Baseline vs Best 100K Model Comparison

Models compared:

- **Baseline**: Moirai2 Small, 100K steps, no preconditioning (Normalized MASE 0.921)
- **Best 100K**: Moirai2 Small + Chebyshev d=4 hint + 10% hint dropout, 100K steps (Normalized MASE 0.853)

Checkpoints:

```
Baseline:  /scratch/gpfs/EHAZAN/jh1161/uni2ts/outputs/pretrain/moirai2_small/lotsa_v1_unweighted/m2_baseline_20260209_114203/checkpoints/epoch_999-step_100000.ckpt
Best 100K: /scratch/gpfs/EHAZAN/jh1161/uni2ts/outputs/pretrain/moirai2_small/lotsa_v1_unweighted/m2_hd10_100k_20260223_112829/checkpoints/epoch_999-step_100000.ckpt
```

Evaluation settings (identical for both): context_length=4000, patch_size=32, batch_size=64, 97 GIFT-Eval configs.

Training note: Baseline uses 10K warmup steps; hint model uses 1K warmup steps. Both use cosine annealing, AdamW (lr=1e-3), 100K total steps (1000 epochs x 100 batches, bs=256) on LOTSA v1.

MASE normalization: All MASE numbers in this notebook are **normalized by the Seasonal Naive model's MASE per config**, consistent with the GIFT-Eval leaderboard. A normalized MASE of 0.80 means the model's error is 80% of the seasonal naive model's error. Lower is better.

Data files (all in this folder):

- `baseline_100k_eval.csv` — GIFT-Eval results for baseline (from `gifteval/results/gifteval_results_epoch_999-step_100000_20260210_203632.csv`)
- `hd10_100k_eval.csv` — GIFT-Eval results for hint+dropout model (from `gifteval/results/gifteval_results_epoch_999-step_100000_20260223_235237.csv`)
- `training_loss_data.json` — Training loss curves extracted from TensorBoard event files

```
Baseline eval: 97 configs, 97 matched to seasonal naive
HD10 eval:     97 configs, 97 matched to seasonal naive

Baseline normalized MASE (geo mean): 0.9213
HD10 normalized MASE (geo mean):     0.8526

(For reference, official Moirai 2.0-R-small = 0.728 on leaderboard)
```

1. Architecture Comparison

Baseline: Standard Moirai2 Small

The baseline is a **causal decoder-only transformer** for time series forecasting:

Component	Details
d_model	384
d_ff	1024
Layers	6 attention layers
Patch size	16 (each patch = 16 time steps)
Parameters	11.39M
Loss	PackedQuantileMAELoss (pinball loss over 9 quantiles)
Scaler	Per-series z-score normalization

Input representation per patch: `[target (16), observation_mask (16)]` = 32 dims, projected to 384 via `in_proj`.

Best 100K: Moirai2 Small + Preconditioning Hint

The hint model adds **one extra input channel** per patch containing the preconditioning filter residual:

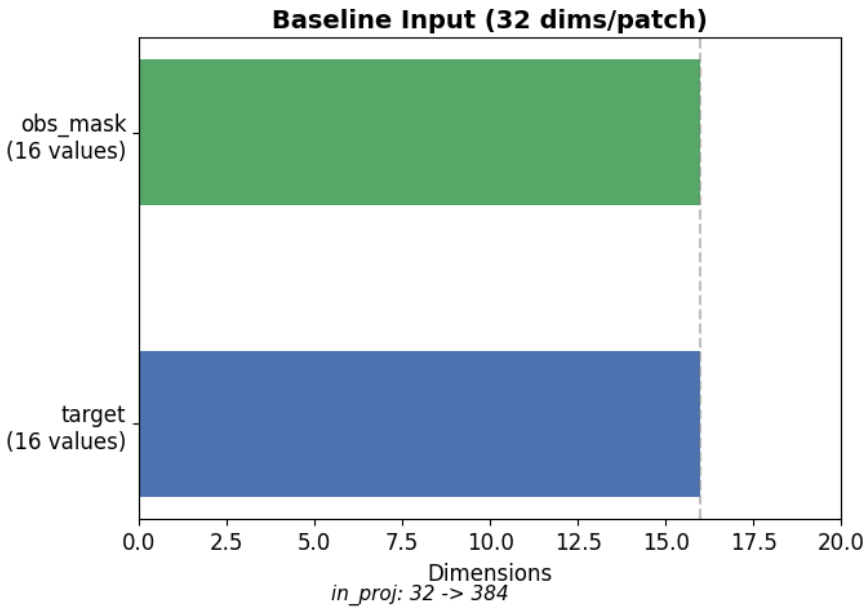
Component	Details
Architecture	Same 6-layer causal transformer
Parameters	11.40M (+12K for wider `in_proj`: 48 -> 384 instead of 32 -> 384)
Hint channel	Chebyshev degree-4 preconditioning filter residual at stride 16
Hint dropout	10% (per-patch, training only)

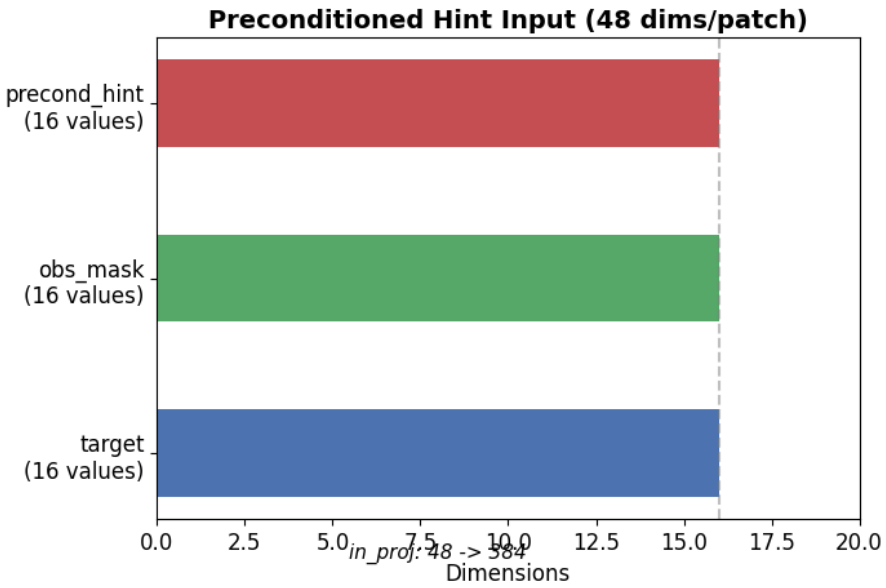
Input representation per patch: `[target (16), observation_mask (16), hint_d4 (16)]` = 48 dims, projected to 384.

The hint channel computes a **causal preconditioning filter residual** (technically a Finite Impulse Response / FIR filter):

```
hint[t] = -1.0 * y[t-32] + 0.125 * y[t-64]
```

This tells the model "how does this time step relate to the same position in previous patches" -- providing explicit inter-patch autocorrelation information. The hint is zeroed in the prediction window (future values unknown).

