Wild-Time: A Benchmark of in-the-Wild Distribution Shift over Time – Appendix

Huaxiu Yao[1*], **Caroline Choi**[1*], **Bochuan Cao**[2], **Yoonho Lee**[1], **Pang Wei Koh**[1], **Chelsea Finn**[1]
[1]Stanford University, [2]Pennsylvania State University
wildtime@googlegroups.com

A Dataset Description

A.1 Yearbook

A.1.1 Setup

Problem Setting. The task is classifying the gender of an American high schooler from a yearbook photo. The input x is a 32×32 grayscale image, and the label y is male or female.

Data. Yearbook is based on the Portraits dataset [16] (MIT license), which collected and processed 37,921 frontal-facing yearbook portraits from $1905 - 2013$ from 128 American high schools in 27 states. The Portraits dataset reflects changing fashion styles and social norms over the decades.

The original Portraits dataset did not evaluate models under a distribution shift setting. We use a subset of the Portraits dataset, consisting of data from $1930 - 2013$. Our fixed time split (Eval-Fix) uses the first 41 years ($1930 - 1970$) for ID, and the remaining 43 years for OOD ($1971 - 2013$). For streaming evaluation (Eval-Stream), we treat each year as a single timestamp. For each timestamp, we randomly allocate 10% of the data to training, and the remaining 90% for validation. For OOD testing, all samples in each year are used. We provide the number of examples allocated to ID Train, ID Test, and OOD Test for each timestamp in Table 1.

The original Portraits dataset is provided as a set of hierarchical directories, organized by year, with PNG images of size 96×96 pixels. To reduce download times and I/O usage, we downsample the images from [16] to 32×32 pixels. We exclude the first 25 years ($1905 - 1929$) due to few samples in these years.

Evaluation Metrics. We evaluate models by their average and worst-time OOD accuracies. The former measures the model's ability to generalize across time, while the latter additionally measures model robustness to trends in time-specific visual patterns.

Eval-Stream evaluates performance across the next 10 years to test on visual trend changes over the decade without resulting in an unreasonably long evaluation time, due to the large number of timestamps in this dataset.

A.1.2 Broader Context

Facial recognition has been widely adopted in recent years. Employed by governments and private companies, facial recognition models are used in smartphones, robotics, advanced human-computer interaction systems. However, human appearance shifts over time due to changing social norms (e.g., the practice of smiling to the camera) and fashion trends (e.g., hair styles, popularity of eyewear). To remain reliable and effective, facial recognition models must be robust to changes in human appearance over time.

[*]Huaxiu Yao and Caroline Choi contributed equally.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

Table 1: Data subset sizes for the Yearbook task.

Years	ID Train	ID Test	OOD Test
1930 - 1934	1,051	120	1,171
1935 - 1939	1,361	154	1,515
1940 - 1944	2,047	230	2,277
1945 - 1949	1,979	222	2,201
1950 - 1954	1,604	181	1,785
1955 - 1959	1,820	205	2,025
1960 - 1964	1,482	167	1,649
1965 - 1969	2,812	315	3,127
1970 - 1974	2,326	260	2,586
1975 - 1979	2,329	261	2,590
1980 - 1984	2,654	298	2,952
1985 - 1989	2,239	251	2,490
1990 - 1994	2,207	249	2,456
1995 - 1999	2,564	287	2,851
2000 - 2004	2,447	274	2,721
2005 - 2009	1,407	159	1,566
2010 - 2013	1,102	125	1,227
Fixed-time split	14,901	1,677	21,439

While Yearbook is not a facial recognition task, the Yearbook dataset can be used to train facial image analysis models that are robust to changes in appearance over time.

A.2 FMoW-Time

A.2.1 Setup

Problem Setting. The task in the FMoW-Time dataset is to classify the functional purpose of a region inside a satellite image. The input x is an 224×224 RGB satellite image, and the label y is one of 62 categories of building or land use. The data was collected from 16 different years. Our fixed time split (Eval-Fix) allocates the first 12 years for training and the last 4 years for testing. For streaming evaluation (Eval-Stream), we treat each year as a single timestamp.

Data. FMoW-Time is based on the Functional Map of the World dataset (license: https://github.com/fMoW/dataset/blob/master/LICENSE) [11], a dataset of satellite images taken from $2002-2018$, from over 200 countries. Each satellite image is labeled according to the functional purpose of the buildings or land depicted in the image.

We adapt the version of the FMoW dataset from the WILDS benchmark [30], which consists of 141,696 RGB satellite images resized to 224×224 pixels. The train/val/test data splits in FMoW-WILDS contain images from disjoint location coordinates, and all splits contain data from all 5 geographic regions. For FMoW-Time, we partition each year's data into train/validation as follows. For $2002 - 2013$, we use the FMoW-WILDS Training (ID) split for training, and the Validation (ID) and Test (ID) splits for validation. For $2013 - 2015$, we use data in the Validation (OOD) split and allocate 90% of the data from each year to train, and the remaining 10% to validation. For $2015 - 2017$, we allocate 90% of the data from each year in the Test (OOD) split to train, and the remaining 10% to validation. Our fixed time split (Eval-Fix) uses the first 11 years ($2002 - 2013$) for training, and the remaining 5 years ($2013 - 2018$) for testing. For streaming evaluation (Eval-Stream), we treat each year as a single timestamp. We provide the number of examples allocated to ID Train, ID Test, and OOD Test at each timestamp in Table 2.

Evaluation Metrics. We evaluate models with the top-1 accuracy, both in terms of the average across all OOD timestamps and the accuracy on the worst timestamp. The former measures the model's ability to reliably generalize across time and the latter more specifically tests the robustness at the most severe shifts. For Eval-Stream, we evaluate performance across the next 6 years.

Table 2: Data subset sizes for the FMoW-Time task.

Year	ID Train	ID Test	OOD Test
2002	1,455	448	1,903
2003	1,985	570	2,555
2004	1,545	450	1,995
2005	2,207	629	2,836
2006	2,765	796	3,561
2007	1,338	349	1,687
2008	1,975	584	2,559
2009	6,454	1,920	8,374
2010	16,498	4,915	21,413
2011	19,237	5,711	24,948
2012	21,404	6,438	27,842
2013	3,465	385	3,850
2014	5,572	620	6,192
2015	8,885	988	9,873
2016	14,363	1,596	15,959
2017	5,534	615	6,149
Eval-Fix split	76,863	22,810	42,023

A.2.2 Broader Context

ML models for satellite imagery can automate applications such as deforestation tracking, population density prediction, crop yield prediction [19, 44, 47]. Visual features in satellite data change over time due to both human and environmental activity, requiring a model that makes predictions for recent images using labeled data from the past. Through such applications, policy and humanitarian efforts would greatly benefit from temporally robust models which can reliably monitor global-scale satellite imagery even when conditions change over time.

A.3 MIMIC-IV

A.3.1 Setup

Problem Setting. The MIMIC-IV dataset contains two tasks: MIMIC-Readmission and MIMIC-Mortality. For both of these tasks, the input x is the concatenated ICD9 codes of diagnosis and treatment for a single patient.

- **MIMIC-Readmission**: the task is predicting hospital readmission for a patient. The label y is whether the patient was readmitted to the hospital within 15 days.
- **MIMIC-Mortality**: the task is predicting in-hospital mortality for each patient. The label y is whether the patient passed away during their hospital stay.

Data. The MIMIC-IV database [27] contains deidentified EHRs of 382,278 patients admitted to the emergency department or intensive care unit (ICU) at the Beth Israel Deaconess Medical Center (BIDMC) from $2008-2019$. To protect patient privacy, the reported admission year is in a three year long date range. Hence, our timestamps are groups of three years: $2008--2010$, $2011--2013$, $2014--2016$, $2017--2019$. We considered ICU patient data sourced from the clinical information system MetaVision at the BIDMC, released in the MIMIC-IV v1.0 dataset, which contains 53,150 patient records. MIMIC-IV requires PhysioNet credentialing for use of human subject data.

We use a subset of the original MIMIC-IV dataset, where we regard each admission as one entry. For each admission, we collect the ICD9 codes of diagnosis and treatment. For each record, we concatenate the corresponding ICD9 codes [38] of diagnosis and treatment. We use the concatenated diagnosis and treatment ICD9 codes as the input feature. Our fixed time split (Eval-Fix) uses the first 6 years ($2008-2013$) for training, and the remaining 6 years for testing ($2014-2019$). For streaming evaluation (Eval-Stream), we treat each three-year block as a single timestamp. We allocate 20% of the data at each timestamp for test, and the rest for training. For OOD testing, all samples in each three-year block are used. We provide the number of examples allocated to ID Train, ID Test, and OOD Test for each timestamp in Table 3.

Table 3: Data subset sizes for the two MIMIC-IV tasks, MIMIC-Mortality and MIMIC-Readmission.

3-Year Block	ID Train	ID Test	OOD Test
2008 - 2010	60,851	15,215	76,066
2011 - 2013	55,714	13,930	69,644
2014 - 2016	53,932	13,485	67,417
2017 - 2019	45,990	11,500	57,490
Eval-Fix split	116,565	29,145	124,907

Evaluation Metrics. For MIMIC-Readmission, we evaluate models by their average and worst-time OOD accuracies. For MIMIC-Mortality, we evaluate models by their average and worst-time ROC-AUC due to label imbalance. The average metric measures the model's ability to generalize across time, while the worst-time metric additionally measures model robustness to temporal distribution shifts in patient data. Eval-Stream evaluates performance across the next 3 years, which represents 25% of all timestamps in the entire dataset.

A.3.2 Broader Context

Many applications of machine learning to clinical healthcare have emerged in the last decade, such as predicting disease risk [32], medication changes [48], patient subtyping [3], in-hospital mortality [18], and length of hospital stay [14]. However, a key obstacle in deploying machine learning-based clinical decision support systems is distribution shift associated with changes in healthcare over time [18]. Existing domain generalization and unsupervised domain adaptation algorithms have been shown to produce less robust models compared to ERM in a variety of tasks (e.g., mortality, length of stay, sepsis, and invasive ventilation prediction) on the MIMIC-IV dataset [18], underscoring the need for better approaches.

The MIMIC-IV Mortality and Readmission tasks evaluate model robustness to temporal shifts in clinical medicine.

A.4 Huffpost

A.4.1 Setup

Problem Setting. The task is classifying the news category of an article from the headline. The input x is a news headline, and the label y is one of 11 news categories.

Data. Huffpost is based on the Kaggle News Category Dataset [36] (license: CC0: Public Domain), which contains approximately 200,000 news headlines and their corresponding news categories from the Huffington Post from $2012 - 2018$. The Kaggle News Category Dataset contains 41 different news categories.

We use a subset of the Huffpost dataset, consisting of 7 years from $2012 - 2018$ and samples from 11 news categories (Black Voices, Business, Comedy, Crime, Entertainment, Impact, Queer Voices, Science, Sports, Tech, Travel). We partition the data by year.

