Response to Reviewer 3 Comments

1. Summary

Thank you very much for reviewing our manuscript and providing valuable comments and suggestions during your busy schedule. Your suggestions are of great significance in guiding our research work, and we would like to express our heartfelt thanks to you. We will respond to your questions and suggestions one by one.

2. Point-by-point response to Comments and Suggestions for Authors

Comments 1: *[Include pseudo-code or an algorithmic summary of the CA-PatchTST model to facilitate reproducibility and better understanding of the workflow.]*

Response 1: We thank the reviewer for this constructive suggestion. To enhance reproducibility and provide a clearer understanding of the model workflow, we have added a comprehensive pseudo-code summary of the proposed CA-PatchTST model as Algorithm 1 in the revised manuscript.

The CA-PatchTST algorithm is included in Section 3.2.6（page18）. It outlines the complete workflow in a structured manner, encompassing the key stages of:

- Input organization and device-based sequence grouping,
- Moving-average-based decomposition into trend and residual components,
- Patch-based encoding and channel-independent Transformer processing,
- Cross-attention fusion across device groups,
- Multi-step forecast generation via additive fusion of trend and residual outputs.

This addition provides a concise yet complete operational view of the model, thus facilitating easier implementation and better comprehension of the overall pipeline, as shown in the pseudocode below.

Algorithm 1 CA-PatchTST for Solar Array Temperature Trend Forecasting

---

1：Input: Multivariate temperature sequence: $X \in \mathbb{R}^{B\times N\times L}$,

Device partition: $\{\mathcal{G}_{\text{BMSP}}, \mathcal{G}_{\text{Bracket}}, \mathcal{G}_{\text{Body}}\}$,

patch length P, stride S, forecast horizon H

2：Organize sequences by device groups: $X_{\text{min}} = X[:, \mathcal{G}_{\text{target}}, :], X_{\text{cross}} = X[:, \mathcal{G} \setminus \mathcal{G}_{\text{target}}, :]$

3：for component $c \in \{\text{main, cross}\}$ do:

4：Apply moving average filtering: $X_{\text{trend}}^{(c)} = \text{MA}(X^{(c)})$

5：Compute residual component: $X_{\text{res}}^{(c)} = X^{(c)} - X_{\text{trend}}^{(c)}$

6：end for

7：for component $c \in \{\text{trend, res}\}$ do:

8：for branch $b \in \{\text{main, cross}\}$ do:

9：Patching: $P_c^{(b)} = \text{Unfold}(X_c^{(b)}, \text{kernel} = P, \text{stride} = S) \in \mathbb{R}^{B\times N_b\times M\times P}$

10：Linear Projection & Position Encoding: $Z_c^{(b)} = P_c^{(b)} W_p + E_{\text{pos}} \in \mathbb{R}^{B\times N_b\times M\times D}$

11：Channel-independent TST Encoding: $Z_c^{(b)} = \text{TSTEncoder}(Z_c^{(b)})$

12: end for

13: end for

14: for component $c \in \{\text{trend}, \text{res}\}$ do:

15: Extract representations: $Z_{\text{main}} = Z_c^{(\text{main})}, Z_{\text{cross}} = Z_c^{(\text{cross})}$

16: Multi-head Cross-Attention:

$$Q = Z_{\text{main}} W_Q, K = Z_{\text{cross}} W_K, V = Z_{\text{cross}} W_V, \text{Attention} = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

17: Residual Connection & Layer Normalization: $Z' = \text{LayerNorm}(\text{Attention} + Z_{\text{main}})$

18: Feed-Forward Network: $Z_c^{\text{out}} = \text{LayerNorm}(\text{FFN}(Z') + Z')$

19: end for

20: for component $c \in \{\text{trend}, \text{res}\}$ do:

21: Flatten and project to forecast horizon: $\hat{Y}_c = \text{Flatten}(Z_c^{\text{out}}) W_o \in \mathbb{R}^{B \times N_{\text{target}} \times H}$

22: end for

23: Additive Fusion: $\hat{Y} = \hat{Y}_{\text{trend}} + \hat{Y}_{\text{res}}$

24: Compute MSE loss: $\mathcal{L} = \frac{1}{B \cdot N_{\text{target}} \cdot H} \sum \| Y - \hat{Y} \|^2$

25: Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$

26: Output: Multi-step forecasts $\hat{Y} \in \mathbb{R}^{B \times N \times H}$

Comments 2: *[While several baselines are used, the paper would be strengthened by including recent transformer-based forecasting models such as Informer, Autoformer, or TimesNet for a fairer and more comprehensive comparison.]*

Response 2: We thank the reviewer for this insightful suggestion to enhance the comprehensiveness of our comparative study. In response, we have expanded our suite of baseline models to include additional, cutting-edge Transformer-based architectures.