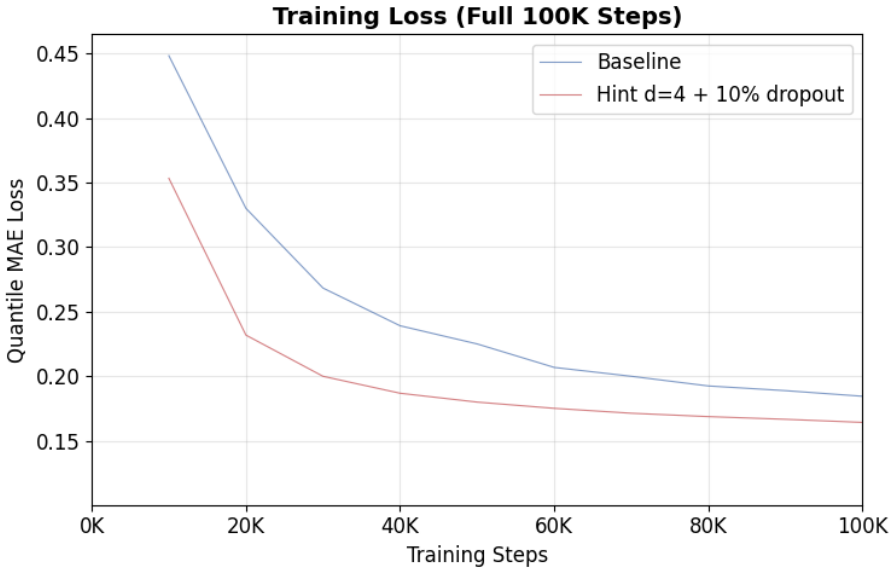




2. Training Loss Comparison

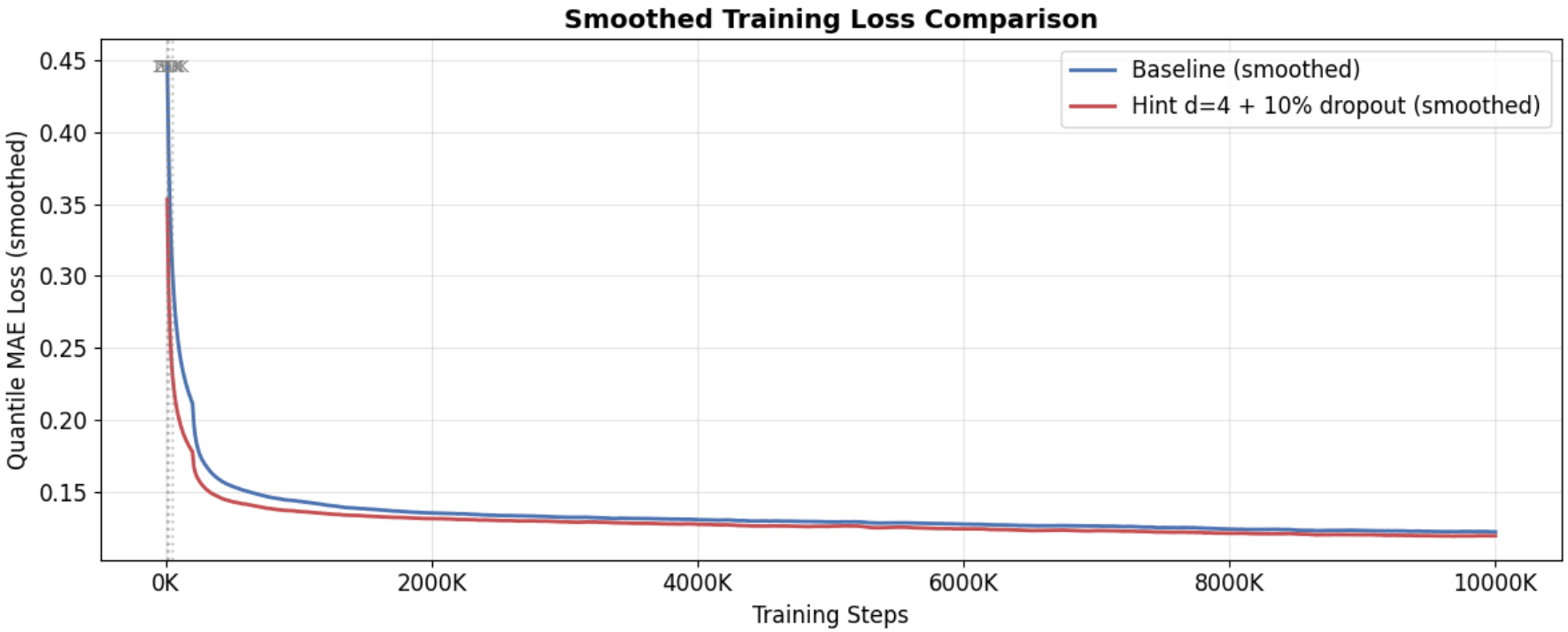
Both models are trained for 100K steps (1000 epochs x 100 batches/epoch, batch_size=256) on LOTSA v1.

The training loss is `PackedQuantileMAELoss` -- pinball loss over 9 quantile levels (0.1 to 0.9). This is the same loss function for both models; the only difference is what inputs the model receives.




```
Baseline:  first epoch loss = 0.4482, final epoch loss = 0.1233
Hint d=4:  first epoch loss = 0.3534, final epoch loss = 0.1204

Delta at final epoch: -0.0029 (-2.4%)
```


Smoothed Training Loss Comparison
Baseline (smoothed)
Hint d=4 + 10% dropout (smoothed)
Quantile MAE Loss (smoothed)
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0K
2000K
4000K
6000K
8000K
10000K
Training Steps

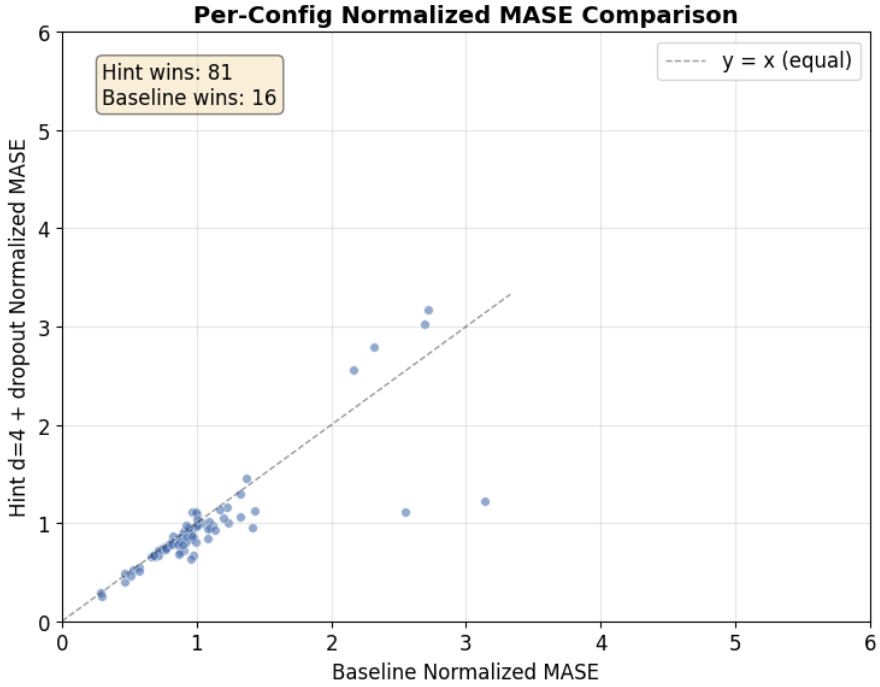
Note: Baseline uses 10K warmup steps, hint model uses 1K warmup steps.
Both use cosine annealing after warmup.