Our fixed time split (Eval-Fix) uses 2016 as the time split, allocating $2012 - 2015$ (4 years) for ID and $2016 - 2018$ (3 years) for OOD. For streaming evaluation (Eval-Stream), we treat each year as a single timestamp. We allocate 10% of the data at each timestamp for test, and the rest for training. For OOD testing, all samples are used. Table 4 lists the number of examples allocated to ID Train, ID Test, and OOD Test for each timestamp.

The News Category Dataset is provided as a CSV file. We exclude news categories which do not appear in all years $2012 - 2018$ to obtain the 11 news categories in Huffpost. We shuffle samples in each year, and randomly select 10% of the samples in each year as ID test and allocate the remaining 90% for training. For OOD testing, all samples in each year are used.

Evaluation Metrics. We evaluate models by their average and worst-time OOD accuracies. The former measures the model's ability to generalize across time, while the latter additionally measures model robustness to trends in time-specific visual patterns.

Table 4: Data subset sizes for the Huffpost task.

Year	ID Train	ID Test	OOD Test
2012	6,701	744	7,446
2013	7,492	832	8,325
2014	9,539	1,059	10,599
2015	11,826	1,313	13,140
2016	10,548	1,172	11,721
2017	7,907	878	8,786
2018	3,501	388	3,890
Eval-Fix split	35,558	3,948	24,397

Eval-Stream evaluates performance across the next 3 years, which represents 42.9% of the timestamps in the entire dataset.

A.4.2 Broader Context

Many language models which deal with information correlated with time exhibit performance degradation in downstream tasks such as Twitter hashtag classification [24] or question answering systems [31]. These performance drops along the temporal dimension reflect changes in the style or content of news that change over time. For instance, American politics in 2022 is more polarized than it was in 2012, according to a study by the Pew Research Center [13]. Models must be robust to such changes in factual knowledge.

A.5 arXiv

A.5.1 Setup

Problem Setting. The task is classifying the primary classification category of a research paper from the title. The input x is the paper title, and the label y is one of 172 paper categories.

Data. arXiv is based on the Kaggle arXiv Dataset [12] (license: CC0: Public Domain), which provides metadata of arXiv preprints from $2007 - 2023$. These include: arXiv id, submitter, authors, title, comments, journal-ref, doi, abstract, categories, and versions.

We use a subset of the Kaggle arXiv dataset for arXiv, which consists of paper titles and theirr corresponding primary categories. Our fixed time split (Eval-Fix) uses 2016 as the time split, allocating data from $2007 - 2016$ (10 years) for ID, and data from $2017 - 2022$ (6 years) for OOD. For streaming evaluation (Eval-Stream), we treat each year as a single timestamp. We allocate 10% of the data at each timestamp for test, and the rest for training. For OOD testing, all samples are used. Table 5 lists the number of examples allocated to ID Train, ID Test, and OOD Test for each timestamp.

The arXiv Dataset metadata is provided as a JSON file. We store only the primary category and the preprint title, and sort the data by update date, partitioning by year. We shuffle samples in each year, and randomly select 10% of the samples in each year as ID test and allocate the remaining 90% for training. For OOD testing, all samples in each year are used.

Evaluation Metrics. We evaluate models by their average and worst-time OOD accuracies. The former measures the model's ability to generalize across time, while the latter additionally measures model robustness to trends in time-specific visual patterns.

Eval-Stream evaluates performance across the next 6 years, which represents 37.5% of all timestamps in the dataset.

A.5.2 Broader Context

Similar to changes in news and current events reflected in the Huffpost dataset, the content of arXiv preprints also change over time as research fields evolve. For example, "neural network attack" was originally a popular keyword in the security community, but it gradually became more prevalent in the machine learning community. As a result, primary categories of arXiv preprints shift over time.

Table 5: Data subset sizes for the arXiv task.

Year	ID Train	ID Test	OOD Test
2007	131,550	14,616	146,167
2008	62,460	6,939	69,400
2009	206,244	22,916	229,161
2010	50,665	5,629	56,295
2011	55,741	6,193	61,935
2012	51,678	5,741	57,420
2013	64,951	7,216	72,168
2014	79,498	8,833	88,332
2015	193,979	21,553	215,533
2016	120,682	13,409	134,092
2017	111,024	12,336	123,361
2018	123,891	13,765	137,657
2019	142,767	15,862	158,630
2020	166,014	18,445	184,460
2021	201,241	22,360	223,602
2022	89,765	9,973	99,739
Eval-Fix split	1,017,448	113,045	927,449

B Algorithm Description

Before introducing all algorithms, we recall that each example is (x, y, t), where x, y, t represent input feature, label, and timestamp, respectively.

B.1 Classical Supervised Learning

- **Empirical Risk Minimization (ERM).** We first consider Empirical Risk Minimization (ERM). This algorithm ignores the time information (t) and minimizes the average training loss

$$\theta^* = \arg\min_{\theta} \ell(x, y; f_\theta) \tag{1}$$

over the entire training dataset.

B.2 Continual Learning

- **Fine-tuning.** In fine-tuning, we use the newly observed labeled examples to continuously fine-tune the learned model without any explicit regularizer between consecutive timestamps.
- **Elastic Weight Consolidation (EWC).** Inspired by synaptic consolidation, EWC slows down the learning process for new tasks based on their relevance to previous tasks. Specifically, when adapting to a new task, EWC's loss function keeps the post-adaptation network parameters close to the parameters learned on previous tasks.
- **Synaptic Intelligence (SI).** Motivated by synaptic dynamics, SI enables deep neural network to learn sequence of tasks by using synaptic state to track the parameter values and maintain online estimation of the importance of past learned experience.
- **Averaged Gradient Episodic Memory (A-GEM).** Gradient Episodic Memory (GEM) leverages an episodic memory to store a selected set of examples from previous tasks in a continual learning setting. When adapting to a new task, the algorithm aims to make the updated model simultaneously perform well on examples in the new task and examples from the episode memory. A-GEM provides an efficient training strategy for Gradient Episodic Memory that significantly improves its computation and memory efficiency. Specifically, instead of making the updated model perform better on each individual previous tasks in the memory, A-GEM aims to produce a model that shows high average performance across the tasks in the episode memory.

B.3 Temporal Invariant Learning

- **CORAL.** CORAL penalizes the differences in the mean and covariance of the feature distributions of each domain. For CORAL, we adapted our implementation from the public repositories for DomainBed and WILDS [30]. CORAL is applicable to all datasets used in Wild-Time.

- **IRM.** Invariant risk minimization aims to learn an invariant predictor that performs well across all domains. The vanilla IRM objective can be reformulated as a bi-level optimization, which is challenging to solve. Following the original paper [1], we adopt IRM-v1 in this paper, an efficient approximation to the original IRM objective for learning invariant predictors.
- **Mixup** is an interpolation-based approach, which generates new training examples by applying the same interpolation strategies on the input features and their corresponding labels [53]. The original training samples are replaced by the newly generated samples for training.
- **LISA.** Motivated by mixup [53], LISA selectively interpolates examples to cancel out domain information. LISA has two variants — intra-label LISA and intra-domain LISA. Intra-label LISA interpolates examples with the same label but from different domains. Intra-domain LISA interpolates examples with the same domain but different labels. Furthermore, as mentioned in [51], intra-LISA performs better in domain shifts without considering domain information. We follow the implementation of Yao et al. [51] and only apply intra-label in Wild-Time.
- **GroupDRO.** GroupDRO uses distributionally robust optimization to optimize the worst-domain loss during the training stage. We follow the implementation of Sagawa et al. [41] and apply group adjustments, strong penalty and early stopping in GroupDRO.

B.4 Self-Supervised Learning

- **SimCLR** [10] is a simple contrastive learning approach for visual recognition. It uses normalized temperature-scaled cross entropy as the loss function and introduces a nonlinear transformation between the learned representation and the contrastive loss. We follow the implementation of Chen et al. [10].
- **SwAV** [7] simultaneously clusters the data and encourages the consistency of cluster assignments generated by different kinds of data augmentations. We follow the implementation of Caron et al. [7].

B.5 Bayesian Learning

- **SWA.** Stochastic Weight Averaging [23] averages multiple parameter values along the trajectory of SGD with almost no computational overhead. This method has been shown to lead to better in-distribution generalization due to its ability to find a better approximation to the posterior distribution over parameters. This property is reflected through the flatness of the learned optima. We follow the official implementation of SWA with the same learning rate as ERM and use default values for other hyperparameters.

C Experimental Details

All reported results are averaged over 3 random seeds. Experiments are conducted on a GPU-cluster with 6 GPU nodes. All classification tasks (i.e., Yearbook, FMoW-Time, MIMIC Mortality, MIMIC Readmission, Precipitation, HuffPost, arXiv) were trained with cross-entropy loss. In our experiments, we tune hyperparameters of all baselines by applying cross-validation with grid search.

For all methods, we use minibatch stochastic optimizers to train models, sampling uniformly from the ID set (in the Eval-Fix setting) or from each timestamp (in the Eval-Stream setting).

We report the number of train iterations used to train baselines for each dataset, under both the Eval-Fix and Eval-Stream settings. A single train iteration corresponds to one update via loss backpropagation. Under the Eval-Fix setting, the number of train iterations is the number of updates to the model on the ID train set. Under the Eval-Stream setting, in which models are trained incrementally, the number of train iterations corresponds to the number of updates to the model at each timestamp.

C.1 Eval-Fix Split Determination

To determine the time splits for the Eval-Fix setting of each dataset, we considered all ID/OOD splits ranging from 40%-60% ID/OOD to 80%-20% ID/OOD. For each of these time splits, we ran ERM on the ID and OOD sets, and selected the split with the largest discrepancy between average ID accuracy and average OOD accuracy.

C.2 Detailed Set Split Strategy

Suppose we have T timestamps. At each timestamp, we randomly sample 90% of the examples for training, and allocate the remaining 10% validation examples for ID evaluation. We detail the difference between the Eval-Fix and Eval-Stream setting as follows:

Eval-Fix Setting. In Eval-Fix, as shown in Figure 1, we have a split timestamp t_s. The ID timestamps are $t < t_s$, and the OOD timestamps are $t \geq t_s$. The training set consists of all training examples from the ID timestamps $t < t_s$. The ID validation set consists of all validation examples from the ID timestamps $t < t_s$. All examples in all test timestamps $t \geq t_s$ are used as the OOD test set.

