
Cross-Market Meta-Learning for Electricity Price Forecasting*

Delaram Pirhayatifard
 Rice University
 Houston, TX
 dp43@rice.edu

Terrence Alsup
 Morgan Stanley
 New York, NY
 Terrence.Alsup@morganstanley.com

Abstract

Day-ahead (DA) electricity price forecasting is usually posed per market, with a separate model fitted to each grid’s own history. This caps accuracy: statistical and tree-based baselines saturate, while higher-capacity sequence models need more training data than a single market’s few years of hourly prices can supply. Such methods also treat each node as an independent series, discarding the spatial dependencies induced by network topology and congestion. Because price formation across independent system operators (ISOs) is driven by shared physics and largely common market design, we instead treat cross-market history as a pretraining corpus. We present MetaGrid, a topology-aware transformer, meta-trained across markets so that a single zone-count-independent parameter set adapts to an unseen grid. On a leak-free benchmark of five U.S. ISOs, where every model receives the same prices, DA load forecasts and calendar features, MetaGrid attains the lowest average MAE, improving on the best baseline by 3.1% and on the same architecture trained from scratch by 7.7%.

1 Introduction

Wholesale electricity markets clear a day-ahead auction that sets locational marginal prices (LMPs) for every hour of the following delivery day. Because bids must be submitted before the auction closes, participants commit to positions without knowing the prices at which they will settle, and a forecast of the next day’s LMPs is what those bidding decisions are made against.

The dominant modeling practice is per-market and per-zone: a model is fitted to one grid’s own history, and each zone is treated as an independent series [20, 10]. Two limitations follow. First, the available sample is small; a single ISO offers a few years of hourly prices, which is enough to saturate gradient-boosted trees [2, 6] but not enough to train a high-capacity sequence model [12]. Second, treating zones independently ignores the physical constraints of a real grid, namely congestion, which can be supplied to the model as a prior rather than left to be learned from the target market’s history alone [8, 19, 13]. Cross-market transfer is known to help single-series price forecasting [4]; the multi-zone setting, where markets differ in zone count and topology, is less well studied.

Both limitations point in the same direction. Price formation across ISOs is governed by the same physics and largely common market design, so history from other markets isn’t out-of-distribution data to be discarded but a pretraining corpus, the premise underlying time-series foundation models [1, 21]. This raises the question that organizes this paper: what has to be shared across markets, and what has to remain market-specific? Grids differ in topology, generation mix and scarcity behavior, so the transferable part of a model must not depend on shape of graph it was trained on.

*Accepted at the NeurIPS 2026 Workshop on AI Foundations for Power Grids.

We propose a model that meta-trains across markets so that adaptation to a held-out grid is itself the training objective [3, 11, 18], with every parameter shared across markets and independent of zone count, so the model transfers to a grid it has never seen. The same view has been applied at test time on graph-structured data [14], where the deployment graph differs from the training graph. Shared-backbone models with task-specific fine-tuning have also been explored for probabilistic forecasting with diffusion decoders [15].

Our architecture, MetaGrid, is a patch-based transformer [12] extended with attention across zones, so that each zone’s representation is formed jointly with the others rather than in isolation. Grid structure enters this attention as a soft, learned prior: adjacency and context correlation are weighted by bounded coefficients learned during meta-training, leaving the model free to use the topology or ignore it. The model predicts quantiles and is trained with the pinball loss [9, 16]. We evaluate on a benchmark spanning five U.S. ISOs. Every model sees only information available before that market’s day-ahead gate closes, and all are scored on same set of hours against raw published prices. Against ten diverse baselines [2, 6, 5, 1, 22], MetaGrid attains the lowest average MAE, improving on the strongest baseline by 3.1% and on the same architecture trained from scratch by 7.7%.

Our contributions are: (i) a five-market benchmark with leakage-free evaluation across ten baselines; (ii) MetaGrid, a meta-trained topology-aware transformer whose every parameter is independent of zone count, so a single initialization applies to an unseen grid of any size; and (iii) evidence that transfer improves accuracy on an unseen grid: on every held-out ISO, pretraining beats target-only training of the same architecture.

2 Problem Setup & Methodology

Problem setup We consider independent system operators (ISOs) indexed by $i \in \mathcal{I}$. ISO i is a graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ with pricing zones $\mathcal{V}_i$, and zone relationships $\mathcal{E}_i$. We represent $\mathcal{E}_i$ by its adjacency matrix A_i . The number of zones $n_i = |\mathcal{V}_i|$ may differ across ISOs. The graph is built from market-map proximity and is a coarse structural prior.