To incorporate the most impactful recent advances, **we selected TimesNet (2023) and iTransformer (2024)**. iTransformer has emerged as a leading benchmark due to its inverted architecture and robust multivariate performance, while TimesNet offers a unique perspective by modeling temporal 2D-variations. We focused on these models as they represent the current state-of-the-art, providing a more rigorous and contemporary benchmark than the earlier Autoformer.

All new baselines have been integrated into our experimental framework and evaluated under the exact same conditions as previously described. The consolidated results, now presented in Section 4.4.2 and Figures 8-9, consistently demonstrate that our CA-PatchTST model maintains a performance advantage across all forecasting horizons and metrics. For instance, it achieves leading MAPE values of 9.26% (24-hour), 13.68% (72-hour), and 24.67% (120-hour), thereby validating its effectiveness against the current state-of-the-art.

The detailed description of these additional baselines and the comprehensive comparative results can be found in Section 4.4.2 (Page 20, Lines 570-593), with performance comparisons visually presented in Figure 8 and Figure 9.

Furthermore, we have added a comprehensive computational efficiency analysis in Section 4.4.2 (Page 21). As summarized in the new Table 3, CA-PatchTST achieves the lowest parameter count, shortest inference time, and minimal memory footprint among the

compared Transformer-based models. This demonstrates its superior efficiency, making it particularly suitable for resource-constrained scenarios without compromising predictive accuracy.

