

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 TSDINO: TEACHER-STUDENT SELF-DISTILLATION FRAMEWORK FOR ROBUST PRE-TRAINING OF TIME- SERIES FOUNDATION MODELS

006
007 **Anonymous authors**
008 Paper under double-blind review

011 ABSTRACT

013 Building time-series foundation models (TSFM) poses challenges in terms of learn-
014 ing stability due to limited data availability and heterogeneous temporal dynamics
015 across various time-series datasets. We propose TSDINO, a teacher-student frame-
016 work for robust pre-training of TSFM based on the principle of self-distillation with
017 no labels. TSDINO offers a model-agnostic approach that combines two comple-
018 mentary objectives: (i) feature preservation under augmentations and (ii) masked
019 patch prediction. A meta-architecture comprising teacher-student networks and
020 projection heads enables adaptation to various models. We evaluate TSDINO on
021 classification and forecasting tasks using diverse publicly available benchmarking
022 datasets. TSDINO consistently achieves competitive zero-shot performance over
023 gradient-based pre-training.

025 1 INTRODUCTION

027 Time series data are ubiquitous and at the operational core of modern sensing and decision-making
028 systems over finance, energy, climate, healthcare, and the web. *Time-Series Foundation Models*
029 (TSFM) (Liang et al., 2024; Das et al., 2024) offer a compelling paradigm by enabling zero-shot or
030 few-shot generalization across domains, tasks, and diverse conditions in time-series analysis. This
031 capability significantly reduces the cost, lead time, and engineering effort required to deploy systems
032 and services that rely on time-series data.

033 Compared to language and vision, where foundation models are well established (Achiam et al.,
034 2023; Liu et al., 2024a; Oquab et al., 2024), time series present unique challenges. Variability in
035 domains and acquisition conditions, such as sampling rates, noise levels, and missing data patterns,
036 introduces highly heterogeneous temporal dynamics. At the same time, training data is often scarce,
037 noisy, or impractical to obtain at scale. These factors make it challenging to train TSFM that produce
038 representations that transfer stably and robustly across tasks, regardless of domain or condition
039 diversity.

040 While the development of TSFM has accelerated, most research has concentrated on architectural
041 innovations rather than training methodologies. Chronos (Ansari et al., 2024) and TimesFM (Das
042 et al., 2024) adopt a Transformer-based architecture, Moirai (Liu et al., 2024b) introduces any-variate
043 attention to take attention over arbitrary numbers of variables, and TinyTimeMixer (TTM) (Ekam-
044 baram et al., 2024) and TSPulse (Ekambaram et al., 2025) explore lightweight MLP-Mixer-style
045 designs.

046 Despite these architectural advances, the refinement of training strategies for TSFM lags behind
047 analogous progress in language and vision, and, basically, using ordinary gradient-based learning is
048 dominant. Further exploring training paradigms is required that go beyond architecture, focusing on
049 robust self-supervised objectives and stability under heterogeneous dynamics.

050 Teacher-student self-distillation framework, such as DINO (Caron et al., 2021), has demonstrated
051 remarkable success in self-supervised pre-training for vision models, where the student network
052 learns to align its predictions with those of a momentum-updated teacher under multiple augmented
053 views of the same image. Unlike contrastive learning methods, the framework does not require large
negative sets; thus, it scales efficiently and produces semantically meaningful clusters, enabling

strong and stable transfer to diverse downstream tasks. Masked image modeling approaches (Zhou et al., 2022b) were also introduced to the framework as patch-level prediction objectives to capture local structures, improving robustness. Oquab et al. (2024) highlights the importance of multi-scale objectives.

In the time-series domain, such teacher-student framework has recently gained attention. Pieper et al. (2023) adapted data2vec-style distillation for time series, while Gao et al. (2024a) explored distillation-enhanced forecasting with momentum contrastive learning. However, these methods target task-specific models requiring supervised learning or fine-tuning and have not been extended to pre-training of TSFMs enabling zero-shot prediction. Moreover, existing approaches often emphasize global invariances but underrepresent local temporal patterns addressed in vision (Zhou et al., 2022b; Oquab et al., 2024).

To address these challenges, we propose **TSDINO**, a *Time-Series-tailored teacher-student framework for robust self-DIstillation with NO labels*. The student is trained to match its outputs with the teacher’s outputs as self-distillation, and the teacher is an exponential moving average (EMA) of the past student, providing a stable target for the student. The matching is conducted with two complementary objectives: (i) *sequence-level feature preservation* under augmentations for global invariance, and (ii) *masked patch prediction* for local structure modeling. Within this framework, while the student processes augmented and masked input compared to the teacher’s input, we propose augmentation and masking to be specialized for time series. The proposed meta-architecture of TSDINO enables adaptation to various models (e.g., TSPulse (Ekambaram et al., 2025) and TTM (Ekambaram et al., 2024)) and tasks (e.g., classification and forecasting). TSDINO provides a robust, stable, scalable, and model-agnostic pre-training framework for TSFMs.

Contributions.

- We present **TSDINO**, a novel model-agnostic teacher-student self-distillation framework for pre-training of TSFMS. TSDINO unifies sequence-level feature preservation with patch-level masked prediction, enabling robust representations for diverse downstream tasks such as classification and forecasting.
- We propose augmentation and masking strategies tailored for time series, departing from standard vision practices, to stabilize learning from time series.
- We demonstrate TSDINO’s effectiveness across multiple tasks, classification, and forecasting. Our results indicate consistent zero-shot improvements over gradient-based pre-training, and ablations confirm the critical role of the unified objective in considering both global and local invariance.

