LEARNING TO STAFF FROM OPERATIONAL PROXIES: TOWARDS WIN-WIN ON-DEMAND STAFFING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-demand staffing plans labor in advance to meet changing business needs. Ideally, it achieves a win-win outcome for businesses and workers: reducing unnecessary labor-hours while meeting business needs and keeping workloads manageable. However, historical schedules record assigned staffing rather than minimum requirements, so learning from them can reproduce overstaffing. In this paper, we propose PROXYSTAFF, a general supervision framework that learns staffing requirements from paired business and scheduling records. An operational proxy relates each role's labor to supported revenue and transactions, treating required roles as complementary stages. Its inverse yields minimum staffing requirements under the proxy. These labels enable different backbone models to learn future hourly staffing needs directly from planning-time information, through training from scratch or fine-tuning pretrained models. To evaluate PROXYSTAFF, we collect two real-world datasets from a restaurant chain with 13 stores and a milk-tea chain with 8 stores. We evaluate 6 backbone models for planning 1 or 7 days in advance, reflecting managers' routine lead times for publishing staffing needs. Across multiple backbones, PROXYSTAFF more often matches human-manager business support with fewer labor-hours than learning historical assignments or forecasting business before converting it to staffing. The results show that PROXYSTAFF can consistently reduce labor-hours by about 20–30% relative to human-manager schedules across both chains and planning horizons while matching or exceeding their business support under the proxy.

1 INTRODUCTION

On-demand staffing plans labor in advance to meet changing business needs across service operations (Ernst et al., 2004; Gurvich et al., 2010). For example, restaurant service relies on chefs, kitchen assistants, and waiters working together. Ideally, staffing achieves a win-win outcome for businesses and workers: reducing unnecessary labor-hours while meeting business needs and keeping workloads manageable. Simply cutting staff can instead overload workers and compromise service, making sufficient staffing central to this goal.

Historical records lack labels for this goal. Typically, managers create schedules recording assigned rather than required staffing: scheduling four workers does not establish whether three could support the same business. Directly imitating historical assignments (Hussein et al., 2017) can therefore reproduce overstaffing, even when the model predicts those assignments accurately. We ask: *how can business and scheduling records supervise staffing that meets business needs with less labor?*

We observe that paired business records provide evidence of what different staffing levels have supported. In Figure 1 (left), kitchen assistants prepare ingredients, chefs cook meals, and waiters serve customers. Each complementary stage must support the business; more chefs cannot compensate for too few waiters. Records pair each role's labor with hourly revenue and transactions, suggesting a way to estimate each stage's support and infer its labor requirements for a given business volume.

We therefore propose PROXYSTAFF, a general supervision framework that constructs requirement labels through an *operational proxy* (Figure 1, middle and right). The proxy estimates each role's revenue and transaction support from historical records (Aragon et al., 2005). Its inverse finds each role's minimum labor supporting both signals in each historical hour. These labels supervise future

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

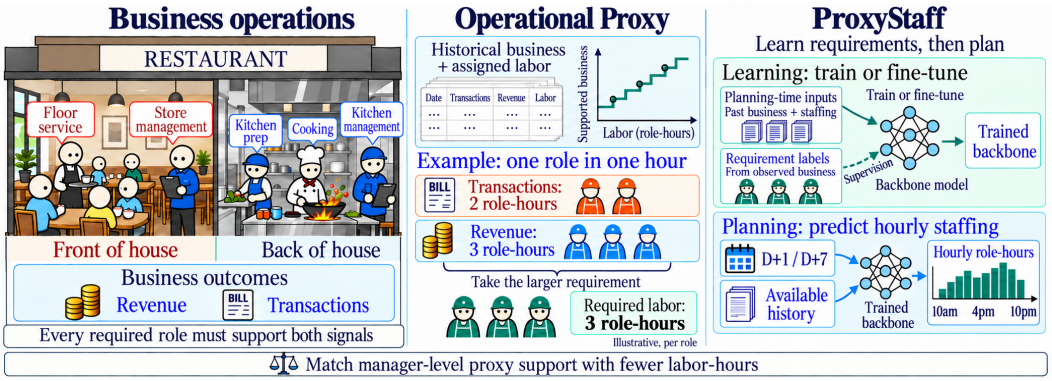


Figure 1: Overview of PROXYSTAFF. Left: complementary restaurant roles jointly support business. Middle: the proxy converts observed revenue and transactions into labels, taking each role's larger requirement to satisfy both signals. Right: these labels train or fine-tune a backbone to predict hourly requirements 1 or 7 days ahead from planning-time information. Numbers are illustrative; requirements are defined under the proxy.

hourly staffing predictions from planning-time information, using different backbones trained from scratch or fine-tuned from pretrained models.

A straightforward solution forecasts business, then converts it into staffing through the proxy. Forecasts are continuous, while staffing levels are discrete: small errors can cross staffing thresholds, while larger ones may leave decisions unchanged. Thus, better business forecasts need not improve staffing (Gneiting, 2011; Donti et al., 2017; Elmachtoub & Grigas, 2022; Mandi et al., 2024). Converting observed business into requirements before training instead aligns the prediction target with staffing. We compare both supervision frameworks using the same proxy and backbones.

To evaluate PROXYSTAFF, we collect two real-world datasets of business and staffing records from 13 restaurant stores and 8 milk-tea stores across two chains. The restaurant data include 5 roles, while the milk-tea data use a single aggregate staff role. We evaluate day-ahead and week-ahead planning (1 and 7 days in advance), reflecting managers' routine lead times for publishing staffing needs. We test whether predictions match manager support with less labor under the same proxy.

We compare 6 backbones under three supervision frameworks: learning historical staffing assignments, forecasting business followed by proxy inversion, and PROXYSTAFF. Within each backbone, the frameworks share planning-time information, fitting budgets, and the calibration procedure. PROXYSTAFF improves joint support–labor attainment over both baselines across multiple backbone models, including N-HiTS (Challu et al., 2023) and fine-tuned Chronos-2 (Ansari et al., 2025). With N-HiTS, it consistently reduces labor-hours by about 20–30% across both chains and planning horizons while matching or exceeding human-manager support under the proxy, demonstrating efficient staffing learned from records without assuming optimal historical assignments.

