Appendix

A. Asymptotic Optimality of MSPRT

In this appendix, we review the asymptotic optimality of the matrix sequential probability ratio test (MSPRT). The whole statements here are primarily based on (Tartakovsky et al., 2014) and references therein. We here provide theorems without proofs, which are given in (Tartakovsky et al., 2014).

A.1. Notation and Basic Assumptions

First, we introduce mathematical notation and several basic assumptions. Let $X^{(0,T)} := \{x^{(t)}\}_{0 \leq t \leq T}$ be a stochastic process. We assume that $K (\in \mathbb{N})$ densities $p_k(X^{(0,T)})$ ($k \in [K] := \{1, 2, ..., K\}$) are distinct. Let $y (\in \mathcal{Y} := [K])$ be a parameter of the densities. Our task is to test K hypotheses $H_k : y = k$ ($k \in [K]$); i.e., to identify which of the K densities p_k is the true one through consecutive observations of $x(t)$.

Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t \geq 0}, P)$ for $t \in \mathbb{Z}_{\geq 0} := \{0, 1, 2, ...\}$ or $t \in \mathbb{R}_{\geq 0} := [0, \infty)$ be a filtered probability space. The sub-σ-algebra $\mathcal{F}_t$ of $\mathcal{F}$ is assumed to be generated by $X^{(0,t)}$. Our target hypotheses are $H_k : P = P_k$ ($k \in [K]$) where P_k are probability measures that are assumed to be locally mutually absolutely continuous. Let $\mathbb{E}_k$ denote the expectation under H_k ($k \in [K]$); e.g., $\mathbb{E}_k[f(X^{(0,t)}] = \int f(X^{(0,t)}) dP_k(X^{(0,t)})$ for a function f. We define the likelihood ratio matrix as

$$\Lambda_{kl}(t) := \frac{dP_k^t}{dP_l^t}(X^{(0,t)}) \quad (t \geq 0)\,, \tag{6}$$

where $\Lambda_{kl}(0) = 1$ P_k-a.s. and P_k^t is the restriction of P_k to $\mathcal{F}_t$. Therefore, the LLR matrix is defined as

$$\lambda_{kl}(t) := \log \Lambda_{kl}(t) \quad (t \geq 0)\,, \tag{7}$$

where $\lambda_{kl}(0) = 0$ P_k-a.s. The LLR matrix plays a crucial role in the MSPRT, as seen in the main text and in the following.

We define a multihypothesis sequential test as $\delta := (d, \tau)$. $d := d(X^{(0,t)})$ is an $\mathcal{F}_t$-measurable terminal decision function that takes values in $[K]$. τ is a stopping time with respect to $\{\mathcal{F}_t\}_{t \geq 0}$ and takes values in $[0, \infty)$. Therefore, $\{\omega \in \Omega | d = k\} = \{\omega \in \Omega | \tau < \infty, \delta \text{ accepts } H_k\}$. In the following, we solely consider statistical tests with $\mathbb{E}_k[\tau] < \infty$ ($k \in [K]$).

Error probabilities. We define three types of error probabilities:

$$\begin{aligned}
&\alpha_{kl}(\delta) := P_k(d = l) \quad (k \neq l, \quad k, l \in [K])\,, \\
&\alpha_k(\delta) := P_k(d \neq k) = \sum_{l (\neq k)} \alpha_{kl}(\delta) \quad (k \in [K])\,, \\
&\beta_l(\delta) := \sum_{k \in [K]} w_{kl} P_k(d = l) \qquad (l \in [K])\,,
\end{aligned} \tag{8}$$

where $w_{kl} > 0$ except for the zero diagonal entries. We further define $\alpha_{\max} := \max_{k,l} \alpha_{kl}$. Whenever $\alpha_{\max} \to 0$, we hereafter assume that for all $k, l, m, n \in [K]$ ($k \neq l$, $m \neq n$),

$$\lim_{\alpha_{\max} \to 0} \frac{\log \alpha_{kl}}{\log \alpha_{mn}} = c_{klmn}\,,$$

where $0 < c_{klmn} < \infty$. This technical assumption means that α_{kl} does not go to zero at an exponentially faster or slower rate than the others.

Classes of tests. We define the corresponding sets of statistical tests with bounded error probabilities.

$$\begin{aligned}
&C(\{\alpha\}) := \{\, \delta \,|\, \alpha_{kl}(\delta) \leq \alpha_{kl}, \quad k \neq l, \quad k, l \in [K]\}\,, \\
&C(\boldsymbol{\alpha}) := \{\, \delta \,|\, \alpha_k(\delta) \leq \alpha_k, \quad k \in [K]\}\,, \\
&C(\boldsymbol{\beta}) := \{\, \delta \,|\, \beta_l(\delta) \leq \beta_l, \quad l \in [K]\}\,.
\end{aligned} \tag{9}$$

Convergence of random variables. We introduce the following two types of convergence for later convenience.

Definition A.1 (Almost sure convergence (convergence with probability one))**.** *Let $\{x^{(t)}\}_{t \geq 0}$ denote a stochastic process. We say that stochastic process $\{x^{(t)}\}_{t \geq 0}$ converges to a constant c almost surely as $t \to \infty$ (symbolically, $x^{(t)} \xrightarrow[t \to \infty]{P-a.s.} c$), if*

$$P\left(\lim_{t \to \infty} x^{(t)} = c\right) = 1\,.$$

Definition A.2 (r-quick convergence)**.** *Let $\{x^{(t)}\}_{t \geq 0}$ be a stochastic process. Let $\mathcal{T}_\epsilon(\{x^{(t)}\}_{t \geq 0})$ be the last entry time of $\{x^{(t)}\}_{t \geq 0}$ into the region $(-\infty, -\epsilon) \cup (\epsilon, \infty)$; i.e.,*

$$\mathcal{T}_\epsilon(\{x^{(t)}\}_{t \geq 0}) = \sup_{t \geq 0} \{ t \,|\, |x^{(t)}| > \epsilon \}, \quad \sup\{\emptyset\} := 0\,. \tag{10}$$

Then, we say that stochastic process $\{x^{(t)}\}_{t \geq 0}$ converges to zero r-quickly, or

$$x^{(t)} \xrightarrow[t \to \infty]{r-\text{quickly}} 0\,, \tag{11}$$

for some $r > 0$, if

$$\mathbb{E}[(\mathcal{T}_\epsilon(\{x^{(t)}\}_{t \geq 0}))^r] < \infty \quad \textit{for all } \epsilon > 0\,. \tag{12}$$

r-quick convergence ensures that the last entry time into the large-deviation region ($\mathcal{T}_\epsilon(\{x^{(t)}\}_{t \geq 0})$) is finite in expectation.

A.2. MSPRT: Matrix Sequential Probability Ratio Test

Formally, the MSPRT is defined as follows:

Definition A.3 (Matrix sequential probability ratio test)**.** *Define a threshold matrix $a_{kl} \in \mathbb{R}$ ($k, l \in [K]$), where the diagonal elements are immaterial and arbitrary, e.g., 0. The MSPRT δ^* of multihypothesis $H_k : P = P_k$ ($k \in [K]$) is defined as*

$$\delta^* := (d^*, \tau^*)$$
$$\tau^* := \min\{\tau_k | k \in [K]\}$$
$$d^* := k \quad \textit{if} \quad \tau^* = \tau_k$$
$$\tau_k := \inf\{t \geq 0 | \min_{\substack{l \in [K] \\ l (\neq k)}} \{\lambda_{kl}(t) - a_{lk}\} \geq 0\} \ \ (k \in [K]) \, .$$

In other words, the MSPRT stops at the smallest t such that for a number of $k \in [K]$, $\lambda_{kl}(t) \geq a_{lk}$ for all $l (\neq k)$. Note that the uniqueness of such k is ensured by the anti-symmetry of λ_{kl}. In our experiment, we use a single-valued threshold for simplicity. A general threshold matrix may improve performance, especially when the dataset is class-imbalanced (Longadge & Dongre, 2013; Ali et al., 2015; Hong et al., 2016). We occasionally use $A_{lk} := e^{a_{lk}}$ in the following.

The following lemma determines the relationship between the thresholds and error probabilities.

Lemma A.1 (General error bounds of the MSPRT (Tartakovsky, 1998))**.** *The following inequalities hold:*

1. $\alpha^*_{kl} \leq e^{-a_{kl}}$ *for* $k, l \in [K]$ $(k \neq l)$,
2. $\alpha^*_k \leq \sum_{l(\neq k)} e^{-a_{kl}}$ *for* $k \in [K]$,
3. $\beta^*_l \leq \sum_{k(\neq l)} w_{kl} e^{-a_{kl}}$ *for* $l \in [K]$.

Therefore,

$$a_{lk} \geq \log(\frac{1}{\alpha_{lk}}) \Longrightarrow \delta^* \in C(\{\alpha\}) \, ,$$
$$a_{lk} \geq a_l = \log(\frac{K-1}{\alpha_{lk}}) \Longrightarrow \delta^* \in C(\boldsymbol{\alpha}) \, ,$$
$$a_{lk} \geq a_k = \log(\frac{\sum_{m(\neq k)} w_{mk}}{\beta_k}) \Longrightarrow \delta^* \in C(\boldsymbol{\beta}) \, .$$

A.3. Asymptotic Optimality of MSPRT in I.I.D. Cases

A.3.1. Under First Moment Condition

Given the true distribution, one can derive a dynamic programming recursion; its solution is the optimal stopping time. However, that recursion formula is intractable in general due to its composite integrals to calculate expectation values(Tartakovsky et al., 2014). Thus, we cannot avoid several approximations unless the true distribution is extremely simple.

To avoid the complications above, we focus on the asymptotic behavior of the MSPRT, where the error probabilities go to zero. In this region, the stopping time and thresholds typically approach infinity because more evidence is needed to make such careful, perfect decisions.

First, we provide the lower bound of the stopping time. Let us define the first moment of the LLR: $I_{kl} := \mathbb{E}_k[\lambda_{kl}(1)]$. Note that I_{kl} is the Kullback-Leibler divergence and hence $I_{kl} \geq 0$.

Lemma A.2 (Lower bound of the stopping time (Tartakovsky et al., 2014))**.** *Assume that I_{lk} is positive and finite for all $k, l \in [K]$ ($k \neq l$). If $\sum_{k \in [K]} \alpha_k \leq 1$, then for all $k \in [K]$,*

$$\inf_{\delta \in C(\{\alpha\})} \mathbb{E}_k[\tau] \geq \max \left[\frac{1}{I_{kl}} \sum_{m \in [K]} \alpha_{km} \log(\frac{\alpha_{km}}{\alpha_{lm}}) \right] \, .$$

The proof follows from Jensen's inequality and Wald's identity $\mathbb{E}_k[\lambda_{kl}(\tau)] = I_{kl} \mathbb{E}_k[\tau]$. However, the lower bound is unattainable in general[2]. In the following, we show that the MSPRT asymptotically satisfies the lower bound.

Lemma A.3 (Asymptotic lower bounds (i.i.d. case) (Tartakovsky, 1998))**.** *Assume the first moment condition $0 < I_{kl} < \infty$ for all $k, l \in [K]$ ($k \neq l$). The following inequalities hold for all $m > 0$ and $k \in [K]$,*

1. *As* $\alpha_{\max} \to 0$,
$$\inf_{\delta \in C(\{\alpha\})} \mathbb{E}_k[\tau]^m \geq \max_{l(\neq k)} \left[\frac{|\log \alpha_{lk}|}{I_{kl}} \right]^m (1 + o(1)) \, .$$
2. *As* $\max_k \alpha_k \to 0$,
$$\inf_{\delta \in C(\boldsymbol{\alpha})} \mathbb{E}_k[\tau]^m \geq \max_{l(\neq k)} \left[\frac{|\log \alpha_l|}{I_{kl}} \right]^m (1 + o(1)) \, .$$
3. *As* $\max_k \beta_k \to 0$,
$$\inf_{\delta \in C(\boldsymbol{\beta})} \mathbb{E}_k[\tau]^m \geq \max_{l(\neq k)} \left[\frac{|\log \beta_k|}{I_{kl}} \right]^m (1 + o(1)) \, .$$

Theorem A.1 (Asymptotic optimality of the MSPRT with the first moment condition (i.i.d. case) (Tartakovsky, 1998; Dragalin et al., 1999; Tartakovsky et al., 2014))**.** *Assume the first moment condition $0 < I_{kl} < \infty$ for all $k, l \in [K]$ ($k \neq l$).*

1. *If $a_{lk} = \log(\frac{1}{\alpha_{lk}})$ for $k.l \in [K]$ ($k \neq l$), then $\delta^* \in C(\{\alpha\})$, and for all $m > 0$ and $k \in [K]$,*
$$\inf_{\delta \in C(\{\alpha\})} \mathbb{E}_k[\tau]^m \sim \mathbb{E}_k[\tau^*]^m \sim \max_{\substack{l \in [K] \\ l(\neq k)}} \left[\frac{|\log \alpha_{lk}|}{I_{kl}} \right]^m \tag{13}$$

[2] We can show that when $K = 2$, the SPRT can attain the lower bound if there are no overshoots of the LLR over the thresholds.

as $\alpha_{\max} \to 0$. *If* $a_{lk} \neq \log(\frac{1}{\alpha_{lk}})$*, the above inequality holds when* $a_{lk} \sim \log(\frac{1}{\alpha_{lk}})$ *and* $\alpha_{kl}(\delta^*) \leq \alpha_{kl}$.[3]

2. *If* $a_{lk} = \log(\frac{K-1}{\alpha_l})$ *for* $k.l \in [K]$ $(k \neq l)$*, then* $\delta^* \in C(\boldsymbol{\alpha})$*, and for all* $m > 0$ *and* $k \in [K]$*,*

$$\inf_{\delta \in C(\boldsymbol{\alpha})} \mathbb{E}_k[\tau]^m \sim \mathbb{E}_k[\tau^*]^m \sim \max_{\substack{l \in [K] \\ l(\neq k)}} \left[\frac{|\log \alpha_l|}{I_{kl}} \right]^m \tag{14}$$

as $\max_k \alpha_k \to 0$. *If* $a_l \neq \log(\frac{K-1}{\alpha_l})$*, the above inequality holds when* $a_{lk} \sim \log(\frac{K-1}{\alpha_l})$ *and* $\alpha_l(\delta^*) \leq \alpha_l$.

