure 5 and Table 2 summarize the results. The generalist achieves 66.6% of specialist performance zero-shot—without any Layout E training data. For context, the specialist requires approximately 30,000 training steps to reach this same performance level, indicating that multi-layout pre-training provides a substantial head start.

5.4 Fine-Tuning Efficiency

While zero-shot performance is operationally meaningful, practitioners may want site-specific adaptation. We investigate fine-tuning the pre-trained generalist on Layout E, comparing two strategies: retaining versus resetting the optimizer state and entropy coefficient.

Resetting proves critical for effective adaptation. With fresh optimizer statistics, fine-tuning reaches 95% of specialist performance in approximately 30,000 steps and ultimately achieves 100.8%—slightly exceeding the specialist, likely due to regularization benefits from pre-training. Retaining optimizer state yields slower adaptation, plateauing at 81.0%.

We attribute this gap to two factors. First, Adam's momentum estimates from multi-layout training reflect a different loss landscape, causing initially misdirected updates. Second, the entropy coefficient α was calibrated for 3–4 turbine layouts (target entropy ≈ -3.5), while Layout E with 6 turbines implies target entropy -6. The mismatched α limits exploration during adaptation.

Table 2: Performance on unseen Layout E.

Model	Mean Return	% of Specialist
Specialist (trained on E)	70.16	100.0
Generalist (zero-shot)	46.73	66.6
Fine-tuned (keep optimizer)	56.83	81.0
Fine-tuned (reset optimizer)	70.72	100.8

Practical implications. These results suggest a deployment strategy: train a generalist on diverse representative layouts (one-time cost), then fine-tune briefly for each new site. Fine-tuning requires $\sim$30,000 steps to reach 95% specialist performance versus $\sim$70,000 steps from scratch—a $2.3\times$ reduction in site-specific sample requirements.

6 Discussion and Conclusion

We presented a transformer-based SAC architecture for wind farm yaw control that generalizes across layouts by treating turbines as tokens with wind-relative positional encoding.

Our experiments demonstrate three key findings. First, multi-layout training incurs minimal cost: the generalist achieves within 0.9% of layout-specific specialists on training layouts. Second, zero-shot transfer is viable: on an unseen 6-turbine layout, the generalist achieves 66.6% of specialist performance without any layout-specific training, indicating the model learns transferable wake physics rather than memorizing index-specific policies. Third, fine-tuning enables rapid adaptation: pre-training reduces site-specific sample requirements by $2.3\times$, reaching specialist-level performance in 30,000 steps versus 70,000 from scratch.

Several limitations constrain our conclusions. We evaluate under fixed wind speed and turbulence intensity; real deployments face time-varying conditions that may require temporal attention mechanisms beyond observation stacking. The DWM wake model, while capturing essential dynamics, omits phenomena present in field conditions. Our largest layout contains only 6 turbines—scaling to utility-scale farms (50–100+ turbines) remains an open challenge, as attention complexity grows quadratically.

Future work should address temporal modeling through recurrent architectures (e.g., GTrXL) for dynamic wind conditions, physics-aware farm decomposition for scaling to large farms, and validation against higher-fidelity simulations and field data. Adaptive entropy coefficients that scale with turbine count could also improve transfer without requiring optimizer resets during fine-tuning.

In summary, transformer-based RL offers a promising path toward layout-agnostic wind farm control. The combination of low generalization cost, viable zero-shot transfer, and efficient fine-tuning makes multi-layout pre-training an attractive alternative to developing bespoke controllers for each new installation.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper. Only add this information once your submission is accepted and deanonymized. The acknowledgments do not count towards the 8–12 page limit.

References

Mahdi Abkar, Navid Zehtabiyan-Rezaie, and Alexandros Iosifidis. Reinforcement learning for wind-farm flow control: Current state and future actions. *Theoretical and Applied Mechanics Letters*, pp. 100475, 2023.

Christian Bak, Frederik Zahle, Robert Bitsche, and et. al. Taeseong Kim. The dtu 10-mw reference wind turbine, 2013. Danish Wind Power Research 2013 ; Conference date: 27-05-2013 Through 28-05-2013.

S. Boersma, B.M. Doekemeijer, P.M.O. Gebraad, P.A. Fleming, J. Annoni, A.K. Scholbrock, J.A. Frederik, and J-W. van Wingerden. A tutorial on control-oriented modeling and control of wind farms. In *2017 American Control Conference (ACC)*, pp. 1–18, 2017. DOI: 10.23919/ACC.2017.7962923.

DTU. Dynamiks, 2024. https://dynamiks.pages.windenergy.dtu.dk/dynamiks/ [Accessed: 20-12-2025].

DTU. Windgym, 2025. Availableat:https://github.com/DTUWindEnergy/WindGym [Accessed: 21-½1-2025].

T. Göçmen, J. Liew, E. Kadoche, N. Dimitrov, R. Riva, S. J. Andersen, A. W.H. Lio, J. Quick, Pierre-Elouan Réthoré, and K. Dykes. Data-driven wind farm flow control and challenges towards field implementation. *Renewable and Sustainable Energy Reviews*, 2024.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. URL http://arxiv.org/abs/1801.01290.

Michael F. Howland, Sanjiva K. Lele, and John O. Dabiri. Wind farm power optimization through wake steering. *Proceedings of the National Academy of Sciences*, 116(29):14495–14500, 2019. DOI: 10.1073/pnas.1903680116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1903680116.

Elie Kadoche, Pascal Bianchi, Florence Carton, Philippe Ciblat, and Damien Ernst. How to craft a deep reinforcement learning policy for wind farm flow control, 2025. URL https://arxiv.org/abs/2506.06204.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020. URL https://arxiv.org/abs/2002.04745.

Table 3: Key hyperparameters.

Embed. dim	128	Learning rate	3×10^{-4}
Attention heads	4	Batch size	256
Layers	2	Buffer size	10^6
History H	15	γ / τ	0.99 / 0.005

Supplementary Materials

The following content was not necessarily subject to peer review.

Content that appears after the references are not part of the "main text," have no page limits, are not necessarily reviewed, and should not contain any claims or material central to the paper. If your paper includes supplementary materials, use the

```
\beginSupplementaryMaterials
```

command as in this example, which produces the title and disclaimer above. If your paper does not include supplementary materials, this command can be removed or commented out.