Supplementary Information for "Electricity Load Forecasting Under Extreme Heat and Cold Waves"

Nan Lu[1,2], Dalin Qin[1,2], Yangze Zhou[1,2], Lei Bai[3], Qingsong Wen[4], Chongqing Kang[5], Yi Wang[1,2*]

[1]Department of Electrical and Computer Engineering, The University of Hong Kong, Hong Kong SAR, China.
[2]The University of Hong Kong Shenzhen Institute of Research and Innovation, Shenzhen, China.
[3]Shanghai AI Laboratory, Shanghai, China.
[4]Squirrel AI Learning, Bellevue, USA.
[5]Department of Electrical Engineering, Tsinghua University, Beijing, China.

*Corresponding author(s). E-mail(s): yiwang@eee.hku.hk;

Contents

Supplementary Notes 3

Supplementary Note 1 — Heat/Cold waves criterion 3
Supplementary Note 2 — Evaluation metrics 4
Supplementary Note 3 — ESDF implementation details 5
Supplementary Note 4 — Baseline settings 11
Supplementary Note 5 — Quantitative evaluation of synthetic sample fidelity 13
Supplementary Note 6 — Load forecasting valuation method 17
Supplementary Note 7 — Proof for conditional reverse process 19

Supplementary Figures 21

Supplementary Figure 1 21
Supplementary Figure 2 22
Supplementary Figure 3 23
Supplementary Figure 4 24
Supplementary Figure 5 25
Supplementary Figure 6 26

Supplementary References 27

Supplementary Notes

Supplementary Note 1 — Heat/Cold waves criterion

To identify heatwave and cold-wave periods, we adopted an operational event-labeling procedure based on the Excess Heat Factor (EHF) / Excess Cold Factor (ECF) framework, which jointly characterizes event intensity and short-term acclimatization. This family of indices has been widely used in prior meteorological and health studies and follows the general formulation in the related literature[1,2].

For each day i, the daily mean temperature T_i is defined in equation (1):

$$T_i = \left(T_{\max,i} + T_{\min,i}\right)/2. \tag{1}$$

where $T_{\max,i}$ and $T_{\min,i}$ denote the maximum and minimum temperatures of the i-th day, respectively. Following the standard formulation, the heatwave significance term $\mathrm{EHI}_{\mathrm{sig},i}$ and acclimatization term $\mathrm{EHI}_{\mathrm{accl},i}$ are defined in equations (2) and (3), respectively:

$$\mathrm{EHI}_{\mathrm{sig},i} = \left(T_i + T_{i+1} + T_{i+2}\right)/3 - T_{95}, \tag{2}$$

$$\mathrm{EHI}_{\mathrm{accl},i} = \left(T_i + T_{i+1} + T_{i+2}\right)/3 - \left(T_{i-1} + \cdots + T_{i-30}\right)/30, \tag{3}$$

where T_{95} denotes the 95th percentile of T_i. The Excess Heat Factor EHF_i is then computed using equation (4):

$$\mathrm{EHF}_i = \max\left(0,\ \mathrm{EHI}_{\mathrm{sig},i} \times \max\left(1, \mathrm{EHI}_{\mathrm{accl},i}\right)\right). \tag{4}$$

Similarly, the cold-wave significance term $\mathrm{ECI}_{\mathrm{sig},i}$ and acclimatization term $\mathrm{ECI}_{\mathrm{accl},i}$ are defined in equations (5) and (6), respectively:

$$\mathrm{ECI}_{\mathrm{sig},i} = \left(T_i + T_{i+1} + T_{i+2}\right)/3 - T_{05}, \tag{5}$$

$$\mathrm{ECI}_{\mathrm{accl},i} = \left(T_i + T_{i+1} + T_{i+2}\right)/3 - \left(T_{i-1} + \cdots + T_{i-30}\right)/30, \tag{6}$$

where T_{05} denotes the 5th percentile of T_i. The Excess Cold Factor ECF_i is then computed using equation (7):

$$\mathrm{ECF}_i = \min\left(0,\ -\mathrm{ECI}_{\mathrm{sig},i} \times \min\left(-1, \mathrm{ECI}_{\mathrm{accl},i}\right)\right). \tag{7}$$

In the original EHF/ECF framework, T_{95} and T_{05} in equations (2) and (5) are defined relative to long-term climatological statistics, typically derived from a multi-year historical archive. However, a harmonized ∼30-year temperature record was not available for all 80 datasets used in this study. Therefore, for cross-region consistency, we operationally approximated these climatological references using statistics computed from the available pre-test training period only for each dataset.

Event detection was performed at the daily level. A a day was labeled as a heatwave day if $\mathrm{EHF}_i > 0$, and as a cold-wave day if $\mathrm{ECF}_i < 0$. A load forecasting sample was then labeled as an extreme-event sample if its target day fell within a heatwave or cold-wave period identified by the above criteria. The same event labels were used consistently for dataset construction, diffusion conditioning, and final out-of-sample evaluation.

Supplementary Note 2 — Evaluation metrics

To compare the performances among different datasets, we use normalized mean absolute error (nMAE) and normalized root mean squared error (nRMSE) as evaluation metrics. Since the raw data have already been normalized during processing, the normalized MAE and RMSE are numerically equivalent to the MAE and RMSE calculated directly from the model outputs. The nRMSE, nMAE, and accuracy are defined in equations (8), (9), and (10), respectively:

$$\text{nRMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{8}$$

$$\text{nMAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{9}$$

$$\text{Accuracy} = 1 - \text{nMAE}. \tag{10}$$

where $\hat{y}_i$ and y_i are the i-th forecasted and observed load values, respectively, and N is the number of samples.

Supplementary Table 1: Configuration details for extreme classifier-guided diffusion synthesis stage

Parameter	Configuration
Architecture	- Denoiser structure: U-Net (4-layer) - Time embedding type: Gaussian Fourier Projection - Time embedding dim: 256 - Number of kernels: 32, 64, 128, 256 - Classifier structure: CNN (2-layer) - Number of kernels: 32, 64 - Pooling: Max pooling - Hidden layer activation function: Sigmoid
Training	- Maximum training epochs: 800 - Batch size: 32 - Initial learning rate: 1×10^{-4} - Optimizer: Adam - Diffusion scheme: Variance Exploding - Discretized diffusion steps: 1000 - Drift coefficient: 0 - Min. std. of diffusion coefficient: 0.01 - Max. std. of diffusion coefficient: 50
Sampling	- Sampling scheme: predictor–corrector sampler - Number of diffusion discretization steps T: 1000 - Classifier guidance strength: selected on validation from $\{5, 10\}$ - Corrector signal-to-noise ratio r: 0.2 - Corrector step size: $2\left(r\|\mathbf{z}\|_2 / \max(\|\mathbf{g}\|_2, 10^{-5})\right)^2$

Supplementary Note 3 — ESDF implementation details

Experimental configurations

For the extreme classifier-guided diffusion synthesis stage, the adopted hyperparameter settings are summarized in Supplementary Table 1.