Our contributions are:

- **A general supervision framework for on-demand staffing.** PROXYSTAFF converts historical business and scheduling records into proxy-derived requirement labels. Different backbones learn required labor directly, without assuming optimal historical schedules.
- **Two real-world staffing datasets and a benchmark.** We collect business and staffing records from 13 restaurant and 8 milk-tea stores across two chains, with a common proxy-based evaluation of business support and labor use.
- **Improved staffing across multiple backbones.** Day-ahead and week-ahead experiments with 6 backbones show that PROXYSTAFF helps multiple models, trained from scratch or fine-tuned, match human-manager business support under the proxy with fewer labor-hours.

Reproducibility. Upon acceptance, we will release code and datasets for replication and reuse.[1]

[1] https://github.com/anonymous_repo

Table 1: Summary of the collected datasets.

Dataset	Stores	Roles	Train / Val. / Test (days)	Open hours
RESTAURANT	13	5	116 / 30 / 30	10:00–22:00
MILKTEA	8	1	61 / 15 / 16	09:00–23:00

2 PROBLEM FORMULATION

We study advance planning of hourly staffing requirements across stores within a chain. Let $i = (s, t)$ denote operating hour t at store s. Hourly revenue y_i^{rev} and transaction count y_i^{txn} characterize the observed business's monetary value and volume; a transaction is a bill or order.

Store s requires roles $\mathcal{R}_s$. A decision assigns h_{ir} role-hours to each $r \in \mathcal{R}_s$: one worker present for the full hour contributes one role-hour, so two chefs contribute two cooking role-hours. We assume that labor adds within each role and that required roles form complementary stages of a service workflow. In a restaurant, kitchen assistants prepare ingredients, chefs turn them into meals, and waiters serve customers (Figure 1, left). Sufficient staffing is needed at every stage: more chefs cannot compensate for too few assistants or waiters. We assess roles separately and require all to support the same business. This assumes non-substitutability, not statistical independence. One aggregate staff role gives the single-stage case.

Managers typically arrange multi-hour shifts. We aggregate their schedules into hourly role labor h_{ir}^{ref}, the Human-Manager Reference for evaluating predictions.

We consider two planning settings reflecting managers' routine lead times for publishing staffing needs. **D+1** plans staffing for target day D after day $D - 1$ closes, while **D+7** plans it after day $D - 7$ closes. Both settings predict one target day at a time using only information available at the planning origin. We predict hourly role requirements that support subsequent business with fewer labor-hours than the Human-Manager Reference. Employee assignment and feasible shift construction are subsequent tasks beyond our scope.

3 BENCHMARK CONSTRUCTION

Our benchmark evaluates staffing without optimal labels using two collected real-world datasets, an operational proxy, and joint support and labor metrics.

3.1 REAL-WORLD OPERATIONAL RECORDS

We collect business and staffing records from a restaurant chain and a milk-tea chain, forming two datasets named RESTAURANT and MILKTEA, respectively. Both contain manager-created schedules, revenue, and transaction counts from 2026, aligned by store, hour, and role.[2] RESTAURANT covers 13 stores over 6 months (January–June 2026) and 5 roles: *floor service*, *store management*, *kitchen preparation*, *cooking*, and *kitchen management*. We use January–April for training, May for validation, and June for testing. MILKTEA covers 8 stores across 6 cities over 3 months (May–July 2026). Staffing is recorded as a single aggregate role. We use May–June for training, July 1–15 for validation, and July 16–31 for testing. Table 1 summarizes the datasets and available calendar days in each split. Of the 13 Restaurant stores, 12 have sufficient history for model evaluation. Both datasets derive reference labor from manager-created schedules.

3.2 EVALUATING STAFFING WITHOUT OPTIMAL LABELS

Without optimal labels, we ask whether staffing matches manager-created schedules' business support with fewer labor-hours. An operational proxy links each role's labor to support, letting us compare plans on the same hours and business outcomes.

From staffing to business support. For each role, we estimate how much revenue and how many transactions a given number of role-hours can support. We pool historical observations within each

[2]The anonymized datasets, code, and benchmark metadata will be released upon paper acceptance.

chain, assuming comparable productivity for the same normalized role and nondecreasing support as its labor increases. These assumptions and the complementary-role interpretation were informed by discussions with the data providers' store managers and human-resources departments.

Let $\mathcal{A}_r$ be the allowed role-hour levels for role r, retained from Train with its minimum staffing level (floor) included. For each positive level b, we collect business-active training hours assigned no more than b role-hours and take the α-quantile of their revenue and transactions separately. The support at level a is the largest such estimate over levels up to a:

$$S_{rm}(a) = \max_{\substack{b \in \mathcal{A}_r \setminus \{0\} \\ b \le a}} Q_\alpha\big(\{y_i^m : i \in \text{Train},\ y_i^{\text{txn}} > 0,\ r \in \mathcal{R}_s,\ h_{ir}^{\text{ref}} \le b\}\big). \tag{1}$$

Here $m \in \{\text{rev}, \text{txn}\}$ identifies the business signal, and Q_α is its empirical quantile. The outer maximum makes support nondecreasing in labor. The proxy is fitted on Train and then held fixed.

We use P95 ($\alpha = .95$) to estimate busy-condition support without treating historical peaks as routine targets. Peaks may reflect temporarily stretched staffing rather than sustainable workloads. P95 leaves a margin relative to the sample maximum, motivated by avoiding staff overload; it does not guarantee manageable workloads. Percentile comparisons and details appear in Appendix A.

A store-hour is supported only if *every required role* supports *both* observed business signals:

$$z_i(\mathbf{h}_i) = \mathbf{1}[S_{rm}(h_{ir}) \ge y_i^m \text{ for all } r \in \mathcal{R}_s,\ m \in \{\text{rev}, \text{txn}\}]. \tag{2}$$

For example, sufficient cooking labor cannot compensate for insufficient floor-service labor. This criterion gives every staffing plan the same operational proxy for evaluation; it does not establish actual service capacity or imply that an hour failing the proxy had unserved customers.