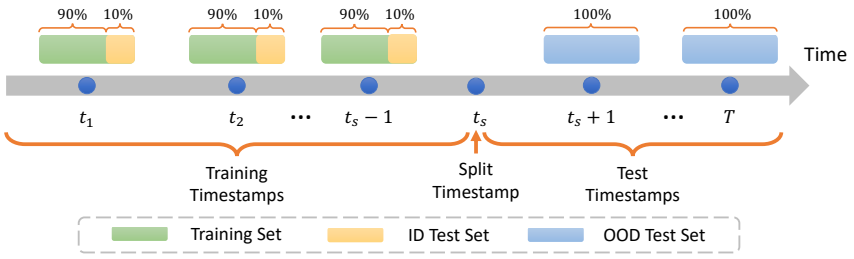


Figure 1: Data split under Eval-Fix setting.

Eval-Stream Setting. In Eval-Stream, at each evaluation timestamp, we evaluate across the next K timestamps. Specifically, at each timestamp $t \in [1, \ldots, T]$, we evaluate our model across the timestamps $\{t+1, \ldots, t+K\}$, which is illustrated in Figure 2.

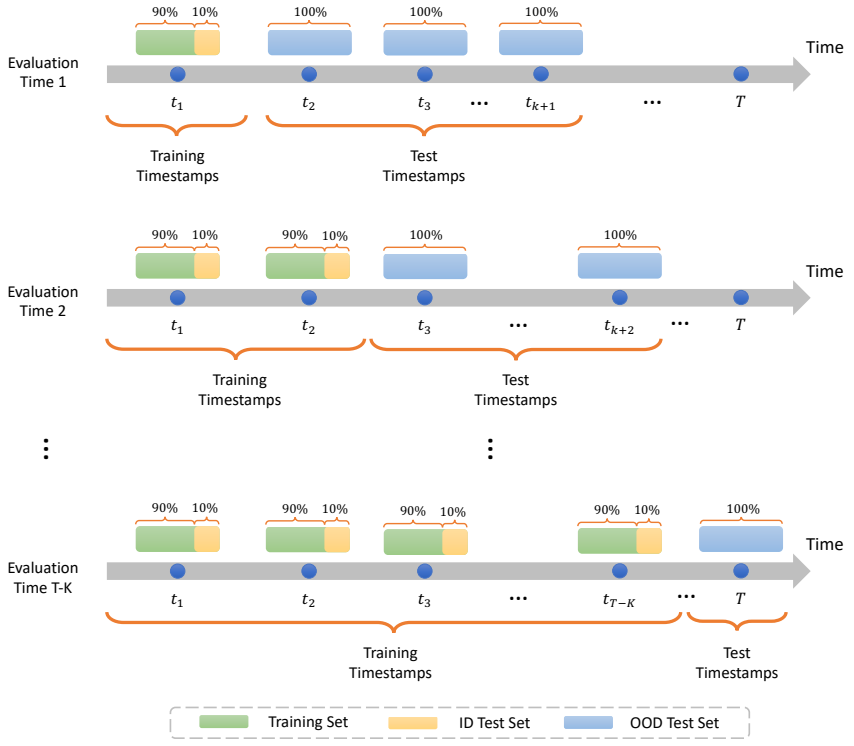


Figure 2: Data split under Eval-Stream setting.

Hence, Eval-Fix can be viewed as a single timestamp evaluation within Eval-Stream, where we evaluate only at t_s and set $n = T - t_s$.

Table 6: Hyperparameters for CORAL, GroupDRO, and IRM baselines on all WildT datasets.

Dataset	CORAL Penalty	IRM Penalty	lr	# Substreams	Substream Size
Yearbook	0.9	1.0	1e-1	10	5
FMoW-Time	0.9	1.0	1e-4	3	3
MIMIC-IV-Mort	1.0	1.0	5e-4	4	3
MIMIC-IV-Readmit	1.0	1.0	5e-4	3	3
HuffPost	0.9	1.0	2e-5	3	2
arXiv	0.9	1.0	2e-5	4	4

C.3 Hyperparameter Settings and Model Architectures

C.3.1 General Settings

Yearbook. We use a 4-layer convolutional network. Each convolutional layer has kernel size 3×3, stride of 1×1, padding of size 1, 32 output channels, a spatial batch norm layer, ReLU activation, and a 2D max pool layer with kernel size 2×2.

We use the Adam optimizer with a fixed learning rate of 10^{-3} and train with a batch size of 32. Baselines were trained for 3000 iterations under the Eval-Fix setting and for 100 iterations under the Eval-Stream setting.

FMoW-Time. Following [30] and [11], we use a DenseNet-121 model [20] pretrained on ImageNet with no L_2 regularization.

We use the Adam optimizer with an initial learning rate of 10^{-4} that decays by 0.96 per epoch and a batch size of 64. Baselines were trained for 3000 iterations for the Eval-Fix setting and for 500 iterations for the Eval-Stream setting.

MIMIC-IV. We use a Transformer, consisting of an encoder and an decoder. Here, we collect the vocabulary based on the ICD9 codes.

We use the Adam optimizer with a learning rate of 5×10^{-4} and a batch size of 128. Baselines were trained for 3000 iterations under the Eval-Fix setting and for 500 iterations under the Eval-Stream setting.

Huffpost. We use a network backbone comprising of a pretrained DistilBERT base model (uncased) from [42] and a fully-connected, classification layer.

We use the AdamW optimizer with a learning rate of 2×10^{-5}, weight decay of 10^{-2}, and train with a batch size of 32. Baselines were trained for 6000 iterations under the Eval-Fix setting and for 1000 iterations under the Eval-Stream setting.

arXiv. We use the same network backbone, optimizer, learning rate, weight decay, and number of train iterations as those used for the Huffpost dataset. We train all baselines with a batch size of 64.

C.3.2 Algorithm-Specific Hyperparameters

Temporally Invariant Methods For GroupDRO, CORAL, and IRM, we follow WILDS [30] and use minibatch stochastic optimizers to train models, sampling uniformly from each substream (i.e., the domain in our temporal robustness setting), regardless of the number of training examples in the substream.

- **GroupDRO.** We adapted the implementation of GroupDRO from Sagawa et al. [41] and Koh et al. [30]. Each example in the minibatch is sampled independently with uniform probabilities across substreams.

  We list the hyperparameters used for GroupDRO on all WildT datasets in Table 6; namely, the number of substreams (e.g., number of groups) and substream size (e.g., group size).

- **CORAL.** We adapted the implementations of DeepCORAL from Gulrajani and Lopez-Paz [17] and Koh et al. [30], and compute CORAL penalties between features from all pairs of substreams, which we treat as groups/domains.

We list the hyperparameters used for CORAL on all WildT datasets in Table 6, which include the CORAL penalty λ_c, number of substreams (e.g., number of groups), and substream size (e.g., group size). CORAL was trained with a penalty of $\lambda_c = 0.1$ on the MIMIC-Mortality task, and $\lambda_c = 1.0$ on the MIMIC-Readmission task. For all remaining datasets, we used a default penalty of $\lambda_c = 0.9$.

- **IRM.** We adapted the implementations of IRM from Arjovsky et al. [1] and Koh et al. [30].

  We list the hyperparameters used for IRM on all WildT datasets in Table 6, which include the IRM penalty λ_i, number of substreams (e.g., number of groups), and substream size (e.g., group size). IRM was trained with a penalty of $\lambda_i = 1.0$ on all datasets.
- **LISA.** We adapted the implementation of LISA from Yao et al. [51] and implemented intra-label LISA, where training samples with the same label are interpolated. For the Yearbook and FMoW-Time datasets, the input image tensors were interpolated. For the arXiv, Huffpost, and MIMIC-IV datasets, the learned feature representations were interpolated.

  All LISA experiments were conducted with $\alpha = 2.0$, where the interpolation ratio $\lambda \in [0, 1]$ is drawn from a Beta(α, α) distribution.
- **Mixup.** For mixup, we use the same hyperparameters as ERM.

Continual Learning Methods

- **A-GEM.** We adapted the implementation of A-GEM from Chaudhry et al. [9] and "Mammoth - An Extendible (General) Continual Learning Framework for Pytorch" [5, 6]. All A-GEM experiments were conducted with a default buffer size of 1000.
- **EWC.** We adapted the implementation of EWC from Kirkpatrick et al. [29], van de Ven and Tolias [45], and van de Ven and Tolias [46].

  For the EWC loss regularization strength, we use a default value of 0.5 for the Yearbook, FMoW-Time, Huffpost, arXiv, and MIMIC-IV-Readmit datasets. For MIMIC-IV-Mortality, we use 1.0.
- **SI.** We adapted the implementation of SI from Zenke et al. [52], van de Ven and Tolias [45], van de Ven and Tolias [46].

  For the SI loss regularization strength λ_s, we use a default value of 0.1 for all datasets.

Self-Supervised Methods

- **SimCLR.** We implement SimCLR using the Lightly framework [22]. We apply SimCLR to learn representations, and then fine-tune the model with the same (labeled) training data.

  For both Yearbook and FMoW-Time, we use the set of image transforms from Chen et al. [10]. Specifically, we sequentially apply the following three random augmentations: random cropping followed by resize back to the original size, color distortions, and Gaussian blur. We list all hyperparameters in Table 7.
- **SwaV.** We implement SwaV using the Lightly framework [22]. We apply SwaV to learn representations, and then fine-tune the model with the same training data. We follow the multi-crop augmentation strategy proposed by Caron et al. [7]. We use 2 views and list all hyperparameters in Table 8.

Bayesian Methods

- **SWA.** We follow the official implementation of SWA [23, 2]. We use the same learning rate as ERM and use default values for other hyperparameters.

D Results Under Eval-Stream Setting

Under Eval-Stream setting, we visualize the average performance and worst-time performance for every timestamp. For each timestamp, we calculate the average/worst performance over the evaluated time window. The results of all tasks are shown in Figure 3. The key observations are very close to the findings under Eval-Fix setting. Additionally, invariant learning approaches performs slightly better than continual learning approaches in most tasks.

Table 7: Hyperparameters for SimCLR on Yearbook and FMoW-Time.

Dataset	Yearbook	FMoW-Time
Prob. Color Jitter	0.8	0.8
Color Jitter Strength	0.5	0.5
Min. Crop Scale	0.08	0.08
Prob. Grayscale	0.2	0.2
Kernel Size	0.1×32	0.1×224
Prob. Vertical Flip	0.5	0
Prob. Horizontal Flip	0.5	0.5
Prob. Rotation (+90)	0.0	0.5
Embedding Dim.	128	128
No. SSL Iters.	2700	1500
No. Finetune Iters.	300	1500

Table 8: Hyperparameters for SwaV on Yearbook and FMoW-Time.

Dataset	Yearbook	FMoW-Time
No. Views	2	2
Crop Sizes	224, 96	224, 96
No. Crops	2, 6	2, 6
Min. Crop Scale	0.08, 0.05	0.08, 0.05
Max. Crop Scale	1.0, 0.14	1.0, 0.14
Prob. Horizontal Flip	0.5	0.5
Prob. Color Jitter	0.8	0.8
Color Jitter Strength	0.8	0.8
Prob. Grayscale	0.2	0.2
Embedding Dim.	128	128
No. Prototypes	32	1024
No. SSL Iters.	2700	1500
No. Finetune Iters.	300	1500

Table 9: Hyperparameters for EWC and SI baselines on all WildT datasets.