2 RELATED WORK

Self-supervised learning for time series. Self-supervised learning (SSL) for time series spans contrastive, generative (e.g., masked modeling), and adversarial paradigms, each with distinct assumptions and augmentation needs (Zhang et al., 2024). Recent works explored self-distillation for time series: Pieper et al. (2023) adopted data2vec-style objectives, Gao et al. (2024a) combined distillation with contrastive learning, and similarity distillation (Hajimoradlou et al., 2022) leveraged pairwise similarities. However, these approaches lack zero-shot capabilities. TSDINO addresses these gaps to enable zero-shot prediction and robust transfer across tasks by unifying sequence-level and patch-level self-distillation.

Self-Distillation in Vision. Teacher-student self-distillation has become a cornerstone of self-supervised learning in vision. DINO (Caron et al., 2021) introduced a non-contrastive approach that aligns student predictions with a momentum teacher across augmented views. iBOT (Zhou et al., 2022b) and DINOv2 (Oquab et al., 2024) extended this paradigm by incorporating masked image modeling, enabling multi-scale representation learning. These methods highlight the importance of teacher-student framework and multi-scale objectives for robust feature learning. TSDINO adapts these principles to time series, addressing unique challenges such as temporal continuity, frequency sensitivity, and heterogeneous dynamics.

108 **Time-series foundation models.** Liang et al. (2024) provides surveys systematizing architectures
 109 of TSFMs. Inspired by large language models, decoder-only models such as TimesFM (Das et al.,
 110 2024) and Chronos (Liang et al., 2024) have demonstrated the feasibility of zero-shot forecasting.
 111 Other architectures, including MLP-Mixer-style designs (TTM (Ekambaram et al., 2024) and
 112 TSPulse (Ekambaram et al., 2025)), have also been explored. However, most efforts focus on architec-
 113 tural innovations rather than training methodologies. TSDINO contributes a robust, model-agnostic
 114 self-distillation framework that complements these models, enhancing their ability to learn robust
 115 representations across tasks and domains.

116
 117 **Tasks in Time-Series Analysis and Modeling.** Time-series analysis encompasses tasks such as
 118 classification, clustering, retrieval, and forecasting, driving applications in finance, energy, healthcare,
 119 and beyond. All of the tasks benefit from robust, transferable representations. TSFMs provide *zero-*
 120 *shot* or few-shot generalization across domains and frequencies in the tasks (Ekambaram et al., 2024;
 121 2025). TSDINO introduces a teacher-student self-distillation pre-training strategy that generally
 122 handles multiple models and tasks.

124 3 SELF-SUPERVISED PRE-TRAINING FOR TIME SERIES FOUNDATION MODELS

125
 126 Our goal is to construct a self-supervised learning algorithm for *Time-Series Foundation Models*
 127 (TSFM) that, given unlabeled time series, learns (i) an encoder for general-purpose representation
 128 of time-series and (ii) a zero-shot forecaster that predicts future values in time-series without task-
 129 specific fine-tuning. The framework is model-agnostic and compatible with various architectures
 130 (TSPulse, TTM etc.) and downstream tasks (classification and forecasting).

131

132 3.1 SELF-SUPERVISED PRE-TRAINING FRAMEWORK FOR TIME SERIES

133
 134 Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ be N unlabeled time series, where each sample $\mathbf{x}_i \in \mathbb{R}^{P \times \tau}$ is a patchfied time
 135 series having $L = P \times \tau$ time steps, where each sequence is partitioned into P non-overlapping
 136 patches having τ consecutive steps, following Nie et al. (2023).

137 Our objective is to learn a prediction model $f_{\theta} : \mathbb{R}^{P \times \tau} \rightarrow \mathbb{R}^d$ that maps a time series to output for
 138 each task $\mathbf{z}_i = f_{\theta}(\mathbf{x}_i)$ described in the subsequent subsection. The model is parameterized by θ .

140 3.2 TASK-SPECIFIC OBJECTIVES

141
 142 **(1) General-purpose representation learning.** The model f_{θ} is an encoder that maps a time
 143 series \mathbf{x}_i to a feature vector \mathbf{z}_i such that $\text{similarity}(\mathbf{z}_i, \mathbf{z}_j) \approx \text{semantic similarity}(\mathbf{x}_i, \mathbf{x}_j)$, enabling
 144 downstream tasks such as clustering, retrieval, and classification without supervision.

145
 146 **(2) Zero-shot forecasting.** The model f_{θ} is a forecaster that given \mathbf{x}_i with time steps $1 : L$, predict
 147 future values $\mathbf{z}_i = \hat{\mathbf{x}}_i$ with time steps $L + 1 : L + 1 + d$ without task-specific fine-tuning.

149 4 TEACHER-STUDENT FRAMEWORK FOR PRE-TRAINING OF TIME SERIES 150 FOUNDATION MODELS

151
 152 We address self-supervised pre-training for TSFMs via a teacher-student framework, where we use
 153 two different networks, student f_{θ_S} and teacher f_{θ_T} , which share the same architecture but have
 154 separate parameters θ_S and θ_T to learn from unlabeled time series.

155
 156 Overall, the student parameter is trained via gradient descent to match the student’s outputs with the
 157 teacher’s outputs as self-distillation, where the teacher is an exponential moving average (EMA) of
 158 the past parameter sequence of the student parameters. The matching between the student and the
 159 teacher outputs is conducted at the sequence level (feature preservation under augmentations) and the
 160 patch level (masked patch prediction), which encourages the model to learn both global invariances
 161 and local temporal structures. The momentum teacher provides stable targets for the student, and
 centering and sharpening of the teacher’s outputs further stabilize training.