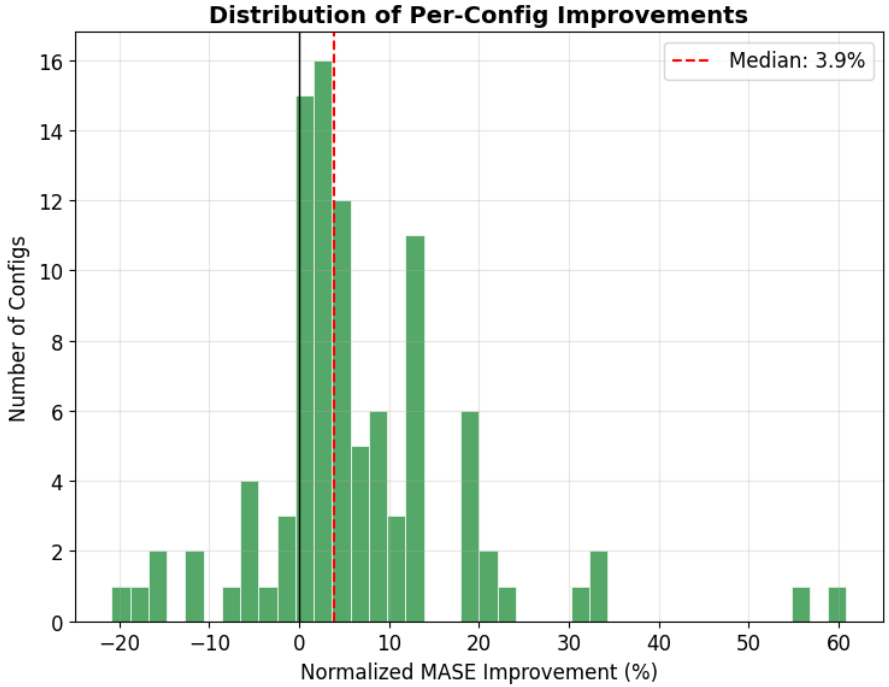
3. GIFT-Eval Results Comparison

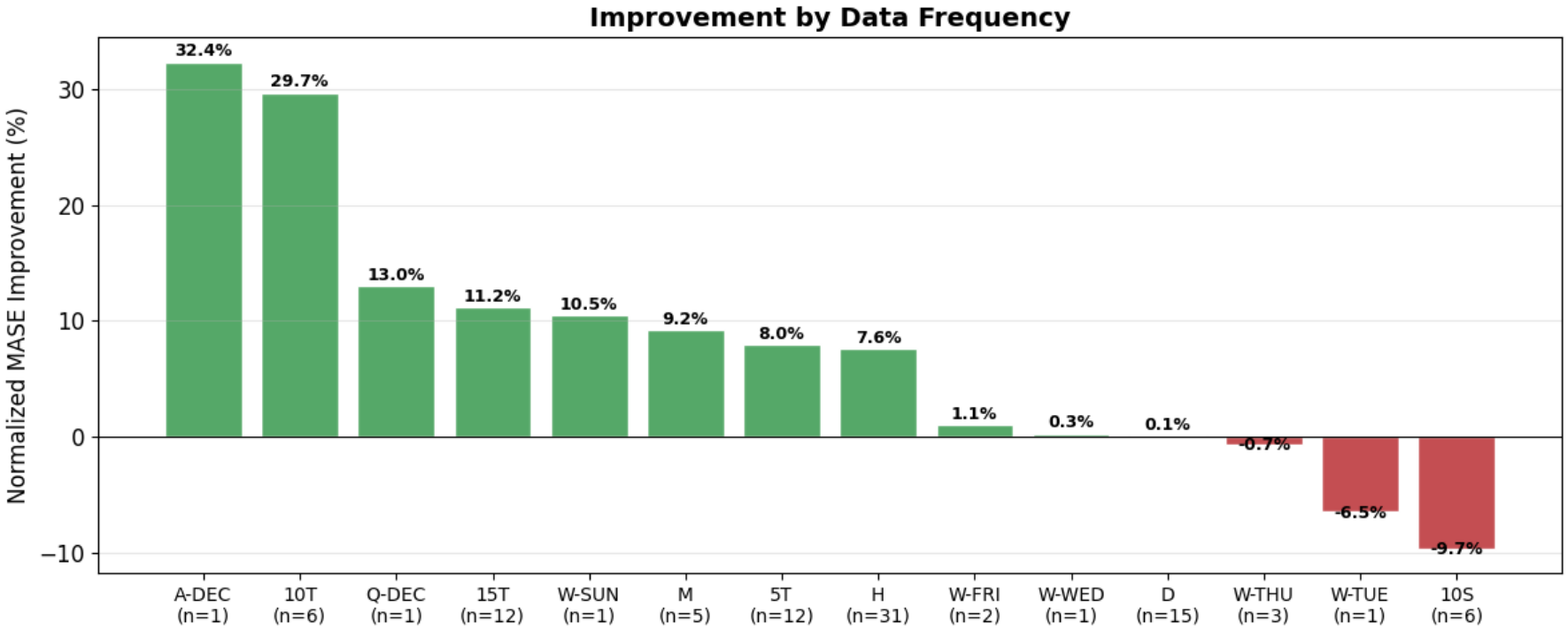
GIFT-Eval is a comprehensive time series forecasting benchmark with 97 configurations across multiple datasets, frequencies (from 10-second to yearly), and forecast horizons (short/medium/long).

Primary metric: Normalized MASE — each config's MASE is divided by the Seasonal Naive model's MASE for that config (consistent with the GIFT-Eval leaderboard). Normalized MASE < 1.0 means the model beats the seasonal naive model.

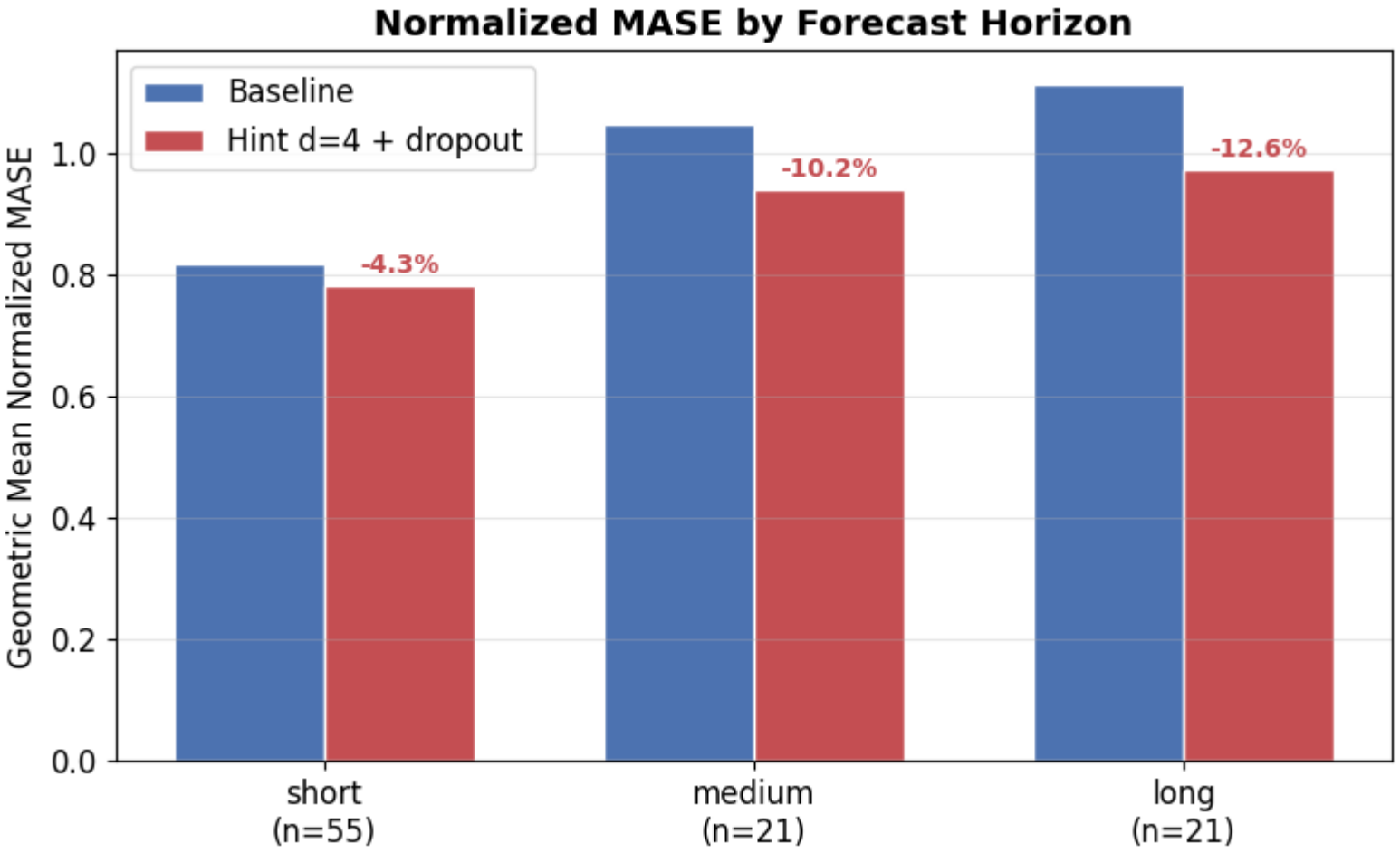
```
============================================================
GIFT-Eval Summary (97 configurations, normalized MASE)
============================================================
Metric                                  Baseline  Hint+Drop
------------------------------------------------------------
Geo Mean norm MASE                        0.9213     0.8526
Arith Mean norm MASE                      0.9958     0.9204
Median norm MASE                          0.9125     0.8416
Configs < 1.0 (beat SN model)                 70         78
Min norm MASE                             0.2888     0.2540
Max norm MASE                             3.1362     3.1705
------------------------------------------------------------
Hint wins / losses                       81 /   16
Relative improvement (geo mean)            7.46%
```