3. *If* $a_{lk} = \log(\frac{\sum_{n(\neq k)} w_{nk}}{\beta_k})$ *for* $k.l \in [K]$ $(k \neq l)$*, then* $\delta^* \in C(\boldsymbol{\beta})$*, and for all* $m > 0$ *and* $k \in [K]$*,*

$$\inf_{\delta \in C(\boldsymbol{\beta})} \mathbb{E}_k[\tau]^m \sim \mathbb{E}_k[\tau^*]^m \sim \max_{\substack{l \in [K] \\ l(\neq k)}} \left[\frac{|\log \beta_k|}{I_{kl}} \right]^m \tag{15}$$

as $\max_k \beta_k \to 0$. *If* $a_{lk} \neq \log(\frac{1}{\beta_k})$*, the above inequality holds when* $a_{lk} \sim \log(\frac{1}{\beta_k})$ *and* $\beta_k(\delta^*) \leq \beta_k$.

Therefore, we conclude that the MSPRT asymptotically minimizes all positive moments of the stopping time including $m = 1$; i.e., asymptotically, the MSPRT makes the quickest decision in expectation among all the algorithms with bounded error probabilities.

A.3.2. Under Second Moment Condition

The *second moment condition*

$$\mathbb{E}_k[\lambda_{kl}(1)]^2 < \infty \quad (k, l \in [K]) \tag{16}$$

strengthens the optimality. We define the *cumulative flaw matrix* as

$$\Upsilon_{kl} := \exp\Big(-\sum_{t=1}^{\infty} \frac{1}{t} [P_l(\lambda_{kl}(t) > 0) + P_k(\lambda_{kl}(t) \leq 0)]\Big) \tag{17}$$

$(k, l \in [K], k \neq l)$. Note that $0 < \Upsilon_{kl} = \Upsilon_{lk} \leq 1$.

Theorem A.2 (Asymptotic optimality of the MSPRT with the second moment condition (i.i.d. case) (Lorden, 1977; Tartakovsky et al., 2014))**.** *Assume that the threshold* $A_{kl} = A_{kl}(c)$ *is a function of a small parameter* $c > 0$*. Then, the error probabilities of the MSPRT are also functions of c; i.e.,* $\alpha_{kl}(\delta^*) =: \alpha^*_{kl}(c)$, $\alpha_k(\delta^*) =: \alpha^*_k(c)$*, and* $\beta_k(\delta^*) =: \beta^*_k(c)$*, and* $A_{lk}(c) \xrightarrow{c \to 0} \infty$ *indicates* $\alpha^*_{kl}(c), \alpha^*_k(c), \beta^*_k(c) \xrightarrow{c \to 0} 0$.

1. *Let* $A_{lk} = B_{lk}/c$ *for any* $B_{kl} > 0$ $(k \neq l)$*. Then, as* $c \to 0$*,*

$$\mathbb{E}_k \tau^*(c) = \inf_{\delta \in C(\{\alpha^*(c)\})} \mathbb{E}_k \tau + o(1) \tag{18}$$

$$\mathbb{E}_k \tau^*(c) = \inf_{\delta \in C(\boldsymbol{\alpha}^*(c))} \mathbb{E}_k \tau + o(1) \tag{19}$$

for all $k \in [K]$*, where* $\boldsymbol{\alpha}^*(c) := (\alpha^*_1(c), ..., \alpha^*_K(c))$.

2. *Let* $A_{lk}(c) = w_{lk}\Upsilon_{kl}/c$*. Then, as* $c \to 0$*,*

$$\mathbb{E}_k \tau^*(c) = \inf_{\delta \in C(\{\boldsymbol{\beta}^*(c)\})} \mathbb{E}_k \tau + o(1) \tag{20}$$

for all $k \in [K]$*, where* $\boldsymbol{\beta}^*(c) := (\beta^*_1(c), ..., \beta^*_K(c))$.

Therefore, the MSPRT δ^* asymptotically minimizes the expected stopping time among all tests whose error probabilities are less than or equal to those of δ^*. We can further generalize Theorem A.2 by introducing different costs c_k for each hypothesis H_k to allow different rates (see (Tartakovsky et al., 2014)).

A.4. Asymptotic Optimality of MSPRT in General Non-I.I.D. Cases

Lemma A.4 (Asymptotic lower bounds (Tartakovsky, 1998))**.** *Assume that there exists a non-negative increasing function* $\psi(t)$ ($\psi(t) \xrightarrow{t \to \infty} \infty$) *and positive finite constants* I_{lk} $(k, l \in [K], k \neq l)$ *such that for all* $\epsilon > 0$ *and* $k, l \in [K]$ $(k \neq l)$*,*

$$\lim_{T \to \infty} P_k \left(\sup_{0 \leq t \leq T} \frac{\lambda_{kl}(t)}{\psi(T)} \geq (1 + \epsilon) I_{kl} \right) = 1\,. \tag{21}$$

Then, for all $m > 0$ *and* $k \in [K]$*,*

$$\inf_{\delta \in C(\{\alpha\})} \mathbb{E}_k[\tau]^m \geq \Psi \left(\max_{\substack{l \in [K] \\ l(\neq k)}} \frac{|\log \alpha_{lk}|}{I_{kl}} \right)^m (1 + o(1)) \quad \text{as} \;\; \alpha_{\max} \to 0\,, \tag{22}$$

$$\inf_{\delta \in C(\boldsymbol{\alpha})} \mathbb{E}_k[\tau]^m \geq \Psi \left(\max_{\substack{l \in [K] \\ l(\neq k)}} \frac{|\log \alpha_l|}{I_{kl}} \right)^m (1 + o(1)) \quad \text{as} \;\; \max_n \alpha_n \to 0\,, \tag{23}$$

$$\inf_{\delta \in C(\boldsymbol{\beta})} \mathbb{E}_k[\tau]^m \geq \Psi \left(\max_{\substack{l \in [K] \\ l(\neq k)}} \frac{|\log \beta_k|}{I_{kl}} \right)^m (1 + o(1)) \quad \text{as} \;\; \max_n \beta_n \to 0\,. \tag{24}$$

where Ψ *is the inverse function of* ψ.

Note that if for all $0 < T < \infty$

$$P_k \left(\sup_{0 \leq t \leq T} |\lambda_{kl}(t)| < \infty \right) = 1 \tag{25}$$

[3] Recall that $\alpha_{kl}(\delta^*) \leq \alpha_{kl}$ is automatically satisfied whenever $a_{lk} \geq \log(1/\alpha_{lk})$ for general distribution because of Lemma A.1. Similar arguments follow for 2. and 3. in Theorem A.1.

and if

$$\frac{\lambda_{kl}(t)}{\psi(t)} \xrightarrow[t\to\infty]{P_k - a.s.} I_{kl} \quad (k, l \in [K], \ \ k \neq l)\,, \tag{26}$$

then (21) holds.

Theorem A.3 (Asymptotic optimality of the MSPRT (Tartakovsky, 1998))**.** *Assume that that there exists a non-negative increasing function $\psi(t)$ ($\psi(t) \xrightarrow{t\to\infty} \infty$) and positive finite constants I_{kl} ($k, l \in [K]$, $k \neq l$) such that for some $r > 0$,*

$$\frac{\lambda_{kl}(t)}{\psi(t)} \xrightarrow[t\to\infty]{P_k - r - quickly} I_{kl} \tag{27}$$

for all $k, l \in [K]$ ($k \neq l$). Let Ψ be the inverse function of ψ. Then,

1. *If $a_{lk} \sim \log(1/\alpha_{lk})$ and $\alpha_{kl}(\delta^*) \leq \alpha_{kl}$ ($k, l \in [K]$, $k \neq l$)*[4]*, then for all $m \in (0, r]$ and $k \in [K]$,*

$$\inf_{\delta \in C(\{\alpha\})} \mathbb{E}_k[\tau]^m \sim \mathbb{E}_k[\tau^*]^m \sim \Psi\left(\max_{\substack{l \in [K] \\ l (\neq k)}} \frac{|\log \alpha_{lk}|}{I_{kl}} \right)^m \quad as \quad \alpha_{\max} \to 0\,. \tag{28}$$

2. *If $a_{lk} \sim \log((K-1)/\alpha_l)$ and $\alpha_k(\delta^*) \leq \alpha_k$ ($k, l \in [K]$, $k \neq l$), then for all $m \in (0, r]$ and $k \in [K]$,*

$$\inf_{\delta \in C(\boldsymbol{\alpha})} \mathbb{E}_k[\tau]^m \sim \mathbb{E}_k[\tau^*]^m \sim \Psi\left(\max_{\substack{l \in [K] \\ l (\neq k)}} \frac{|\log \alpha_{l}|}{I_{kl}} \right)^m \quad as \quad \max_k \alpha_k \to 0\,. \tag{29}$$

3. *If $a_{lk} \sim \log(\sum_{n(\neq k)} w_{nk}/\beta_k)$ and $\beta_k(\delta^*) \leq \beta_k$ ($k, l \in [K]$, $k \neq l$), then for all $m \in (0, r]$ and $k \in [K]$,*

$$\inf_{\delta \in C(\boldsymbol{\beta})} \mathbb{E}_k[\tau]^m \sim \mathbb{E}_k[\tau^*]^m \sim \Psi\left(\max_{\substack{l \in [K] \\ l (\neq k)}} \frac{|\log \beta_{k}|}{I_{kl}} \right)^m \quad as \quad \max_k \beta_k \to 0\,. \tag{30}$$

Therefore, combining Lemma A.4 and Theorem A.3, we conclude that the MSPRT asymptotically minimizes the moments of the stopping time; i.e., asymptotically, the MSPRT makes the quickest decision in expectation among all the algorithms with bounded error probabilities, even without the i.i.d. assumption.

[4] Recall that $\alpha_{kl}(\delta^*) \leq \alpha_{kl}$ is automatically satisfied whenever $a_{lk} \geq \log(1/\alpha_{lk})$ for general distribution because of Lemma A.1. Similar arguments follow for 2. and 3. in Theorem A.3.

B. Supplementary Related Work

We provide additional references. Our work is an interdisciplinary study and potentially bridges various research areas, such as early classification of time series, sequential hypothesis testing, sequential decision making, classification with abstention, and DRE.

Early classification of time series. Many methods have been proposed for early classification of time series: non-deep models are (Xing et al., 2011; McGovern et al., 2011; Xing et al., 2012; Ghalwash & Obradovic, 2012; Ghalwash et al., 2013; Ghalwash et al., 2014; Mori et al., 2016; Karim et al., 2019; Schäfer & Leser, 2020); deep models are (Ma et al., 2016; Wang et al., 2016; Suzuki et al., 2018; Rußwurm et al., 2019); reinforcement learning-based models are (Hartvigsen et al., 2019; Martinez et al., 2020; Wang et al., 2020). There is a wide variety of real-world applications of such models: length adaptive text classification (Huang et al., 2017), early text classification for sexual predator detection and depression detection on social media documents (López-Monroy et al., 2018), early detection of thermoacoustic instability from high-speed videos taken from a combustor (Gangopadhyay et al., 2021), and early river classification through real-time monitoring of water quality (Gupta et al., 2019).

Early exit problem. The *overthinking problem* (Kaya et al., 2019) occurs when a DNN can reach correct predictions before its final layer. Early exit from forward propagation mitigates wasteful computation and circumvents overfitting. (Kaya et al., 2019) proposes the Shallow-Deep Networks, which is equipped with internal layerwise classifiers and observes internal layerwise predictions to trigger an early exit. The early exit mechanism has been applied to Transformer (Vaswani et al., 2017) and BERT (Devlin et al., 2019); e.g., see (Dehghani et al., 2019; Zhou et al., 2020). Owing to early exiting, (Ghodrati et al., 2021) sets a new state of the art for efficient video understanding on the HVU benchmark. However, early exit algorithms are typically given by heuristics. MSPRT-TANDEM can be both the internal classifier and early exit algorithm itself with the theoretically sound background.

Classification with a reject option. Classification with a reject option is also referred to as classification with an abstain option, classification with abstention, classification with rejection, and selective classification. Sequential classification with a reject option (to postpone the classification) can be regarded as early classification of time series (Hatami & Chira, 2013).

SPRT. The SPRT for two-hypothesis testing ("binary SPRT") is optimal for i.i.d. distributions (Wald & Wolfowitz, 1948; Tartakovsky et al., 2014). There are many different proofs: e.g., (Burkholder & Wijsman, 1963; Matches, 1963; Tartakovsky, 1991; Lehmann & Romano, 2006; Shiryaev, 2007; Ferguson, 2014). The Bayes optimality of the binary SPRT for i.i.d. distributions is proved in (Arrow et al., 1949; Ferguson, 2014). The generalization of the i.i.d. MSPRT to non-stationary processes with independent increments is made in (Tartakovskij, 1981; Golubev & Khas'minskii, 1984; Tartakovsky, 1991; Verdenskaya & Tartakovskii, 1992; Tartakovsky, 1998a).

Density ratio estimation. A common method of estimating the density ratio is to train a machine learning model to classify two types of examples in a training dataset and extract the density ratio from the optimal classifier (Sugiyama et al., 2012; Gutmann & Hirayama, 2012; Menon & Ong, 2016).

C. Proof of Theorem 3.1

In this appendix, we provide the proof of Theorem 3.1.