For zone $v \in \mathcal{V}_i$ and time t , let $y_{i,v,t}$ denote the day-ahead price. At forecast issue time o the model receives a price history, the day-ahead load forecast published for the delivery horizon, and calendar features for the context and delivery horizon; we denote this leak-free information by $\mathcal{F}_{i,o}$ (Appendix B). Given a set of quantile levels $\mathcal{T}$ containing the median, the model estimates

$$\hat{q}_{i,v,h}^{(\tau)} \approx Q_\tau(y_{i,v,o+h} \mid \mathcal{F}_{i,o}), \quad h = 1, \dots, H, \quad \tau \in \mathcal{T}. \quad (1)$$

2.1 Shared temporal representation

Price levels and volatility differ widely across markets and zones. We apply reversible instance normalization [7] per zone over the trailing context, inverted exactly after decoding, so the encoder sees shape rather than level. Within each zone the normalized inputs are divided into nonoverlapping temporal patches and mapped to tokens by a shared linear projection. A temporal transformer then processes every zone independently, using the same weights across all zones and ISOs. We use no learned zone identities, so the encoder applies to graphs of different sizes. Let $Z_{i,p} \in \mathbb{R}^{n_i \times d}$ collect the zone representations at patch position p .

2.2 Graph-aware cross-zone attention

At each patch position, cross-zone attention allows information to flow among the rows of $Z_{i,p}$. We augment the attention score with two biases: the adjacency A_i and a price-correlation matrix $C_{i,o}$ computed from the context available at forecast origin o . For head a , the score from zone u to v is

$$s_{i,p,uv}^{(a)} = \frac{\langle q_{i,p,u}^{(a)}, k_{i,p,v}^{(a)} \rangle}{\sqrt{d_a}} + \alpha A_i[u, v] + \beta C_{i,o}[u, v], \quad (2)$$

where d_a is the dimension of each attention head. Attention weights are obtained by applying a softmax over v . The learned scalars α and β are shared across heads. Nonedges are not masked, so A_i acts as a soft structural preference, while $C_{i,o}$ allows cross-zone relationships to vary across forecast origins.

2.3 Gated probabilistic decoder

The encoded zone states are passed to a shared quantile head that directly predicts the required forecast patches for every zone and quantile level. Let $b_{i,v,h}^{(\tau)}$ denote this base prediction. Locational marginal prices decompose into a system-wide energy component and zone-specific components [17]. Rather than imposing that form, we add it as a correction: from the same encoded states we form a market-level term $m_{i,h}$ from the zone-averaged representation and a zone-centered term $r_{i,v,h}$, where we set $\sum_{v \in \mathcal{V}_i} r_{i,v,h} = 0$ so that the two terms are uniquely defined, and predict

$$\hat{q}_{i,v,h}^{(\tau)} = b_{i,v,h}^{(\tau)} + g(m_{i,h} + r_{i,v,h}), \quad (3)$$

where g is a single learned scalar initialized to zero. At $g = 0$ the decoder is exactly the unconstrained head, so the learned $|g|$ reports what the decomposition is worth on a given market. Predicting the decomposed form directly would instead make a common-mode error in $m_{i,h}$ structurally uncorrectable (Appendix B). We train with the mean pinball loss [9]

$$\mathcal{L}_{\mathcal{D}}(\Theta) = \mathbb{E}_{(i,o) \sim \mathcal{D}} \frac{1}{n_i H |\mathcal{T}|} \sum_{v \in \mathcal{V}_i} \sum_{h=1}^H \sum_{\tau \in \mathcal{T}} \rho_{\tau}(y_{i,v,o+h} - \hat{q}_{i,v,h}^{(\tau)}), \quad \rho_{\tau}(u) = \max\{\tau u, (\tau-1)u\}. \quad (4)$$

2.4 Distribution-aware meta-training and adaptation

Training proceeds in two stages. We first minimize $\mathcal{L}_{\mathcal{D}_{\text{src}}}(\Theta)$ on the pooled source-ISO data, providing a shared initialization that captures cross-market temporal and spatial structure. We then refine this initialization with chronological meta-learning episodes. Each episode e is drawn from one source ISO and contains an earlier support set $\mathcal{S}_e$ and a later query set $\mathcal{Q}_e$, with every support origin preceding every query origin. All parameters are adapted per episode. For one inner step, the support update and outer objective are

$$\Theta'_e = \Theta - \eta \nabla_{\Theta} \mathcal{L}_{\mathcal{S}_e}(\Theta), \quad \min_{\Theta} \frac{1}{|\mathcal{M}|} \sum_{e \in \mathcal{M}} \mathcal{L}_{\mathcal{Q}_e}(\Theta'_e), \quad (5)$$

where $\mathcal{M}$ is the collection of episodes and η is the inner-loop step size. Multiple inner steps repeat the support update. We use first-order (model-agnostic meta-learning) MAML, omitting second-order derivatives through the inner update.