Dataset	EWC λ_e	SI λ_s
Yearbook	0.5	0.1
FMoW-Time	0.5	0.1
MIMIC-IV-Mort	1.0	0.1
MIMIC-IV-Readmit	0.5	0.1
HuffPost	0.5	0.1
arXiv	0.5	0.1

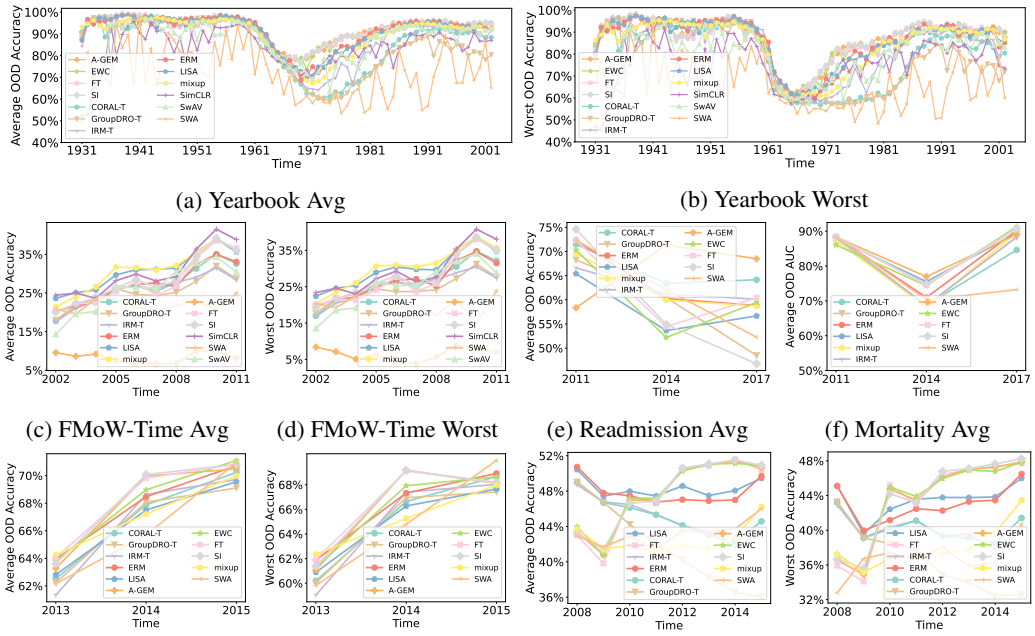
Under the Eval-Stream setting, we further explain why continual learning approaches fail to improve over other baselines in the Eval-Stream setting from the following two reasons: (1) Most existing continual learning approaches focus on backward transfer (i.e., catastrophic forgetting). In Wild-Time, we focus on forward transfer, and evaluate performance on future timestamps (i.e., temporal robustness); (2) For continual learning approaches that also focus on forward transfer (e.g., A-GEM), most of these approaches only show improvements on manually delineated sets of tasks with artificial temporal variations (e.g., Split CUB, Split CIFAR), but are not evaluated on benchmarks with natural temporal distribution shifts, such as Wild-Time. Analogously, we note that invariant learning approaches show improvements in artificial datasets (e.g., ColoredMNIST, Waterbirds [41]), but fail to outperform ERM in benchmarks with natural distribution shifts, e.g., WILDS [30].

E Additional Experiments under Eval-Fix Setting

E.1 Standard Split vs. Mixed Split

We verify that the performance gap between ID and OOD timestamps are not caused by the difficulty of examples from OOD timestamps. First, we analyze the effect of the difficulty of OOD examples. We use two kinds of data splits – standard split and mixed split. In the standard split, the model is trained on timestamps before the split time and then evaluated on examples from future timestamps. In the mixed split, the training data is merged from all timestamps, and the model is evaluated on the original OOD examples. We report the results in Table 10 and observe large performance gaps between standard split and mixed split on all Wild-Time tasks. The observation verifies that the performance gaps between ID and OOD are not caused by the difficulty.

For each Wild-Time dataset, we plot the label distributions over time in Figure 4. We observe that the label distributions change over time for all Wild-Time datasets, as this is a naturally-occurring shift that we aim to tackle with the Wild-Time benchmark.



(a) Yearbook Avg (b) Yearbook Worst

(c) FMoW-Time Avg (d) FMoW-Time Worst (e) Readmission Avg (f) Mortality Avg

(g) : Huffpost Avg (h) : Huffpost Worst (i) arXiv Avg (j) arXiv Worst

Figure 3: Results under Eval-Stream Setting. Note that for MIMIC-Readmission (e) and MIMIC-Mortality (f), we only report the average timestamp performance as we only evaluate on the next timestamp, which is a three-year block.

Table 10: Performance drops of ERM with different splits under Eval-Fix setting. In the standard split, we train the model on timestamps before the split timestamp, and evaluate the model in the future timestamps. In the mixed split, we merge the training data from all timestamps, and evaluate the model on the original OOD set. The large gap between standard split and mixed split indicates that the performance drops between ID and OOD shown in Table 1 in the main paper are not caused by the difficulty of the examples from OOD timestamps.

Dataset	Standard Split		Mixed Split	
	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
Yearbook	81.98%	69.62%	94.57%	78.57%
FMoW-Time	54.07%	46.00%	57.80%	52.00%
MIMIC-Mortality	72.89%	65.80%	91.00%	88.67%
MIMIC-Readmission	61.33%	59.46%	57.18%	54.84%
Huffpost	70.42%	68.71%	78.11%	76.87%
arXiv	45.94%	44.09%	52.12%	50.57%

E.2 Temporal Adaptation of Invariant Learning Methods

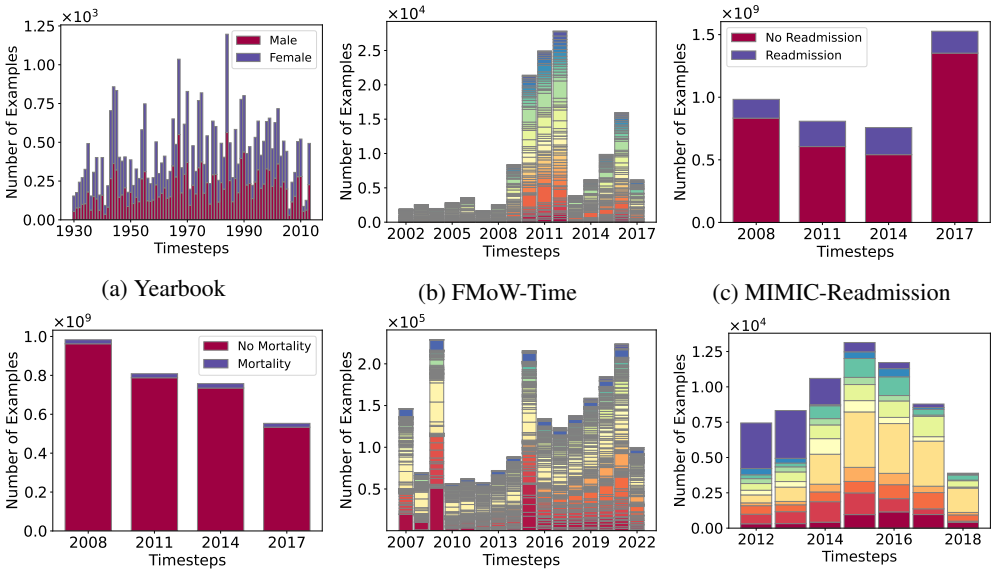
In this section, we provide additional analysis for temporal adaptation, including the analysis of the effectiveness of temporal adaptation, the effect of time window size, and the comparison between overlapping and non-overlapping substreams.

E.2.1 Temporal Adaptation Improves Performance

We compare the temporal adapted invariant learning approaches with the original approaches. The results are listed in Table 11. We observe that temporal adaptation indeed shows improved performance over vanilla invariant learning approaches, verifying the efficacy of the proposed strategy.

E.2.2 Effect of Time Window Size

We include an ablation in which we report the performance of CORAL-T, GroupDRO-T, and IRM-T when the time window size L (defined in Section 4 of the main paper) is reduced. We report baseline



(a) Yearbook (b) FMoW-Time (c) MIMIC-Readmission

(d) MIMIC-Mortality (e) arXiv (f) Huffpost

Figure 4: Label distributions of all Wild-Time classification datasets over time. Note that we include legends for datasets with less than 8 classes due to space limitations.

Table 11: The vanilla versus temporal adapted invariant learning performance of each method evaluated on Wild-Time under the Eval-Fix setting.

	Yearbook (Accuracy (%) ↑)		FMoW-Time (Accuracy (%) ↑)		MIMIC-Readmission (Accuracy (%) ↑)	
	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
GroupDRO	76.19 (1.58)	59.61 (1.09)	42.54 (0.39)	35.17 (1.08)	55.69 (3.53)	54.18 (2.79)
GroupDRO-T	**77.06 (1.67)**	**60.96 (1.83)**	**43.87 (0.55)**	**36.60 (1.25)**	**56.12 (4.35)**	**53.12 (4.41)**
CORAL	76.29 (1.75)	58.54 (2.91)	48.96 (0.23)	40.17 (0.89)	56.62 (3.21)	54.08 (3.50)
CORAL-T	**77.53 (2.15)**	**59.34 (1.46)**	**49.43 (0.38)**	**41.23 (0.78)**	**57.31 (4.45)**	**54.69 (4.36)**
IRM	77.08 (2.05)	63.79 (1.27)	**45.25 (1.01)**	**38.73 (0.69)**	**57.89 (2.76)**	**53.02 (2.53)**
IRM-T	**80.46 (3.53)**	**64.42 (4.38)**	45.00 (1.18)	37.67 (1.17)	56.53 (3.36)	52.67 (5.17)
	MIMIC-Mortality (AUC (%) ↑)		HuffPost (Accuracy (%) ↑)		arXiv (Accuracy (%) ↑)	
	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
GroupDRO	74.93 (3.17)	70.58 (3.46)	68.33 (0.88)	67.42 (1.27)	37.37 (1.09)	36.09 (0.93)
GroupDRO-T	**76.88 (4.74)**	**71.40 (6.84)**	**69.53 (0.54)**	**67.68 (0.78)**	**39.06 (0.54)**	**37.18 (0.52)**
CORAL	76.83 (2.70)	64.62 (5.58)	**70.64 (0.43)**	67.82 (1.16)	40.82 (1.16)	38.16 (0.62)
CORAL-T	**77.98 (2.57)**	**64.81 (10.8)**	70.05 (0.63)	**68.39 (0.88)**	**42.32 (0.60)**	**40.31 (0.61)**
IRM	**76.25 (5.87)**	69.91 (6.02)	**71.69 (1.33)**	**69.49 (1.46)**	35.07 (0.55)	**34.22 (0.63)**
IRM-T	76.16 (6.32)	**70.64 (8.99)**	70.21 (1.05)	68.71 (1.13)	**35.75 (0.90)**	33.91 (1.09)

results in Table 12. We found that reducing L marginally worsens the performance of invariant learning baselines.