Comparing support and labor. Let $\mathcal{I}$ be all evaluated store-hours and $\mathcal{I}_+ = \{i \in \mathcal{I} : y_i^{\text{txn}} > 0\}$ contain the business-active hours. For staffing $\mathbf{h}$, we measure support over $\mathcal{I}_+$:

- $\mathbf{Support_H(h)}$ **(hours):** the fraction of supported hours, with equal weight for each.
- $\mathbf{Support_R(h)}$ **(revenue):** the share of revenue recorded in supported hours.
- $\mathbf{Support_T(h)}$ **(transactions):** the share of transactions recorded in supported hours.

For $k \in \{H, R, T\}$, these metrics can be written as

$$\text{Support}_k(\mathbf{h}) = \frac{\sum_{i \in \mathcal{I}_+} w_i^k z_i(\mathbf{h}_i)}{\sum_{i \in \mathcal{I}_+} w_i^k}, \qquad w_i^H = 1, \quad w_i^R = y_i^{\text{rev}}, \quad w_i^T = y_i^{\text{txn}}. \tag{3}$$

R and T weight busy hours more than H. All require every role to support both signals; R and T do not estimate partially fulfilled orders. Chain-level scores do not guarantee support at every store.

Total labor $L = \sum_{i \in \mathcal{I}} \sum_{r \in \mathcal{R}_s} h_{ir}$ includes all evaluated hours, even those with no transactions. We minimize labor while matching or exceeding all three Human-Manager Reference support scores. Let π combine a model and its rule for converting predictions to role-hours:

$$\min_\pi L(\mathbf{h}^\pi) \quad \text{subject to} \quad \text{Support}_k(\mathbf{h}^\pi) \ge c_k, \quad k \in \{H, R, T\}. \tag{4}$$

Here c_k is the Reference score on the same evaluated hours. An improvement requires meeting all three thresholds with less labor; individual scores also show partial gains. Model calibration uses Validation; Test Reference is used only for scoring.

4 PROXYSTAFF: LEARNING STAFFING REQUIREMENTS

PROXYSTAFF constructs supervision from historical records to match manager-level support with less labor. We motivate direct requirement learning, then describe label construction and use with different backbones (Figure 1).

4.1 FROM BUSINESS PREDICTION TO STAFFING REQUIREMENTS

A straightforward approach forecasts revenue and transactions, then converts them into staffing through the proxy. Metrics such as weighted absolute percentage error (WAPE) assess business

forecasts (Gneiting, 2011), not whether staffing meets manager-level support with less labor. Lower prediction error need not satisfy this staffing objective.

To illustrate this distinction, we compare two business forecasting models: Chronos-2, a pretrained time-series model used here without fine-tuning (Ansari et al., 2025), and N-HiTS, a neural forecasting model trained on historical business records with mean absolute error (MAE) loss (Challu et al., 2023). We convert both models' revenue and transaction forecasts into staffing with the same proxy.

Table 2 reports their Restaurant day-ahead Validation results before staffing calibration. Chronos-2 zero-shot has lower revenue and transaction WAPE than N-HiTS MAE, but falls below the manager's H/R/T support scores. N-HiTS MAE meets all three thresholds while reducing labor by 24.67%. Chronos-2 saves more labor but fails the support criterion, illustrating the mismatch between forecast accuracy and the staffing objective.

Table 2: Lower business prediction error need not meet manager-level support. Restaurant day-ahead Validation with uncalibrated business forecasts. WAPE reports revenue/transaction error (%); Support reports H/R/T (%). ΔL is the labor change from Reference (%). Bold marks lower error or higher support among the two learned forecasts.

Method	WAPE (Rev./Txn.)	Support (H/R/T)	ΔL (%)
Human-Manager	—	81.05/59.84/60.68	0.00
Chronos-2 zero-shot	**27.26/20.31**	75.76/50.38/54.75	−37.86
N-HiTS MAE	33.71/24.33	**84.15/65.37/68.77**	−24.67

This motivates learning requirements directly. Historical business is already observed, so PROXYSTAFF converts it into required role-hours through the same proxy *before* training. Required labor becomes the prediction target. We next define these labels and their use with different backbones.

4.2 CONSTRUCTING STAFFING REQUIREMENT LABELS

For each historical store-hour i and required role r, we find the smallest allowed labor level that supports both the observed revenue and transaction count:

$$d_{ir} = \min\{a \in \mathcal{A}_r : S_{rm}(a) \geq y_i^m \text{ for both } m \in \{\text{rev}, \text{txn}\}\}. \tag{5}$$

This gives a staffing requirement label d_{ir} in role-hours. For example, if transactions require 2 role-hours and revenue requires 3 for the same role, its label is 3: both signals must be supported (Figure 1, middle). If no allowed level suffices, we assign a label above the role's maximum to mark unsupported business within this range (Appendix B).

Because support is nondecreasing in labor, any allowed level at or above d_{ir} supports both signals, while a smaller level does not:

$$S_{rm}(h_{ir}) \geq y_i^m \text{ for both } m \quad \Longleftrightarrow \quad h_{ir} \geq d_{ir}. \tag{6}$$

The label therefore preserves the proxy's support decision while discarding business variation that leaves the required labor unchanged. Since labor adds across complementary roles, choosing each role's minimum yields the least total labor that supports the hour under the proxy, whenever such a staffing plan exists. These labels express requirements under the proxy, without treating the manager's historical assignments as optimal.

4.3 LEARNING REQUIREMENTS WITH DIFFERENT BACKBONES

PROXYSTAFF supplies training pairs (x_i, d_{ir}), where x_i contains information available at the planning origin, including past business, past staffing, and calendar features. A backbone learns to predict required role-hours directly, written as $\widehat{d}_{ir} = f_{\theta,r}(x_i)$. Historical labels use target-day business, unavailable at planning time; target-day manager schedules are never inputs.

The same labels support training from scratch or fine-tuning pretrained models; label construction is independent of architecture and loss. Experiments match inputs, fitting budgets, objectives, and calibration across supervision frameworks (Section 5.1).

The labels align prediction with staffing without guaranteeing gains for every model.

At planning time, the backbone predicts target-day hourly requirements from available history. The supervision applies to D+1 and D+7; the horizon determines the available history. The common procedure in Section 5.1 maps predictions to allowed role-hours. Shift assignment is out of scope.

5 EXPERIMENTS

We test whether PROXYSTAFF helps models match human-manager business support with fewer labor-hours across backbones trained from scratch and fine-tuned pretrained models. Comparisons use both collected datasets and planning horizons; an hourly case study shows how staffing changes save labor.