For an evaluation ISO, the map-derived graph is supplied and $C_{i,o}$ is recomputed from the information available at each origin. Without adaptation data, inference uses the meta-learned initialization Θ . If a causal adaptation set $\mathcal{A}$ preceding the evaluation period is available, we apply the same support update to all parameters using $\mathcal{L}_{\mathcal{A}}(\Theta)$. Although this update has the same form as ordinary fine-tuning, the initialization is different: it has been optimized using losses measured after simulated adaptation to later chronological blocks. The resulting model therefore retains pooled cross-market knowledge while being explicitly trained for rapid distribution-specific adaptation.

3 Experiments

Baselines. We compare against naive forecasts (persistence 24h, 48h), regression/GBM models (XGBoost, LightGBM, quantile regression, LEAR), a tabular foundation model (TabPFN), graph baseline (MTGNN), a time-series model (PatchTST), and a time-series foundation model (Chronos T5-large, adapted). All models receive the same inputs, trailing prices, the day-ahead load forecast and calendar features, and are run with five seeds varying initialization, meta-episode sampling and evaluation-time adaptation. Splits are chronological and identical for every model and ISO: training through 2023-12-31, 2024 for validation, 2025 held out for test. More details on data and implementation in Appendix sections A and C. ²

Metrics. Point accuracy via MAE and probabilistic metric AQL (average quantile loss).

²Code is available at: <https://github.com/delaramphf/MetaGrid.git>. Load and price data are downloaded from that ISO's own market data portal.

Table 1: Per-ISO forecasting performance on the 2025 test set (5 seeds). $\dagger$ denotes deterministic models; bold indicates the best result.

Category	Model	PJM	NYISO	MAE (\$/MWh)		ERCOT	Avg	AQL (\$/MWh)
				CAISO	MISO			
Naive	Persistence (24h) $\dagger$	11.61	11.07	6.65	8.11	12.90	10.068	—
Naive	Persistence (48h) $\dagger$	16.15	16.41	9.19	10.78	15.66	13.639	—
Linear	LEAR $\dagger$	9.24	9.09	6.34	7.64	11.07	8.675	3.544
Linear	Quantile regression	11.00 ± 0.01	9.94 ± 0.01	6.75 ± 0.02	7.69 ± 0.04	13.79 ± 0.02	9.833 ± 0.010	4.009 ± 0.002
GBM	XGBoost	10.09 ± 0.03	9.96 ± 0.06	6.27 ± 0.01	7.90 ± 0.05	11.19 ± 0.06	9.080 ± 0.018	3.757 ± 0.007
GBM	LightGBM	9.77 ± 0.09	10.02 ± 0.07	6.09 ± 0.02	7.89 ± 0.02	11.32 ± 0.22	9.016 ± 0.046	3.684 ± 0.005
Tabular FM	TabPFN	9.19 ± 0.08	8.99 ± 0.10	5.90 ± 0.03	7.94 ± 0.10	10.28 ± 0.02	8.457 ± 0.013	3.391 ± 0.007
Graph	MTGNN	10.46 ± 0.004	13.87 ± 0.2	7.97 ± 0.01	8.01 ± 0.08	15.92 ± 0.03	11.26 ± 0.004	4.625 ± 0.003
Time-series	PatchTST	9.24 ± 0.03	8.91 ± 0.05	5.89 ± 0.01	7.03 ± 0.05	10.59 ± 0.03	8.333 ± 0.014	3.380 ± 0.006
Foundation	Chronos T5-large	9.53 ± 0.003	9.46 ± 0.005	6.10 ± 0.003	6.58 ± 0.012	9.77± 0.009	8.289 ± 0.011	3.427 ± 0.008
Ours	MetaGrid (scratch)	9.11 ± 0.07	9.18 ± 0.12	5.71 ± 0.03	8.37 ± 0.07	11.12 ± 0.26	8.699 ± 0.062	3.652 ± 0.005
Ours	MetaGrid	9.04± 0.009	8.78± 0.06	5.64± 0.06	6.43± 0.005	10.27 ± 0.004	8.032± 0.028	3.263± 0.002