We define the target parameter set as $\Theta^* := \{\boldsymbol{\theta}^* \in \mathbb{R}^{d_\theta} | \hat{\lambda}(X^{(1,t)}; \boldsymbol{\theta}^*) = \lambda(X^{(1,t)}) \ (\forall t \in [T])\}$, and we assume $\Theta^* \neq \emptyset$ throughout this paper. For instance, sufficiently large neural networks can satisfy this assumption. We additionally assume that each $\boldsymbol{\theta}^*$ is separated in Θ^*; i.e., $\exists \delta > 0$ such that $B(\boldsymbol{\theta}^*; \delta) \cap B(\boldsymbol{\theta}^{*\prime}; \delta) = \emptyset$ for arbitrary $\boldsymbol{\theta}^*, \boldsymbol{\theta}^{*\prime} \in \Theta^*$, where $B(\boldsymbol{\theta}; \delta)$ denotes the open ball at center $\boldsymbol{\theta}$ with radius δ.[5]

Theorem C.1 (Consistency of the LSEL)**.** *Let $L(\boldsymbol{\theta})$ and $\hat{L}_S(\boldsymbol{\theta})$ denote $L_{\text{LSEL}}[\hat{\lambda}]$ and $\hat{L}_{\text{LSEL}}(\boldsymbol{\theta}; S)$, respectively. Assume the following three conditions:*

(a) $\forall k, l \in [K]$, $\forall t \in [T]$, $p(X^{(1,t)}|k) = 0 \iff p(X^{(1,t)}|l) = 0$.

(b) $\sup_{\boldsymbol{\theta}} |\hat{L}_S(\boldsymbol{\theta}) - L(\boldsymbol{\theta})| \xrightarrow[M \to \infty]{P} 0$*; i.e., $\hat{L}_S(\boldsymbol{\theta})$ converges in probability uniformly over $\boldsymbol{\theta}$ to $L(\boldsymbol{\theta})$.*[6]

(c) For all $\theta^* \in \Theta^*$, there exist $t \in [T]$, $k \in [K]$ and $l \in [K]$, such that the following $d_\theta \times d_\theta$ matrix is full-rank:

$$\int dX^{(1,t)} p(X^{(1,t)}|k) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{kl}(X^{(1,t)}; \boldsymbol{\theta}^*) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{kl}(X^{(1,t)}; \boldsymbol{\theta}^*)^\top . \tag{31}$$

Then, $P(\hat{\boldsymbol{\theta}}_S \notin \Theta^*) \xrightarrow{M \to \infty} 0$; i.e., $\hat{\boldsymbol{\theta}}_S$ converges in probability into Θ^*.

First, we prove Lemma C.1, which is then used in Lemma C.2. Using Lemma C.2, we prove Theorem 3.1. Our proofs are partly inspired by (Gutmann & Hyvärinen, 2012). Note that for simplicity, we prove all the statements only for an arbitrary $t \in [T]$. The result can be straightforwardly generalized to the sum of the losses with respect to $t \in [T]$. Therefore, we omit $\frac{1}{T}\sum_{t \in [T]}$ from L and $\hat{L}_S$ in the following.

Lemma C.1 (Non-parametric estimation)**.** *Assume that for all $k, l \in [K]$, $p(X^{(1,t)}|k) = 0 \iff p(X^{(1,t)}|l) = 0$. Then, $L[\tilde{\lambda}]$ attains the unique minimum at $\tilde{\lambda} = \lambda$.*

Proof. Let $\phi(X^{(1,t)}) = (\phi_{kl}(X^{(1,t)}))_{k.l \in [K]}$ be an arbitrary perturbation function to $\tilde{\lambda}$. ϕ_{kl} satisfies $\phi_{kl} = -\phi_{lk}$ and is not identically zero if $k \neq l$. For an arbitrarily small $\epsilon > 0$,

$$L[\tilde{\lambda} + \epsilon\phi] = L[\tilde{\lambda}] + \frac{1}{K} \sum_{k \in [K]} \int dX^{(1,t)} p(X^{(1,t)}|k) \Bigg[\epsilon \frac{- \sum_{l (\neq k)} e^{-\tilde{\lambda}_{kl}(X^{(1,t}}}{\sum_{m \in [K]} e^{-\tilde{\lambda}_{km}}}$$
$$+ \frac{\epsilon^2}{2(\sum_{m \in [K]} e^{-\tilde{\lambda}_{km}(X^{(1,t)})})^2}$$
$$\times \Big\{ \sum_{m \in [K]} e^{-\tilde{\lambda}_{km}(X^{(1,t)})} \sum_{l (\neq k)} e^{-\tilde{\lambda}_{kl}(X^{(1,t)})} \phi_{kl}^2(X^{(1,t)}) - (\sum_{l (\neq k)} e^{-\tilde{\lambda}_{kl}(X^{(1,t)}}$$
$$+ \mathcal{O}(\epsilon^3) \, . \tag{32}$$

A necessary condition for the optimality is that the first order terms vanish for arbitrary ϕ. Because

$$\text{(first order)} = -\frac{\epsilon}{K} \int dX^{(1,t)} \sum_{k > l} \phi_{kl}(X^{(1,t)})$$
$$\times \Bigg[\frac{p(X|k)}{\sum_{m \in [K]} e^{-\tilde{\lambda}_{km}(X^{(1,t)})}} e^{-\tilde{\lambda}_{kl}(X^{(1,t)})} - \frac{p(X|l)}{\sum_{m \in [K]} e^{-\tilde{\lambda}_{lm}(X^{(1,t)})}} e^{-\tilde{\lambda}_{lk}(X^{(1}} \tag{33}$$

and $p(X^{(1,t)}|k) = 0 \Leftrightarrow p(X^{(1,t)}|l) = 0$, the following equality holds at the unique extremum:

$$\frac{p(X|k)}{\sum_{m \in [K]} e^{-\tilde{\lambda}_{km}(X^{(1,t)})}} e^{-\tilde{\lambda}_{kl}(X^{(1,t)})} = \frac{p(X|l)}{\sum_{m \in [K]} e^{-\tilde{\lambda}_{lm}(X^{(1,t)})}} e^{-\tilde{\lambda}_{lk}(X}$$
$$\iff e^{\lambda_{kl}} \sum_{m \in [K]} \tilde{\Lambda}_{ml} = \tilde{\Lambda}_{kl}^2 \sum_{m \in [K]} \tilde{\Lambda}_{mk}$$
$$\left(= \tilde{\Lambda}_{kl} \sum_{m \in [K]} \tilde{\Lambda}_{mk} \tilde{\Lambda}_{kl} = \tilde{\Lambda}_{kl} \sum_{m \in [K]} \tilde{\Lambda}_{ml} \right)$$
$$\iff e^{\lambda_{kl}} = \tilde{\Lambda}_{kl}$$
$$\iff \tilde{\lambda}_{kl}(X^{(1,t)}) = \lambda_{kl}(X^{(1,t)}) \, ,$$

where we defined $\tilde{\Lambda}_{kl} := e^{\tilde{\lambda}_{kl}}$ and used $\tilde{\Lambda}_{mk}\tilde{\Lambda}_{kl} = \tilde{\Lambda}_{ml}$. Next, we prove that $\tilde{\lambda}_{kl} = \lambda_{kl}$ is the minimum by showing

[5]This assumption is for simplicity of the proofs. When the assumption above is not true, we conjecture that the consistency holds by assuming the positivity of a projected Hessian of $\hat{\lambda}_{kl}$ w.r.t. $\boldsymbol{\theta}$ at $\partial\Theta^*$ (the boundary of Θ^*). The projection directions may depend on the local curvature of $\partial\Theta^*$ and whether Θ^* is open or closed. We omit further discussions here because they are too complicated but maintain the basis of our statements.

[6]More specifically, $\forall \epsilon > 0, P(\sup_{\boldsymbol{\theta}} |\hat{L}_S(\boldsymbol{\theta}) - L(\boldsymbol{\theta})| > \epsilon) \xrightarrow{M \to \infty} 0$.

that the second order of (32) is positive-definite:

$$
\begin{aligned}
&(\text{second order}) = \frac{\epsilon^2}{2}\frac{1}{K}\sum_{k\in[K]}\int dX^{(1,t)}\frac{p(X^{(1,t)}|k)}{(\sum_{m\in[K]}e^{-\tilde{\lambda}_{km}(X^{(1,t)})})^2}\\
&\times\Big\{\sum_{m\in[K]}e^{-\tilde{\lambda}_{km}(X^{(1,t)})}\sum_{l(\neq k)}\phi_{kl}^2(X^{(1,t)})e^{-\tilde{\lambda}_{kl}(X^{(1,t)})} - (\sum_{m(\neq k)}\phi_{km}(X^{(1,t)})e^{-\tilde{\lambda}_{km}(X^{(1,t)})})^2\Big\}\\
=&\frac{\epsilon^2}{2}\frac{1}{K}\sum_{k\in[K]}\int dX^{(1,t)}\frac{p(X^{(1,t)}|k)}{(\sum_{m\in[K]}e^{-\tilde{\lambda}_{km}(X^{(1,t)})})^2}\\
&\times\Big\{\sum_{l(\neq k)}\phi_{kl}^2(X^{(1,t)})e^{-\tilde{\lambda}_{kl}(X^{(1,t)})}\\
&+\sum_{\substack{m>n\\ m,n\neq k}}(\phi_{km}(X^{(1,t)})-\phi_{kn}(X^{(1,t)}))^2e^{-\tilde{\lambda}_{km}(X^{(1,t)})}e^{-\tilde{\lambda}_{kn}(X^{(1,t)})}\Big\}\\
&>0\,.
\end{aligned}
$$

□

Lemma C.2 (Θ^* minimizes L)**.** *Assume that for all $\theta^*\in\Theta^*$, there exist $k^*\in[K]$ and $l^*\in[K]$, such that the following $d_\theta\times d_\theta$ matrix is full-rank:*

$$
\int dX^{(1,t)}p(X^{(1,t)}|k^*)\nabla_{\boldsymbol{\theta}^*}\hat{\lambda}_{k^*l^*}(X^{(1,t)};\boldsymbol{\theta}^*)\nabla_{\boldsymbol{\theta}^*}\hat{\lambda}_{k^*l^*}(X^{(1,t)};\boldsymbol{\theta}^*)^\top\,. \tag{34}
$$

Then, for any $\boldsymbol{\theta}\notin\Theta^*$,

$$
L(\boldsymbol{\theta})>L(\boldsymbol{\theta}^*)\quad(\forall\boldsymbol{\theta}^*\in\Theta^*)\,,
$$

meaning that $\Theta^*=\mathrm{argmin}_{\boldsymbol{\theta}}L(\boldsymbol{\theta})$.

Proof. Let $\boldsymbol{\theta}^*$ be an arbitrary element in Θ^*. For an arbitrarily small $\epsilon>0$, let $\boldsymbol{\varphi}\in\mathbb{R}^{d_\theta}$ be an arbitrary vector such that $\boldsymbol{\varphi}\neq\mathbf{0}$. Then, in a neighborhood of $\boldsymbol{\theta}^*$,

$$
\begin{aligned}
&L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*+\epsilon\boldsymbol{\varphi})]=L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]\\
&+\frac{1}{K}\sum_{k\in[K]}\int dX^{(1,t)}p(X^{(1,t)}|k)\Big[\epsilon\frac{-\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}(X^{(1,t)})}{\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)}}\\
&+\frac{\epsilon^2}{2(\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)})^2}\Big\{-\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)}\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\omega_{kl}(X^{(1,t)})\\
&+\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)}\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}^2(X^{(1,t)})-(\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}(X^{(1,t)}))^2\Big\}\Big]\\
&+\mathcal{O}(\epsilon^3)\,,
\end{aligned}\tag{35}
$$

where $\rho_{kl}(X^{(1,t)}):=\boldsymbol{\varphi}^\top\cdot\nabla_{\boldsymbol{\theta}}\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)$ and $\omega_{kl}(X^{(1,t)}):=\boldsymbol{\varphi}^\top\cdot\nabla^2_{\boldsymbol{\theta}}\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)\cdot\boldsymbol{\varphi}$. By definition of Θ^*, $\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)=\lambda_{kl}(X^{(1,t)})=\log(p(X^{(1,t)}|k)/p(X^{(1,t)}|l))$. Substituting this into (35), we can see that the first order terms and the second order terms that contain ω_{kl} are identically zero because of the asymmetry of $\hat{\lambda}_{kl}$. Therefore,

$$
\begin{aligned}
&L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*+\epsilon\boldsymbol{\varphi})]=L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]\\
&+\frac{1}{K}\sum_{k\in[K]}\int dX^{(1,t)}p(X^{(1,t)}|k)\frac{\epsilon^2}{2(\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)})^2}\\
&\times\Big\{\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)}\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}^2(X^{(1,t)})-(\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}(X^{(1,t)}))^2\Big\}\\
&+\mathcal{O}(\epsilon^3)\,.
\end{aligned}\tag{36}
$$

Next, we define

$$
I_k:=\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)}\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}^2(X^{(1,t)})-(\sum_{l(\neq k)}e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\rho_{kl}(X^{(1,t)}))^2 \tag{37}
$$

so that

$$
\begin{aligned}
&L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*+\epsilon\boldsymbol{\varphi})]\\
&=L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]+\frac{\epsilon^2}{2}\frac{1}{K}\sum_{k\in[K]}\int dX^{(1,t)}p(X^{(1,t)}|k)\frac{I_k}{(\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)})^2}+\mathcal{O}(\epsilon^3)\\
&=L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]+\frac{\epsilon^2}{2}J[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]+\mathcal{O}(\epsilon^3)\,.
\end{aligned}\tag{38}
$$

Here, we defined

$$
J[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]:=\frac{1}{K}\sum_{k\in[K]}\int dX^{(1,t)}p(X^{(1,t)}|k)\frac{I_k}{(\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)})^2}\,. \tag{39}
$$