For the fidelity-aware representation disentanglement stage, the adopted hyperparameter settings are summarized in Supplementary Table 2. Note that hyperparameter selection followed a two-step procedure to ensure fair comparison with the corresponding plain backbones. We first determined the best-performing configuration of each forecasting backbone in the standard training setting using the validation set. This selected backbone configuration, including the architecture and backbone-specific hyperparameters, was then kept unchanged when constructing the corresponding ESDF-enhanced model. Only Extreme Synthesis-Disentanglement Forecasting (ESDF)-specific hyperparameters were additionally tuned on the validation set. Therefore, the ESDF-enhanced models did not receive an additional backbone re-tuning advantage over their plain counterparts, and the comparison isolates the effect of the proposed ESDF framework itself.

Supplementary Table 2: Configuration details for fidelity-aware representation disentanglement stage

Parameter	Configuration
Training	- Validation set: random 10% of the training set - Maximum training epochs: 2000 - Early stopping epochs: 150 - Batch size: 128 - Initial learning rate: 1×10^{-3} - Optimizer: Adam - Loss weight λ_1: $\in \{0.25, 0.5, 1\}$ - Loss weight λ_2: $\in \{0.25, 0.5, 1\}$ - Loss weight λ_3: $\in \{0.01, 0.1, 0.5\}$
Architecture	- Shared encoder structure: same as baseline models (MLP, LSTM, CNN) - Specific encoder structure: MLP - Hidden size of all encoders: $\in \{64, 96, 128\}$ - Decoder structure: MLP - Hidden activation function: ReLU

Algorithm details for extreme classifier-guided diffusion synthesis

Step 1: Denoiser and classifier training. Let $\mathbf{S}_0$ denote a clean sample drawn from the training dataset $\{\mathbf{S}\}$. The denoising score network is denoted by D_θ, and the extreme-weather classifier is denoted by C_ϕ, where θ and ϕ are their trainable parameters, respectively. For each mini-batch, a diffusion time $t \in (0, 1)$ is randomly sampled. The clean sample $\mathbf{S}_0$ is perturbed using equation (11).

$$\mathbf{S}_t = \mathbf{S}_0 + \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^t \boldsymbol{\epsilon}, \tag{11}$$

where $\boldsymbol{\epsilon}$ denotes Gaussian noise, and $\sigma_{\min}$ and $\sigma_{\max}$ are the minimum and maximum noise levels. The perturbed sample $\mathbf{S}_t$ defined in equation (11) is used to train both the denoising score network and the extreme-weather classifier.

The denoising score network is optimized by minimizing the weighted score-matching loss defined by equation (12).

$$\mathcal{L}_{\text{score}} = \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{\mathbf{S}_0} \mathbb{E}_{\mathbf{S}_t|\mathbf{S}_0} \left[\| D_\theta(\mathbf{S}_t, t) - \nabla_{\mathbf{S}_t} \log P_t(\mathbf{S}_t|\mathbf{S}_0) \|_2^2 \right] \right\}, \tag{12}$$

where $\lambda(t)$ is the time-dependent weighting function and $P_t(\mathbf{S}_t|\mathbf{S}_0)$ denotes the conditional perturbation distribution. The objective in equation (12) encourages D_θ to approximate the score of the perturbed data distribution.

The denoising network parameters are updated using equation (13).

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{score}}, \tag{13}$$

where η is the learning rate. Equations (12) and (13) define the score-matching objective and the corresponding parameter update for the denoising network.

The classifier C_ϕ is trained to distinguish extreme-weather samples from non-extreme samples using perturbed inputs. For a mini-batch of N samples with binary labels c_i, the classifier loss is defined by equation (14).

$$\mathcal{L}_{\text{cls}} = -\frac{1}{N} \sum_{i=1}^{N} \left[c_i \log C_\phi(\mathbf{S}_t) + (1 - c_i) \log \left(1 - C_\phi(\mathbf{S}_t) \right) \right], \tag{14}$$

where N is the mini-batch size and c_i is the binary label of sample i.

The classifier parameters are updated using equation (15).

$$\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{\text{cls}}, \tag{15}$$

where η is the learning rate. The classifier loss in equation (14) and the update rule in equation (15) define the classifier training procedure. After convergence, the trained denoising network parameters θ and classifier parameters ϕ are used for classifier-guided sampling.

Step 2: Classifier-guided predictor-corrector sampling. Synthetic extreme-weather load samples are generated by starting from random noise and progressively applying reverse diffusion under classifier guidance. The reverse process is initialized at diffusion time $t = T$ with a batch of random noise signals $\hat{\mathbf{S}}_T$. At each reverse-time iteration, the predictor update is computed using equation (16).

$$\hat{\mathbf{S}}_{t-\Delta t} \leftarrow \hat{\mathbf{S}}_t - f(\hat{\mathbf{S}}_t, t) + \frac{1}{2} g^2(t) \left[D_\theta(\hat{\mathbf{S}}_t, t) + \nabla_{\hat{\mathbf{S}}_t} C_\phi(\hat{\mathbf{S}}_t) \right], \tag{16}$$

where $f(\hat{\mathbf{S}}_t, t)$ and $g(t)$ denote the drift and diffusion terms of the reverse stochastic differential equation, respectively. The classifier gradient $\nabla_{\hat{\mathbf{S}}_t} C_\phi(\hat{\mathbf{S}}_t)$ in equation (16) guides the reverse trajectory toward the extreme-weather data manifold.

After the predictor update, a corrector step is used to refine the intermediate sample. The corrector update is computed using equation (17).

$$\hat{\mathbf{S}}_{t-\Delta t} \leftarrow \hat{\mathbf{S}}_{t-\Delta t} + \varepsilon \left[D_\theta(\hat{\mathbf{S}}_{t-\Delta t}, t - \Delta t) + \nabla_{\hat{\mathbf{S}}_{t-\Delta t}} C_\phi(\hat{\mathbf{S}}_{t-\Delta t}) \right] + \sqrt{2\varepsilon}\mathbf{z}, \tag{17}$$

where ε is the corrector step size and $\mathbf{z}$ denotes Gaussian noise. The predictor update in equation (16) and the corrector update in equation (17) are repeated until the reverse diffusion time reaches $t = 0$.