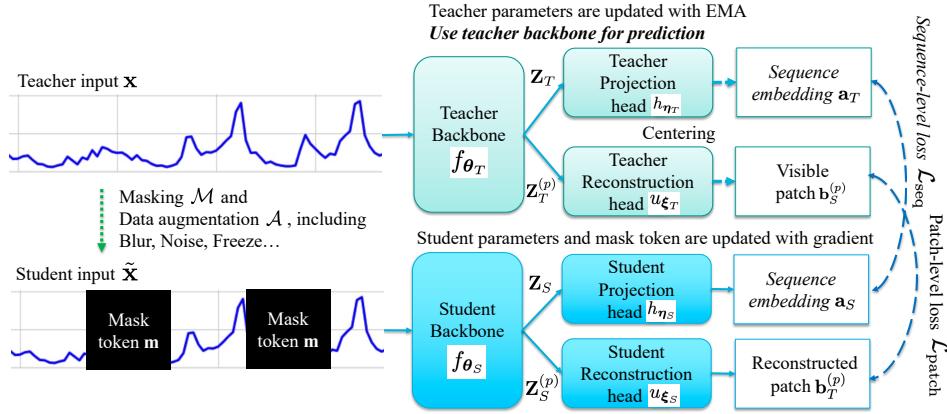


Figure 1: TSDINO training pipeline.

This framework encourages the student to learn robust, invariant representations that generalize across domains and tasks, and the teacher reaches a more stable and robust variant of the student; thus, we finally use the teacher network for prediction.

4.1 OVERVIEW OF TRAINING PIPELINE.

The training pipeline is illustrated in Figure 1.

First, for an input of the student, we generate a binary mask indicator $\mathbf{I}_i \in \{0, 1\}^P$ to each patch randomly, and an augmented and masked view is created:

$$\tilde{\mathbf{x}} \equiv \mathcal{M}(\mathcal{A}(\mathbf{x}), \mathbf{I}), \quad (1)$$

where \mathcal{A} is time-series augmentation and \mathcal{M} perform patch-wise value-fill with the mask indicator \mathbf{I} (further details are in §4.4). An input for the teacher is the original input sequence, in contrast to the vision literature, using the augmentation for both the student and the teacher (Caron et al., 2021; Zhou et al., 2022a) since the uncertainty of time-series data is much larger than that of image data, and strong augmentations may change the semantics of the time series.

The teacher and student respectively process the original input \mathbf{x} and the student input $\tilde{\mathbf{x}}$ to produce embeddings of all P patches, $\{\mathbf{Z}_T, \mathbf{Z}_S\} \in \mathbb{R}^{P \times d}$, as

$$\mathbf{Z}_T \equiv f_{\theta_T}(\mathbf{x}) \quad \text{and} \quad \mathbf{Z}_S \equiv f_{\theta_S}(\tilde{\mathbf{x}}). \quad (2)$$

Then we compute projected sequence embeddings $\{\mathbf{a}_T, \mathbf{a}_S\} \in \mathbb{R}^C$ with the teacher and student as

$$\mathbf{a}_T \equiv h_{\eta_T}(\mathbf{Z}_T) \quad \text{and} \quad \mathbf{a}_S \equiv h_{\eta_S}(\mathbf{Z}_S), \quad (3)$$

where h_{η_T} and h_{η_S} are projection heads for teacher and student respectively. We also compute reconstructed patch embeddings $\{\mathbf{b}_T^{(p)}, \mathbf{b}_S^{(p)}\} \in \mathbb{R}^C$ for masked patches with patch indices $p \in \{p : I_p = 1, I_p \in \mathbf{I}\}$ with the teacher and student as

$$\mathbf{b}_T^{(p)} \equiv u_{\xi_T}(\mathbf{Z}_T^{(p)}) \quad \text{and} \quad \mathbf{b}_S^{(p)} \equiv u_{\xi_S}(\mathbf{Z}_S^{(p)}), \quad \text{where } p \in I, \quad (4)$$

and u_{ξ_T} and u_{ξ_S} are reconstruction heads for teacher and student respectively.

From the outputs of the projection and reconstruction heads, we form the total loss $\mathcal{L}_{SSL}(\mathbf{a}_T, \mathbf{a}_S, \mathbf{b}_T, \mathbf{b}_S)$ and update the student parameters, $\boldsymbol{\Theta}_S \equiv \{\theta_S, \eta_S, \xi_S\}$, by gradient descent on \mathcal{L}_{SSL} as

$$\boldsymbol{\Theta}_S^{(s)} = \boldsymbol{\Theta}_S^{(s-1)} - \rho \nabla_{\boldsymbol{\Theta}_S} \mathcal{L}_{SSL}(\mathbf{a}_T, \mathbf{a}_S, \mathbf{b}_T, \mathbf{b}_S), \quad (5)$$

where ρ is the learning rate. We add the forecasting loss when we use the trained model for forecasting, and the details of \mathcal{L}_{SSL} are given in the subsequent subsection.