E.2.3 Non-Overlapping Time Windows

In the proposed temporal adaptations of the invariant learning methods (CORAL-T, GroupDRO-T, IRM-T), we use overlapping time windows to capture the gradual temporal distribution shift. Here, we run all invariant learning baselines using non-overlapping windows, and report the OOD performance in Table 13. For the Yearbook, Huffpost, arXiv, MIMIC-Mortality, and MIMIC-Readmission, invariant learning baselines generally obtained better performance using overlapping time windows.

Table 12: Performance of the temporally adapted invariant learning baselines when decreasing the length of the time windows, L. We evaluate under the Eval-Fix setting and report the average and standard deviation (value in parentheses), computed over three random seeds. For Yearbook, performance worsens when L is reduced, but improves for FMoW-Time.

	Yearbook (Accuracy (%) ↑)			FMoW-Time (Accuracy (%) ↑)		
	L	OOD Avg.	OOD Worst	L	OOD Avg.	OOD Worst
GroupDRO-T	5	**77.06 (1.67)**	**60.96 (1.83)**	3	43.87 (0.55)	36.60 (1.25)
	4	72.84 (3.04)	56.05 (0.75)	2	**45.17 (0.83)**	**36.97 (1.00)**
	2	73.42 (2.29)	56.99 (2.54)	n/a	n/a	n/a
CORAL-T	5	**77.53 (2.15)**	**59.34 (1.46)**	3	49.43 (0.38)	41.23 (0.78)
	4	77.09 (1.56)	59.17 (1.89)	2	**49.67 (0.49)**	**41.63 (0.32)**
	2	76.92 (1.07)	59.26 (1.38)	n/a	n/a	n/a
IRM-T	5	**80.46 (3.53)**	**64.42 (4.38)**	3	45.00 (1.18)	37.67 (1.17)
	4	79.56 (3.12)	63.70 (3.85)	2	**48.50 (0.10)**	**40.63 (0.59)**
	2	79.47 (2.69)	63.65 (3.91)	n/a	n/a	n/a

Table 13: Performance of CORAL-T, GroupDRO-T, and IRM-T baselines when trained on non-overlapping time substreams. OL: Overlapping; NOL: Non-overlapping

		Yearbook (Accuracy (%) ↑)		FMoW-Time (Accuracy (%) ↑)		MIMIC-Readmission (Accuracy (%) ↑)	
		OOD Avg.	OOD Worst	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
CORAL-T	OL	**77.53 (2.15)**	**59.34 (1.46)**	49.43 (0.38)	41.23 (0.78)	**57.31 (4.45)**	**54.69 (4.36)**
	NOL	75.97 (0.63)	57.47 (0.29)	**49.93 (0.64)**	**42.13 (0.96)**	54.86 (2.93)	51.44 (4.63)
GroupDRO-T	OL	**77.06 (1.67)**	**60.96 (1.83)**	43.87 (0.55)	36.60 (1.25)	**56.12 (4.35)**	**53.12 (4.41)**
	NOL	76.94 (1.87)	58.58 (1.82)	**48.67 (0.57)**	**45.50 (0.62)**	53.96 (3.03)	50.47 (4.43)
IRM-T	OL	**80.46 (3.53)**	**64.42 (4.38)**	45.00 (1.18)	37.67 (1.17)	**56.53 (3.36)**	**52.67 (5.17)**
	NOL	77.21 (2.34)	59.44 (1.72)	**49.67 (0.40)**	**42.50 (1.08)**	54.31 (3.67)	51.08 (5.23)
		MIMIC-Mortality (AUC (%) ↑)		HuffPost (Accuracy (%) ↑)		arXiv (Accuracy (%) ↑)	
		OOD Avg.	OOD Worst	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
CORAL-T	OL	**77.98 (2.57)**	**64.81 (10.8)**	**70.05 (0.63)**	**68.39 (0.88)**	**42.32 (0.60)**	**40.31 (0.61)**
	NOL	71.57 (11.7)	65.77 (15.3)	68.11 (1.40)	66.94 (1.50)	42.07 (0.72)	40.10 (0.72)
GroupDRO-T	OL	**76.88 (4.74)**	**71.40 (6.84)**	**69.53 (0.54)**	**67.68 (0.78)**	**39.06 (0.54)**	**37.18 (0.52)**
	NOL	72.78 (10.7)	67.40 (14.1)	68.41 (0.41)	67.26 (0.49)	36.07 (1.35)	33.98 (1.46)
IRM-T	OL	**76.17 (6.32)**	**70.64 (8.99)**	**70.21 (1.05)**	**68.71 (1.13)**	35.75 (0.90)	33.91 (1.09)
	NOL	73.08 (9.99)	67.69 (13.1)	69.58 (0.79)	68.16 (0.64)	**38.85 (0.44)**	**36.86 (0.42)**

Since, on the aggregate, using overlapping time windows resulted in better performance, we keep the results using non-overlapping windows in Table 2 of the main paper.

E.3 Effect of Model Backbones

We investigate the effect of different models backbones on a Wild-Time image dataset (FMoW-Time) and text dataset (arXiv). Specifically, we use ResNet18 and ResNet50 backbones for FMoW-Time, and BERT and ALBERT backbones for arXiv. We report the performance of ERM and two representative invariant learning and continual learning approaches – LISA and Fine-tuning – under the Eval-Fix setting in Table 14.

These results are consistent with our findings that neither invariant learning nor continual learning approaches make models more robust to temporal distribution shift, even with different backbones.

E.4 Reducing the Number of Training Examples

We analyze the performance of all baselines when reducing the number of training examples. Specifically, under the Eval-Fix setting, we randomly allocate 30% of the data at each training timestamp as training, rather than 90% in our original results (c.f., Table 2 in the main paper). We report all

Table 14: Performance comparison w.r.t. Different backbones.

	Backbone	ERM	Fine-tuning	LISA
FMoW-Time	ResNet18	47.95 (0.39)	40.95 (0.50)	47.59 (0.36)
	ResNet50	52.91 (0.46)	45.68 (0.79)	53.17 (0.85)
	DenseNet101	54.07 (0.25)	44.22 (0.56)	52.33 (0.42)
arXiv	DistilBERT	45.94 (0.97)	50.31 (0.39)	47.82 (0.47)
	BERT	47.51 (1.20)	50.99 (0.52)	49.05 (1.01)
	ALBERT	45.25 (0.65)	49.76 (0.69)	46.01 (0.52)

results in Table 15. We observe that ERM still outperforms invariant learning and continual learning approaches, corroborating our findings in the main paper.

F Datasets without Gradual Temporal Distribution Shifts

In this section, we discuss two additional datasets that were not included in Wild-Time. These datasets do not satisfy the criteria discussed in Section 2.1.

F.1 Drug-BA

F.1.1 Dataset Setup

Problem Setting. The task is predicting the binding affinity of candidate drugs to their target molecules. The input x contains molecular information of both the drug and target molecules, and the label y is the binding affinity value.

Data. The Therapeutics Data Commons (TDC) benchmark (MIT license). TDC offers the BindingDB dataset, which was curated from BindingDB, a public database that features drug-target binding affinities collected from a variety of sources, including patents, journals, and assays. Each entry in BindingDB consists of a small molecule and the corresponding target protein. We exclude data from the year 2021 in the original TDC benchmark as 2021 includes only one month's worth of data.

For Eval-Fix, we use the first 4 years ($2013 - 2016$) for training and allocate 4 years ($2017 - 2020$) for testing. For streaming evaluation (Eval-Stream), we treat each year as a single timestamp. We provide the number of examples allocated to ID Train, ID Test, and OOD Test for each timestamp in Table 16.

Evaluation Metrics. We use Pearson Correlation Coefficient (PCC), which measures the amount of linear correlations between the true values and the predicted values, to evaluate model performance in predicting drug-target binding affinity. Eval-Stream evaluates performance across the next 3 years, which represents 37.5% of all timestamps in the entire dataset.

F.1.2 Baseline Results and Analysis

Experimental Setup. We use the DeepDTA model from [39], which achieves state-of-the-art performance on drug target binding affinity prediction by using CNNs to construct high-level representations of a drug and a target. We use the Adam optimizer with a learning rate of 2×10^{-5} or 5×10^{-5} (for different baselines) and batch size of 256. Baselines were trained for 5000 iterations under the Eval-Fix setting and for 500 iterations under the Eval-Stream setting. In terms of hyperparameters for CORAL, GroupDRO and IRM, the CORAL penalty, IRM penalty, learning rate, the number of substreams and the size of substreams are set as 0.9, 10^{-3}, 5×10^{-5}, 3, 2, respectively. Notice that LISA is only applicable to classification problem, thus we do not evaluate LISA on Drug-BA.

Results. In Drug-BA, similar to Table 10, we reported the results of standard splits and mixed splits and the results of Eval-Stream and Eval-Fix in Figure 5. First, though the performance comparison between standard split and mixed split in the top table of Figure 5, we observe a significant drop between them, where OOD average performance drops from 0.724 (mixed split) to

Table 15: Performance of all baselines when reducing the amount of training data. We randomly allocate 30% of the data at each timestamp to training, rather than 90% in our original benchmark.