In the following, we show that J is positive to obtain $L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*+\epsilon\boldsymbol{\varphi})]>L[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]$. We first see that J is non-negative. Because

$$
\begin{aligned}
I_k=&\sum_{l(\neq k)}(\boldsymbol{\varphi}^\top\cdot\nabla_{\boldsymbol{\theta}}\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*))^2e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\\
&+\sum_{\substack{l>l'\\ l,l'\neq k}}\Big\{\boldsymbol{\varphi}^\top\cdot(\nabla_{\boldsymbol{\theta}}\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)-\nabla_{\boldsymbol{\theta}}\hat{\lambda}_{kl'}(X^{(1,t)};\boldsymbol{\theta}^*))\Big\}^2e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}e^{-\hat{\lambda}_{kl'}(X^{(1,t)};\boldsymbol{\theta}^*)}\\
\geq&\sum_{l(\neq k)}(\boldsymbol{\varphi}^\top\cdot\nabla_{\boldsymbol{\theta}}\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*))^2e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}\,,
\end{aligned}\tag{40}
$$

we can bound J from below:

$$
\begin{aligned}
&J[\hat{\lambda}(X^{(1,t)};\boldsymbol{\theta}^*)]\\
&\geq\frac{1}{K}\sum_{k\in[K]}\sum_{l(\neq k)}\int dX^{(1,t)}p(X^{(1,t)}|k)\frac{e^{-\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*)}}{(\sum_{m\in[K]}e^{-\hat{\lambda}_{km}(X^{(1,t)};\boldsymbol{\theta}^*)})^2}(\boldsymbol{\varphi}^\top\cdot\nabla_{\boldsymbol{\theta}}\hat{\lambda}_{kl}(X^{(1,t)};\boldsymbol{\theta}^*))^2
\end{aligned}\tag{41}
$$

$(\geq 0)\,.$

Note that each term in (41) is non-negative; therefore, J is non-negative. We next show that J is non-zero to prove that $J > 0$. By assumption, $\exists k^*, l^* \in [K]$ such that $\forall \boldsymbol{\varphi} \neq \mathbf{0}$,

$$\begin{aligned} &\boldsymbol{\varphi}^\top \cdot \int dX^{(1,t)} p(X^{(1,t)}|k^*) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{k^*l^*}(X^{(1,t)}; \boldsymbol{\theta}^*) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{k^*l^*}(X^{(1,t)}; \boldsymbol{\theta}^*)^\top \cdot \boldsymbol{\varphi} \\ &= \int dX^{(1,t)} p(X^{(1,t)}|k^*) (\boldsymbol{\varphi}^\top \cdot \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{k^*l^*}(X^{(1,t)}; \boldsymbol{\theta}^*))^2 \neq 0 \, . \\ &\therefore \int dX^{(1,t)} p(X^{(1,t)}|k^*) \frac{e^{-\hat{\lambda}_{k^*l^*}(X^{(1,t)};\boldsymbol{\theta}^*)}}{(\sum_{m \in [K]} e^{-\hat{\lambda}_{k^*m}(X^{(1,t)};\boldsymbol{\theta}^*)})^2} (\boldsymbol{\varphi}^\top \cdot \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{k^*l^*}(X^{(1,t)}; \boldsymbol{\theta}^*))^2 \neq 0 \end{aligned} \tag{42}$$

because

$$\frac{e^{-\hat{\lambda}_{k^*l^*}(X^{(1,t)};\boldsymbol{\theta}^*)}}{(\sum_{m \in [K]} e^{-\hat{\lambda}_{k^*m}(X^{(1,t)};\boldsymbol{\theta}^*)})^2} > 0 \, .$$

Therefore, at least one term in (41) is non-zero, meaning (41) $\neq 0$ and $J[\hat{\lambda}(X^{(1,t)}; \boldsymbol{\theta}^*)] > 0$. Thus, we conclude that $L[\hat{\lambda}(X^{(1,t)}; \boldsymbol{\theta}^* + \epsilon\boldsymbol{\varphi})] > L[\hat{\lambda}(X^{(1,t)}; \boldsymbol{\theta}^*)]$ via (38).

Now, we have proven that $L(\boldsymbol{\theta}) > L(\boldsymbol{\theta}^*)$ in the vicinity of $\boldsymbol{\theta}^*$. For the other $\boldsymbol{\theta} (\notin \Theta^*)$, the inequality $L(\boldsymbol{\theta}) > L(\boldsymbol{\theta}^*)$ immediately follows from Lemma C.1 because $\hat{\lambda}$ is not equal to λ for such $\boldsymbol{\theta} \notin \Theta^*$ and λ is the unique minimum of $L[\hat{\lambda}]$. This concludes the proof. □

Finally, we prove Theorem 3.1 with the help of Lemma C.2.

Proof. To prove the consistency, we show that $P(\hat{\boldsymbol{\theta}}_S \notin \Theta^*) (= P(\{\omega \in \Omega | \hat{\boldsymbol{\theta}}_S(\omega) \notin \Theta^*\})) \xrightarrow{M \to \infty} 0$, where $\hat{\boldsymbol{\theta}}_S$ is the empirical risk minimizer, M is the sample size, P is the probability measure, and Ω is the sample space of the underlying probability space. By Lemma C.2, if $\boldsymbol{\theta} \notin \Theta^*$, then there exists $\delta > 0$ such that $L(\boldsymbol{\theta}) > L(\boldsymbol{\theta}^*) + \delta(\boldsymbol{\theta})$. Therefore,

$$\begin{aligned} &\{\omega \in \Omega | \hat{\boldsymbol{\theta}}_S(\omega) \notin \Theta^*\} \subset \{\omega \in \Omega | L(\hat{\boldsymbol{\theta}}_S(\omega)) > L(\boldsymbol{\theta}^*) + \delta(\hat{\boldsymbol{\theta}}_S)\} \\ &\therefore \; P\left(\hat{\boldsymbol{\theta}}_S \notin \Theta^*\right) \leq P\left(L(\hat{\boldsymbol{\theta}}_S) > L(\boldsymbol{\theta}^*) + \delta(\hat{\boldsymbol{\theta}}_S)\right) \, . \end{aligned} \tag{43}$$

We bound the right-hand side in the following.

$$\begin{aligned} L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*) &= L(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\boldsymbol{\theta}^*) + \hat{L}_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*) \\ &\leq L(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\hat{\boldsymbol{\theta}}_S) + \hat{L}_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*) \, . \end{aligned}$$

Therefore,

$$\begin{aligned} L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*) &= |L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*)| \\ &\leq |L(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\hat{\boldsymbol{\theta}}_S)| + |\hat{L}_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*)| \\ &\leq \; 2 \sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| \, . \end{aligned}$$

Thus,

$$\delta(\hat{\boldsymbol{\theta}}_S) < L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*) \Longrightarrow \delta(\hat{\boldsymbol{\theta}}_S) < \; 2 \sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| \, .$$

Hence,

$$P\Big(L(\hat{\boldsymbol{\theta}}_S) > L(\boldsymbol{\theta}^*) + \delta(\hat{\boldsymbol{\theta}}_S)\Big) \leq P\left(\delta(\hat{\boldsymbol{\theta}}_S) < \; 2 \sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right|\right) \, .$$

By the assumption that $\hat{L}_S(\boldsymbol{\theta})$ converges in probability uniformly over $\boldsymbol{\theta}$ to $L(\boldsymbol{\theta})$, the right-hand side is bounded above by an arbitrarily small $\epsilon > 0$ for sufficiently large M:

$$P\left(\delta(\hat{\boldsymbol{\theta}}_S) < \; 2 \sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right|\right) < \epsilon \, . \tag{44}$$

Combining (43) and (44), we conclude that $\forall \epsilon > 0, \exists n \in \mathbb{N}$ s.t. $\forall M > n, P(\hat{\boldsymbol{\theta}}_S \notin \Theta^*) < \epsilon$. □

D. Modified LSEL and Logistic Loss

In this appendix, we first discuss the effect of the prior ratio term $\log(\hat{p}(y = k)/\hat{p}(y = l)) =: \log \hat{\nu}_{kl}$ in the M-TANDEM and M-TANDEMwO formula (Appendix D.1). We then define the logistic loss used in the main text (Appendix D.2).

D.1. Modified LSEL and Consistency

In the main text, we ignore the prior ratio term $\log \hat{\nu}_{kl}$ (Section 3.4). Strictly speaking, this is equivalent to the definition of the following *modified LSEL (modLSEL)*:

$$L_{\mathrm{modLSEL}}[\tilde{\lambda}] := \frac{1}{T} \sum_{t \in [T]} \underset{\substack{(X^{(1,t)}, y) \\ \sim P(X^{(1,t)}, y)}}{\mathbb{E}} \left[\log(1 + \sum_{k (\neq y)} \nu_{yk}^{-1} e^{-\tilde{\lambda}_{yk}(X^{(1,t)})}) \right]$$
$$= \frac{1}{T} \sum_{t \in [T]} \sum_{y \in [K]} \int dX^{(1,t)} p(X^{(1,t)} | y) p(y) \log(1 + \sum_{k (\neq y)} \nu_{yk}^{-1} e^{-\tilde{\lambda}_{yk}(X^{(1,t)})}) \tag{45}$$

where $\nu_{kl} = p(y = k)/p(y = l)$ $(k, l \in [K])$ is the prior ratio matrix. The empirical approximation of L_{modLSEL} is

$$\hat{L}_{\mathrm{modLSEL}}(\boldsymbol{\theta}; S) := \frac{1}{MT} \sum_{i \in [M]} \sum_{t \in [T]} \log(1 + \sum_{k (\neq y_i)} \hat{\nu}_{y_i k}^{-1} e^{-\hat{\lambda}_{y_i k}(X_i^{(1,t)}; \boldsymbol{\theta})}) \tag{46}$$

where $\hat{\nu}_{kl} := M_k / M_l$ $(k, l \in [K])$. M_k denotes the sample size of class k, i.e., $M_k := |\{i \in [M] | y_i = k\}|$. (45) is a generalization of the logit adjustment (Menon et al., 2021) to the LSEL and helps us to train neural networks on imbalanced datasets.

We can prove the consistency even for the modified LSEL, given an additional assumption (d):

Theorem D.1 (Consistency of the modLSEL)**.** *Let $L(\boldsymbol{\theta})$ and $\hat{L}_S(\boldsymbol{\theta})$ denote $L_{modLSEL}[\hat{\lambda}(\cdot; \boldsymbol{\theta})]$ and $\hat{L}_{modLSEL}(\boldsymbol{\theta}; S)$, respectively. Let $\hat{\boldsymbol{\theta}}_S$ be the empirical risk minimizer of $\hat{L}_S$; namely, $\hat{\boldsymbol{\theta}}_S := \mathrm{argmin}_{\boldsymbol{\theta}} \hat{L}_S(\boldsymbol{\theta})$. Let $\Theta^* := \{\boldsymbol{\theta}^* \in \mathbb{R}^{d_\theta} | \hat{\lambda}(X^{(1,t)}; \boldsymbol{\theta}^*) = \lambda(X^{(1,t)})\ (\forall t \in [T])\}$ be the target parameter set. Assume, for simplicity of proof, that each $\boldsymbol{\theta}^*$ is separated in Θ^*; i.e., $\exists \delta > 0$ such that $B(\boldsymbol{\theta}^*; \delta) \cap B(\boldsymbol{\theta}^{*\prime}; \delta) = \emptyset$ for arbitrary $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}^{*\prime} \in \Theta^*$, where $B(\boldsymbol{\theta}; \delta)$ denotes an open ball at center $\boldsymbol{\theta}$ with radius δ. Define*

$$\hat{L}'_S(\boldsymbol{\theta}) := \frac{1}{MT} \sum_{i \in [M]} \sum_{t \in [T]} \log \left(1 + \sum_{k (\neq y_i)} \nu_{y_i k}^{-1} e^{-\hat{\lambda}_{y_i k}(X_i^{(1,t)}; \boldsymbol{\theta})} \right) \tag{47}$$

($\hat{\nu}$ is replaced by ν in $\hat{L}_S$). Assume the following three conditions:

(a) $\forall k, l \in [K]$, $\forall t \in [T]$, $p(X^{(1,t)} | k) = 0 \iff p(X^{(1,t)} | l) = 0$.

(b′) $\sup_{\boldsymbol{\theta}} |\hat{L}'_S(\boldsymbol{\theta}) - L(\boldsymbol{\theta})| \xrightarrow[M \to \infty]{P} 0$; i.e., $\hat{L}'_S(\boldsymbol{\theta})$ converges in probability uniformly over $\boldsymbol{\theta}$ to $L(\boldsymbol{\theta})$.[7]

(c) For all $\theta^* \in \Theta^*$, there exist $t \in [T]$, $k \in [K]$ and $l \in [K]$, such that the following $d_\theta \times d_\theta$ matrix is full-rank:

$$\int dX^{(1,t)} p(X^{(1,t)} | k) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{kl}(X^{(1,t)}; \boldsymbol{\theta}^*) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{kl}(X^{(1,t)}; \boldsymbol{\theta}^*)^\top . \tag{48}$$

(d) $\sup_{\boldsymbol{\theta}} |\hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta})| \xrightarrow[M \to \infty]{P} 0$.

Then, $P(\hat{\boldsymbol{\theta}}_S \notin \Theta^*) \xrightarrow{M \to \infty} 0$; i.e., $\hat{\boldsymbol{\theta}}_S$ converges in probability into Θ^*.

(b) is now modified to (b′) and (d) is added to the assumptions of Theorem 3.1. (b′) can be satisfied under the standard assumptions of the uniform law of large numbers. (d) may be proven under some appropriate assumptions, but we simply accept it here.

Proof. We prove all the statements only for an arbitrary $t \in [T]$ and omit $\frac{1}{T} \sum_{t \in [T]}$ from L and $\hat{L}_S$, as is done in the proof of Theorem 3.1. In the same way as Appendix C, we first provide two lemmas:

Lemma D.1 (Non-parametric estimation)**.** *Assume that for all $k, l \in [K]$, $p(X^{(1,t)} | k) = 0 \iff p(X^{(1,t)} | l) = 0$. Then, $L[\tilde{\lambda}]$ attains the unique minimum at $\tilde{\lambda} = \lambda$.*

Lemma D.2 (Θ^* minimizes L)**.** *Assume that for all $\theta^* \in \Theta^*$, there exist $k^* \in [K]$ and $l^* \in [K]$, such that the following $d_\theta \times d_\theta$ matrix is full-rank:*

$$\int dX^{(1,t)} p(X^{(1,t)} | k^*) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{k^* l^*}(X^{(1,t)}; \boldsymbol{\theta}^*) \nabla_{\boldsymbol{\theta}^*} \hat{\lambda}_{k^* l^*}(X^{(1,t)}; \boldsymbol{\theta}^*)^\top . \tag{49}$$

Then, for any $\boldsymbol{\theta} \notin \Theta^*$,

$$L(\boldsymbol{\theta}) > L(\boldsymbol{\theta}^*) \quad (\forall \boldsymbol{\theta}^* \in \Theta^*) ,$$

meaning that $\Theta^* = \mathrm{argmin}_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$.