216 The teacher is not directly optimized by gradients but is maintained as an exponential moving average
 217 (EMA) of the student parameter sequence to provide stable targets. At step s the teacher parameters,
 218 $\Theta_T \equiv \{\theta_T, \eta_T, \xi_T\}$, are updated as
 219

$$\Theta_T^{(s)} = m(s)\Theta_T^{(s-1)} + (1 - m(s))\Theta_S^{(s-1)}, \quad (6)$$

220 where the momentum $m(s)$ is typically scheduled to increase toward 1 over training (cosine schedule).
 221 EMA serves as a low-cost mechanism to stabilize self-supervised training and improve representation
 222 quality without introducing additional optimization complexity. EMA ensures that the teacher evolves
 223 slowly, producing consistent targets for the student and reducing training instability, and the teacher
 224 acts as a temporal ensemble of past student models, improving generalization and reducing variance,
 225 similar to Polyak averaging.
 226

227 Empirically, prior work (Caron et al., 2021; Oquab et al., 2024) shows that EMA-based teachers
 228 consistently outperform their student counterparts on downstream tasks, validating its effectiveness.
 229 Therefore, at inference, we use the teacher network f_{θ_T} for prediction, as it is a more stable and
 230 robust variant of the student.
 231

232 4.2 DETAILED LOSS FUNCTIONS FOR TRAINING STUDENT.

233 We use cross-entropy (CE) loss for both sequence-level and patch-level objectives, following Caron
 234 et al. (2021); Zhou et al. (2022b); Oquab et al. (2024). The self-supervised objective for student
 235 parameters combines: $\mathcal{L}_{\text{SSL}} \equiv \mathcal{L}_{\text{seq}} + \alpha \mathcal{L}_{\text{patch}}$, where:
 236

$$\mathcal{L}_{\text{seq}} \equiv \frac{1}{N} \sum_{i=1}^N \text{CE}(\{\mathbf{a}_T\}_i, \{\mathbf{a}_S\}_i), \quad (7)$$

$$\mathcal{L}_{\text{patch}} \equiv \frac{1}{\sum_{i=1}^N \sum_{p=1}^P I_{i,p}} \sum_{i=1}^N \sum_{p=1}^P I_{i,p} \text{CE}(\{\mathbf{b}_T\}_{i,p}, \{\mathbf{b}_S\}_{i,p}). \quad (8)$$

237 Here, α controls the contribution of masked prediction. We also add KoLeo regularization (Sablay-
 238 rolles et al., 2018; Oquab et al., 2024) to promote spreading of features and avoid collapse, improving
 239 embedding quality. The forecasting loss for the forecasting task is MSE, $\frac{1}{N} \sum_{i=1}^N |\mathbf{z}_i - \hat{\mathbf{x}}_i|^2$.
 240

241 4.3 MODEL-AGNOSTIC META-ARCHITECTURE WITH HEADS.

242 The proposed **TSDINO** adopts a unified and modulated model architecture for both of the teacher and
 243 student. They are sharing the same architecture but have independent parameters. Our model-agnostic
 244 meta-architecture consists of three main components: (i) backbone networks $f_{\theta_T}, f_{\theta_S}$ that extracts
 245 embeddings from input patches, (ii) projection heads h_{η_T}, h_{η_S} that map embeddings to a normalized
 246 feature space for self-distillation, and (iii) reconstruction heads u_{ξ_T}, u_{ξ_S} that reconstructs masked
 247 token targets. This design is agnostic to the specific model choice (e.g., TTM, TSPulse) as it is
 248 backbone for the meta model and supports various downstream tasks (classification and forecasting).
 249 The teacher further has centering and sharpening components to stabilize training.
 250

251 **Projection and Reconstruction Heads.** Projection heads $h_{\eta_T}, h_{\eta_S} : \mathbb{R}^{P \times d} \rightarrow \mathbb{R}^C$ are three-layer
 252 MLPs with GELU activation, normalization, and a softmax layer at the top layer and have student
 253 and teacher concatenated embeddings over patches after the backbone as their inputs. Reconstruction
 254 heads $u_{\xi_T}, u_{\xi_S} : \mathbb{R}^d \rightarrow \mathbb{R}^C$ are also three-layer MLPs with GELU activation, normalization, and a
 255 softmax layer at the top layer, and have student and teacher patch embeddings after the backbone as
 256 their inputs. The heads for teachers h_{η_T}, u_{ξ_T} further have a centering and sharpening component,
 257 which centers the teacher outputs and sharpens before computing the distillation loss to prevent
 258 representation collapse and stabilize training (Oquab et al., 2024).
 259

260 For forecasting, we utilize the decoder from Tiny-Tim-Mixers (TTM) (Ekambaran et al., 2024).
 261

262 4.4 AUGMENTATION TECHNIQUES AND MASKING STRATEGIES FOR TIME SERIES

263 The teacher network always receives the original sequence x , while the student network processes the
 264 augmented and masked view \tilde{x} . This asymmetric design stabilizes training by providing consistent
 265

270 targets while allowing the student to learn invariances under perturbations. For student input, we
 271 apply data augmentation \mathcal{A} and patch-wise temporal masking \mathcal{M} .
 272

273 **Data Augmentation for Student.** We define an augmentation operator \mathcal{A} that stochastically
 274 applies a set of transformations: $\mathcal{A}(x) = \mathcal{T}_k \circ \dots \circ \mathcal{T}_1(x)$, where $\{\mathcal{T}_i\}$ are sampled from a pool of
 275 augmentations. Specifically, we use Gaussian noise: $\mathcal{T}(x) = x + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, Salt-and-
 276 pepper noise: randomly replace points with $\min(x)$ or $\max(x)$ with probability p , Gaussian blur:
 277 apply convolution with a Gaussian kernel to smooth high-frequency components, FreezeSection:
 278 randomly select a consecutive segment $[j, k]$ and set $x_{j:k} = x_j$, and Random time shift circularly
 279 shift the sequence by step δ .