	Yearbook (Accuracy (%) ↑)			FMoW-Time (Accuracy (%) ↑)		
	ID Avg.	OOD Avg.	OOD Worst	ID Avg.	OOD Avg.	OOD Worst
Fine-tuning	46.29 (1.17)	52.00 (5.00)	44.10 (2.15)	40.30 (0.14)	39.34 (0.57)	30.91 (0.29)
EWC	45.50 (0.00)	48.84 (0.01)	42.86 (0.01)	40.89 (0.66)	39.99 (0.66)	31.24 (0.43)
SI	49.42 (6.73)	47.03 (17.2)	45.52 (4.61)	41.03 (0.19)	39.65 (0.53)	30.91 (0.29)
A-GEM	45.50 (0.00)	46.98 (3.57)	44.95 (3.45)	40.52 (0.56)	39.60 (0.41)	31.05 (0.38)
ERM	93.96 (1.72)	**77.05 (5.13)**	**60.72 (2.92)**	51.74 (0.77)	**50.63 (0.56)**	40.29 (0.94)
GroupDRO-T	77.56 (11.5)	60.45 (7.10)	47.03 (7.99)	40.16 (0.68)	39.52 (0.86)	31.62 (0.29)
mixup	92.88 (2.35)	77.31 (2.60)	61.56 (3.00)	54.23 (0.26)	53.77 (0.23)	**43.04 (0.77)**
LISA	92.51 (4.03)	74.17 (5.22)	57.39 (1.77)	51.34 (0.14)	50.76 (0.83)	40.12 (1.42)
CORAL-T	75.35 (19.7)	59.66 (10.1)	44.00 (10.58)	44.15 (0.93)	43.85 (0.46)	34.59 (0.25)
IRM-T	77.04 (17.7)	60.45 (7.10)	47.03 (7.99)	45.19 (0.94)	44.11 (1.23)	36.26 (1.36)
	MIMIC-Readmission (Accuracy (%) ↑)			MIMIC-Mortality (AUC (%) ↑)		
	ID Avg.	OOD Avg.	OOD Worst	ID Avg.	OOD Avg.	OOD Worst
Fine-tuning	74.22 (2.96)	64.20 (3.73)	62.33 (5.25)	87.92 (0.92)	59.20 (0.71)	50.00 (1.35)
EWC	74.49 (1.41)	66.75 (0.99)	65.93 (1.26)	87.94 (0.08)	60.07 (2.45)	51.21 (3.11)
SI	74.22 (2.96)	64.20 (3.73)	62.33 (5.25)	87.92 (0.92)	59.20 (0.71)	50.00 (1.35)
A-GEM	80.58 (0.14)	**69.90 (0.01)**	**68.48 (0.01)**	70.27 (17.5)	53.68 (3.76)	48.00 (1.83)
ERM	70.49 (2.47)	55.28 (2.54)	51.69 (5.47)	89.53 (0.82)	71.06 (7.63)	65.76 (10.2)
GroupDRO-T	74.36 (2.65)	59.90 (15.3)	54.92 (21.3)	89.48 (0.85)	**73.28 (7.58)**	68.25 (9.97)
mixup	71.82 (3.61)	41.57 (1.12)	30.29 (0.00)	89.48 (1.14)	71.32 (8.55)	65.65 (11.4)
LISA	67.50 (2.22)	40.48 (0.68)	30.29 (0.00)	90.01 (0.32)	73.37 (10.5)	**68.97 (14.5)**
CORAL-T	74.48 (1.72)	45.00 (4.50)	34.58 (7.05)	89.12 (1.43)	71.55 (10.4)	66.01 (13.6)
IRM-T	74.36 (1.90)	52.00 (14.4)	44.24 (20.0)	88.24 (1.59)	73.13 (10.1)	68.66 (13.5)
	HuffPost (Accuracy (%) ↑)			arXiv (Accuracy (%) ↑)		
	ID Avg.	OOD Avg.	OOD Worst	ID Avg.	OOD Avg.	OOD Worst
Fine-tuning	13.11 (1.03)	14.12 (3.27)	12.89 (2.46)	50.34 (0.13)	**48.88 (0.26)**	**46.72 (0.25)**
EWC	13.26 (1.27)	13.65 (1.51)	12.37 (1.03)	50.31 (0.17)	48.56 (0.05)	46.38 (0.11)
SI	13.06 (1.05)	14.22 (3.26)	12.95 (2.44)	50.35 (0.13)	**48.88 (0.25)**	**46.72 (0.26)**
A-GEM	13.07 (0.69)	**15.53 (2.18)**	**13.43 (2.03)**	50.36 (0.18)	48.79 (0.32)	46.53 (0.39)
ERM	15.86 (1.45)	12.32 (2.64)	11.32 (2.08)	53.55 (0.21)	46.07 (0.53)	44.16 (0.50)
GroupDRO-T	14.23 (1.05)	11.82 (1.07)	11.03 (0.76)	50.01 (0.03)	39.71 (0.63)	37.79 (0.65)
mixup	15.49 (0.92)	13.35 (1.45)	11.92 (1.08)	52.66 (0.13)	45.98 (0.47)	44.00 (0.45)
LISA	14.95 (0.68)	13.26 (3.62)	11.93 (2.54)	49.17 (0.43)	47.66 (0.27)	45.71 (0.30)
CORAL-T	16.56 (0.56)	13.15 (5.17)	11.82 (4.29)	52.60 (0.06)	42.72 (0.27)	40.72 (0.24)
IRM-T	14.06 (0.65)	11.39 (0.41)	11.05 (0.47)	46.20 (0.12)	35.85 (0.70)	34.14 (0.75)

0.357 (standard split). Second, the performance per test time in Figure 5(b) further indicates such a sudden performance drop between Oracle ID and ERM in 2017. These results suggest that the Drug-BA dataset violates our criterion about gradual temporal distribution shifts. Thus, we exclude it in the official Wild-Time benchmark.

F.1.3 Broader Context

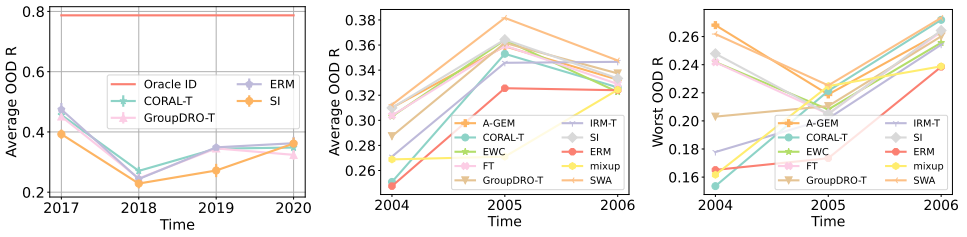
Drug discovery brings new candidate medications to potentially billions of people, allowing people to live longer and healthier lives. Traditional methods of drug discovery are via high-throughput, wet-lab experiments [21], which are expensive, time-consuming, and limited in their ability to search over large sets of drug candidates. Virtual screening is a computational pre-screening process in which the binding activity of a drug candidate with the target protein of a disease is predicted [8, 43]. Recently, there has been a surge of interest in applying machine learning to virtual screening, which can reduce costs and increase the search space to avoid missing potential drug candidates. Recent binding activity prediction models investigate binding pairs between existing compounds and target proteins [34, 39, 49, 50]. In practice, new target proteins or new classes of compounds appear over time, requiring machine learning models that are robust to subtle domain shifts across time.

Table 16: Data subset sizes for the Drug-BA task.

Year	ID Train	ID Test	OOD Test
2013	9,121	2,281	11,402
2014	16,148	4,038	20,186
2015	24,251	6,063	30,314
2016	23,095	5,774	28,869
2017	41,203	10,301	51,504
2018	32,924	8,231	41,155
2019	33,607	8,402	42,009
2020	5,557	1,390	6,947
Eval-Fix split	72,615	18,156	141,615

Dataset	Standard Split		Mixed Split	
	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
Drug-BA	0.357	0.244	0.724	0.710

(a) Performance drops of ERM with different splits on Drug-BA dataset.



(b) Eval-Fix (c) Drug-BA Avg (Eval-Stream) (d) Drug-BA Worst (Eval-Stream)

Figure 5: Results on Drug-BA. (b) out-of-distribution performance per test timestamp under Eval-Fix setting; (b) (c): results under Eval-Stream setting.

F.2 Precipitation

F.2.1 Dataset Setup

Problem Setting. The task is classifying the precipitation level of a region. The input x is tabular data consisting of 123 meteorological features (1 categorical feature and 122 continuous features). The label y is one of 9 precipitation classes.

Data. Precipitation is based on the Shifts Precipitation Prediction dataset [33] (Apache-2.0 license), which collected and processed tabular Precipitation data from the Yandex Precipitation Service to provide a domain shift benchmark for two tasks: temperature prediction (scalar regression) and precipitation classification (multi-class classification). The Shifts Precipitation Prediction dataset contains 10 million 129-column entries, consisting of 123 heterogeneous meteorological features, 4 meta-data attributes (e.g., time, latitude, longitude, and climate type), and 2 targets (temperature and precipitation class). The data is distributed uniformly between September 1, 2018 to September 1, 2019 and is partitioned by both time and climate type.

We use a subset of the original Shifts Precipitation Prediction dataset, using measurements taken from October 2018 - August 2019. The Precipitation dataset consists of 123 heterogeneous meteorological features, 1 target (precipitation class), and 1 metadata attribute (time). We partition the dataset by month. Our fixed time split (Eval-Fix) uses data from October 2018 - April 2019 (7 months) for ID, and data from May 2019 - August 2019 (4 months) for OOD. For streaming evaluation (Eval-Stream), we treat each month as a single timestamp. We allocate 10% of the data at each timestamp for test, and the rest for training. For OOD testing, all samples are used. Table 17 lists the number of examples allocated to ID Train, ID Test, and OOD Test for each timestamp.

The Shifts Precipitation Prediction dataset is provided as a CSV file. We ignore the latitude, longitude, and climate type metadata and filter out samples where at least one of the meteorological features is NaN. We shuffle Precipitation measurements in each month, and randomly select 10% of the

Table 17: Data subset sizes for the Precipitation task.

Month	ID Train	ID Test	OOD Test
Sep 2018	698,134	77,570	775,705
Oct 2018	714,265	79,362	793,628
Nov 2018	613,885	68,209	682,095
Dec 2019	707,274	78,586	785,861
Jan 2019	739,325	82,147	821,473
Feb 2019	665,745	73,971	739,717
Mar 2019	729,527	81,058	810,586
Apr 2019	691,366	76,818	768,185
May 2019	673,058	74,784	747,843
Jun 2019	548,793	60,976	609,770
Jul 2019	680,152	75,572	755,725
Aug 2019	681,035	75,670	756,706
Eval-Fix split	4,868,155	540,903	3,638,229

measurements in each month as test-ID and allocate the remaining 90% for training. For OOD testing, all samples in each month are used.

Evaluation Metrics. We evaluate models by their average and worst-time OOD accuracies. The former measures the model's ability to generalize across time, while the latter additionally measures model robustness to trends in seasonal Precipitation patterns.

Eval-Stream evaluates performance across the next 4 months, which represents 33.3% of all timestamps in the entire dataset and tests a model's robustness to shifting meteorological measurements from seasonal Precipitation changes.

F.3 Results

Experimental Setup. We follow [33] and use a FTTransformer [28], which is well-suited for deep learning with tabular data. We use all default architecture settings for the FTTransformer, except that the deep MLP in our FTTransformer has 2 layers, each of size 32 units, and uses LeakyReLU activation.

We use the Adam optimizer with a fixed learning rate of 10^{-3} and train with a batch size of 128. Baselines were trained for 5000 iterations under the Eval-Fix setting and for 500 iterations under the Eval-Stream setting. In terms of hyperparameters for CORAL, GroupDRO and IRM, the CORAL penalty, IRM penalty, learning rate, the number of substreams and the size of substreams are set as 0.9, 1.0, 10^{-3}, 3, 4, respectively.

Results. Similar to Table 10, we reported the results of standard splits and mixed splits and the results of Eval-Stream and Eval-Fix in Figure 6. According to the performance between different splits, we can not observe clear performance gaps between standard split and mixed split. Thus, precipitation dataset violates our first dataset selection criterion, and we decide not to include this dataset in the Wild-Time benchmark.