The final generated samples are denoted by equation (18).

$$\hat{\mathbf{S}}_0 = (\mathbf{X}_{\text{syn}}, \mathbf{Y}_{\text{syn}}), \tag{18}$$

where $\mathbf{X}_{\text{syn}}$ and $\mathbf{Y}_{\text{syn}}$ represent the synthetic input features and the corresponding synthetic load targets, respectively. The output definition in equation (18) is used as the synthetic extreme-weather dataset for subsequent representation disentanglement and forecasting.

Algorithm details for fidelity-aware representation disentanglement and forecasting

Step 1: Representation disentanglement network training. Let $(\mathbf{X}_{\text{real}}, \mathbf{Y}_{\text{real}})$ denote the real training samples and $(\mathbf{X}_{\text{syn}}, \mathbf{Y}_{\text{syn}})$ denote the synthetic samples generated by the classifier-guided diffusion model. The representation disentanglement network contains a shared encoder, type-specific encoders, restorers and forecasters. The shared encoder is parameterized by θ_{shared}, the type-specific encoders are parameterized by θ_{type}, the restorers are parameterized by ϑ_{type}, and the forecasters are parameterized by ψ_{type}, where type $\in \{\text{real}, \text{syn}\}$.

For each training epoch, the real and synthetic samples are first mapped to shared representations using equation (19).

$$\mathbf{R}_{\text{type}}^{\text{shared}} = E_{\text{shared}}\left(\mathbf{X}_{\text{type}}; \theta_{\text{shared}}\right), \qquad \text{type} \in \{\text{real}, \text{syn}\}, \tag{19}$$

where $E_{\text{shared}}(\cdot)$ denotes the shared encoder. The real and synthetic samples are also mapped to type-specific representations using equation (20).

$$\mathbf{R}_{\text{type}}^{\text{specific}} = E_{\text{type}}\left(\mathbf{X}_{\text{type}}; \theta_{\text{type}}\right), \qquad \text{type} \in \{\text{real}, \text{syn}\}, \tag{20}$$

where $E_{\text{type}}(\cdot)$ denotes the type-specific encoder for either real or synthetic samples. The shared representations in equation (19) are used to capture information common to real and synthetic data, whereas the type-specific representations in equation (20) are used to capture domain-dependent variations.

The input samples are reconstructed from the shared and type-specific representations using equation (21).

$$\widetilde{\mathbf{X}}_{\text{type}} = G_{\text{type}}\left(\left[\mathbf{R}_{\text{type}}^{\text{shared}}, \mathbf{R}_{\text{type}}^{\text{specific}}\right]; \vartheta_{\text{type}}\right), \qquad \text{type} \in \{\text{real}, \text{syn}\}, \tag{21}$$

where $G_{\text{type}}(\cdot)$ denotes the restorer and $[\cdot, \cdot]$ denotes feature concatenation. The corresponding load targets are predicted using equation (22).

$$\widetilde{\mathbf{Y}}_{\text{type}} = F_{\text{type}}\left(\left[\mathbf{R}_{\text{type}}^{\text{shared}}, \mathbf{R}_{\text{type}}^{\text{specific}}\right]; \psi_{\text{type}}\right), \qquad \text{type} \in \{\text{real}, \text{syn}\}, \tag{22}$$

where $F_{\text{type}}(\cdot)$ denotes the forecaster for real or synthetic samples.

The representation disentanglement penalty is computed using equation (23).

$$L_{\text{RD}} = \sum_{\text{type} \in \{\text{real}, \text{syn}\}} \left\| \left(\mathbf{R}_{\text{type}}^{\text{shared}}\right)^{\top} \mathbf{R}_{\text{type}}^{\text{specific}} \right\|_F^2 + \left\| \mu\left(\mathbf{R}_{\text{real}}^{\text{shared}}\right) - \mu\left(\mathbf{R}_{\text{syn}}^{\text{shared}}\right) \right\|_2^2, \tag{23}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\mu(\cdot)$ denotes the mini-batch mean of the corresponding representation. The first term in equation (23) penalizes dependence between shared and type-specific representations, and the second term encourages the shared representations of real and synthetic samples to remain aligned.

The restoring error is computed using equation (24).

$$L_{\text{RS}} = \sum_{\text{type} \in \{\text{real}, \text{syn}\}} \frac{1}{N_{\text{type}}} \left\| \widetilde{\mathbf{X}}_{\text{type}} - \mathbf{X}_{\text{type}} \right\|_2^2, \tag{24}$$

where N_{type} is the number of samples of the corresponding type in the mini-batch. The forecasting error is computed using equation (25).

$$L_{\text{FD}} = \sum_{\text{type} \in \{\text{real}, \text{syn}\}} \frac{1}{N_{\text{type}}} \left\| \widetilde{\mathbf{Y}}_{\text{type}} - \mathbf{Y}_{\text{type}} \right\|_2^2. \tag{25}$$

The overall training objective is defined by equation (26).

$$\min_{\theta_{\text{shared}}, \theta_{\text{type}}, \vartheta_{\text{type}}, \psi_{\text{type}}} \lambda_1 L_{\text{RD}} + \lambda_2 L_{\text{RS}} + \lambda_3 L_{\text{FD}}, \tag{26}$$

where λ_1, λ_2 and λ_3 are non-negative coefficients that control the relative contributions of representation disentanglement, restoration and forecasting. The network parameters are updated by minimizing the objective in equation (26) over training epochs. After convergence, the trained parameters θ_{shared}, θ_{type}, ϑ_{type} and ψ_{type} are retained for forecasting.

Step 2: Forecasting for unseen samples. Let $\mathbf{X}_{\text{unseen}}$ denote an unseen input sample. The unseen sample is mapped to the shared representation using equation (27).

$$\mathbf{R}_{\text{unseen}}^{\text{shared}} = E_{\text{shared}}\left(\mathbf{X}_{\text{unseen}}; \theta_{\text{shared}}\right). \tag{27}$$

The unseen sample is mapped to the real-specific representation using equation (28).

$$\mathbf{R}_{\text{unseen}}^{\text{specific}} = E_{\text{real}}\left(\mathbf{X}_{\text{unseen}}; \theta_{\text{real}}\right). \tag{28}$$

The real-specific encoder is used in equation (28) because the final forecasting task is evaluated on real unseen load samples. The load forecasting result is computed using equation (29).

$$\mathbf{Y}_{\text{unseen}} = F_{\text{real}}\left(\left[\mathbf{R}_{\text{unseen}}^{\text{shared}}, \mathbf{R}_{\text{unseen}}^{\text{specific}}\right]; \psi_{\text{real}}\right), \tag{29}$$

where $F_{\text{real}}(\cdot)$ denotes the forecaster associated with the real-data branch. The output $\mathbf{Y}_{\text{unseen}}$ in equation (29) is used as the final load forecasting result.