280 **Patch-wise temporal masking.** We use patch-wise temporal masks to encourage capturing local
 281 temporal structures. Let $\{\mathbf{x}^{(p)}\}_{p=1}^P$ be the sequence of patches, each $\mathbf{x}^{(p)} \in \mathbb{R}^\tau$. We sample patch
 282 index p until a target mask ratio $r_{\text{mask}} \in (0, 1)$ is reached, yielding the masked and unmasked
 283 index set for student I . We replace masked patches with a single learnable embedding (mask token)
 284 $\mathbf{m} \in \mathbb{R}^\tau$ to simplify implementation and provide an explicit signal of missingness. The mask token is
 285 also updated with student update rule (gradient): $\mathbf{m}^{(s)} = \mathbf{m}^{(s-1)} - \rho \nabla_{\mathbf{m}} \mathcal{L}_{\text{SSL}}(\mathbf{a}_T, \mathbf{a}_S, \mathbf{b}_T, \mathbf{b}_S)$.
 286

287 5 EXPERIMENTS

288 **Scope of Experiments.** The goal of this paper is *to introduce a novel pre-training framework for*
 289 *TSFMs* and demonstrate its effectiveness, *not to propose a new TSFM*. Specifically, we aim to show
 290 that the proposed method can maximize the performance of existing TSFM architectures in zero-shot
 291 scenarios. Therefore, for each task, we adopt an existing TSFM as the backbone network in TSDINO
 292 and replace only the pre-training procedure from gradient-based pre-training to TSDINO, ensuring
 293 that any performance gains are attributable to the learning method rather than architectural changes.
 294 Consequently, the only difference between our method and the baseline lies in the pre-training
 295 procedure, while the model architecture remains identical.
 296

297 We first describe the pre-training setup and show evaluation results of **TSDINO** on primary tasks for
 298 TSFMs in zero-shot scenarios: (i) *classification* and (ii) *forecasting*. We follow standard protocols
 299 used in prior work on forecasting and representation learning benchmarks (Zhou et al., 2021; Wu
 300 et al., 2021; Zeng et al., 2023; Nie et al., 2023; Ekambaram et al., 2024; 2025; Bagnall et al., 2018).
 301

302 **Pre-training Setup and Training Dataset.** TSDINO pre-training utilizes a subset of 1B time
 303 points drawn from the combined dataset of the Monash (Godahewa et al., 2021) and LibCity (Wang
 304 et al., 2021; Woo et al., 2024) data collections. We leverage the same pre-training data as used in
 305 TTM (Ekambaram et al., 2024) and TSPulse (Ekambaram et al., 2025), following their data selection
 306 and pre-processing procedures. All datasets used are selected under permissive licenses that allow
 307 both open-source and commercial use. We first pretrain the model using the objectives in §4.2 with
 308 the dataset. After pre-training, we *freeze* the backbone network weights and evaluate in zero-shot
 309 downstream tasks. Pre-training uses a single \times A100-80GB GPU, which took around 12 hours.
 310

311 5.1 CLASSIFICATION

312 Time-series classification is a fundamental task in time-series analysis. Our objective is to evaluate
 313 zero-shot performance without task-specific fine-tuning. For this purpose, we consider two settings:
 314 (i) similarity search and retrieval (§5.1.1) and (ii) classification with a frozen backbone (§5.1.2).
 315 Similarity search and retrieval finds the time series from a single reference example and is a pure
 316 zero-shot task, which can be seen as zero-shot classification. Classification with a frozen backbone is
 317 not a pure zero-shot task but is a common evaluation protocol in representation learning; thus, we
 318 will investigate it for reference.
 319

320 **Baselines** We use TSPulse (Ekambaram et al., 2025) as the backbone network for both the student
 321 and teacher networks of TSDINO due to its strong performance in classification and publicly available
 322 implementation, which enables us to conduct local pre-training. We compared TSDINO with TSPulse
 323 trained using the original gradient-based pre-training, referred to as **Gradient**. Note that TSDINO
 and **Gradient** share the same model architecture, differing only in the pre-training strategy. Also,

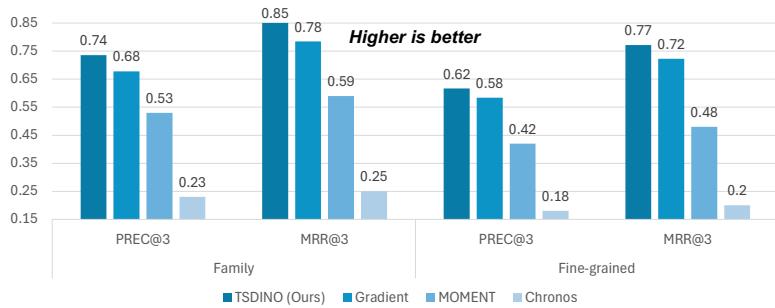


Figure 2: Zero-shot search / retrieval results on UCR dataset. The figure shows PREC@K and MRR@K (K = 3) for TSDINO compared to baselines (Gradient, Chronos, MOMENT).

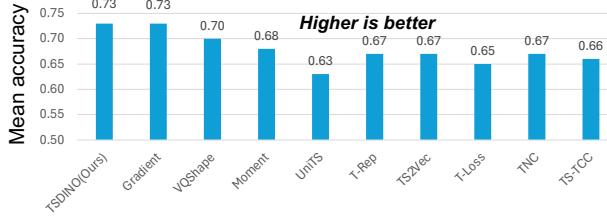


Figure 3: Classification accuracy on UEA datasets.