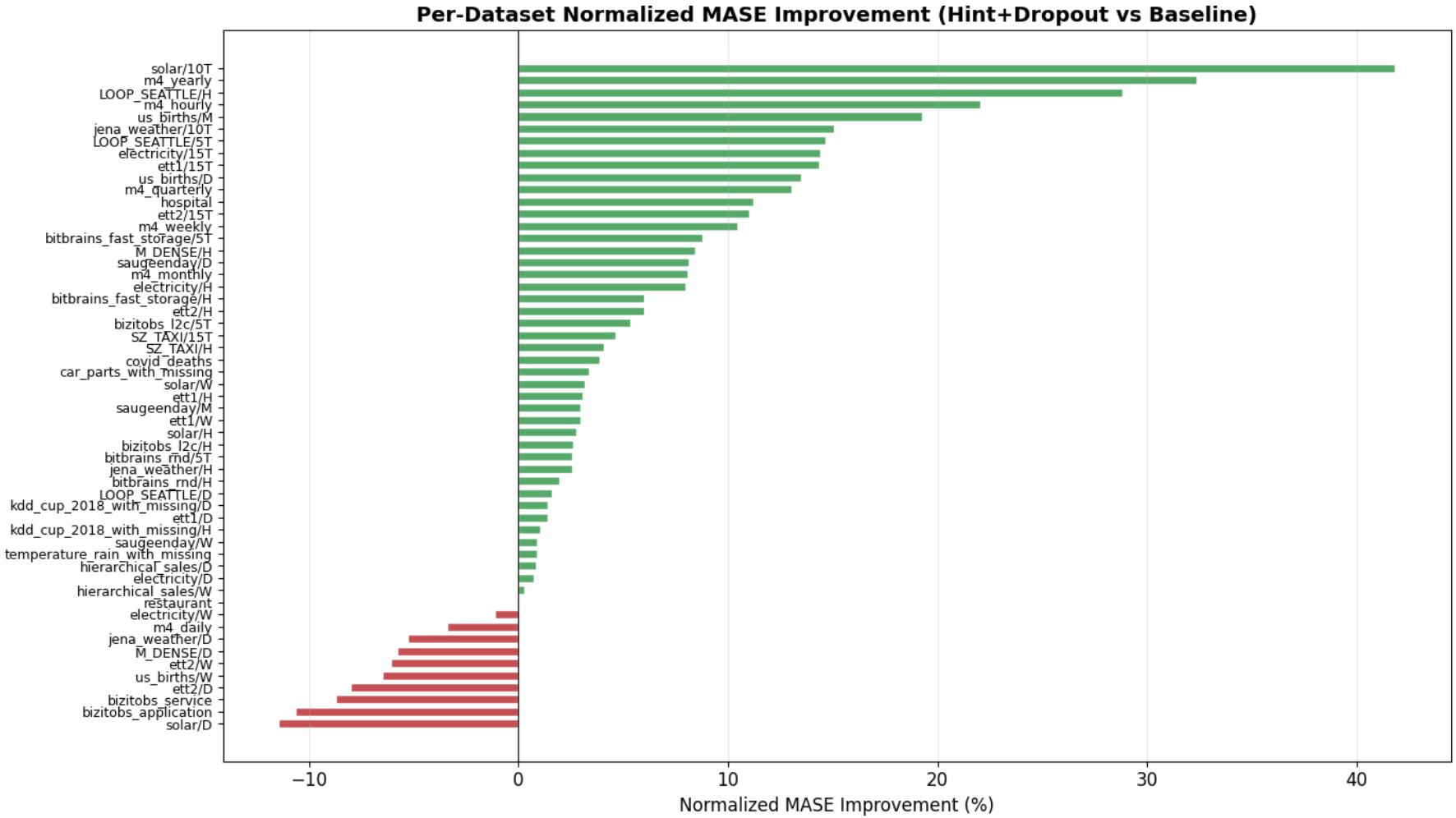
Frequency	N	Geo norm MASE (Baseline)	Geo norm MASE (Hint+Drop)	Improvement (%)	Wins	Losses
A-DEC	1	1.41	0.96	32.39	1	0
10T	6	1.28	0.90	29.72	5	1
Q-DEC	1	0.89	0.78	13.03	1	0
15T	12	1.02	0.90	11.19	12	0
W-SUN	1	0.97	0.87	10.47	1	0
M	5	0.86	0.78	9.19	5	0
5T	12	0.80	0.74	7.95	11	1
H	31	0.84	0.77	7.62	29	2
W-FRI	2	1.03	1.02	1.06	1	1
W-WED	1	0.73	0.73	0.29	1	0
D	15	0.74	0.74	0.09	10	5
W-THU	3	0.96	0.97	-0.67	2	1
W-TUE	1	0.92	0.98	-6.51	0	1
10S	6	1.98	2.17	-9.68	2	4



Horizon	N	Geo norm MASE (Baseline)	Geo norm MASE (Hint+Drop)	Improvement (%)	Wins	Losses
short	55	0.82	0.78	4.29	45	10
medium	21	1.05	0.94	10.24	18	3
long	21	1.11	0.97	12.64	18	3

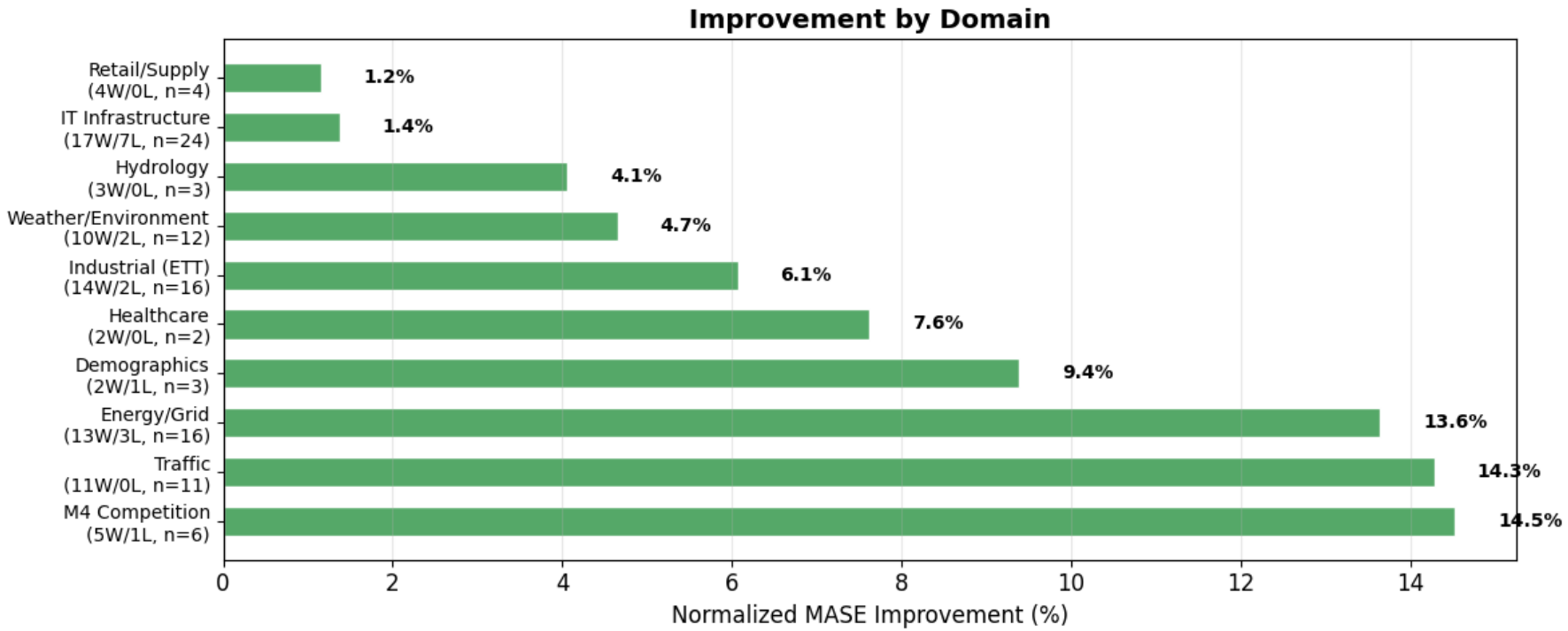
```
================================================================================
TOP 15 BIGGEST IMPROVEMENTS (Hint+Dropout over Baseline)
================================================================================
                     Config  Baseline norm MASE  Hint+Drop norm MASE  Improvement (%)
         solar/10T/10T/long              3.1362               1.2271          60.8735
        solar/10T/10T/medium              2.5478               1.1157          56.2089
      LOOP_SEATTLE/H/H/long              0.9549               0.6297          34.0553
      m4_yearly/A-DEC/short              1.4148               0.9566          32.3877
    LOOP_SEATTLE/H/H/medium              0.9777               0.6756          30.9024
          m4_hourly/H/short              1.0798               0.8416          22.0610
   electricity/15T/15T/long              1.4282               1.1283          21.0011
     LOOP_SEATTLE/H/H/short              0.8648               0.6843          20.8722
  LOOP_SEATTLE/5T/5T/medium              0.9021               0.7218          19.9887
 electricity/15T/15T/medium              1.3272               1.0656          19.7096
        us_births/M/M/short              0.9922               0.8011          19.2620
    LOOP_SEATTLE/5T/5T/long              0.8758               0.7087          19.0806
  jena_weather/10T/10T/long              1.1374               0.9243          18.7316
         ett1/15T/15T/long              1.2305               1.0075          18.1193
jena_weather/10T/10T/medium              1.0860               0.9353          13.8711

================================================================================
TOP 10 WORST REGRESSIONS
================================================================================
                        Config  Baseline norm MASE  Hint+Drop norm MASE  Improvement (%)
bizitobs_application/10S/medium              2.3150               2.7980         -20.8615
  bizitobs_application/10S/long              2.1614               2.5606         -18.4706
      bizitobs_service/10S/long              2.7192               3.1705         -16.5966
           solar/10T/10T/short               0.9659               1.1088         -14.7934
    bizitobs_service/10S/medium              2.6934               3.0303         -12.5049
              solar/D/D/short                0.9958               1.1096         -11.4249
               ett2/D/D/short                1.0024               1.0825          -7.9854
       us_births/W/W-TUE/short               0.9212               0.9812          -6.5095
            ett2/W/W-THU/short               1.3670               1.4503          -6.0938
             M_DENSE/D/D/short               0.4651               0.4921          -5.8004
```