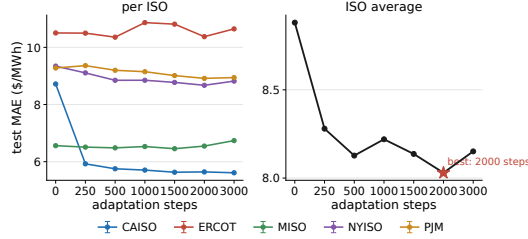


Figure 1: Test MAE vs. adaptation steps on the held-out target, per ISO (left) and averaged (right). Zero step is zero-shot transfer.

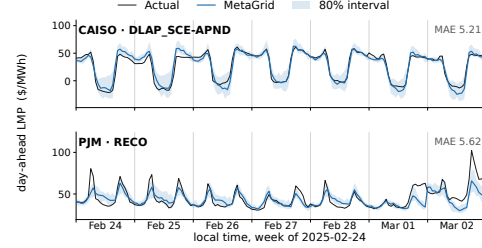


Figure 2: One week of DA LMP test forecast against realized LMP; the band is the 10-90% quantile.

Performance on held-out target ISO against the baselines Table 1 reports our model against every baseline category we ran at full adaptation data, averaged over all five ISOs. Our model achieves the best MAE and AQL of any model in this comparison, while using fewer parameters than any competing neural model. MetaGrid uses 7.8M parameters which is within $2\times$ of PatchTST and roughly $90\times$ smaller than Chronos-T5-large, yet attains 3.1% lower MAE and 3.5% lower average quantile loss than the strongest baseline. MetaGrid wins on 4 out of 5 ISOs, except for ERCOT, where TSFM baselines outperform MetaGrid. Figure 2 illustrates one week of DA forecast for CAISO and PJM compared to the ground truth. MetaGrid tracks the ground-truth time series at most time steps t .

The gain is not simply due to model scale: removing the graph-aware cross-zone module increases MAE by 0.39 \$/MWh and degrades performance on all five ISOs (Appendix D). Also, since $C_{i,o}$ is recomputed at each origin while A_i is fixed, the zone relationships the model uses change over time instead of staying fixed, and the learned weights differ by market: NYISO, with persistent congestion, relies more on adjacency ($\alpha = 0.71$) while MISO leans on correlation ($\beta = 0.54$, Table 4).

Pretrained vs. scratch: effect of adaptation data Figure 1 varies adaptation from 0 to 2000 steps, with the target-only model as reference and the optimal step selected on validation. Zero-shot transfer is already competitive in some markets: MISO achieves \$6.43/MWh without seeing MISO prices during training. Overall, pretraining is most effective when combined with adaptation.

4 Conclusion

We propose MetaGrid, which employs the operating histories of established ISOs as a pretraining corpus for a grid the model has never seen. Because price formation is driven by shared physics and largely common market design, a representation learned across markets transfers to an unseen one. Rather than fitting any single market, we meta-train for adaptation itself, so the training objective rewards few-shot generalization to a new ISO instead of in-distribution accuracy. The architecture is zone-count agnostic, so one initialization transfers across grids of different size and topology. Network structure enters as a soft, learned prior. Future work includes adding more exogenous features and extending the benchmark from zonal settlement points to nodal LMPs.

References

- [1] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.
- [2] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.
- [3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.
- [4] Salih Gunduz, Umut Ugurlu, and Ilkay Oksuz. Transfer learning for electricity price forecasting. *Sustainable Energy, Grids and Networks*, 34:100996, 2023.
- [5] Noah Hollmann, Samuel Müller, Katharina Eggersperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.
- [6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
- [7] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International conference on learning representations*, 2021.
- [8] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [9] Roger Koenker and Gilbert Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.
- [10] Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021.
- [11] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.
- [12] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.
- [13] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. Optimal power flow using graph neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5930–5934. IEEE, 2020.
- [14] Delaram Pirhayatifard and Arlei Silva. Cross-domain graph anomaly detection via test-time training with homophily-guided self-supervision. *Transactions on Machine Learning Research*, 2025.
- [15] Delaram Pirhayatifard, Mohammad Taha Toghiani, Guha Balakrishnan, and César A Uribe. Improving denoising diffusion probabilistic models via exploiting shared representations. In *2023 57th Asilomar Conference on Signals, Systems, and Computers*, pages 789–793. IEEE, 2023.
- [16] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

Table 2: Dataset specs. Samples are zone-hours of day-ahead LMP.