We skip the proofs because they are completely parallel to the proof of Lemma C.1 and Lemma C.2.

To prove the consistency, we show that $P(\hat{\boldsymbol{\theta}}_S \notin \Theta^*) (= P(\{\omega \in \Omega | \hat{\boldsymbol{\theta}}_S(\omega) \notin \Theta^*\})) \xrightarrow{M \to \infty} 0$, where $\hat{\boldsymbol{\theta}}_S$ is the empirical risk minimizer on the random training set S, M is the sample size, P is the probability measure, and Ω is the sample space of the underlying probability space. By

[7] Specifically, $\forall \epsilon > 0, P(\sup_{\boldsymbol{\theta}} |\hat{L}'_S(\boldsymbol{\theta}) - L(\boldsymbol{\theta})| > \epsilon) \xrightarrow{M \to \infty} 0$.

Lemma D.2, if $\boldsymbol{\theta} \notin \Theta^*$, then there exists $\delta > 0$ such that $L(\boldsymbol{\theta}) > L(\boldsymbol{\theta}^*) + \delta(\boldsymbol{\theta})$. Therefore,

$$\{\omega \in \Omega | \hat{\boldsymbol{\theta}}_S(\omega) \notin \Theta^*\} \subset \{\omega \in \Omega | L(\hat{\boldsymbol{\theta}}_S(\omega)) > L(\boldsymbol{\theta}^*) + \delta(\hat{\boldsymbol{\theta}}_S)\}$$
$$\therefore \; P\left(\hat{\boldsymbol{\theta}}_S \notin \Theta^*\right) \leq P\left(L(\hat{\boldsymbol{\theta}}_S) > L(\boldsymbol{\theta}^*) + \delta(\hat{\boldsymbol{\theta}}_S)\right) . \tag{50}$$

We bound the right-hand side in the following.

$$\begin{aligned} L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*) &= L(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\boldsymbol{\theta}^*) + \hat{L}_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*) \\ &\leq L(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\hat{\boldsymbol{\theta}}_S) + \hat{L}_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*) \\ &= L(\hat{\boldsymbol{\theta}}_S) - \hat{L}'_S(\hat{\boldsymbol{\theta}}_S) + \hat{L}'_S(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\hat{\boldsymbol{\theta}}_S) \\ &\quad + \hat{L}_S(\boldsymbol{\theta}^*) - \hat{L}'_S(\boldsymbol{\theta}^*) + \hat{L}'_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*) . \end{aligned}$$

Therefore,

$$\begin{aligned} L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*) &= |L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*)| \\ &\leq |L(\hat{\boldsymbol{\theta}}_S) - \hat{L}'_S(\hat{\boldsymbol{\theta}}_S)| + |\hat{L}'_S(\hat{\boldsymbol{\theta}}_S) - \hat{L}_S(\hat{\boldsymbol{\theta}}_S)| \\ &\quad + |\hat{L}_S(\boldsymbol{\theta}^*) - \hat{L}'_S(\boldsymbol{\theta}^*)| + |\hat{L}'_S(\boldsymbol{\theta}^*) - L(\boldsymbol{\theta}^*)| \\ &\leq \; 2\sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| + 2\sup_{\boldsymbol{\theta}} \left| \hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| \end{aligned}$$

Thus,

$$\delta(\hat{\boldsymbol{\theta}}_S) < L(\hat{\boldsymbol{\theta}}_S) - L(\boldsymbol{\theta}^*) \Longrightarrow \delta(\hat{\boldsymbol{\theta}}_S) < \; 2\sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| + 2\sup_{\boldsymbol{\theta}} \left| \hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right|$$

Hence,

$$P(L(\hat{\boldsymbol{\theta}}_S) > L(\boldsymbol{\theta}^*) + \delta(\hat{\boldsymbol{\theta}}_S)) \leq P\Big(\delta(\hat{\boldsymbol{\theta}}_S) < \; 2\sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| + 2\sup_{\boldsymbol{\theta}} \left| \hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| \Big) .$$

Recall that by assumption, $2\sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right|$ and $2\sup_{\boldsymbol{\theta}} \left| \hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right|$ converge in probability to zero; hence, $2\sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| + 2\sup_{\boldsymbol{\theta}} \left| \hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right|$ converge in probability to zero because in general,

$$a_n \xrightarrow[n\to\infty]{P} 0 \;\text{ and }\; b_n \xrightarrow[n\to\infty]{P} 0 \Longrightarrow a_n + b_n \xrightarrow[n\to\infty]{P} 0 ,$$

where $\{a_n\}$ and $\{b_n\}$ are sequences of random variables. By definition of convergence in probability, for sufficiently large sample sizes M,

$$P\Big(\delta(\hat{\boldsymbol{\theta}}_S) < \; 2\sup_{\boldsymbol{\theta}} \left| L(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| + 2\sup_{\boldsymbol{\theta}} \left| \hat{L}'_S(\boldsymbol{\theta}) - \hat{L}_S(\boldsymbol{\theta}) \right| \Big) < \epsilon . \tag{51}$$

Combining (50) and (51), we conclude that $\forall \epsilon > 0, \exists n \in \mathbb{N}$ s.t. $\forall M > n, P(\hat{\boldsymbol{\theta}}_S \notin \Theta^*) < \epsilon$. □

D.2. Logistic Loss and Consistency

We use the following logistic loss for DRME in the main text:

$$\hat{L}_{\text{logistic}}(\boldsymbol{\theta}; S) := \frac{1}{KT} \sum_{k\in[K]} \sum_{t\in[T]} \frac{1}{M_k} \sum_{i\in I_k} \frac{1}{K-1} \sum_{l(\neq k)} \log(1 + e^{-\hat{\lambda}_{kl}(X_i^{(1,t)};\boldsymbol{\theta})}) . \tag{52}$$

Note that $\hat{L}_{\text{logistic}}$ resembles the LSEL but is defined as the sum of the logarithm of the exponential (sum-log-exp), not log-sum-exp. We can prove the consistency and the proof is completely parallel to, and even simpler than, that of Theorem 3.1; therefore, we omit the proof to avoid redundancy. $\hat{L}_{\text{logistic}}$ approaches

$$L_{\text{logistic}}[\lambda] = \frac{1}{KT} \sum_{k\in[K]} \sum_{t\in[T]} \mathbb{E}_{X^{(1,t)} \sim p(X^{(1,t)}|y=k)} \left[\frac{1}{K-1} \sum_{l(\neq k)} \log(1 + e^{-\lambda_{kl}(X^{(} \right. \tag{53}$$

as $M \to \infty$.

Additionally, we can define the modified logistic loss as

$$\hat{L}_{\text{modlogistic}}(\boldsymbol{\theta}; S) = \frac{1}{MT} \sum_{i\in[M]} \sum_{t\in[T]} \frac{1}{K-1} \sum_{l(\neq y_i)} \log(1 + \hat{\nu}_{y_i l}^{-1} e^{-\hat{\lambda}_{y_i l}(X_i^{(1,t)}} \tag{54}$$

We can prove the consistency in a similar way to Theorem D.1. $\hat{L}_{\text{modlogistic}}$ approaches

$$L_{\text{modlogistic}}[\lambda] = \frac{1}{T} \sum_{t\in[T]} \mathbb{E}_{(X^{(1,t)},y) \sim p(X^{(1,t)},y)} \left[\frac{1}{K-1} \sum_{l(\neq y)} \log(1 + \nu_{yl}^{-1} e^{-\lambda_{yl}(X^{(1,t)}} \right. \tag{55}$$

as $M \to \infty$.

Our empirical study shows that the LSEL is better than the logistic loss (Figure 2 and Appendix E), potentially because the LSEL weighs hard classes more than the logistic loss (Section 3.3.2).

E. Performance Comparison of LSEL and Logistic Loss

Figure 6 provides the performance comparison of the LSEL (2) and the logistic loss (52). The LSEL is consistently better than or comparable with the logistic loss, which is potentially because of the hard class weighting effect.

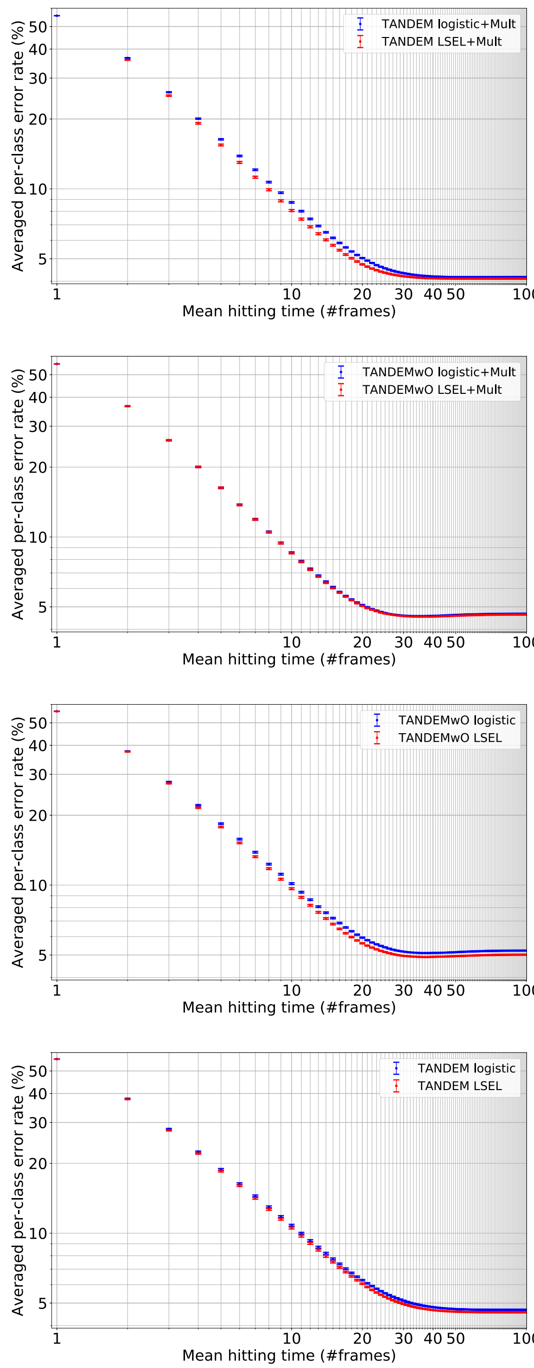

Figure 6. **LSEL v.s. Logistic Loss.** The LSEL is consistently better than or at least comparable with the logistic loss. The dataset is NMNIST-100f. The error bar is the SEM. "TANDEM" means that the model is trained with the M-TANDEM formula, "TANDEMwO" means that the model is trained with the M-TANDEMwO formula, and "Mult" means that the multiplet loss is simultaneously used.

F. M-TANDEM vs. M-TANDEMwO Formulae

The M-TANDEM and M-TANDEMwO formulae enable to efficiently train RNNs on long sequences, which often cause vanishing gradients (Hochreiter et al., 2001). In addition, if a class signature is localized within a short temporal interval, not all frames can be informative (Xing et al., 2011; McGovern et al., 2011; Ghalwash & Obradovic, 2012; Ghalwash et al., 2013; Ghalwash et al., 2014; Karim et al., 2019). The M-TANDEM and M-TANDEMwO formulae alleviate these problems.

Figure 7 highlights the differences between the M-TANDEM and M-TANDEMwO formulae. The M-TANDEM formula covers all the timestamps, while the M-TANDEMwO formula only covers the last $N+1$ timestamps. In two-hypothesis testing, the M-TANDEM formula is the canonical generalization of Wald's i.i.d. SPRT, because for $N=0$ (i.i.d.), the M-TANDEM formula reduces to $\hat{\lambda}_{1,2}(X^{(1,T)}) = \sum_{t=1}^{T} \log(p(x^{(t)}|1)/p(x^{(t)}|2))$, which is used in the classical SPRT (Wald, 1945), while the M-TANDEMwO formula reduces to a sum of frame-by-frame scores when $N=0$.

Figure 8 compares the performance of the M-TANDEM and M-TANDEMwO formulae on three datasets: NMNIST, NMNIST-H, and NMNIST-100f. NMNIST (Ebihara et al., 2021) is similar to NMNIST-H but has much weaker noise. On relatively short sequences (NMNIST and NMNIST-H), the M-TANDEMwO formula is slightly better than or much the same as the M-TANDEM formula. On longer sequences (NMNIST-100f), the M-TANDEM formula outperforms the M-TANDEMwO formula; the latter slightly and gradually increases the error rate in the latter half of the sequences. In summary, the performance of the M-TANDEM and M-TANDEMwO formulae depends on the sequence length of the training datasets, and we recommend using the M-TANDEM formula as the first choice for long sequences ($\gtrsim 100$ frames) and the M-TANDEM formula for short sequences (~ 10 frames).