TSPulse has a single version, making reproducibility straightforward, which also allows us to completely follow the comparison with the compared methods in the paper. Following the paper, we use: Chronos (Ansari et al., 2024), and MOMENT (Goswami et al., 2024) in §5.1.1. In §5.1.2, we compare with **Data-specific models**. T-Rep (Fraikin et al., 2023), textitTS2Vec (Yue et al., 2022), T-Loss (Franceschi et al., 2019), TS-TCC (Eldele et al., 2021), TNC (Tonekaboni et al., 2021). **Zero-shot forecasters (TSFMs, reference only)**. TSPulse (Ekambaram et al., 2025), **VQShape** (Wen et al., 2024), MOMENT (Goswami et al., 2024), and UniTS (Gao et al., 2024b).

5.1.1 TIME-SERIES SIMILARITY SEARCH & RETRIEVAL AS ZERO-SHOT CLASSIFICATION

Setup We evaluate time-series similarity search & retrieval as zero-shot classification, where we find the time series from a single reference example. We use a synthetic dataset and a real dataset based on the UCR dataset (Dau et al., 2019), following the setting in (Ekambaram et al., 2025). After pre-training, we freeze the encoder, and retrieval is performed using cosine similarity of embeddings. We report PREC@K and MRR@K (K = 3). Higher values indicate better performance. No fine-tuning or additional adaptation is applied.

Results Figure 2 shows retrieval performance. TSDINO consistently improves the gradient-based pre-training and outperforms the baselines, demonstrating that the proposed method yields representations that better capture semantic similarity.

5.1.2 FINE-TUNED CLASSIFICATION WITH FROZEN BACKBONE

Setup For reference, while it is not zero-shot task, we conducted evaluation on fine-tuned classification, following the setting in (Ekambaram et al., 2025). We evaluate on 29 datasets from the UEA Multivariate Time Series Classification Archive (Bagnall et al., 2018). Standard train/test splits are used. We report classification mean accuracy over classes. Higher values indicate better performance.

Results As shown in Fig. 3, TSDINO achieves higher accuracy than most of the baselines and performs comparably to Gradient. Fig. 4 visualizes results across datasets. Among 29 datasets, TSDINO improves Gradient in 16 datasets, indicating that their overall performance is comparable, but TSDINO yields a different local optimum of the weights from gradient-based pre-training.

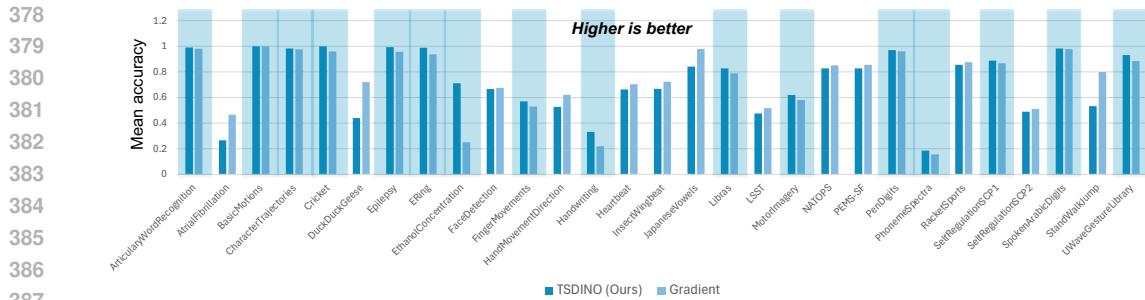


Figure 4: Per-dataset classification accuracy on UEA datasets.

Table 1: Comparison of MSE in zero-shot forecasting (best in **bold**, second-best underlined).

Dataset	TSDINO (Ours)	Gradient	TTM (From paper, For ref.)
ETTH1	<u>0.399</u>	0.404	0.395
ETTH2	0.337	0.346	<u>0.345</u>
ETTM1	0.382	0.410	<u>0.386</u>
ETTM2	0.278	0.305	<u>0.281</u>
Weather	<u>0.239</u>	0.260	0.237
Electricity	<u>0.221</u>	0.231	0.205
Traffic	<u>0.567</u>	0.579	0.564

5.2 FORECASTING

Time-series forecasting is also a core task in time-series analysis. We evaluate TSDINO’s ability to perform zero-shot forecasting on unseen datasets without task-specific fine-tuning.

Baselines We use TTM (Ekambaram et al., 2024) as the backbone network for both the student and teacher networks of TSDINO due to its strong performance in zero-shot forecasting and publicly available implementation, which enables us to conduct local pre-training. We compared TSDINO with TTM trained using the original gradient-based pre-training, referred to as Gradient. Note that TSDINO and Gradient share the same model architecture, differing only in the pre-training strategy. TTM has multiple versions, making reproducibility challenging; therefore, we report both our reproduced results Gradient and the reference values from the original paper, referred to as TTM (From paper, For ref.).

Setup We evaluate zero-shot forecasting on widely used multivariate benchmarks: *ETTh1*, *ETTh2*, *ETTm1*, *ETTm2*, *Electricity*, and *Traffic* (Zhou et al., 2021; Nie et al., 2023). These datasets cover diverse temporal patterns and scales, including strong seasonality (Electricity), high-dimensional traffic flows, and energy-related series with varying granularity. We follow standard preprocessing and splits from prior work (Nie et al., 2023; Wu et al., 2021; Zeng et al., 2023; Ekambaram et al., 2023), and evaluate multiple forecast horizons $H \in \{96, 192, 336, 720\}$. We use Mean Squared Error (MSE) as evaluation metrics, reporting average performance across all forecast horizons. Lower values indicate better performance.