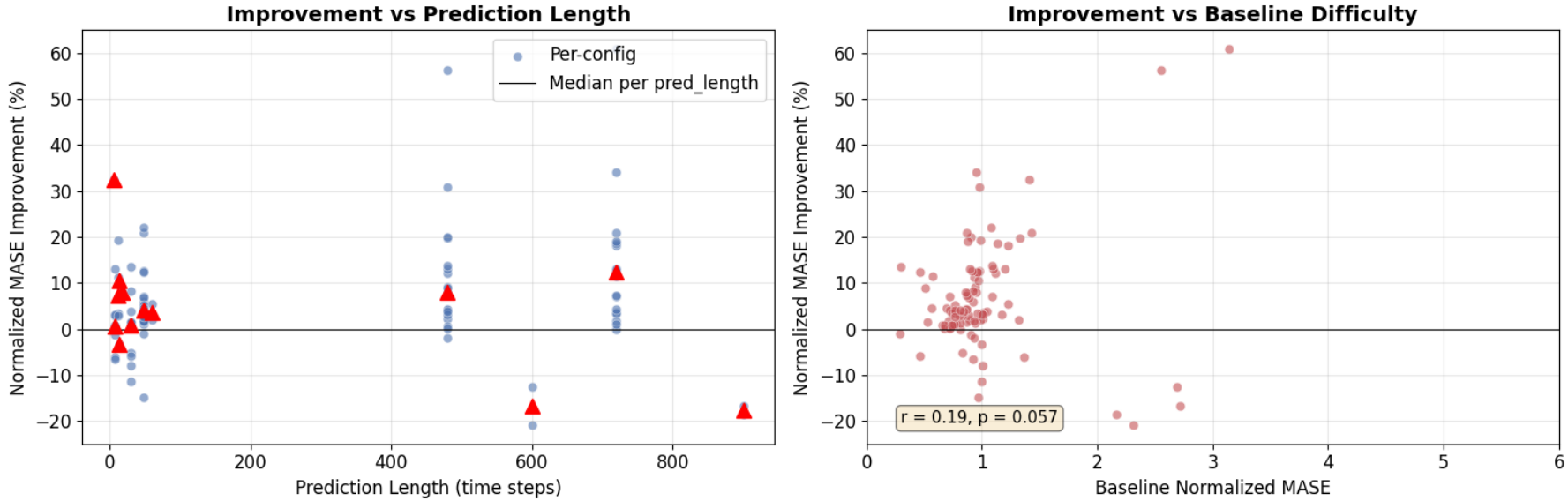


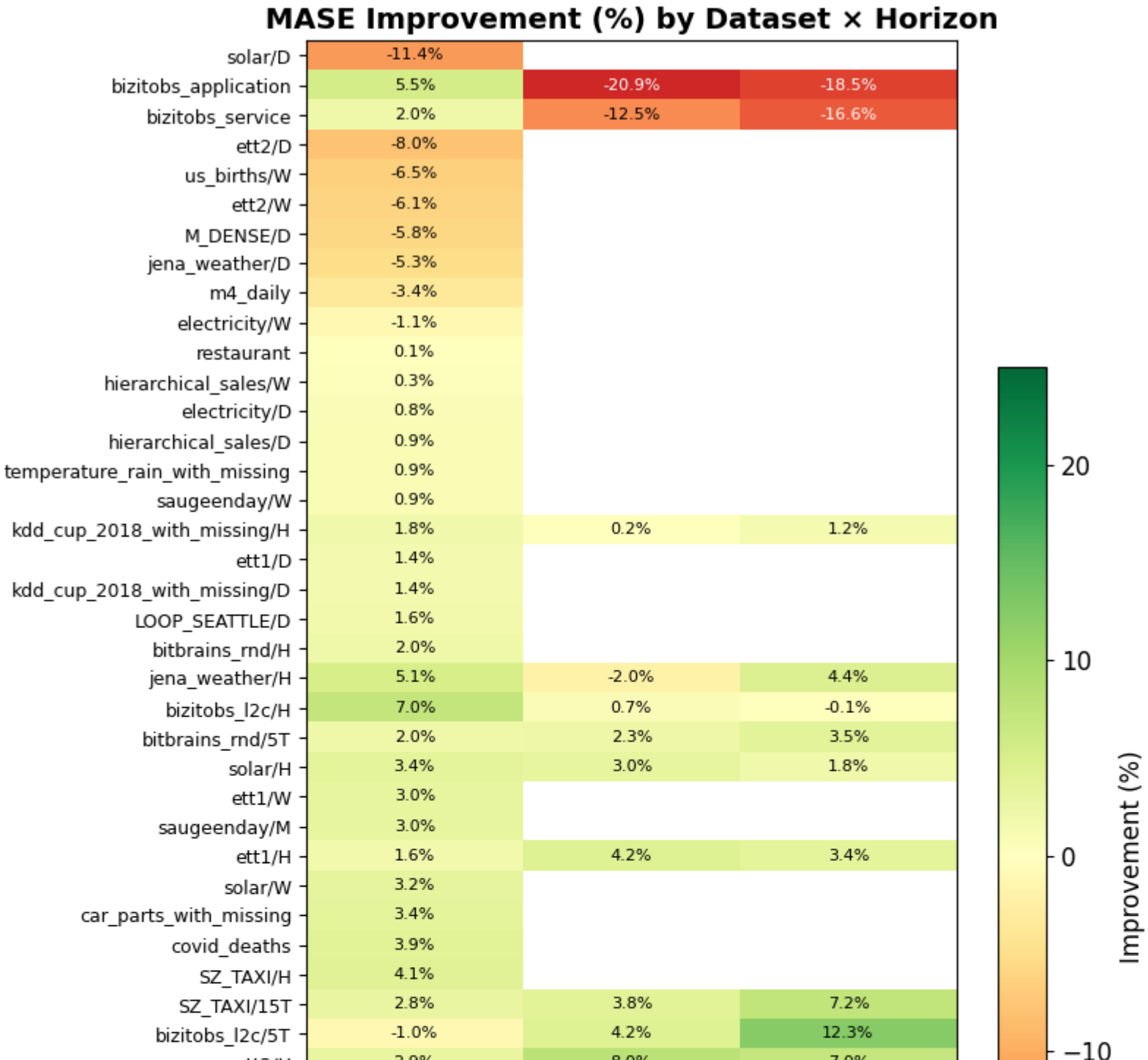
3.1 Where Do Hints Help Most?

To understand *why* hints help, we categorize datasets by domain and analyze patterns in the improvements.



Domain	N	Datasets	Geo norm MASE (Baseline)	Geo norm MASE (Hint)	Improvement (%)
M4 Competition	6	6	1.021	0.872	14.527
Traffic	11	5	0.823	0.706	14.287
Energy/Grid	16	8	1.110	0.959	13.647
Demographics	3	3	0.645	0.584	9.393
Healthcare	2	2	0.985	0.910	7.619
Industrial (ETT)	16	8	1.004	0.943	6.070
Weather/Environment	12	6	0.794	0.757	4.656
Hydrology	3	3	0.819	0.786	4.061
IT Infrastructure	24	10	0.937	0.924	1.377
Retail/Supply	4	4	0.702	0.693	1.158


Improvement vs Prediction Length
Per-config
Median per pred_length
Normalized MASE Improvement (%)
Prediction Length (time steps)
Improvement vs Baseline Difficulty
Normalized MASE Improvement (%)
r = 0.19, p = 0.057
Baseline Normalized MASE


MASE Improvement (%) by Dataset × Horizon
solar/D -11.4%
bizitobs_application 5.5% -20.9% -18.5%
bizitobs_service 2.0% -12.5% -16.6%
ett2/D -8.0%
us_births/W -6.5%
ett2/W -6.1%
M_DENSE/D -5.8%
jena_weather/D -5.3%
m4_daily -3.4%
electricity/W -1.1%
restaurant 0.1%
hierarchical_sales/W 0.3%
electricity/D 0.8%
hierarchical_sales/D 0.9%
temperature_rain_with_missing 0.9%
saugeenday/W 0.9%
kdd_cup_2018_with_missing/H 1.8% 0.2% 1.2%
ett1/D 1.4%
kdd_cup_2018_with_missing/D 1.4%
LOOP_SEATTLE/D 1.6%
bitbrains_rnd/H 2.0%
jena_weather/H 5.1% -2.0% 4.4%
bizitobs_l2c/H 7.0% 0.7% -0.1%
bitbrains_rnd/5T 2.0% 2.3% 3.5%
solar/H 3.4% 3.0% 1.8%
ett1/W 3.0%
saugeenday/M 3.0%
ett1/H 1.6% 4.2% 3.4%
solar/W 3.2%
car_parts_with_missing 3.4%
covid_deaths 3.9%
SZ_TAXI/H 4.1%
SZ_TAXI/15T 2.8% 3.8% 7.2%
bizitobs_l2c/5T -1.0% 4.2% 12.3%
20
10
0
−10
Improvement (%)

3.2 Analysis: Where and Why Preconditioning Hints Help

Strong improvements (>10%):

- **Energy/solar at sub-hourly frequency**: Solar 10T medium/long see 56–61% improvement — the largest gains in the benchmark. Solar energy has strong diurnal periodicity at exactly the patch-boundary scale, which the preconditioning filter directly encodes.
- **Traffic (LOOP_SEATTLE)**: 19–34% improvement. Traffic has strong daily/weekly periodicity. The hint at stride=16 captures the repeating pattern across patches.
- **Electricity sub-hourly**: 19–21% improvement at 15T medium/long. Similar periodic structure.
- **ETT sub-hourly**: 13–18% improvement at 15T long. Industrial transformer data has temporal regularity.