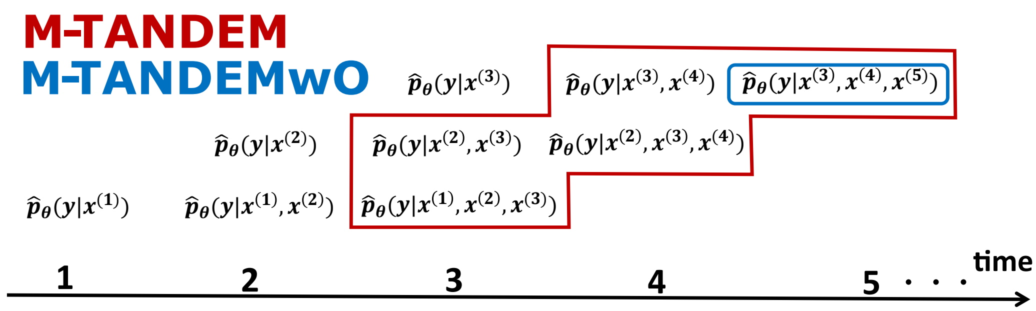

Figure 7. **M-TANDEM v.s. M-TANDEMwO with $N = 2$.** The posterior densities encircled in red and blue are used in the M-TANDEM and M-TANDEMwO formulae, respectively. We can see that the M-TANDEM formula covers all the frames, while the M-TANDEMwO formula covers only the last $N + 1$ frames.

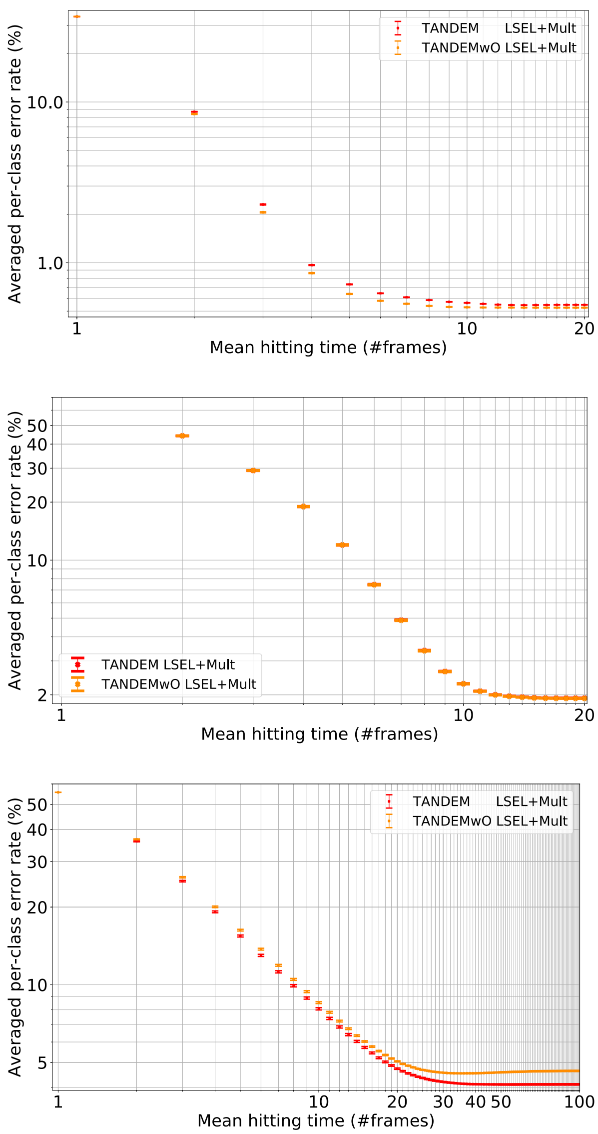

Figure 8. **M-TANDEM vs. M-TANDEMwO. Top: NMNIST. Middle: NMNIST-H. Bottom: NMNIST-100f.** TANDEM and TANDEMwO means that the model is trained with the M-TANDEM and M-TANDEMwO formulae, respectively. Mult means that the multiplet loss is simultaneously used.

G. Proofs Related to Guess-Aversion

G.1. Proof of Theorem 3.2

Proof. For any $k, l \in [K]$ $(k \neq l)$ and any $\boldsymbol{s} \in \mathcal{S}_k$, $e^{-(s_k - s_l)}$ is less than 1 by definition of $\mathcal{S}_k$. Therefore, for any $k \in [K]$, any $\boldsymbol{s} \in \mathcal{S}_k$, any $\boldsymbol{s}' \in \mathcal{A}$, and any cost matrix C,

$$\ell(\boldsymbol{s}, k; C) := C_k \log(1 + \sum_{l(\neq k)} e^{-(s_k - s_l)}) < C_k \log(1 + \sum_{l(\neq k)} 1) = \ell(\boldsymbol{s}', k; C)\,.$$

□

G.2. NGA-LSEL Is Not Guess-Averse

The NGA-LSEL is $\ell(\boldsymbol{s}, y; C) = \sum_{k(\neq y)} C_{y,l} \log(1 + \sum_{l(\neq k)} e^{s_l - s_k})$ (Section 3.3.3). We prove that the NGA-LSEL is not guess-averse by providing a counter example.

Proof. Assume that $K = 3$, $C_{kl} = 1$ $(k \neq l)$, $\boldsymbol{s}(X_i^{(1,t)}) = (3, 2, -100)^\top$, and $y_i = 1$. Then,

$$\ell(\boldsymbol{s}(X_i^{(1,t)}), y_i; C) = \log(1 + e^{s_1 - s_2} + e^{s_3 - s_2}) + \log(1 + e^{s_1 - s_3} + e^{s_2 - s_3})$$
$$= \log(1 + e^1 + e^{-102}) + \log(1 + e^{103} + e^{102})$$
$$> \log(3) + \log(3) = \ell(\boldsymbol{0}, y_i; C)\,.$$

□

G.3. Another cost-sensitive LSEL

Alternatively to $\hat{L}_{\mathrm{CLSEL}}$, we can define

$$\hat{L}_{\mathrm{LSCEL}}(\boldsymbol{\theta}, C; S) := \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} \log(1 + \sum_{l(\neq y_i)} C_{y_i l} e^{-\hat{\lambda}_{y_i l}(X_i^{(1,t)}; \boldsymbol{\theta})})\,. \tag{56}$$

$\hat{L}_{\mathrm{LSCEL}}$ reduces to $\hat{L}_{\mathrm{modLSEL}}$ when $C_{kl} = \hat{\nu}_{kl}^{-1}$. The following theorem shows that $\hat{L}_{\mathrm{LSCEL}}$ is guess-averse:

Theorem G.1. *$\hat{L}_{\mathrm{LSCEL}}$ is guess-averse, provided that the log-likelihood vector*

$$\Big(\log \hat{p}_{\boldsymbol{\theta}}(X^{(1,t)}|y = 1), \log \hat{p}_{\boldsymbol{\theta}}(X^{(1,t)}|y = 2), \ldots, \log \hat{p}_{\boldsymbol{\theta}}(X^{(1,t)}|y = K) \Big)^\top \in \mathbb{R}^K$$

is regarded as the score vector $\boldsymbol{s}(X^{(1,t)})$.

Proof. We use Lemma 1 in (Beijbom et al., 2014):

Lemma G.1 (Lemma 1 in (Beijbom et al., 2014))**.** *Let $\ell(\boldsymbol{s}, y; C) = \gamma(\sum_{k \in [K]} C_{yk} \phi(s_y - s_k))$, where $\gamma : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing function and $\phi : \mathbb{R} \to \mathbb{R}$ is a function such that for any $v > 0$, $\phi(v) < \phi(0)$. Then, ℓ is guess-averse.*

The statement of Theorem G.1 immediately follows by substituting $\gamma(v) = \log(1 + v)$ and $\phi(v) = e^{-v}$ into Lemma G.1. □

G.4. Cost-Sensitive Logistic Losses Are Guess-Averse

We additionally prove that the cost-sensitive logistic losses defined below are also guess-averse, which may be of independent interest. We define a cost-sensitive logistic loss as

$$\hat{L}_{\text{C-logistic}}(\boldsymbol{\theta}, C; S) := \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} \frac{1}{K-1} \sum_{l(\neq y_i)} C_{y_i l} \log \Big(1 + e^{-\hat{\lambda}_{y_i l}(X_i^{(1,t} \tag{57}$$

$\hat{L}_{\text{C-logistic}}$ reduces to $\hat{L}_{\text{logistic}}$ (defined in Appendix D.2) if $C_{kl} = C_k = M/KM_k$. $\hat{L}_{\text{C-logistic}}$ is guess-averse:

Theorem G.2. *$\hat{L}_{\text{C-logistic}}$ is guess-averse, provided that the log-likelihood vector*

$$\Big(\log \hat{p}(X^{(1,t)}|y = 1), \log \hat{p}(X^{(1,t)}|y = 2), \ldots, \log \hat{p}(X^{(1,t)}|y = K) \Big)^\top \in \mathbb{I} \tag{58}$$

is regarded as the score vector $\boldsymbol{s}(X^{(1,t)})$.

Proof. The proof is parallel to that of Theorem 3.2. For any $k, l \in [K]$ $(k \neq l)$ and any $\boldsymbol{s} \in \mathcal{S}_k$, $e^{-(s_k - s_l)}$ is less than 1 by definition of $\mathcal{S}_k$. Therefore, for any $k, l \in [K]$, any $\boldsymbol{s} \in \mathcal{S}_k$, any $\boldsymbol{s}' \in \mathcal{A}$, and any cost matrix C,

$$\ell(\boldsymbol{s}, k; C) := \frac{1}{K-1} \sum_{l(\neq k)} \log(1 + e^{-(s_k - s_l)})^{C_{kl}} < \frac{1}{K-1} \sum_{l(\neq k)} \log(1 + 1)$$

□

We also define

$$\hat{L}_{\text{logistic-C}}(\boldsymbol{\theta}, C; S) := \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=1}^{T} \frac{1}{K-1} \sum_{l(\neq y_i)} \log \Big(1 + C_{y_i l} e^{-\hat{\lambda}_{y_i l}(X_i^{(1,t} \tag{59}$$

$\hat{L}_{\text{logistic-C}}$ reduces to $\hat{L}_{\text{modlogistic}}$ if $C_{kl} = \hat{\nu}_{kl}^{-1}$. $\hat{L}_{\text{logistic-C}}$ is guess-averse:

Theorem G.3. *$\hat{L}_{\text{logistic-C}}$ is guess-averse, provided that the log-likelihood vector*

$$\Big(\log \hat{p}(X^{(1,t)}|y = 1), \log \hat{p}(X^{(1,t)}|y = 2), \ldots, \log \hat{p}(X^{(1,t)}|y = K) \Big)^\top \in \mathbb{I} \tag{60}$$

is regarded as the score vector $\boldsymbol{s}(X^{(1,t)})$.

To prove Theorem G.3, we first show the following lemma:

Lemma G.2. *Let*

$$\ell(\boldsymbol{s}, k, ; C) = \gamma\Big(\prod_{l \in [K]} (1 + C_{kl} \phi(s_k - s_l)) \Big)\,,$$

where $\gamma : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing function and $\phi : \mathbb{R} \to \mathbb{R}$ is a function such that for any $v > 0$, $\phi(v) < \phi(0)$. Then, ℓ is guess-averse.

Proof. For any $\boldsymbol{s} \in \mathcal{S}_k$ and $l \in [K]$,

$$\phi(s_k - s_l) < \phi(0)\,,$$

because $\phi(v) < \phi(0)$ and $s_k > s_l$ for all $v > 0$ and $l \in [K]$ ($l \neq k$). Therefore,

$$\prod_{l \in [K]} (1 + C_{kl}\phi(s_k - s_l)) < \prod_{l \in [K]} (1 + C_{kl}\phi(0))\,,$$

because $C_{kl} \geq 0$ for all $k, l \in [K]$ and $C_{kl} \neq 0$ for at least one $l (\neq k)$. Hence, for any $k \in [K]$, any $\boldsymbol{s} \in \mathcal{S}_k$, any $\boldsymbol{s}' \in \mathcal{A}$, and any cost matrix C, the monotonicity of γ shows that

$$\ell(\boldsymbol{s}, k; C) = \gamma\left(\prod_{k \in [K]} (1 + C_{yk}\phi(s_y - s_k))\right) < \gamma\left(\prod_{k \in [K]} (1 + C_{yk}\phi(0))\right) = \ell(\boldsymbol{s}', k; C)\,.$$

□

Proof of Theorem G.3. The statement immediately follows from Lemma G.2 by substituting $\gamma(v) = \log(v)$ and $\phi(v) = e^{-v}$. □

H. Ablation Study of Multiplet Loss and LSEL

Figure 9 shows the ablation study comparing the LSEL and the multiplet loss. The combination of the LSEL and the multiplet loss is statistically significantly better than either of the two losses (Appendix L). The multiplet loss also performs better than the LSEL. However, the independent use of the multiplet loss has drawbacks: The multiplet loss optimizes *all* the posterior densities output from the temporal integrator (magenta circles in Figure 4), while the LSEL uses the *minimum* posterior densities required to calculate the LLR matrix via the M-TANDEM or M-TANDEMwO formula. Therefore, the multiplet loss can lead to a suboptimal minimum for estimating the LLR matrix. In addition, the multiplet loss tends to suffer from the overconfidence problem (Guo et al., 2017), causing extreme values of LLRs.

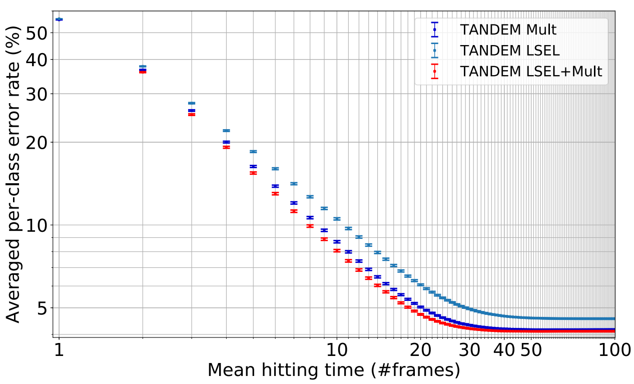

Figure 9. **Ablation Study of LSEL and Multiplet Loss on NMNIST-100f.** The error bars show SEM. The combination of the LSEL with the multiplet loss gives the best result. The other two curves represent the models trained with the LSEL only and the multiplet loss only. Details of the experiment and the statistical tests are given in Appendices I and L.