5.1 EXPERIMENTAL SETUP

Datasets and planning settings. We use the chronological splits of Restaurant and MilkTea described in Section 3, forecasting staffing at observed stores 1 day (D+1) and 7 days (D+7) ahead. Both horizons are evaluated on the same target hours using only information available at their planning origins. Models fit Train for Validation and refit Train plus Validation for Test; the operational proxy remains fixed after fitting on Train.

Matched supervision. We assess the labels through three supervision frameworks per backbone:

- **Staffing supervision:** learn manager role-hours; map predictions to allowed staffing.
- **Business + proxy:** forecast revenue and transactions; invert the same proxy for minimum staffing.
- **PROXYSTAFF:** predict proxy-derived requirements and map to allowed staffing.

All three receive the same past business *and* staffing and calendar information, with matched fitting budgets, training objectives, and calibration procedures within each backbone. Comparing with staffing supervision tests whether inferred requirements improve on learning historical assignments. Comparing with Business + proxy tests whether learning requirements directly improves on converting predicted business; both frameworks use the same proxy.

Backbones. We evaluate XGBoost (Chen & Guestrin, 2016), MLP, residual MLP (ResMLP), N-HiTS (Challu et al., 2023), TiDE (Das et al., 2023), and fine-tuned Chronos-2 (Ansari et al., 2025), covering trees, neural models trained from scratch, and a pretrained time-series model.

Common training and calibration. All frameworks use the same quantile-regression objective (Koenker & Bassett, 1978). Validation calibration selects the prediction settings that minimize labor while meeting all three Human-Manager Reference support thresholds, following Equation 4. These settings remain fixed for Test, and predictions are converted to allowed role-hour levels. Training and calibration are shared experimental choices, separate from PROXYSTAFF's label construction. Appendix B specifies the matched controls and calibration.

Reporting. We compare backbones using Reference-normalized support and labor saving:

$$C_{\min} = \min_{k \in \{H,R,T\}} \frac{\mathrm{Support}_k(\mathbf{h})}{\mathrm{Support}_k(\mathbf{h}^{\mathrm{ref}})}, \qquad \mathrm{Saved} = 1 - \frac{L(\mathbf{h})}{L(\mathbf{h}^{\mathrm{ref}})}. \tag{7}$$

An improvement requires $C_{\min} \geq 1$ and positive Saved: all three Reference support thresholds are met with less labor. We compute both metrics per seed and report their mean and sample standard deviation over 3 seeds. Mean attainment does not imply success in every run; Appendix C reports individual support metrics and seed-level success counts.

5.2 DOES PROXYSTAFF IMPROVE STAFFING?

Table 3 compares the three supervision frameworks under the same backbones, testing whether the proposed labels improve staffing beyond historical-assignment learning and business forecasting.

Table 3: Changing supervision improves joint support–labor attainment across most backbones. Test mean ± sample standard deviation (%, 3 seeds); D+1/D+7: day/week ahead. $C_{\min}$: minimum hour/revenue/transaction support ratio to Human-Manager Reference; Saved: relative labor reduction. Blue shading pairs $C_{\min} > 100\%$ with its Saved value, indicating mean support above Reference, not an every-seed guarantee. Bold means: highest $C_{\min}$ per row and highest Saved *among shaded pairs only*. Highlighting uses unrounded means; Reference is fixed across seeds.

Backbone	Lead	Staffing target $C_{\min}$ ↑	Staffing target Saved↑	Business + proxy $C_{\min}$ ↑	Business + proxy Saved↑	PROXYSTAFF $C_{\min}$ ↑	PROXYSTAFF Saved↑
Human-Manager Reference		100.0	0.0	100.0	0.0	100.0	0.0
Restaurant							
Tree-based models							
XGBoost	D+1	93.0±0.1	8.5±0.1	98.5±0.2	33.6±0.3	**99.8**±0.3	31.0±0.3
	D+7	94.5±0.1	6.1±0.1	96.5±0.3	34.5±0.1	**98.6**±0.8	32.0±0.1
Neural models trained from scratch							
MLP	D+1	94.9±0.1	7.3±0.5	100.0±0.1	31.6±0.9	**100.9**±0.0	**28.5**±0.6
	D+7	94.5±0.2	6.2±0.5	100.9±0.2	**30.7**±0.3	**101.0**±0.2	28.9±0.5
ResMLP	D+1	94.6±0.4	7.0±0.4	99.0±1.8	32.8±1.3	**101.5**±0.7	**28.6**±0.2
	D+7	92.0±1.2	8.2±1.4	100.9±0.7	**30.3**±0.7	**101.5**±0.6	27.8±0.7
N-HiTS	D+1	88.0±5.0	17.9±4.3	93.1±5.6	34.9±3.0	**104.1**±1.1	**23.7**±1.7
	D+7	87.7±4.1	18.2±3.5	91.3±6.2	35.5±3.2	**103.7**±0.8	**24.2**±1.3
TiDE	D+1	99.5±1.4	4.1±5.4	99.3±3.0	37.4±3.2	**99.8**±0.0	2.8±2.7
	D+7	**100.1**±0.3	**4.4**±4.5	99.4±2.4	37.4±1.3	99.8±0.0	2.8±2.7
Pretrained models with full fine-tuning							
Chronos-2	D+1	99.1±0.2	-1.4±0.3	100.1±3.4	**27.3**±3.5	**104.4**±1.9	4.6±3.6
	D+7	99.4±0.5	-1.5±0.4	96.2±4.1	31.6±2.9	**104.4**±2.3	**4.4**±5.7
MilkTea							
Tree-based models							
XGBoost	D+1	99.3±0.2	7.5±0.5	102.8±0.1	**36.8**±1.5	**103.3**±0.2	32.4±0.6
	D+7	99.3±0.2	9.5±0.5	101.8±0.8	**38.4**±0.8	**102.7**±0.2	33.8±0.7
Neural models trained from scratch							
MLP	D+1	99.6±0.3	7.3±1.3	104.5±0.1	**28.7**±0.8	**105.9**±0.2	6.7±0.6
	D+7	99.2±0.7	12.9±1.2	104.1±0.2	**29.4**±0.6	**104.8**±0.3	11.1±2.1
ResMLP	D+1	95.2±0.4	19.4±1.5	99.4±1.7	36.3±2.5	**104.5**±0.5	**17.1**±0.6
	D+7	96.3±1.2	18.5±3.4	89.6±3.2	45.9±2.9	**102.3**±0.4	**22.0**±2.9
N-HiTS	D+1	98.7±1.0	7.3±3.1	**104.8**±0.1	26.3±2.3	103.5±0.6	**31.9**±0.8
	D+7	99.3±0.8	9.1±0.6	**105.3**±0.2	22.9±0.2	104.0±0.3	**30.1**±1.2
TiDE	D+1	98.6±3.2	10.8±3.7	101.9±0.3	**47.8**±0.3	**102.0**±1.0	2.3±2.6
	D+7	96.3±1.9	12.9±2.0	100.9±2.0	**49.8**±2.5	**102.4**±0.2	2.1±1.8
Pretrained models with full fine-tuning							
Chronos-2	D+1	99.9±0.7	5.3±2.7	**103.4**±1.3	**29.5**±3.3	100.4±2.1	23.5±6.0
	D+7	101.1±0.2	0.2±2.0	**103.9**±0.6	**28.1**±1.4	101.6±1.1	24.5±4.8