Moderate improvements (3–10%):

- **M4 competition datasets**: Hourly (+22%), yearly (+32%), quarterly (+13%), monthly (+8%). These are diverse time series with varying autocorrelation — hints help on average.
- **Weather/jena**: 13–19% at 10T, more modest at hourly. Hints capture temperature cycles.
- **Hydrology, healthcare, demographics**: Mixed, 3–19%.

Regressions (<0%):

- **bizitobs (application/service) at 10S**: −15% to −21%. These are the most fine-grained series (10-second intervals). The preconditioning filter at stride=16 looks back 160 seconds and 480 seconds — these may not be meaningful timescales for application monitoring, where patterns are more bursty than periodic.
- **solar/D short, solar/10T short**: −11% to −15%. Interestingly, solar *short* horizon hurts while medium/long helps massively. At short horizons, the model may not need inter-patch structure — the immediate context is sufficient.
- **Weekly/daily data (ett2/W, us_births/W, M_DENSE/D)**: −5% to −8%. Lower-frequency data has fewer patches in the context window, so the preconditioning filter has less material to work with.

Key insight: Preconditioning hints help most when **(a)** the data has strong inter-patch autocorrelation at the stride-16 timescale, and **(b)** the prediction horizon is long enough that the model benefits from explicit structural information. Short-horizon predictions on irregular/bursty data see the least benefit.

4. How Hint Dropout Works

Hint dropout is a regularization technique applied **only during training** that randomly zeros out entire hint channels for individual patches. Here's how it works:

Mechanism

During each training forward pass:

1. The hint channel `hint_d4[t]` is computed for every patch
2. For each patch independently, a random number is drawn from Uniform(0, 1)
3. If that number < 0.1 (the 10% dropout rate), the **entire** hint vector for that patch is set to zero
4. The remaining 90% of patches keep their full hint information

```
# From module.py — the actual implementation:
if training_mode and self.hint_dropout > 0:
    for i in range(len(hint_channels)):
        hint_mask = torch.rand(
            hint_channels[i].shape[:-1], device=hint_channels[i].device
        ) < self.hint_dropout
        hint_channels[i] = hint_channels[i].masked_fill(hint_mask.unsqueeze(-1), 0.0)
```

Key design choices:

- **Per-patch, not per-element**: The entire 16-dimensional hint vector for a patch is either kept or zeroed. This is analogous to spatial dropout in CNNs — it forces the model to not rely on any single patch's hint.
- **Training only**: At inference, all hints are always present (no dropout).
- **Independent per patch**: Each patch in the sequence independently decides whether to drop its hint.

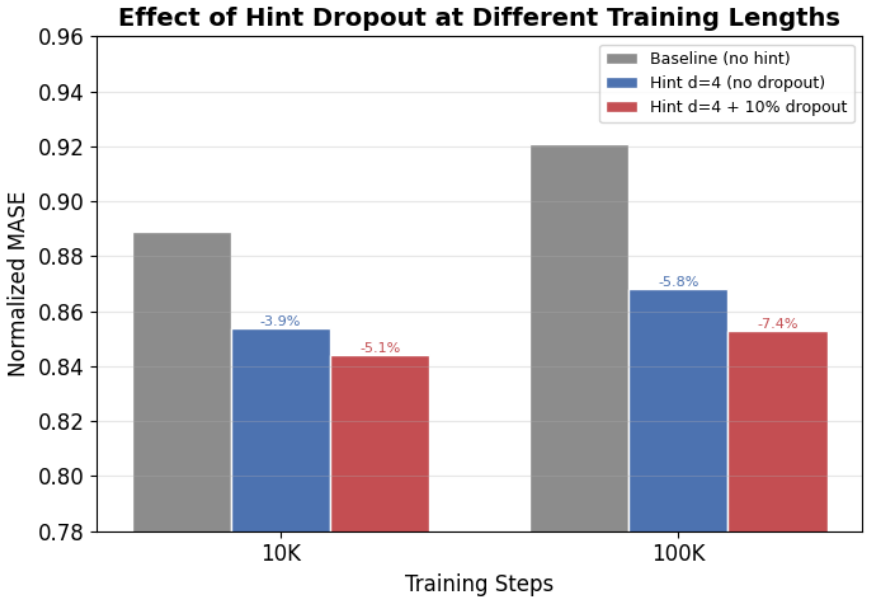
Why it matters at 100K steps

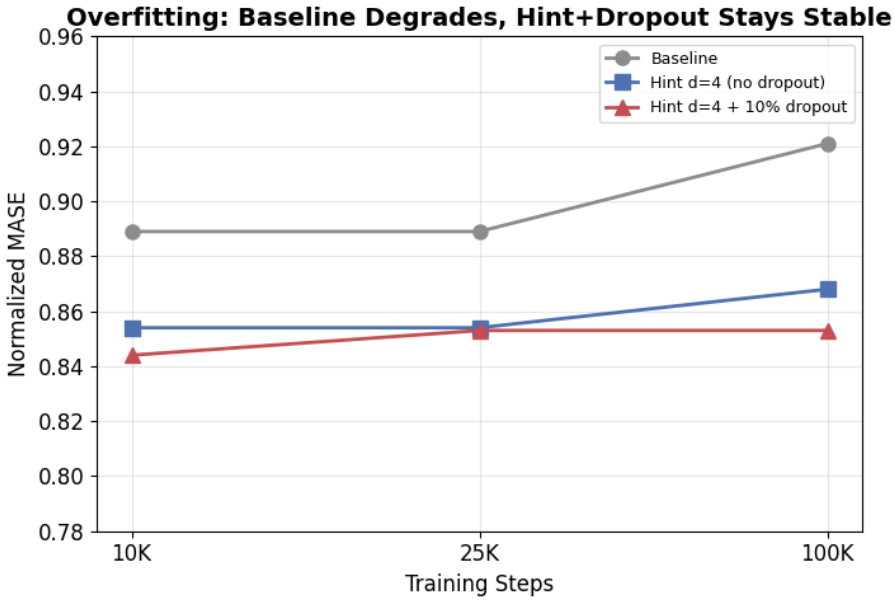
Training	Without dropout	With 10% dropout
10K steps	Norm MASE 0.854	Norm MASE 0.844 (-1.2%)
100K steps	**Norm MASE 0.868**	**Norm MASE 0.853 (-1.8%)**
vs 100K baseline	-5.77%	**-7.45%**

At short training (10K), hints already act as a regularizer — the model hasn't memorized the training data yet, so adding dropout is only marginally helpful.

At long training (100K), the baseline **overfits** (norm MASE degrades from 0.889 at 10K to 0.921 at 100K). Without dropout, the hint model also partially overfits by learning to rely too heavily on the hint signal. With 10% dropout, the model is forced to sometimes make predictions without hints, which:

1. Prevents over-reliance on the hint channel
2. Encourages the model to learn both hint-based and hint-free prediction strategies
3. Acts as an ensemble effect — at inference, the full hint is always available, giving the model strictly more information than it had during any single training step





5. Inference Pipeline Comparison

Baseline Pipeline

```
Raw series --> z-score normalize --> [target, mask] --> in_proj (32->384) --> 6-layer causal transformer --> out_proj --> quantile predictions
```

Preconditioning Hint Model Pipeline

```
Raw series --> z-score normalize --> compute preconditioning hint (causal, backward-looking only)
                                                |
                                                v
                                  [target, mask, hint_d4] --> in_proj (48->384) --> 6-layer causal transformer --> out_proj --> quantile predictions
                                                ^
                                   hint is ZEROED in prediction window
                                   (future values unknown)
```

What the preconditioning hint provides

The Chebyshev degree-4 preconditioning filter with stride=16 (= patch_size) computes:

```
hint[t] = -1.0 * y[t-32] + 0.125 * y[t-64]
```

This is the **residual** of applying a causal preconditioning filter -- it captures inter-patch autocorrelation at exactly the patch boundary. Because stride=16 matches the patch size, each coefficient connects corresponding positions across patches:

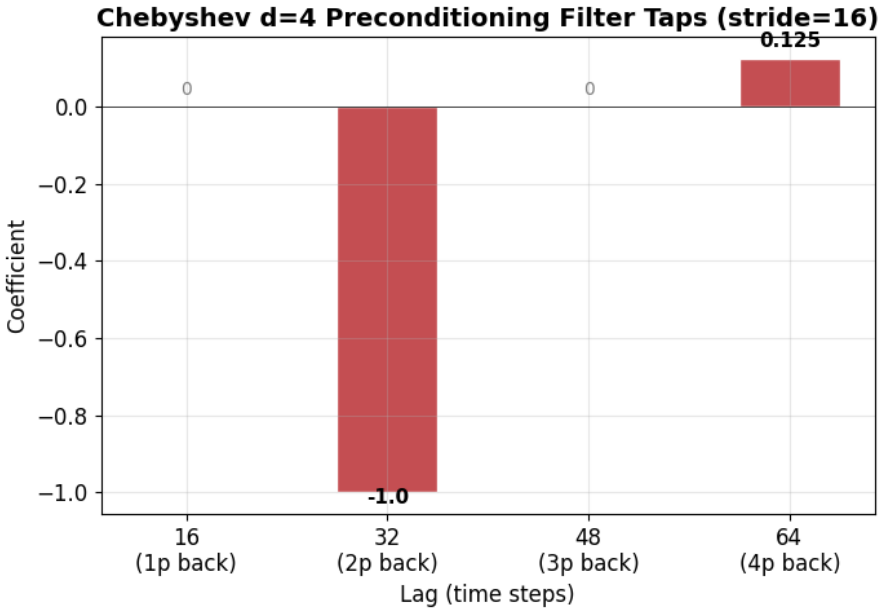
- `y[t-32]`: same position, 2 patches back
- `y[t-64]`: same position, 4 patches back