Results Table 1 summarizes zero-shot forecasting performance. TSDINO consistently improves Gradient across all datasets, demonstrating the effectiveness of the proposed pre-training method. Compared to TTM (From paper, For ref.), TSDINO achieves competitive performance.

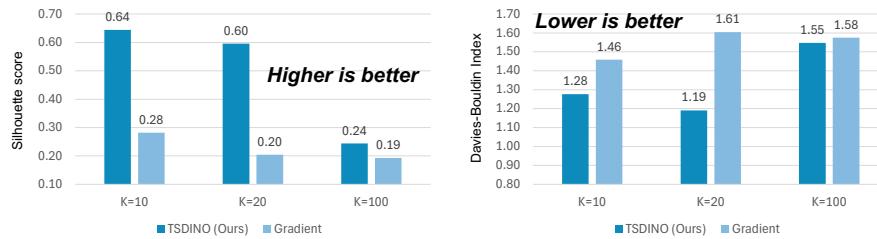
5.3 ABLATION STUDY

5.3.1 LEARNING OBJECTIVES

We measure the contribution of loss components in §4.2, TSDINO (Ours): Sequence-level loss + Patch-level loss, Sequence: Sequence-level loss only, and Patch: Patch-level loss only, in terms of

432 Table 2: Ablation study: For ZS classification, we report PREC@3 and MRR@3 (higher is better).
 433 For For FT classification, we report mean accuracy (higher is better). For forecasting, we report MSE
 434 (lower is better). Best in **bold** and second-best underlined.

436	437	Task	Teacher-student EMA		438
439	440	TSDINO (Ours)	Sequence	Patch	441
ZS Classification (Family, PREC@3)		0.735	0.700	<u>0.701</u>	0.678
ZS Classification (Family, MRR@3)		0.850	<u>0.822</u>	0.814	0.784
ZS Classification (Fine-grained, PREC@3)		0.616	0.580	0.571	<u>0.584</u>
ZS Classification (Fine-grained, MRR@3)		0.771	<u>0.737</u>	0.718	0.723
FT Classification (Mean accuracy)		0.730	0.707	<u>0.713</u>	0.730
Forecasting (MSE, Lower is better)		0.346	0.358	0.358	0.362



445 Figure 5: Clustering metrics comparison using k -means on frozen features ($k = 10, 20, 100$).
 446 TSDINO improves the gradient-based pre-training, Gradient, in silhouette score and Davies-
 447 Bouldin index, indicating better clustering quality.

448 performance improvement from the gradient-based pre-training in tasks, zero-shot (ZS) classification
 449 in §5.1.1, fine-tuned (FT) classification in §5.1.2, and forecasting in §5.2, as shown in Table 2. Both
 450 sequence-level and patch-level losses contribute to performance gains, with the full combination
 451 yielding the best results overall. This indicates that learning both global invariances and local temporal
 452 structures is crucial for effective time-series representation learning. The teacher-student framework
 453 and EMA weight updates basically contribute to improving performance compared to the ordinary
 454 gradient-based training.

455 5.3.2 CLUSTERING RESULTS OF EMBEDDINGS

456 To investigate the quality of learned representations, we evaluate clustering performance on em-
 457 beddings. We apply k -means on embeddings ($k = \#$ classes) and compute the silhouette score and
 458 Davies-Bouldin Index. A higher silhouette score and lower Davies-Bouldin index indicate better
 459 clustering quality. We compare TSDINO with the gradient-based pre-training, Gradient, using
 460 TSPulse.

461 **Results** TSDINO improved clustering metrics compared to Gradient by a good margin, indi-
 462 cating that its representations effectively capture semantic similarity. This nature aligns with good
 463 zero-shot performances in various downstream tasks.

464 6 CONCLUSION

465 We presented TSDINO, a self-supervised pre-training framework for time-series foundation models
 466 based on the teacher-student self-distillation framework. Our design incorporates time-series aug-
 467 mentation and patch-wise masking to respect temporal continuity and structure. Extensive experiments
 468 demonstrate that TSDINO learns robust, generalizable features that excel in zero-shot tasks, including
 469 classification and forecasting across diverse benchmark datasets. Ablations confirm the importance
 470 of each component in our objectives.

486 **Reproducibility statement** We wrote the details of experimental settings, computing resources,
 487 and datasets in §5. We plan to release the code after the paper is accepted.
 488

489 REFERENCES
 490

491 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman,
 492 Diogo Almeida, Janko Altenschmidt, Sam Altman, and Shyamal Anadkat , et al. Gpt-4 technical
 493 report. *arXiv preprint arXiv:2303.08774*, 2023.

494 Abdul Fatir Ansari, Lorenzo Stella, Caner Türkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen,
 495 Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, et al. Chronos: Learning
 496 the language of time series. *arXiv:2403.07815*, 2024. URL <https://arxiv.org/abs/2403.07815>.

497 Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul
 498 Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv
 499 preprint arXiv:1811.00075*, 2018.

500 Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and
 501 Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pp. 9650–
 502 9660, 2021.

503 Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for
 504 time-series forecasting. In *ICML*, 2024. URL <https://arxiv.org/abs/2310.10688>.

505 Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh
 506 Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive.
 507 *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019. doi: 10.1109/JAS.2019.1911747.

508 Vijay Ekambaram, Arindam Jati, Nam Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam.
 509 Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting. In *Proceedings of
 510 the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 459–469, 2023.