Supplementary Note 4 — Baseline settings

Hyperparameter search strategy

To ensure a fair and transparent comparison across forecasting models, we adopted a two-level hyperparameter selection strategy. First, a set of shared training hyperparameters was fixed across all baselines and the proposed method, including the data split, input setting, optimization protocol, and training budget. This design avoids confounding effects caused by inconsistent training procedures and ensures that performance differences are not driven by model-specific training setups.

Second, model-specific hyperparameters were selected on the validation set. For relatively simple forecasting baselines, such as MLP, LSTM, and CNN, we performed a search over a predefined candidate space and selected the configuration with the lowest validation loss. For more structurally complex baselines, such as Informer and Autoformer, the number of possible hyperparameter combinations is substantially larger. To avoid an excessively large and uneven tuning budget across methods, we adopted the default or recommended hyperparameter settings provided in the official implementations of these models, while keeping the same shared training protocol as for all other baselines.

Shared hyperparameters

Unless otherwise specified, all forecasting baselines and the proposed method were trained under the same experimental protocol. The shared settings were chosen to ensure consistency across models and to prevent differences in performance from arising from mismatched training budgets or input configurations. For all models, 10% of the training set was randomly sampled as the validation set. The maximum number of training epochs was set to 2000, with an early stopping patience of 150 epochs. The lookback length was 168 hours, the batch size was 128, and the initial learning rate was 1×10^{-3}. The Adam optimizer was used for training.

In addition, all models were evaluated using the same forecasting inputs and exogenous variables under the same train/validation/test split.

Specific hyperparameters

The model-specific hyperparameters and their candidate values or adopted configurations are summarized in Supplementary Table 3.

Supplementary Table 3: Model-specific hyperparameters for forecasting baselines. For simple baselines, the best configuration was selected based on validation loss from the predefined candidate space. For complex Transformer-based baselines, the adopted settings follow the default or recommended configurations from the official implementations.

Baseline	**Tuning strategy**	**Hyperparameter**	**Candidate values**
MLP	Search	Hidden size	{64, 96, 128}
LSTM	Search	LSTM hidden size	{64, 128}
		FC layer hidden size	{64, 128}
CNN	Search	Kernel size (1st layer)	{32, 64}
		Kernel size (2nd layer)	{32, 64}
		FC layer hidden size	{64, 128}
N-BEATS	Search	Block number	2
		Stack number	{2, 3}
		Number of FC layers	4
		FC layer hidden size	{64, 128}
ImpactNet	Search	RCU kernel size	{3, 5, 7}
		RCU channels	{64, 96}
		FC layer hidden size	{64, 96, 128}
Informer	Official	d_model	128
		n_heads	8
		e_layers	3
		d_layers	2
		d_ff	256
		attn	prob
		embed	fixed
		activation	gelu
Autoformer	Official	factor	5
		d_model	128
		n_heads	8
		e_layers	3
		d_layers	2
		d_ff	256
		attn	prob
		embed	fixed
		activation	gelu
		output_attention	False
		distil	True
		mix	True
		moving_avg	25

Supplementary Note 5 — Quantitative evaluation of synthetic sample fidelity

Descriptor groups

Two complementary groups of descriptors are considered for evaluating synthetic events. Distributional descriptors summarize the marginal properties of each event, covering central tendency, variability, extreme quantiles, and the dependence between load and temperature. The features used to compute the statistical distances are listed in Table 4. Event-level descriptors characterize the morphology of an extreme event, covering the timing and magnitude of load peaks, ramping behavior, persistence above high-load thresholds, and temporal alignment between load and temperature extremes. The features used to compute the event-level distances are listed in Table 5.

Together, these two descriptor groups provide a compact but interpretable representation of whether synthetic samples are faithful to real extreme events not only in distributional terms, but also in temporal structure.

Supplementary Table 4: Distributional descriptors used for fidelity evaluation.

Descriptor	Definition
Load mean	Mean load over the event window.
Load standard deviation	Standard deviation of load over the event window.
Load minimum	Minimum load over the event window.
Load maximum	Maximum load over the event window.
Load 90th percentile	90th percentile of load over the event window.
Load 95th percentile	95th percentile of load over the event window.
Load 99th percentile	99th percentile of load over the event window.
Cumulative load	Sum of load over the event window.
Temperature mean	Mean temperature over the event window.
Temperature standard deviation	Standard deviation of temperature over the event window.
Temperature minimum	Minimum temperature over the event window.
Temperature maximum	Maximum temperature over the event window.
Temperature 1st percentile	1st percentile of temperature over the event window.
Temperature 5th percentile	5th percentile of temperature over the event window.
Temperature 95th percentile	95th percentile of temperature over the event window.
Temperature 99th percentile	99th percentile of temperature over the event window.
Load–temperature correlation	Pearson correlation between load and temperature over the event window.

Comparison synthesis methods

To contextualize the quantitative comparisons, several baseline generators with different levels of structural prior and distributional control are considered.

Supplementary Table 5: Event-level descriptors used for fidelity evaluation.

Descriptor	Definition
Load peak	Maximum load within the event window.
Load peak hour	Time index of the maximum load within the event window.
Mean absolute load ramp	Mean absolute difference between consecutive load values within the event window.
Maximum absolute load ramp	Maximum absolute difference between consecutive load values within the event window.
Duration above load q95	Number of time steps with load above the event-specific 95th percentile threshold.
Excess above load q95	Cumulative exceedance above the event-specific 95th percentile threshold.
Temperature extreme	Minimum temperature for cold waves or maximum temperature for heat waves within the event window.
Temperature extreme hour	Time index of the temperature extreme within the event window.
Peak-hour gap	Absolute difference between the load peak hour and the temperature extreme hour.

The unconditional baseline removes classifier guidance during sampling. The diffusion model is trained on the original imbalanced dataset and generates samples without explicit steering toward extreme-event regions. It therefore reflects the generative behavior of a standard unconditional denoising diffusion model.

The direct baseline trains the diffusion model only on the subset of historical extreme samples. Unlike the proposed approach, it does not use the full data manifold together with classifier-based guidance, but instead attempts to learn the extreme-event distribution from a much smaller training set.

The synthetic minority over-sampling technique baseline synthesizes new extreme samples through linear interpolation between neighboring real extreme samples in the original sample space. It enriches the minority class by constructing convex combinations of observed extreme events, but does not explicitly model temporal generation dynamics.

The Dirichlet baseline extends pairwise interpolation by combining multiple neighboring extreme samples using random Dirichlet weights. Compared with the synthetic minority over-sampling technique baseline, it provides more diverse convex mixtures while still remaining a local interpolation-based augmentation strategy.