ISO	Zones	Date range	Freq.	Samples
CAISO	4	2021-05-01 – 2025-12-31	1 h	163,776
ERCOT	8	2021-01-01 – 2025-12-31	1 h	350,592
MISO	8	2023-01-01 – 2025-12-31	1 h	210,432
NYISO	15	2021-01-01 – 2025-12-31	1 h	657,360
PJM	21	2021-01-01 – 2025-12-31	1 h	920,304

- [17] Fred C Schweppe, Michael C Caramanis, Richard D Tabors, and Roger E Bohn. *Spot pricing of electricity*. Springer Science & Business Media, 2013.
- [18] Mohammad Taha Toghiani, Sebastian Perez-Salazar, and César A Uribe. On first-order meta-reinforcement learning with moreau envelopes. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 4176–4181. IEEE, 2023.
- [19] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [20] Rafał Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International journal of forecasting*, 30(4):1030–1081, 2014.
- [21] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.
- [22] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 753–763, 2020.

A Data

All five markets publish hourly day-ahead locational marginal prices (LMPs) at the zone or hub level. We use published zone and hub definitions, so each ISO is a small graph whose vertices are the settlement points market participants actually transact against.

CAISO settles at four default load aggregation points, one per investor-owned utility service territory, giving the smallest and most homogeneous graph in the study. **ERCOT** settles at eight load zones inside a single interconnection with no external ties, so its zones are electrically tightly coupled and its prices are the most volatile of the five. **MISO** is represented by eight regional hubs spanning a long north–south footprint. **NYISO** uses eleven internal load zones plus four proxy interfaces to neighboring control areas; it is the most strongly radial of the five, with a persistent downstate–upstate congestion pattern. **PJM** is the largest, with twenty-one transmission-owner zones across thirteen states, and contributes the most zones.

Splits are chronological and identical for every model: training through 2023-12-31, validation over calendar 2024, and testing over calendar 2025. MISO’s record begins in 2023, so it contributes a shorter training window than the others. Table 2 summarizes the panel.

B Additional method details

Leak-free construction. Every example is anchored to the market’s DA publication gate. The trailing context ends strictly before the gate, and the full delivery day is forecast at a 24-hour horizon.

Patching. Within each zone, normalized prices, system-level load forecasts, missingness indicators, and calendar features are aligned and divided into nonoverlapping temporal patches, and a shared linear projection maps each patch to a token. Calendar features are the sine and cosine of local hour-of-day and of local day-of-week. Because no learned zone identities are used, the encoder applies unchanged to graphs of different sizes.

Why the correction is additive. An alternative is to predict the decomposed form directly, replacing the head with $m_{i,h} + r_{i,v,h}$. Because r is centered across zones, any common-mode error in m is then structurally uncorrectable, and building quantiles from cumulative positive increments propagates median error outward. The additive gate in Eq. 3 avoids both: the base head remains free to correct m , and the shift is applied identically at every quantile level.

Zone-count independence. No parameter depends on $n_i = |\mathcal{V}_i|$: the patch projection, the transformer weights, the cross-zone attention and the quantile head are shared across all zones and ISOs, and no zone identities are learned. Cross-zone attention is permutation-equivariant along the zone axis, and the priors A_i and $C_{i,o}$ enter Eq. 2 as additive scores rather than as parameters. The entire parameter vector Θ therefore transfers unchanged to a grid of any size, so the inner loop can update all of Θ without introducing anything market-specific.

C Implementation details

Architecture. Patch length 16 hours over a 336-hour (14-day) context, model width 384, six attention heads, four transformer layers, feed-forward width 1536, dropout 0.1. Nine quantile levels, $\tau \in \{0.1, \dots, 0.9\}$. The attention prior weights are parameterized as $\gamma \sigma(\cdot)$ with $\gamma = 2$ and logits initialized to -4 , placing α and β at ≈ 0.036 ; the decoder gate g is initialized to 0. 7.8M parameters in total.

Meta-training. 2000 outer steps of first-order MAML, five inner steps at $\eta = 10^{-3}$ with the inner gradient norm clipped to 1. Outer optimizer AdamW at 3×10^{-4} with 5% linear warmup followed by cosine decay, batch size 32 episodes, eight chronological blocks per source ISO. Non-finite episodes are skipped rather than applied. The inner loop updates all parameters; first-order MAML keeps the per-episode cost at one extra forward-backward pass per inner step.