511 Vijay Ekambaram, Arindam Jati, Pankaj Dayama, Sumanta Mukherjee, Nam H. Nguyen, Wesley M.
 512 Gifford, and Jayant Kalagnanam. Tiny time mixers (ttms): Fast pre-trained models for enhanced
 513 zero/few-shot forecasting of multivariate time series. *arXiv preprint arXiv:2401.03955*, 2024. URL
 514 <https://arxiv.org/abs/2401.03955>.

515 Vijay Ekambaram, Subodh Kumar, Arindam Jati, Sumanta Mukherjee, Tomoya Sakai, Pankaj
 516 Dayama, Wesley M Gifford, and Jayant Kalagnanam. Tspulse: Dual space tiny pre-trained models
 517 for rapid time-series analysis. *arXiv preprint arXiv:2505.13033*, 2025.

518 Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and
 519 Cuntai Guan. Time-series representation learning via temporal and contextual contrasting. *arXiv
 520 preprint arXiv:2106.14112*, 2021.

521 Archibald Fraikin, Adrien Bennetot, and Stéphanie Allassonnière. T-rep: Representation learning for
 522 time series using time-embeddings. *arXiv preprint arXiv:2310.04486*, 2023.

523 Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation
 524 learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

525 Haozhi Gao, Qianqian Ren, and Jinbao Li. Distillation enhanced time series forecasting network
 526 with momentum contrastive learning. *Information Sciences*, 675:120712, 2024a.

527 Shanhua Gao, Teddy Koker, Owen Queen, Tom Hartvigsen, Theodoros Tsiligkaridis, and Marinka
 528 Zitnik. Units: A unified multi-task time series model. *Advances in Neural Information Processing
 529 Systems*, 37:140589–140631, 2024b.

530 Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-
 531 Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

540 Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski.
 541 MOMENT: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*,
 542 2024. URL <https://arxiv.org/abs/2402.03885>. ICML 2024.

543
 544 Ainaz Hajimoradlou, Leila Pishdad, Frederick Tung, and Maryna Karpusha. Self-supervised time
 545 series representation learning with temporal-instance similarity distillation. In *ICML 2022 Pre-
 546 training Workshop*, 2022. URL <https://openreview.net/forum?id=nhtkdCvVLih>.

547
 548 Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and
 549 Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. *arXiv preprint
 550 arXiv:2403.14735*, 2024. URL <https://arxiv.org/abs/2403.14735>. KDD'24 tuto-
 551 rial/survey.

552
 553 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, henggang Zhao, Chengqi
 554 Deng, Chenyu Zhang, and Chong Ruan , et al. Deepseek-v3 technical report. *arXiv preprint
 555 arXiv:2412.19437*, 2024a.

556
 557 Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao
 558 Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. Moirai-moe: Empowering time series
 559 foundation models with sparse mixture of experts. *arXiv preprint arXiv:2410.10469*, 2024b. URL
 560 <https://arxiv.org/abs/2410.10469>.

561
 562 Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64
 563 words: Long-term forecasting with transformers. In *ICLR*, 2023. URL <https://arxiv.org/abs/2211.14730>.

564
 565 Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov,
 566 Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning
 567 robust visual features without supervision. In *arXiv*, 2024. URL <https://arxiv.org/abs/2304.07193>.

568
 569 Felix Pieper, Konstantin Ditschuneit, Martin Genzel, Alexandra Lindt, and Johannes Otterbach.
 570 Self-distilled representation learning for time series. *arXiv preprint arXiv:2311.11335*, 2023. URL
 571 <https://arxiv.org/pdf/2311.11335.pdf>.

572
 573 Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for
 574 similarity search. *arXiv preprint arXiv:1806.03198*, 2018.

575
 576 Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for
 577 time series with temporal neighborhood coding. In *ICLR*, 2021. URL <https://arxiv.org/abs/2106.00750>.

578
 579 Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. Libcity: An open
 580 library for traffic prediction. In *Proceedings of the 29th international conference on advances in
 581 geographic information systems*, pp. 145–148, 2021.

582
 583 Yunshi Wen, Tengfei Ma, Lily Weng, Lam Nguyen, and Anak Agung Julius. Abstracted shapes as
 584 tokens-a generalizable and interpretable model for time-series classification. *Advances in Neural
 585 Information Processing Systems*, 37:92246–92272, 2024.

586
 587 Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo.
 588 Unified training of universal time series forecasting transformers. *PMLR*, 2024.

589
 590 Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers
 591 with auto-correlation for long-term series forecasting. In *NeurIPS*, 2021. URL <https://arxiv.org/abs/2106.13008>.

592
 593 Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and
 594 Bixiong Xu. Ts2vec: Towards universal representation of time series. *AAAI*, 2022. URL
 595 <https://arxiv.org/abs/2106.10466>.

596
 597 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series
 598 forecasting? *AAAI*, 2023. URL <https://arxiv.org/abs/2205.13504>.

594 Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y. Zhang, Yux-
595 uan Liang, Guansong Pang, Dongjin Song, and Shirui Pan. Self-supervised learning for time series
596 analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine*
597 *Intelligence*, 2024. URL <https://arxiv.org/abs/2306.10125>. arXiv:2306.10125.

598 Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Zhou, Junshuai Zhang, and Jinhai
599 Li. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*,
600 2021. URL <https://arxiv.org/abs/2012.07436>.

602 Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot:
603 Image bert pre-training with online tokenizer. In *ICLR*, 2022a. URL <https://arxiv.org/abs/2111.07832>.

605 Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT:
606 Image BERT pre-training with online tokenizer. In *International Conference on Learning Repre-*
607 *sentations (ICLR)*, 2022b. URL <https://arxiv.org/abs/2111.07832>.

609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647