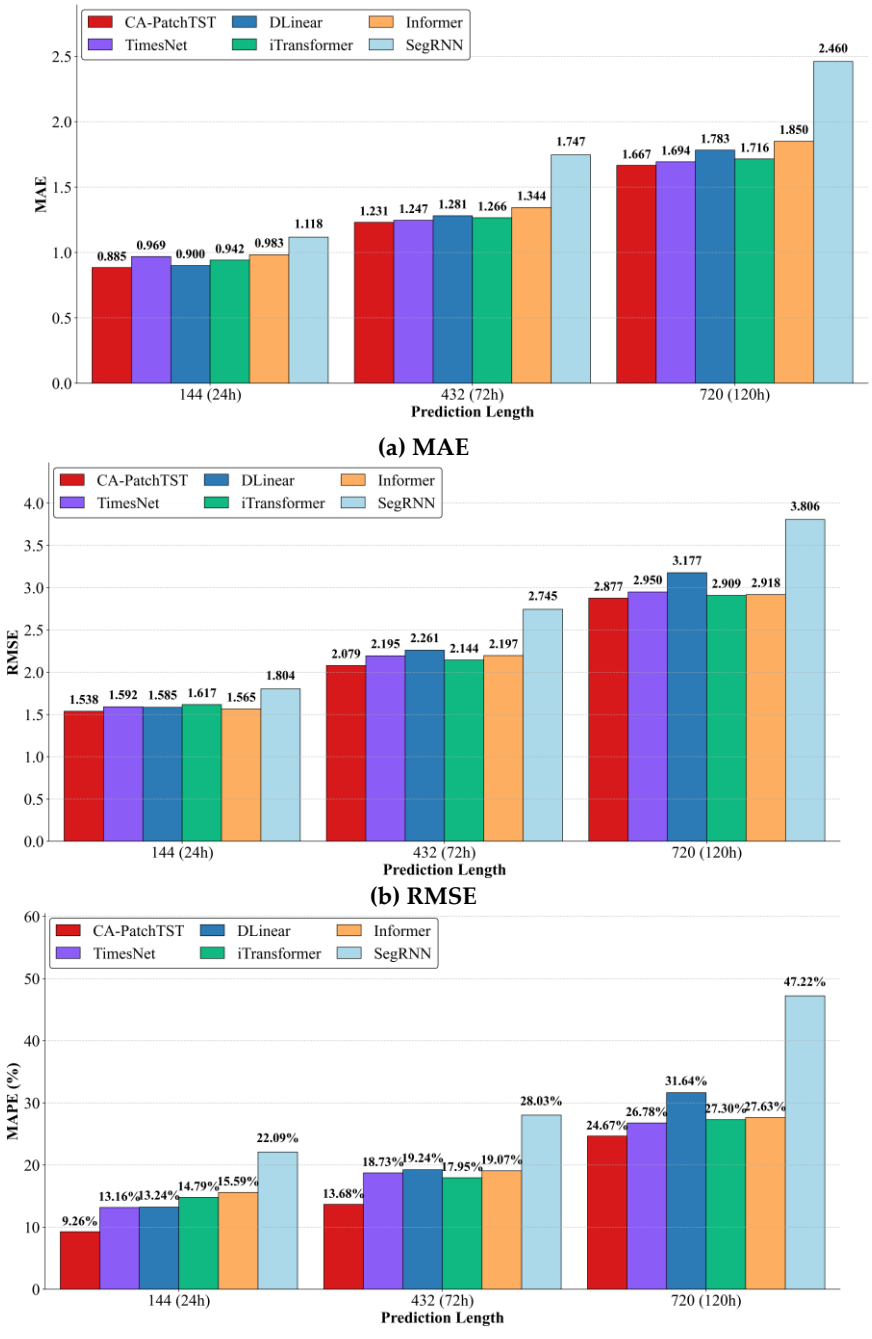


Figure 8. Performance Comparison of Different Models.

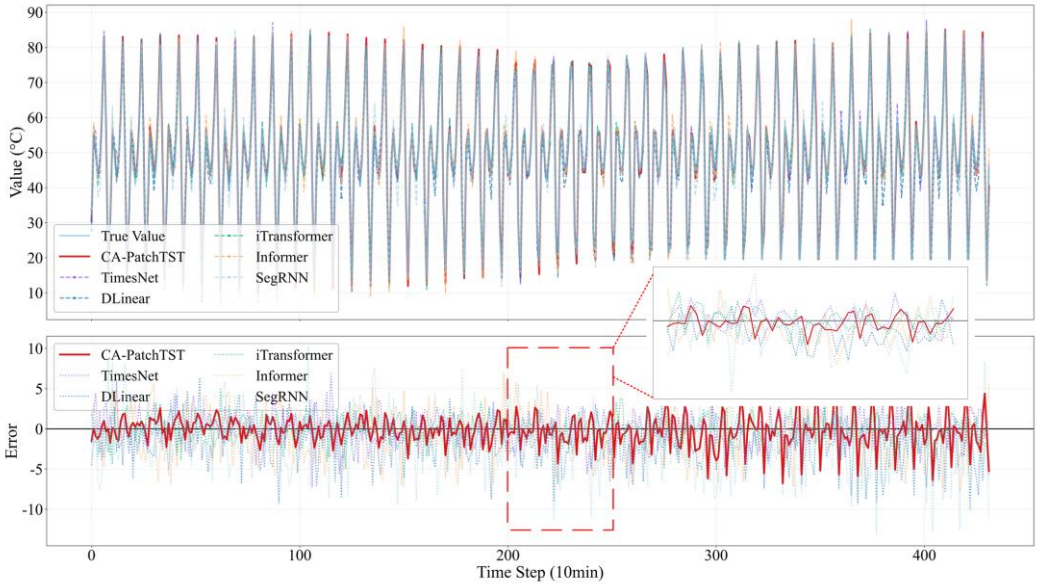


Figure 9. Comparative Forecasting Performance of THT10002 Parameter across Different Models.

Comments 3: *[The authors briefly mention model lightweighting and adaptive learning. Expanding on how these could be implemented (e.g., pruning, quantization, or online adaptation) would provide valuable insights for future research and practical deployment.]*

Response 3: We thank the reviewer for the valuable suggestion regarding model light-weighting and adaptive learning. In response, we have expanded the discussion of future work in the Conclusions section to provide more concrete implementation details. Specifically, we now outline plans for model light-weighting through pruning, quantization, and knowledge distillation to optimize the efficiency-accuracy trade-off, as well as online adaptation using continual learning with experience replay and anomaly-aware updates to maintain performance under dynamic environmental conditions. These refinements help clarify our research roadmap and strengthen the practical relevance of the study.

The detailed future work discussion is included in Section 5 (Page 24, Lines 737-743) of the revised manuscript. The revised content is as follows:*"[In future work, we will advance our prototype toward operational deployment by focusing on model light-weighting and online adaptation. For light-weighting, we will implement pruning, quantization, and knowledge distillation to optimize the efficiency-accuracy trade-off. For online adaptation, we will develop continual learning with experience replay and anomaly-aware updates to maintain performance under environmental shifts. These co-designed improvements will enhance robustness and enable long-term autonomous satellite power management.]"*