Adaptation and selection. For a held-out ISO we fine-tune all parameters with AdamW at 3×10^{-4} , batch size 32, and record checkpoints at various steps. The number of adaptation steps is selected per ISO on validation MAE; the test year is never consulted for any selection. Each held-out ISO is pretrained on the remaining four, so no target-market data enters pretraining.

D Ablation on graph-aware module

Table 3 removes the graph-aware cross-zone module from MetaGrid. Removing the module leaves MAE 0.39 \$/MWh worse. The model uses grid topology with α . So the gains come from cross-market pretraining as well as structural priors over the network. Table 4 reports the learned values of α , which weights the adjacency bias in cross-zone attention, β , which weights the context-correlation bias, and g , which scales the locational correction in the decoder. Starting them from two very different initializations gives nearly the same learned values on every market, so these weights are learned from data rather than inherited from initialization.

Table 3: MetaGrid with and without the graph-aware cross-zone module. Test MAE in \$/MWh; the average column is the mean $\pm$ standard deviation over 5 seeds.

Model	PJM	NYISO	CAISO	MISO	ERCOT	Avg
MetaGrid	9.04	8.78	5.64	6.43	10.27	8.03 ± 0.028
without graph-aware module	9.43	9.51	5.84	6.87	10.43	8.42 ± 0.007

E Ablation on meta-objective

Table 5 separates the two things cross-market training provides. Replacing the meta-objective with ordinary pooled pretraining over the same four source markets, in the same architecture and with the same adaptation protocol, costs 0.29 \$/MWh, while removing cross-market pretraining altogether costs 0.67: overall the benefit comes from the source data as well as the meta-objective.

Table 4: Learned attention-prior weights α (adjacency) and β (context correlation) and decoder gate g per held-out ISO, under two initializations of α and β .

Market	init $\alpha = \beta = 0.036$			init $\alpha = \beta = 1.000$		
	α	β	g	α	β	g
PJM	0.612 ± 0.021	0.388 ± 0.017	0.734 ± 0.028	0.634 ± 0.029	0.401 ± 0.022	0.719 ± 0.033
NYISO	0.714 ± 0.026	0.352 ± 0.019	0.681 ± 0.031	0.698 ± 0.031	0.366 ± 0.024	0.695 ± 0.036
CAISO	0.283 ± 0.018	0.507 ± 0.023	0.402 ± 0.024	0.301 ± 0.025	0.492 ± 0.027	0.415 ± 0.029
MISO	0.164 ± 0.014	0.541 ± 0.021	0.286 ± 0.019	0.187 ± 0.020	0.533 ± 0.026	0.298 ± 0.025
ERCOT	0.441 ± 0.023	0.463 ± 0.020	0.588 ± 0.026	0.428 ± 0.028	0.470 ± 0.025	0.601 ± 0.031

Table 5: Contribution of cross-market pretraining and of the meta-objective. All three arms share the architecture, the source markets and the evaluation-time adaptation protocol; they differ only in how the initialization is obtained (Test MAE in \$/MWh).

Initialization	PJM	NYISO	CAISO	MISO	ERCOT	Avg
MetaGrid (FOMAML)	9.04 ± 0.09	8.78 ± 0.06	5.64 ± 0.06	6.43 ± 0.05	10.27 ± 0.04	8.03 ± 0.08
pooled pretraining	9.11 ± 0.08	8.89 ± 0.02	5.95 ± 0.02	6.97 ± 0.04	10.67 ± 0.13	8.32 ± 0.02
no pretraining	9.11 ± 0.07	9.18 ± 0.12	5.71 ± 0.03	8.37 ± 0.07	11.12 ± 0.26	8.70 ± 0.06

Domain Evaluation Checklist

- 1. Physics feasibility under full AC power flow.** N/A. We forecast prices and produce no operating point or dispatch decision; the adjacency prior is a statistical structure over pricing zones, not a network model.
- 2. Out-of-distribution evaluation.** Yes. All results are leave-one-ISO-out: each target market is unseen in pretraining and differs in zone count, topology and generation mix (Table 1).
- 3. Documented failure modes.** Partially. We report per-ISO results and identify three: MetaGrid loses on ERCOT (Table 1), adaptation degrades MISO (Figure 1), and spikes are underpredicted (Figure 2).
- 4. System-level effects.** No. We report forecast accuracy only; propagation into dispatch and hedging decisions is future work.
- 5. Data and code availability.** Yes. All price and load-forecast data are public ISO market-portal downloads; codes are available in the public GitHub repository (Section 3).