I. Details of Experiment and More Results

Our computational infrastructure is DGX-1. The fundamental libraries used in the experiment are Numpy (Harris et al., 2020), Scipy (Virtanen et al., 2020), Tensorflow 2.0.0 (Abadi et al., 2015) and PyTorch 1.2 (Paszke et al., 2019).

The train/test splitting of NMNIST-H and NMNIST-100f follows the standard one of MNIST (LeCun et al., 2010). The validation set is separated from the last 10,000 examples in the training set. The train/test splitting of UCF101 and HMDB51 follows the official splitting #1. The validation set is separated from the training set, keeping the class frequency. All the videos in UCF101 and HMDB51 are clipped or repeated to make their temporal length equal (50 and 79, respectively). See also our official code. All the pixel values are divided by 127.5 and then subtracted by 1 before training the feature extractor.

Hyperparameter tuning is performed with the TPE algorithm (Bergstra et al., 2011), the default algorithm of Optuna (Akiba et al., 2019). For optimizers, we use Adam, Momentum, (Loshchilov & Hutter, 2019) or RMSprop (Graves, 2013). Note that Adam and Momentum are not the originals ((Kingma & Ba, 2014) and (Rumelhart et al., 1986)), but AdamW and SGDW (Loshchilov & Hutter, 2019), which have a decoupled weight decay from the learning rate.

To obtain arbitrary points of the SAT curve, we compute the thresholds of MSPRT-TANDEM as follows. First, we compute all the LLR trajectories of the test examples. Second, we compute the maximum and minimum values of $|\hat{\lambda}(X_i^{(1,t)})|$ with respect to $i \in [M]$ and $t \in [T]$. Third, we generate the thresholds between the maximum and minimum. The thresholds are linearly uniformly separated. Forth, we run the MSPRT and obtain a two-dimensional point for each threshold ($x =$ mean hitting time, $y =$ averaged per-class error rate). Finally, we plot them on the speed-accuracy plane and linearly interpolate between two points with neighboring mean hitting times. If all the frames in a sequence are observed, the threshold of MSPRT-TANDEM is immediately collapsed to zero to force a decision.

I.1. Common Feature Extractor

NMNIST-H and NMNIST-100f We first train the feature extractor to extract the bottleneck features, which are then used to train the temporal integrator, LSTM-s/m, and EARLIEST. Therefore, all the models in Figure 5 (MSPRT-TANDEM, NP test, LSTM-s/m, and EARLIEST) share exactly the same feature extractor, ResNet-110 (He et al., 2016a;b) with the bottleneck feature dimensions set to 128. The total number of trainable parameters is 6,904,608.

Tables 1 and 2 show the search spaces of hyperparameters. The batch sizes are 64 and 50 for NMNIST-H and NMNIST-100f, respectively. The numbers of training iterations are 6,200 and 40,000 for NMNIST-H and NMNIST-100f, respectively. For each *tuning trial*, we train ResNet and evaluate its averaged per-class accuracy on the validation set per 200 training steps, and after all training iterations, we save the best averaged per-class accuracy in that tuning trial. After all tuning trials, we choose the best hyperparameter combination, which is shown in cyan letters in Tables 1 and 2. We train ResNet with those hyperparameters to extract 128-dimensional bottleneck features, which are then used for the temporal integrator, LSTM-s/m, and EARLIEST. The averaged per-class accuracies of the feature extractors trained on NMNIST-H and on NMNIST-100f are $\sim 84\%$ and 43%, respectively. The approximated runtime of a single tuning trial is 4.5 and 35 hours for NMNIST-H and NMNIST-100f, respectively, and the GPU consumption is 31 GBs for both datasets.

UCF101 and HMDB51 We use a pre-trained model without fine-tuning (Microsoft Vision ResNet-50 version 1.0.5 (Lenyk & Park)). The final feature dimensions are 2048.

Table 1. **Hyperparameter Search Space of Feature Extractor in Figure 5: NMNIST-H.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	$\{10^{-2}, 5{*}10^{-3}, 10^{-3}, 5{*}10^{-4}, 10^{-4}\}$
WEIGHT DECAY	$\{10^{-3}, 10^{-4}, 10^{-5}\}$
OPTIMIZER	{ ADAM, MOMENTUM, RMSPROP }
# TUNING TRIALS	96

Table 2. **Hyperparameter Search Space of Feature Extractor in Figure 5: NMNIST-100f.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	$\{10^{-2}, 5{*}10^{-3}, 10^{-3}, 5{*}10^{-4}, 10^{-4}, 10^{-5}\}$
WEIGHT DECAY	$\{10^{-3}, 10^{-4}, 10^{-5}\}$
OPTIMIZER	{ ADAM, MOMENTUM, RMSPROP }
# TUNING TRIALS	7

I.2. Figure 5: NMNIST-H

MSPRT-TANDEM and NP test The approximation formula is the M-TANDEMwO formula. The loss function consists of the LSEL and the multiplet loss. The temporal integrator is Peephole LSTM (Gers & Schmidhuber, 2000) with the hidden state dimensions set to 128 followed by a fully-connected layer to output logits for classification. The temporal integrator has 133,760 trainable parameters. The batch size is fixed to 500. The number of training iterations is 5,000.

Table 3 shows the search space of hyperparameters. For each tuning trial, we train the temporal integrator and evaluate its mean averaged per-class accuracy[8] per every 50 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Table 3 in cyan letters.

After fixing the hyperparameters, we then train LSTM arbitrary times. During each statistics trial, we train LSTM with the best fixed hyperparameters and evaluate its mean averaged per-class accuracy at every 50 training iterations. After all training iterations, we save the best weight parameters in terms of the mean averaged per-class accuracy. After all statistics trials, we can plot the SAT "points" with integer hitting times. The approximated runtime of one statistic trial is 3.5 hours, and the GPU consumption is 1.0 GB.

Table 3. **Hyperparameter Search Space of MSPRT-TANDEM in Figure 5: NMNIST-H.** The best hyperparameter combination is highlighted in cyan. γ is defined as $L_{\rm total} = L_{\rm mult} + \gamma L_{\rm LSEL}$.

ORDER	{0,1,5,10, 15,19 }
LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^2, 10^3}
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	500

LSTM-s/m The backbone model is Peephole LSTM with the hidden state dimensions set to 128 followed by a fully-connected layer to output logits for classification. LSTM has 133,760 trainable parameters. The batch size is fixed to 500. The number of training iterations is 5,000.

Tables 4 and 5 show the search spaces of hyperparameters. For each tuning trial, we train LSTM and evaluate its mean averaged per-class accuracy per every 50 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 4 and 5 in cyan letters.

After fixing the hyperparameters, we then train LSTM arbitrary times. During each statistics trial, we train LSTM with the best fixed hyperparameters and evaluate its mean averaged per-class accuracy per every 50 training iterations. After all training iterations, we save the best weight parameters in terms of the mean averaged per-class accuracy. After all statistics trials, we can plot the SAT "points" with integer hitting times. The approximated runtime of one statistics trial is 3.5 hours, and the GPU consumption is 1.0 GB.

Table 4. **Hyperparameter Search Space of LSTM-s in Figure 5: NMNIST-H.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\rm total} = L_{\rm cross\text{-}entropy} + \gamma L_{\rm ranking}$ (Ma et al., 2016).

LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^2 }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	500

Table 5. **Hyperparameter Search Space of LSTM-m in Figure 5: NMNIST-H.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\rm total} = L_{\rm cross\text{-}entropy} + \gamma L_{\rm ranking}$ (Ma et al., 2016).

LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^2 }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	500

EARLIEST The main backbone is LSTM (Hochreiter & Schmidhuber, 1997) with the hidden state dimensions set to 128. The whole architecture has 133,646 trainable parameters. The batch size is 1000. The number of training iterations is 20,000. EARLIEST has a parameter λ (Not to be confused with the LLR matrix) that controls the speed-accuracy tradeoff. A larger λ gives faster and less accurate decisions, and a smaller λ gives slower and more accurate decisions. We train EARLIEST with two different λ's: 10^{-2} and 10^2.

Tables 6 and 7 show the search spaces of hyperparameters. For each tuning trial, we train EARLIEST and evaluate its averaged per-class accuracy per every 500 training iterations. After all training iterations, we save the best averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 6 and 7 in cyan letters.

After fixing the hyperparameters, we then train EARLIEST arbitrary times. During each statistics trial, we train EARLIEST with the best fixed hyperparameters and evaluate its mean averaged per-class accuracy per every 500 training iterations. After all training iterations, we save the best weight parameters in terms of the mean averaged per-class accuracy. After all statistics trials, we can plot the SAT "points." Note that EARLIEST cannot change the decision policy after training, and thus one statistics trial gives only one point on the SAT graph; therefore, several statistics

[8] 1. Compute LLRs for all frames; 2. Run MSPRT with threshold = 0 and compute framewise averaged per-class accuracy; 3. Compute the arithmetic mean of the framewise averaged per-class accuracy.

trials give only one point with an error bar. The approximated runtime of one statistics trial is 12 hours, and the GPU consumption is 1.4 GBs.

Table 6. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-2}$ in Figure 5: NMNIST-H.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	500

Table 7. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{2}$ in Figure 5: NMNIST-H.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	500

I.3. Figure 5: NMNIST-100f

MSPRT-TANDEM and NP test The approximation formula is the M-TANDEM formula. The loss function consists of the LSEL and the multiplet loss. The temporal integrator is Peephole LSTM (Gers & Schmidhuber, 2000) with the hidden state dimensions set to 128 followed by a fully-connected layer to output logits for classification. The temporal integrator has 133,760 trainable parameters. The batch size is fixed to 100. The number of training iterations is 5,000.

Table 8 shows the search space of hyperparameters. For each tuning trial, we train the temporal integrator and evaluate its mean averaged per-class accuracy per every 200 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Table 8 in cyan letters. The approximated runtime of one statistics trial is 1 hour, and the GPU consumption is 8.7 GBs.

LSTM-s/m The backbone model is Peephole LSTM with the hidden state dimensions set to 128 followed by a fully-connected layer to output logits for classification. LSTM has 133,760 trainable parameters. The batch size is fixed to 500. The number of training iterations is 5,000.

Tables 9 and 10 show the search spaces of hyperparameters. For each tuning trial, we train LSTM and evaluate its mean averaged per-class accuracy per every 100 training iterations.

Table 8. **Hyperparameter Search Space of MSPRT-TANDEM in Figure 5: NMNIST-100f.** The best hyperparameter combination is highlighted in cyan. γ is defined as $L_{\text{total}} = L_{\text{mult}} + \gamma L_{\text{LSEL}}$.

ORDER	{ 0, 25, 50, 75, 99 }
LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^{2}, 10^{3}}
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	200

After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 9 and 10 in cyan letters.

The approximated runtime of one statistics trial is 5 hours, and the GPU consumption is 2.6 GBs.

Table 9. **Hyperparameter Search Space of LSTM-s in Figure 5: NMNIST-100f.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\text{total}} = L_{\text{cross-entropy}} + \gamma L_{\text{ranking}}$ (Ma et al., 2016).

LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^{2} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	200

Table 10. **Hyperparameter Search Space of LSTM-m in Figure 5: NMNIST-100f.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\text{total}} = L_{\text{cross-entropy}} + \gamma L_{\text{ranking}}$ (Ma et al., 2016).

LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^{2} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	200

EARLIEST The main backbone is LSTM (Hochreiter & Schmidhuber, 1997) with the hidden state dimensions set to 128. The whole architecture has 133,646 trainable parameters. The batch size is 1000. The number of training iterations is 20,000. We train EARLIEST with two different λ's: 10^{-2} and 10^{-4}.

Tables 11 and 12 show the search spaces of hyperparameters. For each tuning trial, we train EARLIEST and evaluate its averaged per-class accuracy per every 500 training iterations.

After all training iterations, we save the best averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 11 and 12 in cyan letters. The approximated runtime of a single tuning trial is 14 hours, and the GPU consumption is 2.0 GBs.

Table 11. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-2}$ in Figure 5: NMNIST-100f.** The best hyperparameter combination is highlighted in cyan.

Learning rate	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
Optimizer	{ Adam, RMSprop }
# tuning trials	200

Table 12. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-4}$ in Figure 5: NMNIST-100f.** The best hyperparameter combination is highlighted in cyan.

Learning rate	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
Optimizer	{ Adam, RMSprop }
# tuning trials	200

I.4. Figure 5: UCF101

MSPRT-TANDEM and NP test The approximation formula is the M-TANDEM formula. The loss function consists of the LSEL and the multiplet loss. The temporal integrator is Peephole LSTM (Gers & Schmidhuber, 2000) with the hidden state dimensions set to 256 followed by a fully-connected layer to output logits for classification. The temporal integrator has 2,387,456 trainable parameters. The batch size is fixed to 256. The number of training iterations is 10,000. We use the effective number (Cui et al., 2019) as the cost matrix of the LSEL, instead of $1/M_k$, to avoid over-emphasizing the minority class and to simplify the parameter tuning (only one extra parameter β is introduced).

Table 13 shows the search space of hyperparameters. For each tuning trial, we train the temporal integrator and evaluate its mean averaged per-class accuracy per every 200 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Table 13 in cyan letters. The approximated runtime of one statistics trial is 8 hours, and the GPU consumption is 16–32 GBs.

Table 13. **Hyperparameter Search Space of MSPRT-TANDEM in Figure 5: UCF101.** The best hyperparameter combination is highlighted in cyan. γ is defined as $L_{\text{total}} = L_{\text{mult}} + \gamma L_{\text{LSEL}}$. β controls the cost matrix (Cui et al., 2019).

Order	{ 0, 10, 25, 40, 49 }
Learning rate	{ 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^2 }
Optimizer	{ Adam, RMSprop }
β	{ 0.99, 0.999, 0.9999, 0.99999, 1. }
# tuning trials	100

LSTM-s/m The backbone model is Peephole LSTM with the hidden state dimensions set to 256 followed by a fully-connected layer to output logits for classification. LSTM has 2,387,456 trainable parameters. The batch size is fixed to 256. The number of training iterations is 5,000.