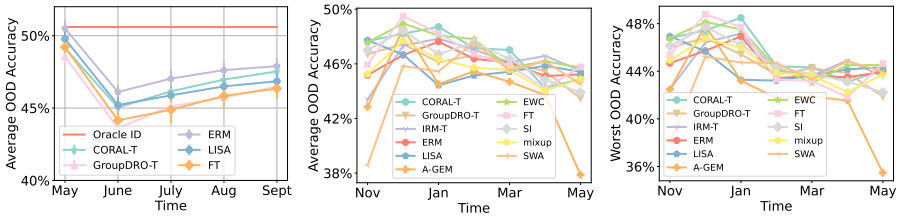
F.3.1 Broader Context

Precipitation forecasting enhances public health, safety, and economic prosperity. Extreme Precipitation warnings can save lives and reduce property damage. Forecasts on temperature and precipitation are crucial to agriculture, and hence to traders on commodity markets. On a daily basis, many people use Precipitation forecasts on a daily basis. Precipitation forecasting comprises a large part of the economy: the United States alone spent 5.1 billion on Precipitation forecasting in 2009, resulting in benefits estimated to be 6 times as much [40].

The Precipitation dataset, which contains heterogeneous tabular data, exhibits data that changes over time due to seasonal changes in Precipitation patterns. In addition, the distribution of the measurement locations are distributed non-uniformly across the planet. Certain climate regions, such as the polar caps, are under-represented, presenting further challenges [33].

Dataset	Standard Split		Mixed Split	
	OOD Avg.	OOD Worst	OOD Avg.	OOD Worst
Precipitation	46.08%	44.15%	47.56%	45.51%

(a) Performance drops of ERM with different splits on Precipitation dataset.



(b) Eval-Fix (c) PCPN Avg (Eval-Stream) (d) PCPN Worst (Eval-Stream)

Figure 6: Results on Precipitation. (b) out-of-distribution performance per test timestamp under Eval-Fix setting; (b) (c): results under Eval-Stream setting.

G User Guide and Maintenance Plan

G.1 User Guide of Wild-Time

Licenses. The Wild-Time datasets and baselines are freely available for research purposes. Though Drug-BA and Precipitation are not included in the formal Wild-Time benchmark, we still include these datasets in the Wild-Time package. All code for Wild-Time is available under the MIT license. We list the licenses for each Wild-Time dataset below:

- Yearbook: MIT License
- FMoW-Time: The Functional Map of the World Challenge Public License
- MIMIC-IV (Readmission and Mortality): PhysioNet Credentialed Health Data License 1.5.0
- Drug-BA: MIT License
- Precipitation: CC BY-NC 4.0
- Huffpost: CC0: Public Domain
- arXiv: CC0: Public Domain

Hosting Platform. We will use GitHub as the hosting platform of code. We provide (1) detailed data preprocessing scripts to help users process the data from scratch, and (2) preprocessed data from each curated dataset except MIMIC-IV.

Dependencies. Wild-Time is built upon Python 3.8+, and depends on PyTorch, PyTorch Tabular, PyTorch Transformers, PyTDC, Huggingface-Hub. Additionally, it uses numpy, scipy, and scikit-learn for data manipulation.

G.2 Using the Wild-Time Package

In this section, we discuss our open-sourced Python package that provides a simple interface to use the Wild-Time benchmark. Our Python package allows the users to use our datasets with a few lines of code. In addition, users can easily construct their own datasets or baselines on top of our package. Specifically, Figure 7 shows how to use APIs to load the Wild-Time datasets and train a baseline. Beyond the current APIs, we plan to provide standardized evaluation of methods using our dataset in the future.

G.3 Maintenance Plan

Wild-Time will be maintained by the authors of this paper. The group can be contacted by raising an issue on the GitHub or by writing to the first authors. The dataset is currently hosted on Google Drive storage. The Wild-Time benchmark may be updated at the discretion of the authors. Updates may include adding more diverse baseline methods, datasets, and tasks, or updating infrastructure to improve efficiency. Updates which correct errors will replace previous versions of the datasets.

```
>>> import argparse
>>> from WildTime import dataloader, baseline_trainer
# Load the corresponding config for a specific baseline and dataset
>>> from WildTime.configs.eval_fix.configs_fmow import configs_fmow_ewc
>>> configs = argparse.Namespace(**configs_fmow_ewc)

# If you only need data, you only need the get_data method
>>> fmow_data = dataloader.getdata("fmow", configs)

# If you need to run a baseline, use the following method
>>> baseline_trainer.train(configs)
```

Figure 7: Dataset initialization and baseline training.

We welcome contributions to the Wild-Time benchmark. Other parties may update the Wild-Time benchmark by submitting a pull request on GitHub. We are releasing the Wild-Time benchmark under the open-source MIT License. We permit other parties to create new datasets from the Wild-Time benchmark, given that the changes are documented and the Wild-Time benchmark is referenced.

G.4 Author Statement

To the best of our knowledge, the released dataset and benchmark does not violate any existing licenses. However, if such a violation were to exist, the authors claim responsibility for resolving these issues.

H Discussion

H.1 Limitations

One limitation of this paper is that we do not categorize covariant shift and concept drift over time. Though, we've seen some sudden distribution shifts occur in our benchmark, we currently do not find a good way to precisely identify the reasons of sudden distribution shifts and further categorize them. We will focus on this in the next version.

H.2 Ethics Discussion

The Wild-Time benchmark includes the Yearbook dataset, which is an adaptation of the Portraits dataset [16]. The task is binary gender prediction from yearbook photos of American high schoolers. We recognize the harmful ramifications of binary gender prediction. A binary gender prediction task excludes nonbinary individuals, may misgender transgender individuals, and may reinforce problematic gender norms.

The FMoW-Time dataset, adapted from the WILDS benchmark [30], involves geographic region prediction from satellite imagery and has applications to remote sensing. We recognize the privacy and surveillance issues surrounding remote sensing. We remark that FMoW-Time uses a lower image resolution than other publicly available satellite data, such as Google Maps. We also recognize that the FMoW-Time dataset raises issues of systematic bias and fairness. Specifically, the WILDS benchmark [30] found that models performed poorly on satellite images from Africa. As remote sensing is used for development and humanitarian purposes, poor model performance in certain geographic regions can harm certain populations. These issues are discussed in more detail in the UNICEF discussion paper by Berman et al. [4].

The MIMIC dataset, adapted from the MIMIC-IV database [26], involves predicting patient mortality and readmission to the ICU. Ethical challenges associated with using artificial intelligence (AI) in healthcare include (1) informed consent to use, (2) safety and transparency, (3) fairness and algorithmic biases, and (4) data privacy [15]. The MIMIC-IV database adopted a permissive access scheme, allowing for broad reuse of data. With regards to patient privacy, we note that the MIMIC-IV database includes de-identified patient data [26]. We also note that the authors of Wild-Time followed

proper credentialing protocol to access the MIMIC-IV dataset. To protect patient confidentiality, we do not release the MIMIC dataset. Instead, we provide instructions for how users can get credentialed on PhysioNet to download the MIMIC-IV dataset and provide a script to generate the MIMIC dataset. We recognize considerations of fairness and algorithmic bias for the MIMIC task. Several studies have found that AI algorithms exhibit biases with respect to ethnicity and gender [35, 37]. Phenotype-related data in healthcare can similarly lead to biased models. This can result in incorrect diagnoses for certain subpopulations, endangering their safety. Finally, we emphasize the importance of robust and interpretable AI, especially in healthcare, where human safety is at stake. We hope that the MIMIC task can help lay the groundwork for further research in this direction. We refer readers to [25] for an in-depth discussion of the ethical issues surrounding AI in healthcare.

H.3 Comments on Designing Temporally Robust Models

In our experiments, we found that most existing approaches can not effectively mitigate natural temporal distribution shifts. We believe that there are two important aspects to consider in resolving natural distribution shift:

- **Learning changeable temporal invariance**. To build a robust model, it would be useful to learn invariance, which captures features in the data that remain invariant across different distributions. However, this is difficult to do when temporal distribution shift happens, as such invariance can also change over time, where one kind of invariance is only suitable for a specific time window. Capturing the correlations between different time windows and determining when and how to update the invariant model are crucial.
- **Leveraging supervised and unsupervised adaptation**. In addition to maintaining a temporally invariant model, adapting to new timestamps is also necessary in tackling temporal distribution shifts. Here, we can leverage labeled data from timestamps in the near past and unlabeled observations from the current timestamp to fine-tune the model. How to combine temporal invariance with supervised and unsupervised adaptation to achieve effective adaptation remains an open problem.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[2] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. *arXiv preprint arXiv:1806.05594*, 2018.

[3] Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 65–74, 2017.

[4] Gabrielle Berman, Sara de la Rosa, Tanya Accone, et al. Ethical considerations when using geospatial technologies for evidence generation. 2018.

[5] Matteo Boschini, Lorenzo Bonicelli, Pietro Buzzega, Angelo Porrello, and Simone Calderara. Class-incremental continual learning into the extended der-verse. *arXiv preprint arXiv:2201.00766*, 2022.

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15920–15930. Curran Associates, Inc., 2020.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[8] Kristy A Carpenter, David S Cohen, Juliet T Jarrell, and Xudong Huang. Deep learning and virtual drug screening. *Future medicinal chemistry*, 10(21):2557–2567, 2018.

Table 18: The in-distribution versus out-of-distribution test performance of each method evaluated on Wild-Time under the Eval-Fix setting. The average and standard deviation (value in parentheses) are computed over three random seeds. We bold the best OOD performance for each dataset.