The Gaussian baseline generates synthetic samples by adding Gaussian noise to real extreme samples, followed by clipping or normalization when needed to preserve the valid data range. It provides a simple perturbation-based augmentation baseline without explicit generative modeling.

Distributional discrepancy measures

For each descriptor group, the empirical distributions of synthetic samples are compared with two reference sets, namely all real samples and real extreme samples. The

first metric is the Wasserstein distance (WD). For a given descriptor, the first-order Wasserstein distance between the empirical distributions of the reference samples and the synthetic samples is defined in equation (30):

$$W_1(P_r, P_s) = \inf_{\gamma \in \Pi(P_r, P_s)} \int |x - y| \, d\gamma(x, y), \tag{30}$$

where P_r and P_s denote the empirical distributions of the descriptor for the reference and synthetic samples, respectively, and $\Pi(P_r, P_s)$ is the set of all couplings with marginals P_r and P_s. In practice, the Wasserstein distance in equation (30) is computed separately for each descriptor and then averaged within each descriptor group.

The second metric is maximum mean discrepancy (MMD), which is used to compare the joint distributions of descriptor vectors with a radial basis function kernel. Given descriptor vectors $\{x_i\}_{i=1}^n$ from the synthetic samples and $\{y_j\}_{j=1}^m$ from the reference samples, the squared maximum mean discrepancy is defined in equation (31):

$$\mathrm{MMD}^2(X, Y) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{i'=1}^{n} k(x_i, x_{i'}) + \frac{1}{m^2} \sum_{j=1}^{m} \sum_{j'=1}^{m} k(y_j, y_{j'}) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(x_i, y_j), \tag{31}$$

where $k(\cdot, \cdot)$ is the radial basis function kernel. Lower maximum mean discrepancy values in equation (31) indicate greater similarity between the synthetic and reference descriptor distributions.

Diversity computation

To quantify the variability of the generated extreme samples, we measure diversity in the same descriptor space used for the distributional discrepancy analysis. Let $\{z_i\}_{i=1}^{N}$ denote the descriptor vectors of the synthetic extreme samples, where each $z_i \in \mathbb{R}^d$ concatenates the selected distributional and event-level descriptors. We define diversity as the average pairwise Euclidean distance among synthetic samples, as shown in (32):

$$\text{Diversity}(Z) = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \|z_i - z_j\|_2. \tag{32}$$

A larger diversity value in (32) indicates that the generated samples span a broader range of extreme-event characteristics, whereas a smaller value suggests that the samples are more concentrated and potentially less varied. To ensure comparability across descriptors with different physical scales, all descriptor dimensions are standardized using the statistics of the real training data before distance computation.

Supplementary Note 6 — Load forecasting valuation method

This work assesses the economic value of improved load forecasting for electricity balance from two complementary perspectives: the potential socioeconomic exposure associated with under-forecasting, quantified using the value of lost load (VOLL), and the operational balancing cost incurred in electricity markets.

This supplementary note provides the mathematical formulations for both valuation metrics. Let y_t denote the actual load at timestep t, and let $\hat{y}_t^{\text{base}}$ and $\hat{y}_t^{\text{our}}$ denote the day-ahead forecasts generated by a baseline method and the proposed method, respectively.

We first quantify socioeconomic risk using VOLL. The potential unserved-load exposure at timestep t is defined as the positive deviation between actual load and forecasted load, as shown in equation (33). This quantity captures the extent of under-forecasting before accounting for available operating reserves, demand response, or other remedial actions. It should therefore be interpreted as an upper-bound exposure measure rather than as the amount of load curtailment that would necessarily occur in system operation.

$$\begin{aligned} \Delta r_t^{\text{base}} &= \max(y_t - \hat{y}_t^{\text{base}}, 0), \\ \Delta r_t^{\text{our}} &= \max(y_t - \hat{y}_t^{\text{our}}, 0). \end{aligned} \tag{33}$$

where Δr_t^{base} and Δr_t^{our} denote the potential unserved-load exposure at timestep t for the baseline method and the proposed method, respectively.

The VOLL-based reduction in potential socioeconomic exposure achieved by the proposed method relative to a baseline is then calculated using equation (34):

$$\Delta R^{\text{soci}} = \text{VOLL} \sum_{t=1}^{T} \left(\Delta r_t^{\text{base}} - \Delta r_t^{\text{our}} \right), \tag{34}$$

where VOLL is a standard regulatory metric representing the economic cost per unit of unserved energy, and T is the number of timesteps considered. Since the calculation is performed before reserve adjustment, ΔR^{soci} measures the reduction in upper-bound exposure associated with fewer and smaller under-forecasting errors, rather than a directly observed outage cost. The country-level values and their ratios to total electricity costs in the Europe dataset are listed in Supplementary Table 6.

We then evaluate the financial impact of forecasting accuracy on operational balancing costs under a standard two-settlement electricity market. Unlike the VOLL-based metric, this calculation focuses on the market cost of correcting day-ahead forecast deviations through real-time procurement. Let p_t^{da} and p_t^{rt} denote the day-ahead and real-time electricity prices, respectively. The total energy procurement costs over the period T for the baseline and the proposed method are modeled using equation (35):

$$
\begin{aligned}
C^{\text{oper,base}} &= \sum_{t=1}^{T} \left[\hat{y}_t^{\text{base}} p_t^{\text{da}} + (y_t - \hat{y}_t^{\text{base}}) p_t^{\text{rt}}\right] \\
&= \sum_{t=1}^{T} \left[-\hat{y}_t^{\text{base}} \Delta p_t + y_t p_t^{\text{rt}}\right], \\
C^{\text{oper,our}} &= \sum_{t=1}^{T} \left[\hat{y}_t^{\text{our}} p_t^{\text{da}} + (y_t - \hat{y}_t^{\text{our}}) p_t^{\text{rt}}\right] \\
&= \sum_{t=1}^{T} \left[-\hat{y}_t^{\text{our}} \Delta p_t + y_t p_t^{\text{rt}}\right],
\end{aligned}
\tag{35}
$$

where $\Delta p_t = p_t^{\text{rt}} - p_t^{\text{da}}$ is the spread between real-time and day-ahead prices.

The operational balancing cost reduction achieved by the proposed method relative to a baseline is calculated using equation (36):

$$
\begin{aligned}
\Delta C^{\text{oper}} &= C^{\text{oper,base}} - C^{\text{oper,our}} \\
&= \sum_{t=1}^{T} \left(\hat{y}_t^{\text{our}} - \hat{y}_t^{\text{base}}\right) \left(p_t^{\text{rt}} - p_t^{\text{da}}\right) \\
&= \sum_{t=1}^{T} \Delta p_t \Delta y_t,
\end{aligned}
\tag{36}
$$

where $\Delta y_t = \hat{y}_t^{\text{our}} - \hat{y}_t^{\text{base}}$ denotes the difference in day-ahead scheduled volume between the proposed method and the baseline.