Learning requirements instead of historical assignments. With MLP, ResMLP, and N-HiTS, PROXYSTAFF meets Reference support on average with less labor on both chains and at both planning horizons. Staffing supervision falls below Reference support in all of these settings. This supports the claim that historical records can provide more useful supervision through inferred requirements than through the recorded assignments themselves.

Learning requirements instead of forecasting business. N-HiTS illustrates why converting observed business into labels before learning can improve staffing. On Restaurant, business supervision saves more labor but misses support; PROXYSTAFF meets support while retaining substantial savings. On MilkTea, both frameworks meet support, but PROXYSTAFF saves more labor. Across both chains and horizons, PROXYSTAFF with N-HiTS consistently saves about 20–30% relative to Reference. With the proxy held fixed, these gains come from changing the prediction target.

Applicability across backbones. The gains extend to fine-tuned Chronos-2, which meets Reference support on average with less labor on both chains and horizons. Together with the models trained from scratch, this supports using PROXYSTAFF as a general supervision framework. Its advantage is not universal: Business + proxy saves more labor for some backbones when both meet support, and TiDE does not consistently achieve the joint support–labor improvement. Benefits thus span multiple architectures but vary across backbones and seeds.

Reliability across seeds. The gains extend beyond mean attainment. For each of MLP, ResMLP, and N-HiTS, PROXYSTAFF meets all three Reference support thresholds with less labor in all 12 runs: 2 chains $\times$ 2 horizons $\times$ 3 seeds. For N-HiTS, the standard deviation of labor saving is only 0.8–1.7 percentage points across the four settings. Chronos-2 succeeds in 11 of 12 runs, with one MilkTea D+1 seed falling below Reference support. These results support consistency across training seeds on the historical splits; they do not establish reliability in independent deployments. Appendix C reports the individual support metrics and counts.

5.3 CASE STUDY: HOURLY STAFFING DECISIONS

Figure 2 illustrates how meeting role-specific requirements can reduce total labor on a selected Restaurant day with flexible-worker shifts. We compare Human-Manager Reference, PROXYSTAFF, and business forecasting followed by proxy conversion, using the same N-HiTS backbone and day-ahead settings as Table 3. The partly flat Reference curves reflect full-time labor reconstructed over the operating window; flexible workers retain their recorded intervals.

The gray boundary marks each role's minimum requirement under the frozen proxy, computed from realized revenue and transactions for evaluation; curves below it indicate insufficient staffing. PROXYSTAFF raises staffing around meals and reduces it during quiet hours. It uses fewer cooking hours than Reference throughout the day while adding servers at the lunch peak. Compared with Business + proxy, it avoids large peak-hour allocations while meeting every role's requirement. Total

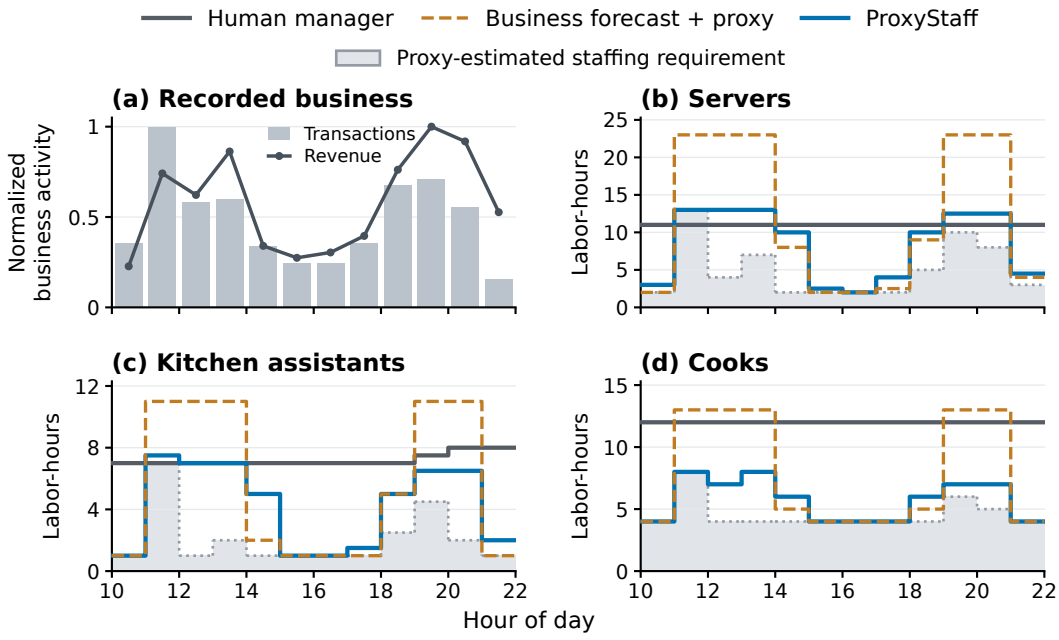


Figure 2: A restaurant day with flexible staffing (June 26, 2026). (a) Transactions and revenue, each normalized to its daily peak to withhold absolute store-level business volumes for confidentiality. (b–d) Labor-hours in each one-hour interval for servers, kitchen assistants, and cooks, using N-HiTS for both learned methods. The gray area's upper boundary is the proxy-estimated requirement from realized business: a curve at or above it satisfies that role's requirement. Store-hour support requires all five roles to satisfy their requirements, including the two management roles not plotted.

labor falls by 35.1% relative to Reference and 22.6% relative to Business + proxy. Both learned methods achieve 100% support under the proxy.