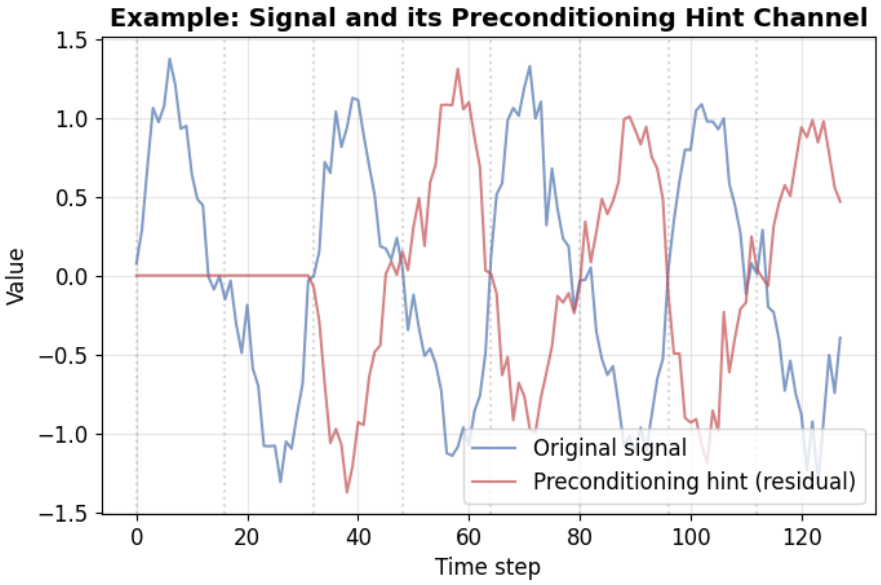
(The Chebyshev d=4 polynomial has 4 coefficient slots at lags 16, 32, 48, 64, but the even-indexed coefficients are zero -- only the lags at 2 and 4 patches back are active.)

The transformer's attention already models patch-to-patch relationships, but it must discover these patterns from raw data. The hint **explicitly encodes** this structure, providing an inductive bias that the model can learn to use or ignore.

Why hint mode works (but reversal doesn't)

- **Reversal mode**: Predict in preconditioned space, then analytically undo the filter. Any prediction error gets amplified through the inverse -- errors compound across patches.
- **Hint mode**: Predictions stay in original z-scored space. The hint is just extra information. The model predicts exactly the same target as baseline, with strictly more input information.





Autoregressive Generation and Hints

Moirai2 generates forecasts **autoregressively**: it predicts `num_predict_token = 4` patches at a time, then feeds those predictions back as "observed" context for the next forward pass. This repeats until the full prediction horizon is filled.

Key question: Are hints recomputed from predicted values during autoregressive generation?

No. The hint channels for all prediction-window patches are **zeroed at the start** and never updated. Specifically:

1. Before generation begins, `expand_target` is prepared with context values filled in and prediction slots empty
2. The hint is computed from the context window only — prediction-window positions are zeroed via `h.masked_fill(input_mask.unsqueeze(-1), 0.0)` (see `module.py`)

3. During the autoregressive loop (`forecast.py:374-429`), predicted quantile values are written back into `expand_target` at the next patch positions, and the prediction mask is updated
4. But the hint channels for those newly-filled patches remain zero — they are never recomputed from predictions

```
Step 1:  [ctx_1, ctx_2, ..., ctx_N, ???, ???, ???, ???]   ← context filled, prediction empty
         [h_1,   h_2,   ..., h_N,   0,   0,   0,   0]   ← hints: context computed, prediction zeroed

Step 2:  [ctx_1, ctx_2, ..., ctx_N, p_1, p_2, p_3, p_4, ???, ???, ???, ???]
         [h_1,   h_2,   ..., h_N,   0,   0,   0,   0,   0,   0,   0,   0]  ← still zeroed!

Step 3:  [ctx_1, ctx_2, ..., ctx_N, p_1, p_2, ..., p_8, ???, ???, ???, ???]
         [h_1,   h_2,   ..., h_N,   0,   0,   ..., 0,   0,   0,   0,   0]  ← still zeroed
```

This is consistent with training, where hint dropout (10%) forces the model to sometimes predict without hints. The model learns a dual strategy: use hints when available (context), predict without them when not (prediction window).

6. Summary

Key takeaways

1. **The preconditioning hint model achieves -7.45% lower MASE vs baseline at 100K steps** (normalized MASE 0.853 vs 0.921), with only +12K extra parameters (+0.1%).

2. **The baseline overfits at 100K** (normalized MASE degrades from 0.889 at 10K to 0.921 at 100K). The hint+dropout model does not — it maintains 0.853, even better than the 10K baseline.

3. **Hint dropout is critical at long training**. Without it, hint d=4 at 100K gives -5.77%; with 10% dropout it gives -7.45%. The dropout prevents the model from over-relying on the hint signal.

4. **Improvements are concentrated in periodic, sub-hourly data at long horizons**. Solar, traffic, electricity, and ETT datasets at 5T/10T/15T/H frequencies see 10-61% improvement. The preconditioning filter at stride=16 directly encodes inter-patch autocorrelation that matches the natural periodicity of these domains.

5. **Regressions occur on bursty/irregular data and short horizons**. IT monitoring data at 10S frequency (−15% to −21%) and short-horizon predictions on low-frequency data (−5% to −8%) see degradation. The stride-16 preconditioning filter taps don't align with meaningful timescales for these series.

6. **Hints are context-only during autoregressive generation**. The model generates 4 patches per step, feeds predictions back, but hint channels remain zeroed for all prediction patches. Hint dropout during training prepares the model for this.

7. **The architecture change is minimal**: same transformer, same loss, same everything except `in_proj` goes from 32 to 48 dims to accommodate the hint channel.

Caveats

- **LR schedule differs**: Baseline uses 10K warmup, this hint model uses 1K warmup. A matched-schedule run is in progress (job 5087530) to isolate the hint effect.
- **Win rate is 81/16**, but 5 of those 16 losses are on 10S IT monitoring data (bizitobs). If this domain is out of scope, the win rate is 81/11.

MASE normalization note: All MASE numbers above are normalized by the Seasonal Naive model per config, consistent with the GIFT-Eval leaderboard. For reference, the official Moirai 2.0-R-small achieves normalized MASE = 0.728 on the leaderboard.

7. Lessons Learned from 50+ Experiments

The following lessons are distilled from over 50 preconditioning experiments across 4 polynomial families, 7 degrees, 5 dropout rates, multiple strides, and several architectural variants. All MASE numbers are **normalized by the Seasonal Naive model per config** (geometric mean across 97 GIFT-Eval configurations), consistent with the GIFT-Eval leaderboard.

Lesson 1: Reversal Fails — Hint Mode Is the Only Way

Standard preconditioning (filter the input, train, analytically reverse at inference) **does not work** for multi-step forecasting. Forecast errors compound through the inverse filter:

Approach	Norm MASE	vs Baseline
Reversal d=2 s=16	0.875	-1.56%
Reversal d=4 s=16	0.893	+0.53%
First differencing (coeff -1.0)	1.139	+28.2%
Dual-head (raw + precond outputs)	1.114	+25.3%
Hint d=4 s=16	**0.854**	**-3.84%**
Hint d=6 s=16	**0.847**	**-4.71%**

The fundamental issue: any prediction error in preconditioned space gets amplified when reversing. Higher-degree polynomials make this worse (d=4 reversal is already worse than baseline). Hint mode avoids reversal entirely — the model predicts in raw z-scored space with the filter residual as extra information.

Lesson 2: Stride Must Match Patch Size

The preconditioning filter stride must equal the patch size (16) so that filter taps connect corresponding positions across patches, aligning with how the transformer's attention processes temporal information:

Stride	Context	Norm MASE	Note
s=1	Intra-patch correlations	~1.14	Catastrophic failure
s=4	hint d=6 s=4	0.883	-0.60% — mediocre
s=8	Multi-stride d=4 s=16+s=8	0.863	-2.93%
s=16	**Inter-patch, patch-aligned**	**0.847**	**-4.71% (d=6)**

Stride=1 creates intra-patch correlations that confuse the patch embedding. Stride=16 ensures each filter tap references the same position in a previous patch, which is semantically meaningful for the transformer.