Supplementary Table 6: Mitigated socioeconomic risks and their ratios to total electricity costs in the Europe dataset compared to the baseline with the highest accuracy.

Country	VOLL (EUR megawatt-hour^{-1} (MWh^{-1}))	Avoided Loss (Million EUR)	Avoided Load Loss MWh)	Total Cost (Million EUR)	Ratio (%)
Netherlands	15737.0	11875.94	213.05	186.89	87.72
Poland	12125.0	53115.36	372.59	644.02	172.85
Croatia	5788.0	4301.79	40.21	24.9	61.93
Germany	12610.0	43650.68	983.12	550.44	55.99
Belgium	13455.0	11083.08	167.93	149.12	88.8
Latvia	6373.0	310.01	22.06	1.98	8.96
Italy	16210.0	4754.74	742.7	77.07	10.38
Hungary	7444.0	12746.56	91.75	94.89	103.42
Romania	8880.0	18770.21	124.4	166.68	133.99
France	8193.0	218675.82	1139.39	1791.61	157.24
Slovenia	7495.0	2955.3	30.69	22.15	72.17
Ireland	9614.0	6546.68	68.93	62.94	91.31
Sweden	4957.0	36511.95	322.15	180.99	56.18
Lithuania	7444.0	1572.57	36.24	11.71	32.3
Denmark	11254.0	15697.46	67.04	176.66	263.52

Supplementary Note 7 — Proof for conditional reverse process

The reverse process recovers data from noise, with the objective of steering the generation process by the condition c so that the generated samples resemble heatwaves or cold waves. Directly modeling the conditional reverse transition is difficult under data scarcity. We therefore express the conditional reverse transition through a Bayesian decomposition. For a generic reverse step from τ to $\tau - \Delta\tau$, the conditional transition is written as equation (37):

$$p_\theta\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} \mid \hat{\mathbf{S}}_\tau, c\right) = p_\theta\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} \mid \hat{\mathbf{S}}_\tau\right) \exp\left[\log p_\phi\left(c \mid \hat{\mathbf{S}}_{\tau-\Delta\tau}\right) - \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right)\right], \tag{37}$$

where $p_\theta\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} \mid \hat{\mathbf{S}}_\tau\right)$ denotes the unconditional reverse transition and $p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right)$ denotes the probability assigned by the classifier to condition c.

The unconditional reverse transition is modeled as a Gaussian distribution derived from the reverse stochastic diffusion equation, as shown in equation (38):

$$p_\theta\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} \mid \hat{\mathbf{S}}_\tau\right) = \mathcal{N}\left(\hat{\mathbf{S}}_{\tau-\Delta\tau}; \boldsymbol{\mu}_\theta\left(\hat{\mathbf{S}}_\tau, \tau\right), \boldsymbol{\Sigma}_\tau\right), \tag{38}$$

where $\boldsymbol{\mu}_\theta\left(\hat{\mathbf{S}}_\tau, \tau\right)$ is the denoising mean predicted by the reverse model and $\boldsymbol{\Sigma}_\tau$ is the covariance matrix of the reverse transition. Equivalently, under the reverse stochastic differential equation, the mean can be written as equation (39):

$$\boldsymbol{\mu}_\theta\left(\hat{\mathbf{S}}_\tau, \tau\right) = \hat{\mathbf{S}}_\tau - \left[\mathbf{f}\left(\hat{\mathbf{S}}_\tau, \tau\right) - g_\tau^2 \nabla_{\hat{\mathbf{S}}_\tau} \log p_\tau\left(\hat{\mathbf{S}}_\tau\right)\right] \Delta\tau, \tag{39}$$

where $\mathbf{f}\left(\hat{\mathbf{S}}_\tau, \tau\right)$ is the drift term, g_τ is the diffusion coefficient, and $\nabla_{\hat{\mathbf{S}}_\tau} \log p_\tau\left(\hat{\mathbf{S}}_\tau\right)$ is the score function estimated by the score network.

To obtain a practical conditional sampler, we approximate the classifier term in equation (37) using a first-order Taylor expansion around $\hat{\mathbf{S}}_\tau$, as shown in equation (40):

$$\log p_\phi\left(c \mid \hat{\mathbf{S}}_{\tau-\Delta\tau}\right) = \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right) + \left(\hat{\mathbf{S}}_{\tau-\Delta\tau} - \hat{\mathbf{S}}_\tau\right)^{\mathrm{T}} \nabla_{\hat{\mathbf{S}}_\tau} \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right) + \mathcal{O}(\Delta\tau), \tag{40}$$

where $\nabla_{\hat{\mathbf{S}}_\tau} \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right)$ is the classifier gradient that guides the reverse sampling process toward condition c.

Substituting equation (40) into equation (37) gives the approximation in equation (41):

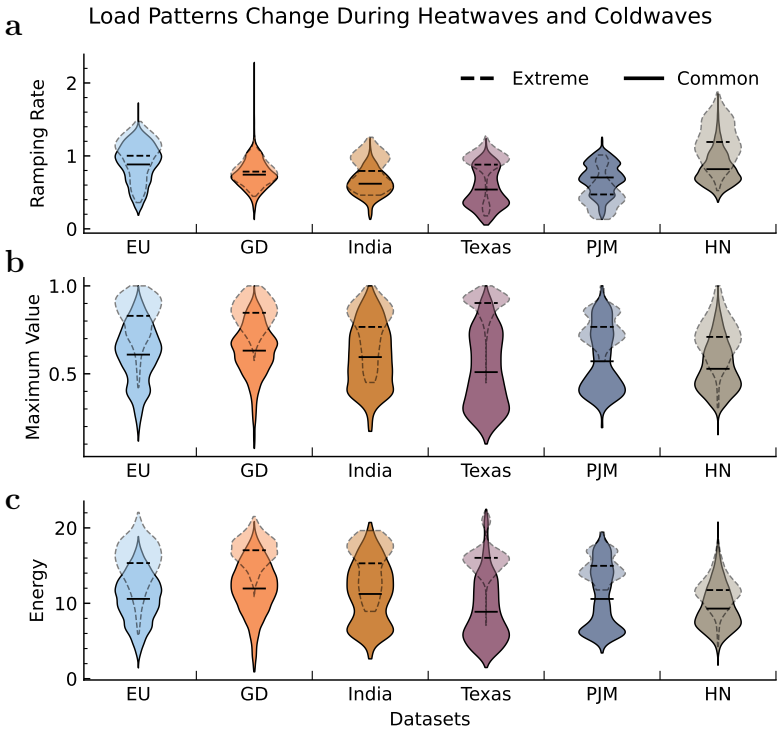
$$
\begin{aligned}
p_\theta\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} \mid \hat{\mathbf{S}}_\tau, c\right) \propto \exp\Bigg[& -\frac{1}{2}\left(\hat{\mathbf{S}}_{\tau-\Delta\tau}-\boldsymbol{\mu}_\theta\right)^{\mathrm{T}} \boldsymbol{\Sigma}_\tau^{-1}\left(\hat{\mathbf{S}}_{\tau-\Delta\tau}-\boldsymbol{\mu}_\theta\right) \\
& +\hat{\mathbf{S}}_{\tau-\Delta\tau}^{\mathrm{T}} \nabla_{\hat{\mathbf{S}}_\tau} \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right)\Bigg] \\
\propto \exp\Bigg[& -\frac{1}{2}\left(\hat{\mathbf{S}}_{\tau-\Delta\tau}-\boldsymbol{\mu}_\theta-\boldsymbol{\Sigma}_\tau \nabla_{\hat{\mathbf{S}}_\tau} \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right)\right)^{\mathrm{T}} \boldsymbol{\Sigma}_\tau^{-1} \\
& \left(\hat{\mathbf{S}}_{\tau-\Delta\tau}-\boldsymbol{\mu}_\theta-\boldsymbol{\Sigma}_\tau \nabla_{\hat{\mathbf{S}}_\tau} \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right)\right)\Bigg] .
\end{aligned}
\tag{41}
$$