At 11:00, the manager's server allocation falls below the proxy-estimated requirement despite the recorded business activity. Employees may have temporarily absorbed a higher workload to sustain service, although the records do not directly establish overload or fatigue. PROXYSTAFF adds service labor during this hour while reducing labor elsewhere. This illustrates why recorded business alone does not establish sufficient staffing for manageable workloads, and why labor savings can coexist with adding staff where needed. This case illustrates how savings arise; Table 3 reports performance across stores and days.

6 RELATED WORK

Staffing under demand uncertainty. Staffing models address assignment constraints (Ernst et al., 2004), multi-period service and congestion (Thompson, 1993; Green et al., 2001), demand uncertainty (Gurvich et al., 2010), and load-dependent productivity (Smirnov & Huchzermeier, 2020). Warehouse staffing also uses offline reinforcement learning with simulator-based evaluation (Kujanpää et al., 2026). We study how paired business and scheduling records can supervise staffing without optimal schedules or detailed service trajectories. Our two collected datasets provide a benchmark for jointly evaluating business support and labor use under a common operational proxy.

Learning from imperfect records. Beyond action imitation (Hussein et al., 2017), T-REX learns rewards from ranked demonstrations (Brown et al., 2019); DemoDICE combines expert and imperfect demonstrations (Kim et al., 2022). PROXYSTAFF requires neither rankings nor an identified expert subset. Building on conditional-quantile frontiers (Aragon et al., 2005), it estimates operational support across complementary roles and business signals. Its inverse supplies staffing requirement labels for learning from records without imitating managers' assignments.

Decision-aligned supervision. Decision-focused learning aligns prediction with decision quality (Donti et al., 2017; Mandi et al., 2024) through decision-error surrogates (Elmachtoub & Grigas, 2022), differentiable optimization (Amos & Kolter, 2017; Agrawal et al., 2019; Wilder et al., 2019), or learned losses (Shah et al., 2022; Mandi et al., 2022). PROXYSTAFF pursues this alignment through label construction: it converts observed business into staffing requirements before training. Standard prediction losses then operate in role-hours without differentiating through a staffing optimizer; the supervision remains independent of backbone architecture.

Temporal models. Basis expansions in N-BEATS (Oreshkin et al., 2020), patching in PatchTST (Nie et al., 2023), and pretraining in TimesFM (Das et al., 2024) advance forecasting architectures. PROXYSTAFF complements these advances by changing what models learn to predict. Its labels support both training from scratch and fine-tuning pretrained models. Our comparisons within each backbone show staffing improvements across multiple architectures under matched conditions.

7 CONCLUSION AND FUTURE WORK

We presented PROXYSTAFF, a general supervision framework that converts historical business and scheduling records into staffing requirement labels through an operational proxy. This allows models to learn required labor without treating managers' assignments as optimal. We also contribute two real-world datasets collected from 13 restaurant and 8 milk-tea stores, together with a benchmark for evaluating business support and labor use. Experiments show that these labels help multiple backbones, trained from scratch or fine-tuned, match human-manager support under the proxy with fewer labor-hours. These contributions offer a path toward win-win on-demand staffing using existing operational records.

These findings remain conditional on the operational proxy. Extending hourly requirements into feasible shifts may reduce savings because minimum shift lengths couple neighboring hours (Ernst et al., 2004). Future work should test alternative proxy assumptions, extend the model beyond complementary roles to substitutable roles and coupled workflows, and assess service quality and worker well-being in prospective deployments.

ETHICS STATEMENT

Labor optimization can increase work intensity, queues, or income instability even when a proxy reports improvement. We therefore evaluate support alongside labor and distinguish observational support from service capacity or worker well-being. The inputs contain anonymized store identifiers and no employee identities; raw mappings and credentials are excluded. Commercial disclosure and data licensing govern release.

Deployment requires human oversight and constraints absent from this hourly requirement study, including breaks, minimum paid time, role qualifications, worker preferences, and predictable income. Independent service and worker-outcome monitoring should precede staffing changes. An observational proxy alone should not trigger removal of scheduled workers.

AI USE STATEMENT

The authors led the research, provided the core ideas, domain assumptions, and data, and directed the methodological decisions. Generative AI tools assisted with refining the mathematical formulation and experimental design, data preparation, implementation, result analysis, literature review, and manuscript and figure preparation. AI-assisted code and numerical results were checked through tests and reproducible evaluations. The authors take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Upon acceptance, we will release both collected datasets, code, and experiment configurations, together with instructions for reproducing the reported results. Sections 3–5 describe benchmark construction, PROXYSTAFF, and the experimental protocol. Appendix A explains the proxy percentile choice; Appendix B provides implementation and calibration details; and Appendix C reports individual support metrics and seed-level attainment.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145, 2017. URL https://proceedings.mlr.press/v70/amos17a.html.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, Mononito Goswami, Shubham Kapoor, Danielle C. Maddix, Pablo Guerron, Tony Hu, Junming Yin, Nick Erickson, Prateek Mutalik Desai, Hao Wang, Huzefa Rangwala, George Karypis, Yuyang Wang, and Michael Bohlke-Schneider. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025. doi: 10.48550/arXiv.2510.15821.

Yves Aragon, Abdelaati Daouia, and Christine Thomas-Agnan. Nonparametric frontier estimation: A conditional quantile-based approach. *Econometric Theory*, 21(2):358–389, 2005. doi: 10.1017/S0266466605050206.

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 783–792, 2019. URL https://proceedings.mlr.press/v97/brown19a.html.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.v37i6.25854.

Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016. doi: 10.1145/2939672.2939785.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with TiDE: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023. doi: 10.48550/arXiv.2304.08424. URL https://arxiv.org/abs/2304.08424.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/3fc2c60b5782f641f76bcefc39fb2392-Abstract.html.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1): 9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Andreas T. Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004. doi: 10.1016/S0377-2217(03)00095-X.