	Yearbook (Accuracy (%) ↑)			FMoW-Time (Accuracy (%) ↑)		
	ID Avg.	OOD Avg.	OOD Worst	ID Avg.	OOD Avg.	OOD Worst
Fine-tuning	95.43 (1.65)	81.98 (1.52)	**69.62 (3.38)**	47.67 (0.81)	44.22 (0.56)	36.59 (0.98)
EWC	96.36 (0.47)	80.07 (0.22)	66.61 (1.95)	47.42 (0.64)	44.02 (0.65)	36.42 (1.56)
SI	96.40 (0.83)	78.70 (3.78)	65.18 (2.44)	47.41 (0.36)	44.25 (0.97)	37.14 (1.34)
A-GEM	97.18 (0.43)	81.04 (1.40)	67.07 (2.23)	47.09 (0.36)	44.10 (0.65)	36.02 (0.61)
ERM	97.99 (1.40)	79.50 (6.23)	63.09 (5.15)	58.07 (0.15)	**54.07 (0.25)**	46.00 (0.79)
GroupDRO-T	96.04 (0.45)	77.06 (1.67)	60.96 (1.83)	46.57 (0.57)	43.87 (0.55)	36.60 (1.25)
mixup	96.42 (0.26)	76.72 (1.35)	58.70 (1.36)	56.93 (0.50)	53.67 (0.49)	44.57 (1.02)
LISA	96.56 (0.97)	83.65 (4.61)	68.53 (5.79)	55.10 (0.26)	52.33 (0.42)	43.30 (0.75)
CORAL-T	98.19 (0.58)	77.53 (2.15)	59.34 (1.46)	52.60 (0.10)	49.43 (0.38)	41.23 (0.78)
IRM-T	97.02 (1.52)	80.46 (3.53)	64.42 (4.38)	46.60 (1.40)	45.00 (1.18)	37.67 (1.17)
SimCLR	96.11 (0.92)	78.59 (2.72)	60.15 (3.48)	46.91 (0.65)	44.76 (0.17)	37.00 (0.44)
SwaV	96.24 (0.58)	78.38 (1.86)	60.73 (1.08)	47.31 (0.46)	44.92 (0.81)	37.17 (0.52)
SWA	98.46 (0.15)	**84.25 (3.06)**	67.90 (4.34)	58.05 (0.14)	54.06 (0.23)	**46.01** (0.78)

	MIMIC-Readmission (Accuracy (%) ↑)			MIMIC-Mortality (AUC (%) ↑)		
	ID Avg.	OOD Avg.	OOD Worst	ID Avg.	OOD Avg.	OOD Worst
Fine-tuning	69.71 (10.8)	62.19 (3.71)	59.57 (4.43)	89.99 (0.98)	63.37 (1.91)	52.45 (2.64)
EWC	77.78 (0.38)	**66.40 (0.09)**	**64.69 (0.01)**	89.53 (0.65)	62.07 (1.52)	50.41 (2.03)
SI	71.28 (6.22)	62.60 (3.27)	61.13 (3.39)	89.25 (0.84)	61.76 (0.58)	50.19 (1.25)
A-GEM	73.56 (3.25)	63.95 (0.14)	62.66 (1.23)	88.74 (0.17)	61.78 (0.27)	50.40 (0.51)
ERM	73.00 (2.94)	61.33 (3.45)	59.46 (3.66)	90.89 (0.59)	72.89 (8.96)	65.80 (12.3)
GroupDRO-T	69.70 (4.71)	56.12 (4.35)	54.69 (4.36)	89.22 (0.46)	**76.88 (4.74)**	**71.40 (6.84)**
mixup	70.08 (2.14)	58.82 (4.03)	57.30 (4.77)	89.75 (1.04)	73.69 (7.83)	66.83 (11.1)
LISA	70.52 (1.10)	56.90 (0.95)	54.01 (0.92)	89.29 (0.47)	76.34 (8.94)	71.14 (12.4)
CORAL-T	70.18 (4.72)	57.31 (4.45)	54.69 (4.36)	88.77 (0.97)	77.98 (2.57)	64.81 (10.8)
IRM-T	72.33 (1.50)	56.53 (3.36)	52.67 (5.17)	89.49 (0.17)	76.17 (6.32)	70.64 (8.99)
SWA	72.62 (3.60)	59.88 (5.48)	57.68 (6.36)	89.53 (1.96)	69.53 (1.60)	60.83 (2.73)

	HuffPost (Accuracy (%) ↑)			arXiv (Accuracy (%) ↑)		
	ID Avg.	OOD Avg.	OOD Worst	ID Avg.	OOD Avg.	OOD Worst
Fine-tuning	76.79 (0.51)	69.59 (0.10)	68.91 (0.49)	51.42 (0.15)	50.31 (0.39)	48.19 (0.41)
EWC	76.26 (0.32)	69.42 (1.00)	68.61 (0.98)	51.34 (0.13)	**50.40 (0.11)**	**48.18 (0.18)**
SI	76.97 (0.30)	70.46 (0.27)	69.05 (0.52)	51.52 (0.19)	50.21 (0.40)	48.07 (0.48)
A-GEM	77.15 (0.07)	70.22 (0.50)	69.15 (0.88)	51.57 (0.18)	50.30 (0.37)	48.14 (0.40)
ERM	79.40 (0.05)	70.42 (1.15)	68.71 (1.36)	53.78 (0.16)	45.94 (0.97)	44.09 (1.05)
GroupDRO-T	78.04 (0.26)	69.53 (0.54)	67.68 (0.78)	49.78 (0.22)	39.06 (0.54)	37.18 (0.52)
mixup	80.15 (0.17)	**71.18 (1.17)**	68.89 (0.38)	51.40 (0.20)	45.12 (0.71)	43.23 (0.75)
LISA	78.20 (0.53)	69.99 (0.60)	68.04 (0.75)	50.72 (0.31)	47.82 (0.47)	45.91 (0.42)
CORAL-T	78.19 (0.31)	70.05 (0.63)	68.39 (0.88)	53.25 (0.12)	42.32 (0.60)	40.31 (0.61)
IRM-T	78.38 (0.51)	70.21 (1.05)	68.71 (1.13)	46.30 (0.53)	35.75 (0.90)	33.91 (1.09)
SWA	80.40 (0.22)	70.98 (0.05)	**69.52 (0.10)**	51.42 (0.30)	44.36 (0.77)	42.54 (0.68)

[9] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Hkf2_sC5FX`.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[11] Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[12] Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of arxiv as a dataset, 2019.

[13] Drew DeSilver. The polarization in today's congress has roots that go back decades. 2022.

[14] Roelof GA Ettema, Linda M Peelen, Marieke J Schuurmans, Arno P Nierich, Cor J Kalkman, and Karel GM Moons. Prediction models for prolonged intensive care unit stay after cardiac surgery: systematic review and validation study. *Circulation*, 122(7):682–689, 2010.

[15] Sara Gerke, Timo Minssen, and Glenn Cohen. Ethical and legal challenges of artificial intelligence-driven healthcare. In *Artificial intelligence in healthcare*, pages 295–336. Elsevier, 2020.

[16] Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1–7, 2015.

[17] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[18] Lin Lawrence Guo, Stephen R Pfohl, Jason Fries, Alistair EW Johnson, Jose Posada, Catherine Aftandilian, Nigam Shah, and Lillian Sung. Evaluation of domain generalization and adaptation on improving model robustness to temporal dataset shift in clinical medicine. *Scientific reports*, 12(1):1–10, 2022.

[19] Matthew C Hansen, Peter V Potapov, Rebecca Moore, Matt Hancher, Svetlana A Turubanova, Alexandra Tyukavina, David Thau, Stephen V Stehman, Scott J Goetz, Thomas R Loveland, et al. High-resolution global maps of 21st-century forest cover change. *science*, 342(6160): 850–853, 2013.

[20] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[21] James P Hughes, Stephen Rees, S Barrett Kalindjian, and Karen L Philpott. Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.

[22] Philipp Wirth Jeremy Prescott Malte Ebner et al. Igor Susmelj, Matthias Heller. Lightly. *GitHub. Note: https://github.com/lightly-ai/lightly*, 2020.

[23] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

[24] Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. *arXiv preprint arXiv:2110.08534*, 2021.

[25] Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of ai ethics guidelines. *Nature Machine Intelligence*, 1(9):389–399, 2019.

[26] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv. *version 0.4). PhysioNet. https://doi. org/10.13026/a3wn-hq05*, 2020.

[27] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. Mimic-iv, 2021. URL `https://physionet.org/content/mimiciv/1.0/`.

[28] Manu Joseph. Pytorch tabular: A framework for deep learning with tabular data, 2021.

[29] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[30] Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[31] Bill Yuchen Lin, Sida Wang, Xi Victoria Lin, Robin Jia, Lin Xiao, Xiang Ren, and Wen-tau Yih. On continual model refinement in out-of-distribution data streams. *arXiv preprint arXiv:2205.02014*, 2022.

[32] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1903–1911, 2017.

[33] Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark JF Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, et al. Shifts: A dataset of real distributional shift across multiple large-scale tasks. *arXiv preprint arXiv:2107.07455*, 2021.

[34] Eric J Martin, Valery R Polyakov, Xiang-Wei Zhu, Li Tian, Prasenjit Mukherjee, and Xin Liu. All-assay-max2 pqsar: activity predictions as accurate as four-concentration ic50s for 8558 novartis assays. *Journal of chemical information and modeling*, 59(10):4450–4459, 2019.

[35] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[36] Rishabh Misra and Jigyasa Grover. *Sculpting Data for ML: The first act of Machine Learning*. 01 2021. ISBN 978-0-578-83125-1.

[37] Peter A Noseworthy, Zachi I Attia, LaPrincess C Brewer, Sharonne N Hayes, Xiaoxi Yao, Suraj Kapa, Paul A Friedman, and Francisco Lopez-Jimenez. Assessing and mitigating bias in medical artificial intelligence: the effects of race and ethnicity on a deep learning model for ecg analysis. *Circulation: Arrhythmia and Electrophysiology*, 13(3):e007988, 2020.

[38] World Health Organization. *International Statistical Classification of Diseases and Related Health Problems: Alphabetical index*, volume 3. World Health Organization, 2004.

[39] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[40] R Powers and D Beede. Fostering innovation, creating jobs, driving better decisions: The value of government data. *Office of the Chief Economist, Economics and Statistics Administration, US Department of Commerce*, 2014.

[41] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *ICLR*, 2020.

[42] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[43] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43 (6):1947–1958, 2003.

[44] Tobias G. Tiecke, Xianming Liu, Amy Zhang, Andreas Gros, Nan Li, Gregory Yetman, Talip Kilic, Siobhan Murray, Brian Blankespoor, Espen B. Prydz, and Hai-Anh H. Dang. Mapping the world population one building at a time. *arXiv preprint arXiv: Arxiv-1712.05839*, 2017.

[45] Gido M van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018.

[46] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[47] Sherrie Wang, William Chen, Sang Michael Xie, George Azzari, and David B Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 12(2): 207, 2020.

[48] Chaoqi Yang, Cao Xiao, Lucas Glass, and Jimeng Sun. Change matters: Medication change prediction with recurrent residual networks. *arXiv preprint arXiv:2105.01876*, 2021.

[49] Huaxiu Yao, Long-Kai Huang, Linjun Zhang, Ying Wei, Li Tian, James Zou, Junzhou Huang, et al. Improving generalization in meta-learning via task augmentation. In *International Conference on Machine Learning*, pages 11887–11897. PMLR, 2021.

[50] Huaxiu Yao, Ying Wei, Long-Kai Huang, Ding Xue, Junzhou Huang, and Zhenhui Jessie Li. Functionally regionalized knowledge transfer for low-resource drug discovery. *Advances in Neural Information Processing Systems*, 34, 2021.

[51] Huaxiu Yao, Yu Wang, Sai Li, Linjun Zhang, Weixin Liang, James Zou, and Chelsea Finn. Improving out-of-distribution robustness via selective augmentation. In *ICML*, 2022.

[52] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017.

[53] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. 2018.