where terms independent of $\hat{\mathbf{S}}_{\tau-\Delta\tau}$ are absorbed into the proportionality constant.

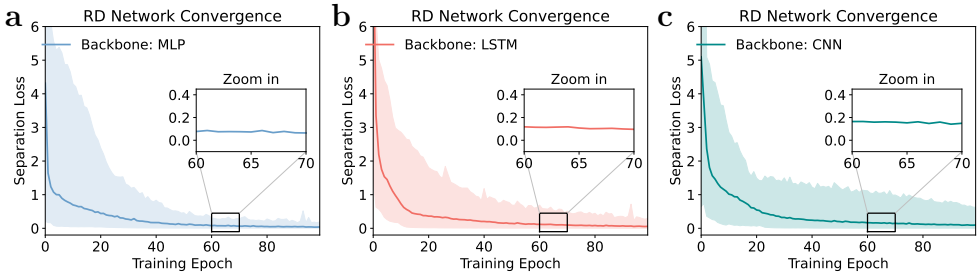
Equation (41) shows that the conditional reverse transition remains Gaussian, but its mean is shifted by the classifier gradient. The resulting classifier-guided reverse transition is therefore written as equation (42):

$$
p_\theta\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} \mid \hat{\mathbf{S}}_\tau, c\right)=\mathcal{N}\left(\hat{\mathbf{S}}_{\tau-\Delta\tau} ; \boldsymbol{\mu}_\theta+\boldsymbol{\Sigma}_\tau \nabla_{\hat{\mathbf{S}}_\tau} \log p_\phi\left(c \mid \hat{\mathbf{S}}_\tau\right), \boldsymbol{\Sigma}_\tau\right) . \tag{42}
$$

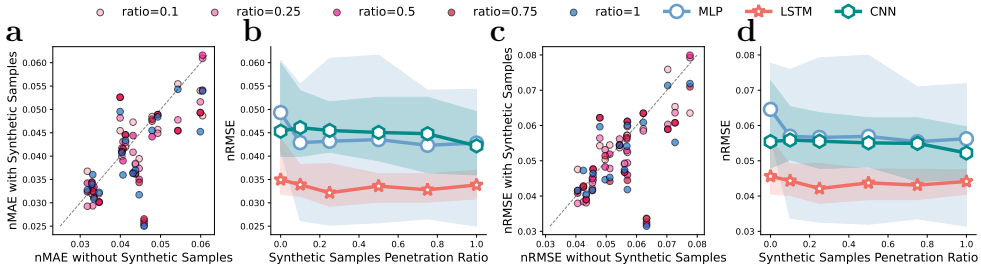
Supplementary Figures



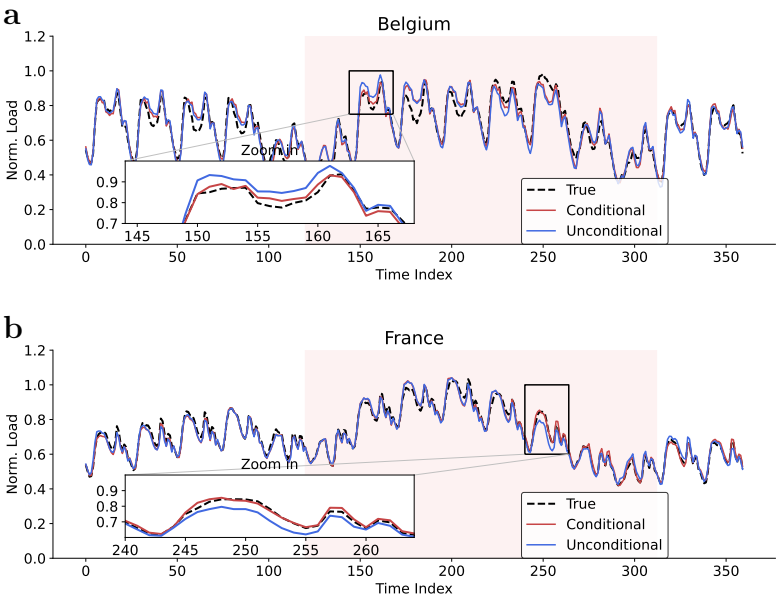
Supplementary Fig. 1: Load pattern changes under heat and cold waves. Statistical distributions of electricity-load patterns under normal conditions and extreme-temperature events. **a**, Distribution of total daily energy consumption. **b**, Distribution of extreme load values. **c**, Distribution of the total load variation rate. Violin plots summarize the sample distributions of each metric across normal and heat/cold-wave conditions.