Tilmann Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106(494):746–762, 2011. doi: 10.1198/jasa.2011.r10138.

Linda V. Green, Peter J. Kolesar, and João Soares. Improving the SIPP approach for staffing service systems that have cyclic demands. *Operations Research*, 49(4):549–564, 2001. doi: 10.1287/opre.49.4.549.11228.

Itai Gurvich, James Luedtke, and Tolga Tezcan. Staffing call centers with uncertain demand forecasts: A chance-constrained optimization approach. *Management Science*, 56(7):1093–1115, 2010. doi: 10.1287/mnsc.1100.1173.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 50(2):21:1–21:35, 2017. doi: 10.1145/3054912.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=BrPdX1bDZkQ.

Roger Koenker and Gilbert Bassett, Jr. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. doi: 10.2307/1913643.

Kalle Kujanpää, Yuying Zhu, Kristina Klinkner, and Shervin Malmasi. Learning to staff: Offline reinforcement learning and fine-tuned LLMs for warehouse staffing optimization. *arXiv preprint arXiv:2603.24883*, 2026. doi: 10.48550/arXiv.2603.24883.

Jayanta Mandi, Víctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14935–14947, 2022. URL https://proceedings.mlr.press/v162/mandi22a.html.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024. doi: 10.1613/jair.1.15320.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2211.14730.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. Decision-focused learning without decision-making: Learning locally optimized decision losses. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0904c7edde20d7134a77fc7f9cd86ea2-Abstract-Conference.html.

Dmitry Smirnov and Arnd Huchzermeier. Analytics for labor planning in systems with load-dependent service times. *European Journal of Operational Research*, 287(2):668–681, 2020. doi: 10.1016/j.ejor.2020.04.036.

Gary M. Thompson. Accounting for the multi-period impact of service when determining employee requirements for labor scheduling. *Journal of Operations Management*, 11(3):269–287, 1993. doi: 10.1016/0272-6963(93)90004-9.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1658–1665, 2019. doi: 10.1609/aaai.v33i01.33011658.

A WHY USE P95 RATHER THAN HISTORICAL PEAKS?

The proxy requires at least 400 cumulative business-active observations for each retained positive role-hour level and sets $S_{rm}(0) = 0$. Support is stepwise between allowed levels. The Train-fitted proxy remains fixed during model calibration and evaluation.

We audit candidate percentiles on held-out months *within Train*: January–March to April for Restaurant and May to June for MilkTea, using the same action grid and evidence threshold. Table 4 evaluates the manager's recorded Reference labor against the corresponding realized business, not a learned policy. Each hour passes only if all roles support both signals.

A larger α credits the same labor with more business and is therefore more permissive. It is neither a joint-support probability nor a comfort label. Compared with the historical maximum, a lower percentile leaves a buffer: whenever both inverses exist on the same grid, P95 requests at least as much labor as P100 for the same target.

Table 4: Human-Manager Reference support on held-out Train months. Entries: passing-hour and associated revenue/transaction shares (%). All roles must pass both signals. Bold: selected percentile.

Percentile α	RESTAURANT	MILKTEA
.70	49.2/20.3/22.0	53.2/30.4/30.3
.80	58.7/30.0/31.1	60.3/38.9/39.1
.90	71.8/48.1/47.6	70.2/54.5/55.2
.95	**81.4/64.5/64.0**	**75.8/65.6/66.1**
1.00	95.4/98.0/97.5	81.7/83.1/83.3

P70–P90 classify much of Reference business as unsupported. We use P95 as a more permissive routine-support convention without relying on sample maxima. Held-out P100 remains below 100% because later business can exceed earlier support estimates or labor can fall below the retained action levels.

B IMPLEMENTATION AND REPRODUCIBILITY

Within each backbone, all supervision frameworks use the same planning-time inputs, fitting budget, training objective, and 3 seeds. Input histories and backbone adaptations are shared across frameworks; output dimensions follow their respective prediction targets. Complete configurations will be released with the code.

Both horizons use the same target hours, excluding each split's first 7 calendar days so fitting precedes the earliest D+7 planning origin. Restaurant store management, kitchen management, and kitchen preparation have a one-role-hour floor; other roles and MilkTea allow zero. When no allowed staffing level supports the observed business, the requirement label is $\max \mathcal{A}_r + 1$, marking it as beyond the allowed range.

All frameworks use pinball loss (Koenker & Bassett, 1978) at $\mathcal{Q} = \{.5, .7, .85, .95\}$. Validation selects one quantile and a nonnegative scaling factor per chain, backbone, and framework to minimize mean labor while meeting all Reference support thresholds across both horizons and all seeds. This pair is shared across horizons and seeds and remains fixed for Test; settings that fail the criterion remain in the comparison.

GPU training used an NVIDIA V100. Upon acceptance, we will release both datasets, the code, and experiment configurations to support replication and reuse.[3]

[3] https://github.com/anonymous_repo

C Individual Support Metrics and Seed-Level Attainment

Tables 5–6 unpack the main table into hour, revenue, and transaction support, total labor, and seed-level success counts. This distinguishes meeting the objective on average from doing so consistently across runs. MLP, ResMLP, and N-HiTS with ProxyStaff achieve joint support–labor improvement in all 3 seeds on both chains and horizons. Chronos-2 succeeds in 11 of 12 runs; its MilkTea D+1 setting misses Reference support in one seed despite meeting the thresholds on average. Seed variation reflects training randomness, not independent datasets.

Table 5: Restaurant Test: detailed results for Table 3. H/R/T: mean hour/revenue/transaction support (%, 3 seeds). Labor: mean ± SD in thousands of role-hours; D+1/D+7: day/week ahead. As in Table 3, blue marks $C_{\min} > 100\%$; bold support triplets identify the highest $C_{\min}$, and bold labor the lowest mean among shaded methods for each backbone and lead time. Seeds: runs meeting all Reference support thresholds with less labor; bold 3/3 means all seeds succeed. Markers use unrounded values.