Lesson 3: Multi-Scale d=4+d=6 Is Uniquely Optimal

Adding a second hint channel at a different polynomial degree captures complementary spectral information. But the specific combination d=4+d=6 is a sharp optimum:

Multi-Scale Config	Norm MASE	vs Baseline
d=4 + d=6	**0.835**	**-6.01%**
d=4 + d=2 + d=6 (triple)	0.854	-3.87%
d=4 + d=5	0.875	-1.58%
d=4 + d=8	0.868	-2.30%
d=4 + d=6 + d=8 (triple)	0.877	-1.32%
d=6 + d=8 (no d=4)	0.879	-1.08%

d=4 and d=6 capture complementary frequency bands — d=4 provides moderate smoothing (max|c|=1.0) while d=6 captures higher-order spectral detail (max|c|=1.5). Adding d=5 is redundant (too similar to d=4), d=8 is too aggressive, and triple combinations add noise.

Lesson 4: Primary Degree Ordering Matters

In multi-scale configurations, the first hint channel gets preferential treatment from the input projection. Swapping d=4 and d=6 as primary/secondary is catastrophic:

Config	Norm MASE	vs Baseline
d=4 primary + d=6 extra	**0.835**	**-6.01%**
d=6 primary + d=4 extra	0.885	-0.45%

Swapping costs 5.6% in MASE. d=4 is the anchor degree for multi-scale hints.

Lesson 5: Degree Sweep Is Non-Monotonic

The relationship between polynomial degree and performance is not monotonic — there are two local optima at d=4 and d=6:

Degree	Chebyshev Coefficients (non-zero)	Norm MASE	vs Baseline
d=2	c1=-0.5	0.870	-2.13%
d=3	c1=-0.75	0.861	-3.07%
d=4	**c1=-1.0, c3=0.125**	**0.854**	**-3.84%**
d=5	c1=-1.25, c3=0.3125	0.864	-2.71%
d=6	**c1=-1.5, c3=0.5625, c5=-0.03**	**0.847**	**-4.71%**
d=7	c1=-1.75, c3=0.875, c5=-0.109	0.860	-3.17%
d=8	c1=-2.0, c3=1.25, c5=-0.25, c7=0.016	0.874	-1.65%

For even-degree Chebyshev polynomials, the coefficients at even array positions (0, 2, 4, ...) are zero because T_n(z) contains only even powers of z. This means the effective filter taps are sparser than the degree suggests. d=8 degrades (max|c|=2.0) likely due to numerical precision issues in bf16 training.

Lesson 6: Hint Dropout — Hurts at 10K, Critical at 100K

Hint dropout (randomly zeroing entire hint patches during training) has opposite effects depending on training length:

At 10K steps (short training):

Model	No Dropout	10% Dropout	Effect
d=4	0.854	0.844	Helps (-1.2%)
d=6	0.847	0.877	Hurts (+3.6%)
ms d=4+d=6	0.835	0.845	Hurts (+1.2%)

At 100K steps (long training):

Model	No Dropout	10% Dropout	Effect
d=4	0.868 (-5.77%)	0.853 (-7.45%)	Critical (+1.68%)

At short training, hints already act as regularizers, so additional dropout is redundant or harmful. At long training, dropout prevents the model from over-relying on hints, maintaining their regularization benefit against overfitting. **For production deployments with long training, hint dropout is essential.**

Lesson 7: Polynomial Family Comparison

Four polynomial families tested at d=6, stride=16:

Family	max|coeff|	Norm MASE	vs Baseline	Description
L2-optimized	**~0.28**	**0.843**	**-5.13%**	Minimizes L2 norm of coefficients
Chebyshev	1.50	0.847	-4.71%	Standard orthogonal polynomials
Lyapunov	~0.23	0.857	-3.51%	Minimizes Lyapunov exponent
Legendre	1.36	0.866	-2.60%	Orthogonal on [-1,1]

Surprise: L2-optimized (mildest coefficients) beats Chebyshev at d=6 by 0.44%. But L2-opt **fails at d=4** (0.896, +0.89% vs baseline) — not enough discriminative power at low degree.

For multi-scale, Chebyshev dominates: Pure Chebyshev d=4+d=6 (0.835, -6.01%) dramatically outperforms L2-opt multi-scale (0.864, -2.75%) and mixed Cheb+L2-opt (0.864, -2.73%). The strong d=4 anchor requires Chebyshev's stronger coefficients.

Lesson 8: Learnable Coefficients Underperform Fixed Polynomials

When allowed to learn optimal FIR coefficients via backpropagation (initialized from Chebyshev d=6), the model converges to **near-identity filters** (~4x weaker than Chebyshev d=2):

Config	Norm MASE	vs Baseline
Fixed Chebyshev d=6	0.847	-4.71%
Learnable (initialized d=6)	**0.860**	**-3.19%**

The learned coefficients essentially undo the preconditioning, optimizing for training loss reduction rather than generalizable spectral features. Fixed polynomial coefficients provide **inductive bias that the model cannot discover on its own**.

Lesson 9: Benefits Decay with Training Length

Hint preconditioning benefit is largest early in training and shrinks over time:

Training Steps	ms d=4+d=6 Norm MASE	vs Baseline	Relative Improvement
10K	0.835	vs 0.889	**-6.01%**
25K	0.874	vs 0.889	-1.69%

Similarly for single-scale d=5: -2.71% at 10K but +0.25% at 25K (worse than baseline!). The hint provides inductive bias that is most valuable in the **low-data/early-training regime**. At longer training, the model learns spectral patterns from raw data alone, making hints partially redundant.

However, 100K tells a different story because the baseline itself overfits (0.921 vs 0.889 at 10K). Here, hints provide regularization against overfitting: d=4+dropout gives -7.45% at 100K, the **largest relative improvement of any experiment**.

Lesson 10: Anomaly Filtering Is a Major Confound

Setting `anomaly_zscore_threshold=8.0` (filtering extreme outlier sequences) had a larger impact than most preconditioning experiments:

Model	Norm MASE	Note
Published Moirai 1.1 Small	0.946	Official benchmark
Our 10K baseline (with zscore=8.0)	0.889	**-6.3% vs published**

This single setting accounts for most of the gap with published numbers. **All valid preconditioning comparisons must use identical zscore threshold on both baseline and experimental runs** — which we do. But it means our baselines are already strong, making improvements harder.

Lesson 11: Flash-STU Hybrid Underperforms

Parallel STU+Attention architecture (spectral transform units alongside attention in each layer, zero-init tanh gate, approx mode with K=24 Hankel filters):

Model	Params	Norm MASE	vs 10K Baseline
Flash-STU v2 (parallel, approx)	11.75M	0.933	**+5.01%**
Baseline	11.05M	0.889	—

The STU hybrid significantly underperforms baseline at 10K steps. Possible causes: zero-init gates need longer training to open; reduced d_ff (1024->940) to fit extra params hurts attention capacity. The spectral inductive bias of STU does not appear complementary to attention for time series forecasting at this model scale and training length.

7. Suggested Next Steps

Immediate (pending jobs)

1. **Wait for matched-schedule result** (job 5087530 -> 5087878): The hint model with 10K warmup (matching baseline) will give a clean apples-to-apples comparison. This eliminates the LR schedule confound.
2. **Wait for multi-scale d=4+d=6 at 100K with dropout** (job 5092070): Combines our best 10K architecture (norm MASE 0.835) with the dropout regularization that proved critical at 100K.

Short-term research

1. **Adaptive stride / frequency-aware hints**: The biggest failure mode is the mismatch between stride=16 and the data's natural periodicity. For 10S data, stride=16 means looking back 160 seconds — possibly meaningless for application monitoring. A frequency-adaptive stride (e.g., stride=1 period) could recover these regressions without hurting existing gains.

2. **Autoregressive hint propagation**: Currently hints are zeroed for all prediction patches. An alternative: recompute hints from predicted values during autoregressive generation. This would provide the model with the same preconditioning filter signal in the prediction window that it had during context. Risk: error propagation through the filter. But since the filter is simple (2 non-zero taps), errors should be bounded.

3. **Closing the gap with official Moirai2**: Our best model (norm MASE 0.853) vs official Moirai 2.0-R-small (0.728). The ~15% gap is likely due to (a) official model trained with data weighting, (b) longer training, (c) possibly different hyperparameters. Understanding and closing this gap is important for publication.

Longer-term directions

1. **Hint dropout curriculum**: Instead of constant 10% dropout, anneal dropout from high (50%) to low (5%) during training. Early training gets regularization; late training gets full signal.

2. **Scale to base/large models**: All experiments so far are on Moirai2 Small (11.4M params). The hint adds +0.1% parameters regardless of model size, so the overhead shrinks at larger scale. Does the relative improvement hold?

3. **Combine with Flash-STU hybrid**: The STU branch provides spectral inductive bias; preconditioning hints provide autocorrelation bias. These are complementary — STU captures global spectral structure while hints capture local inter-patch structure. Worth testing whether the combination is additive.