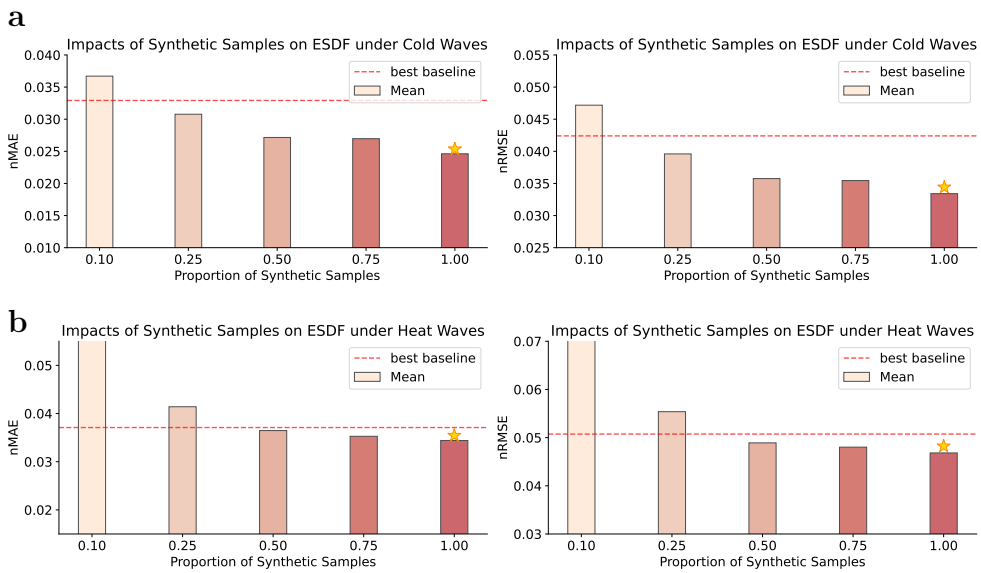
Supplementary Fig. 2: Convergence curves of the proposed representation disentanglement (RD) network with different backbones. The solid line represents the mean convergence curve, and the shaded area indicates the range. **a**, Convergence curve with the multilayer perceptron (MLP) backbone. **b**, Convergence curve with the long short-term memory (LSTM) backbone. **c**, Convergence curve with the convolutional neural network (CNN) backbone. Separation loss refers to the penalty incurred when the similarity and orthogonality constraints are violated during training. Regardless of the backbone used, the separation loss converges to a small value after 100 training epochs, indicating that the RD network can map real and synthetic samples to the desired disentangled representations.



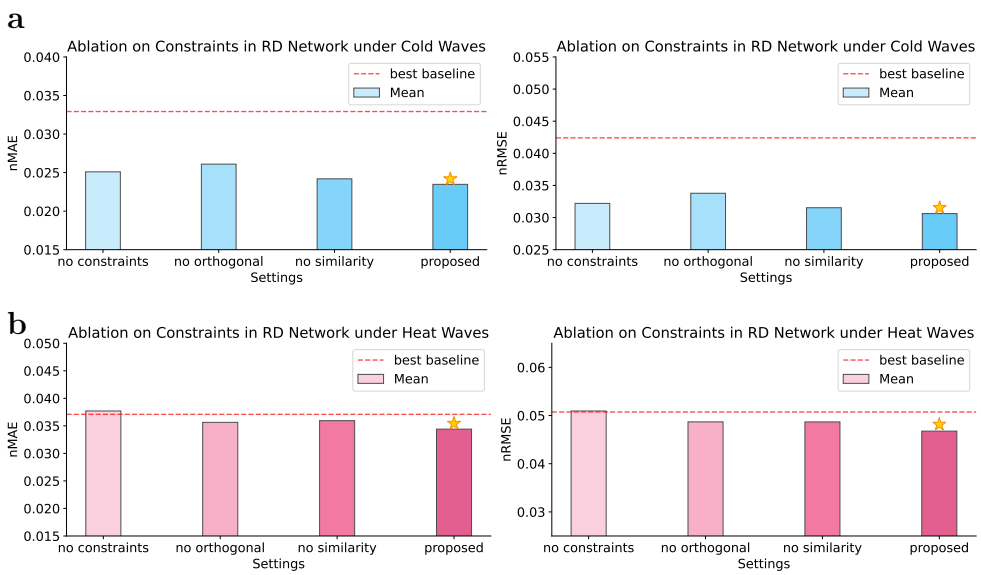
Supplementary Fig. 3: Impacts of direct training with different ratios of generated samples on forecasting accuracy. Performance comparison of forecasting models trained with different ratios of generated samples, evaluated using normalized mean absolute error (nMAE) and normalized root mean square error (nRMSE). **a**, Scatter plot of nMAE values obtained under different generated-sample ratios. **b**, Relationship between the generated-sample ratio and nMAE. **c**, Scatter plot of nRMSE values obtained under different generated-sample ratios. **d**, Relationship between the generated-sample ratio and nRMSE. Directly using generated samples generally improves forecasting accuracy during training, indicating that the generated samples contain useful information for the forecasting task. However, without the uninformative distillation module, the direct use of generated samples can still degrade accuracy in some experiments.



Supplementary Fig. 4: Ablation on classifier-guided deviation for Belgium and France. Forecasting curves comparing the model with and without classifier-guided deviation during extreme-temperature events. **a**, Forecasting curves for Belgium. **b**, Forecasting curves for France. The shaded areas mark extreme-temperature periods. Classifier-guided generation improves forecasting accuracy during these periods, whereas its impact is limited under normal conditions, indicating its specific contribution to enhancing forecasting robustness during critical periods.



Supplementary Fig. 5: Impacts of different proportions of synthetic samples on ESDF forecasting accuracy under cold waves and heat waves. a, Forecasting accuracy under cold-wave events with different proportions of synthetic samples. **b**, Forecasting accuracy under heat-wave events with different proportions of synthetic samples. The star indicates the best-performing setting. Unlike directly using generated samples for training, distillation through the representation disentanglement (RD) network enables a larger volume of generated data to yield stable improvements in forecasting accuracy. This suggests that estimation error is reduced without introducing additional fidelity gaps. Notably, when the proportion of generated samples exceeds 50% of the original dataset, the forecasting error drops below that of the best baseline model. These results demonstrate the effectiveness of the proposed ESDF framework.



Supplementary Fig. 6: Ablation of constraints in the representation disentanglement (RD) network under cold waves and heat waves. a, Ablation results for the RD-network constraints under cold-wave events. **b**, Ablation results for the RD-network constraints under heat-wave events. The star indicates the best-performing setting. Removing either the orthogonality constraint or the similarity constraint increases the forecasting error. Under cold-wave conditions, using either constraint alone is insufficient for effective representation distillation. The lowest forecasting error is achieved only when both constraints are applied simultaneously, supporting the effectiveness of the proposed method.

Supplementary References

[1] Oliveira, A., Lopes, A. & Soares, A. Excess heat factor climatology, trends, and exposure across European functional urban areas. *Weather Clim. Extrem.* **36**, 100455 (2022).

[2] Li, A., Toll, M. & Bentley, R. The risk of energy hardship increases with extreme heat and cold in Australia. *Commun. Earth Environ.* **5**, 595 (2024).