		D+1			D+7		
Backbone	Supervision	H/R/T↑	Labor↓	Seeds	H/R/T↑	Labor↓	Seeds
Human-Manager Reference		80.7/57.8/59.9	68.64	–	80.7/57.8/59.9	68.64	–
Tree-based models							
XGBoost	Staffing target	79.2/53.8/56.5	62.82±0.07	0/3	79.8/54.6/57.3	64.46±0.09	0/3
	Business + proxy	80.1/57.0/60.5	45.57±0.22	0/3	79.4/55.8/59.7	44.99±0.09	0/3
	ProxyStaff	**80.7/57.7/61.3**	47.36±0.23	1/3	**80.1/57.0/60.6**	46.66±0.06	0/3
Neural models trained from scratch							
MLP	Staffing target	79.9/54.9/57.5	63.60±0.32	0/3	79.7/54.6/57.3	64.41±0.36	0/3
	Business + proxy	80.7/58.0/61.6	46.97±0.62	2/3	81.4/59.0/62.9	**47.55**±0.20	**3/3**
	ProxyStaff	**81.4/59.7/63.6**	**49.05**±0.44	**3/3**	**81.5/59.5/63.4**	48.77±0.37	**3/3**
ResMLP	Staffing target	79.8/54.7/57.4	63.80±0.26	0/3	78.9/53.2/56.0	62.98±0.96	0/3
	Business + proxy	80.4/57.6/61.2	46.10±0.90	1/3	81.4/59.5/63.0	**47.84**±0.51	**3/3**
	ProxyStaff	**81.9/60.4/64.3**	**48.99**±0.12	**3/3**	**81.9/60.3/64.2**	49.58±0.48	**3/3**
N-HiTS	Staffing target	77.4/50.9/53.9	56.35±2.94	0/3	77.3/50.7/53.8	56.17±2.43	0/3
	Business + proxy	77.9/53.9/58.8	44.68±2.07	0/3	77.3/52.8/57.6	44.27±2.19	0/3
	ProxyStaff	**84.0/63.5/67.2**	**52.34**±1.19	**3/3**	**83.7/63.2/66.9**	**52.00**±0.90	**3/3**
TiDE	Staffing target	82.0/57.6/59.7	65.79±3.71	1/3	**82.2/58.0/60.0**	**65.65**±3.08	1/3
	Business + proxy	80.7/58.2/61.9	42.95±2.19	1/3	80.4/58.3/62.0	42.96±0.89	2/3
	ProxyStaff	**82.2/57.9/59.8**	66.68±1.83	0/3	82.2/57.9/59.8	66.70±1.84	0/3
Pretrained models with full fine-tuning							
Chronos-2	Staffing target	81.0/57.3/59.8	69.59±0.21	0/3	81.3/57.5/59.9	69.64±0.29	0/3
	Business + proxy	81.1/60.0/63.3	**49.93**±2.43	2/3	78.8/55.9/59.6	46.93±1.96	0/3
	ProxyStaff	**84.2/63.6/67.2**	65.51±2.49	**3/3**	**84.2/63.6/67.4**	**65.60**±3.91	**3/3**

Table 6: MilkTea Test: detailed results for Table 3. Metrics, units, and highlighting follow Table 5. Seeds counts successful runs out of 3.

		D+1			D+7		
Backbone	Supervision	H/R/T↑	Labor↓	Seeds	H/R/T↑	Labor↓	Seeds
Human-Manager Reference		86.7/68.8/69.6	3.12	–	86.7/68.8/69.6	3.12	–
Tree-based models							
XGBoost	Staffing target	86.6/68.4/69.1	2.88±0.01	0/3	86.7/68.3/69.1	2.82±0.02	0/3
	Business + proxy	89.1/73.4/74.0	**1.97**±0.05	**3/3**	88.3/71.6/72.2	**1.92**±0.02	**3/3**
	PROXYSTAFF	**89.6/74.2/74.8**	2.11±0.02	**3/3**	**89.1/73.2/73.8**	2.06±0.02	**3/3**
Neural models trained from scratch							
MLP	Staffing target	86.9/68.5/69.3	2.89±0.04	0/3	86.8/68.3/69.1	2.71±0.04	0/3
	Business + proxy	90.6/76.0/76.7	**2.22**±0.02	**3/3**	90.3/75.3/76.0	**2.20**±0.02	**3/3**
	PROXYSTAFF	**91.8/78.2/79.0**	2.91±0.02	**3/3**	**90.9/76.7/77.4**	2.77±0.07	**3/3**
ResMLP	Staffing target	85.1/65.5/66.3	2.51±0.05	0/3	85.5/66.2/67.1	2.54±0.10	0/3
	Business + proxy	86.3/69.3/70.0	1.99±0.08	2/3	82.0/61.7/62.5	1.69±0.09	0/3
	PROXYSTAFF	**90.6/75.9/76.7**	**2.58**±0.02	**3/3**	**88.7/72.5/73.4**	**2.43**±0.09	**3/3**
N-HiTS	Staffing target	86.5/67.9/68.7	2.89±0.10	0/3	86.7/68.3/69.1	2.83±0.02	0/3
	Business + proxy	**90.9/76.6/77.4**	2.30±0.07	**3/3**	**91.3/77.3/78.1**	2.40±0.01	**3/3**
	PROXYSTAFF	89.8/74.7/75.4	**2.12**±0.02	**3/3**	90.2/75.1/75.8	**2.18**±0.04	**3/3**
TiDE	Staffing target	86.6/67.8/68.8	2.78±0.12	1/3	85.7/66.3/67.3	2.71±0.06	0/3
	Business + proxy	88.3/70.8/71.6	**1.63**±0.01	**3/3**	87.9/70.0/70.8	**1.56**±0.08	2/3
	PROXYSTAFF	**88.5/71.1/71.8**	3.05±0.08	2/3	**88.8/71.6/72.3**	3.05±0.06	2/3
Pretrained models with full fine-tuning							
Chronos-2	Staffing target	87.0/68.8/69.6	2.95±0.08	1/3	87.7/70.1/70.9	3.11±0.06	2/3
	Business + proxy	**89.7/74.7/75.4**	**2.20**±0.10	**3/3**	**90.1/75.2/75.9**	**2.24**±0.04	**3/3**
	PROXYSTAFF	87.3/70.1/71.1	2.38±0.19	2/3	88.1/71.6/72.6	2.35±0.15	**3/3**