Tables 14 and 15 show the search spaces of hyperparameters. For each tuning trial, we train LSTM and evaluate its mean averaged per-class accuracy per every 200 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 14 and 15 in cyan letters. The approximated runtime of one statistics trial is 3 hours, and the GPU consumption is 2.6 GBs.

Table 14. **Hyperparameter Search Space of LSTM-s in Figure 5: UCF101.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\text{total}} = L_{\text{cross-entropy}} + \gamma L_{\text{ranking}}$ (Ma et al., 2016).

Learning rate	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^2 }
Optimizer	{ Adam, RMSprop }
# tuning trials	100

Table 15. **Hyperparameter Search Space of LSTM-m in Figure 5: UCF101.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\text{total}} = L_{\text{cross-entropy}} + \gamma L_{\text{ranking}}$ (Ma et al., 2016).

Learning rate	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^2 }
Optimizer	{ Adam, RMSprop }
# tuning trials	100

EARLIEST The main backbone is LSTM (Hochreiter & Schmidhuber, 1997) with the hidden state dimensions set to 256. The whole architecture has 2,387,817 trainable parameters. The batch size is 256. The number of training iterations is 5,000. We train EARLIEST with two different λ's: 10^{-1} and 10^{-10}.

Tables 16 and 17 show the search spaces of hyperparameters. For each tuning trial, we train EARLIEST and evaluate its averaged per-class accuracy per every 500 training iterations. After all training iterations, we save the best averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 16 and 17 in cyan letters. The approximated runtime of a single tuning trial is 0.5 hours, and the GPU consumption is 2.0 GBs.

Table 16. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-1}$ in Figure 5: UCF101.** The best hyperparameter combination is highlighted in cyan.

Learning rate	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
Optimizer	{ Adam, RMSprop }
# tuning trials	100

Table 17. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-10}$ in Figure 5: UCF101.** The best hyperparameter combination is highlighted in cyan.

Learning rate	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
Optimizer	{ Adam, RMSprop }
# tuning trials	100

I.5. Figure 5: HMDB51

MSPRT-TANDEM and NP test The approximation formula is the M-TANDEM formula. The loss function consists of the LSEL and the multiplet loss. The temporal integrator is Peephole LSTM (Gers & Schmidhuber, 2000) with the hidden state dimensions set to 256 followed by a fully-connected layer to output logits for classification. The temporal integrator has 2,374,656 trainable parameters. The batch size is fixed to 128. The number of training iterations is 10,000. We use the effective number (Cui et al., 2019) as the cost matrix of the LSEL, instead of $1/M_k$, to avoid over-emphasising the minority class and to simplify the parameter tuning (only one extra parameter β is introduced).

Table 18 shows the search space of hyperparameters. For each tuning trial, we train the temporal integrator and evaluate its mean averaged per-class accuracy per every 200 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Table 18 in cyan letters. The approximated runtime of one statistics trial is 8 hours, and the GPU consumption is 16–32 GBs.

Table 18. **Hyperparameter Search Space of MSPRT-TANDEM in Figure 5: HMDB51.** The best hyperparameter combination is highlighted in cyan. γ is defined as $L_{\text{total}} = L_{\text{mult}} + \gamma L_{\text{LSEL}}$. β controls the cost matrix (Cui et al., 2019).

Order	{ 0, 10, 40, 60, 78 }
Learning rate	{ 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^{2} }
Optimizer	{ Adam, RMSprop }
β	{ 0.99, 0.999, 0.9999, 0.99999, 1. }
# tuning trials	100

LSTM-s/m The backbone model is Peephole LSTM with the hidden state dimensions set to 256 followed by a fully-connected layer to output logits for classification. LSTM has 2,374,656 trainable parameters. The batch size is fixed to 128. The number of training iterations is 10,000.

Tables 19 and 20 show the search spaces of hyperparameters. For each tuning trial, we train LSTM and evaluate its mean averaged per-class accuracy per every 200 training iterations. After all training iterations, we save the best mean averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 19 and 20 in cyan letters. The approximated runtime of one statistics trial is 3 hours, and the GPU consumption is 2.6 GBs.

Table 19. **Hyperparameter Search Space of LSTM-s in Figure 5: HMDB51.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\text{total}} = L_{\text{cross-entropy}} + \gamma L_{\text{ranking}}$ (Ma et al., 2016).

Learning rate	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Weight decay	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^{2} }
Optimizer	{ RMSprop }
# tuning trials	100

EARLIEST The main backbone is LSTM (Hochreiter & Schmidhuber, 1997) with the hidden state dimensions set to 256. The whole architecture has 2,374,967 trainable parameters. The batch size is 256. The number of training iterations is 5,000. We train EARLIEST with two different λ's: 10^{-1} and 10^{-10}.

Tables 21 and 22 show the search spaces of hyperparameters. For each tuning trial, we train EARLIEST and evaluate its

Table 20. **Hyperparameter Search Space of LSTM-m in Figure 5: HMDB51.** The best hyperparameter combination is highlighted in cyan. γ controls the strength of monotonicity and is defined as $L_{\text{total}} = L_{\text{cross-entropy}} + \gamma L_{\text{ranking}}$ (Ma et al., 2016).

LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-2}, 10^{-1}, 1, 10, 10^{2} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	100

averaged per-class accuracy per every 500 training iterations. After all training iterations, we save the best averaged per-class accuracy. After all tuning trials, we select the best combination of the hyperparameters, which is shown in Tables 21 and 22 in cyan letters. The approximated runtime of a single tuning trial is 0.5 hours, and the GPU consumption is 2.0 GBs.

Table 21. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-1}$ in Figure 5: HMDB51.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	100

Table 22. **Hyperparameter Search Space of EARLIEST with $\lambda = 10^{-10}$ in Figure 5: HMDB51.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	100

I.6. Figure 1: LLR Trajectories

We plot ten different i's randomly selected from the validation set of NMNIST-100f. The base temporal integrator is selected from the models used for NMNIST-100f in Figure 5. More example trajectories are shown in Figure 10 in Appendix J.

I.7. Figure 9: Ablation Study of LSEL and Multiplet Loss

The training procedure is totally similar to that of Figure 5. Tables 23 and 24 show the search spaces of hyperparameters. "TANDEM LSEL+Mult" is the same model as "MSPRT-TANDEM" in Figure 5 (NMNIST-100f).

Table 23. **Hyperparameter Search Space of "TANDEM Mult" in Figure 9.** The best hyperparameter combination is highlighted in cyan.

ORDER	{ 0, 25, 50, 75, 99 }
LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	N/A
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	200

Table 24. **Hyperparameter Search Space of "TANDEM LSEL" in Figure 9.** The best hyperparameter combination is highlighted in cyan.

ORDER	{ 0, 25, 50, 75, 99 }
LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^{2}, 10^{3}}
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	200

I.8. Figure 2 Top: LSEL v.s. Binary DRE-based Losses

We define the loss functions used in Figure 2. Conventional DRE losses are often biased (LLLR), unbounded (LSIF and DSKL), or numerically unstable (LSIF, LSIFwC, and BARR), especially when applied to multiclass classification, leading to suboptimal performances (Figure 2).) Because conventional DRE losses are restricted to binary DRE, we modify two LSIF-based and three KLIEP-based loss functions for DRME. The logistic loss we use is introduced in Appendix D.

The original LSIF (Kanamori et al., 2009) is based on a kernel method and estimates density ratios via minimizing the mean squared error between p and $\hat{r}q$ ($\hat{r} := \hat{p}/\hat{q}$). We define a variant of LSIF for DRME as

$$\hat{L}_{\text{LSIF}} := \sum_{t \in [T]} \sum_{\substack{k,l \in [K] \\ (k \neq l)}} \left[\frac{1}{M_l} \sum_{i \in I_l} \hat{\Lambda}_{kl}^2(X_i^{(1,t)}) - \frac{1}{M_k} \sum_{i \in I_k} \hat{\Lambda}_{kl}(X_i^{(1,t)}) \right] , \tag{61}$$

where the likelihood ratio is denoted by $\hat{\Lambda}(X) := e^{\hat{\lambda}(X)} = \hat{p}(X|k)/\hat{p}(X|l)$. Because of the k,l-summation, $\hat{L}_{\text{LSIF}}$ is symmetric with respect to the denominator and numerator, unlike the original LSIF. Therefore, we can expect that $\hat{L}_{\text{LSIF}}$ is more stable than the original one. However, $\hat{L}_{\text{LSIF}}$ inherits the problems of the original LSIF; it is unbounded and numerically unstable. The latter is due to dealing with $\hat{\Lambda}$ directly, which easily explodes when the denominator is small. This instability is not negligible, especially when LSIF is used with DNNs. The following LSIF with Constraint (LSIFwC)[9] avoids the explosion by adding a constraint:

$$\hat{L}_{\text{LSIFwC}} := \sum_{t \in [T]} \sum_{\substack{k,l \in [K] \\ (k \neq l)}} \left[\frac{1}{M_l} \sum_{i \in I_l} \hat{\Lambda}_{kl}^2(X_i^{(1,t)}) - \frac{1}{M_k} \sum_{i \in I_k} \hat{\Lambda}_{kl}(X_i^{(1,t)}) + \gamma \left| \frac{1}{M_l} \sum_{i \in I_l} \hat{\Lambda}_{kl}(X_i^{(1,t)}) - 1 \right| \right] , \tag{62}$$

where $\gamma > 0$ is a hyperparameter. $\hat{L}_{LSIFwC}$ is symmetric and bounded for sufficiently large γ. However, it is still numerically unstable due to $\hat{\Lambda}$. Note that the constraint term is equivalent to the probability normalization $\int dx\, p(x|l)\, (\hat{p}(x|k)/\hat{p}(x|l)) = 1$.

DSKL (Khan et al., 2019), BARR (Khan et al., 2019), and LLLR (Ebihara et al., 2021) are based on KLIEP (Sugiyama et al., 2008), which estimates density ratios via minimizing the Kullback-Leibler divergence (Kullback & Leibler, 1951) between p and $\hat{r}q$ ($\hat{r} := \hat{p}/\hat{q}$). We define a variant of DSKL for DRME as

$$\hat{L}_{\text{DSKL}} := \sum_{t \in [T]} \sum_{\substack{k,l \in [K] \\ (k \neq l)}} \left[\frac{1}{M_l} \sum_{i \in I_l} \hat{\lambda}_{kl}(X_i^{(1,t)}) - \frac{1}{M_k} \sum_{i \in I_k} \hat{\lambda}_{kl}(X_i^{(1,t)}) \right] \tag{63}$$

The original DSKL is symmetric, and the same is true for $\hat{L}_{\text{BARR}}$, while the original KLIEP is not. $\hat{L}_{\text{BARR}}$ is relatively stable compared with $\hat{L}_{\text{LSIF}}$ and $\hat{L}_{\text{LSIFwC}}$ because it does not include $\hat{\Lambda}$ but $\hat{\lambda}$; still, $\hat{L}_{\text{DSKL}}$ is unbounded and can diverge. BARR for DRME is

$$\hat{L}_{\text{BARR}} := \sum_{t \in [T]} \sum_{\substack{k,l \in [K] \\ (k \neq l)}} \left[-\frac{1}{M_k} \sum_{i \in I_k} \hat{\lambda}_{kl}(X_i^{(1,t)}) + \gamma \left| \frac{1}{M_l} \sum_{i \in I_l} \hat{\Lambda}_{kl}(X_i^{(1,t)}) - \right. \right. \tag{64}$$

where $\gamma > 0$ is a hyperparameter. $\hat{L}_{\text{BARR}}$ is symmetric and bounded because of the second term but is numerically unstable because of $\hat{\Lambda}$. LLLR for DRME is

$$\hat{L}_{\text{LLLR}} := \sum_{t \in [T]} \sum_{\substack{k,l \in [K] \\ (k \neq l)}} \frac{1}{M_k + M_l} \left[\sum_{i \in I_l} \sigma(\hat{\lambda}_{kl}(X_i^{(1,t)})) + \sum_{i \in I_k} (1 - \sigma(\hat{\lambda}_{kl}(X_i \right. \tag{65}$$

where σ is the sigmoid function. LLLR is symmetric, bounded, and numerically stable but is biased in the sense that it does not necessarily converge to the optimal solution λ; i.e., the probability normalization constraint, which is explicitly included in the original KLIEP, cannot be exactly satisfied. Finally, the logistic loss is defined as (52).

All the models share the same feature extractor (Appendix I.1) and temporal integrator (Appendix I.2) without the M-TANDEM(wO) approximation or multiplet loss. The search spaces of hyperparameters are given in Tables 25–31. The other setting follows Appendix I.2.

Table 25. **Hyperparameter Search Space of LSIF in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

Learning rate	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
Weight decay	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
Optimizer	{ Adam, RMSprop, Momentum }
# tuning trials	150

I.9. Figure 2 Bottom: LSEL v.s. Logistic Loss

The experimental condition follows that of Appendix I.3. The logistic loss is defined as (52). The hyperparameters are given in Tables 32 and 33.

[9]Do not confuse this with cLSIF (Kanamori et al., 2009).

Table 26. **Hyperparameter Search Space of LSIFwC in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
WEIGHT DECAY	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-4}, 10^{-3}, 10^{-2}, 1, 10 }
OPTIMIZER	{ ADAM, RMSPROP, MOMENTUM }
# TUNING TRIALS	150

Table 27. **Hyperparameter Search Space of DSKL in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
WEIGHT DECAY	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP, MOMENTUM }
# TUNING TRIALS	150

Table 28. **Hyperparameter Search Space of BARR in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
WEIGHT DECAY	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-4}, 10^{-3}, 10^{-2}, 1, 10 }
OPTIMIZER	{ ADAM, RMSPROP, MOMENTUM }
# TUNING TRIALS	150

Table 29. **Hyperparameter Search Space of LLLR in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
WEIGHT DECAY	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP, MOMENTUM }
# TUNING TRIALS	150

Table 30. **Hyperparameter Search Space of Logistic Loss in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
WEIGHT DECAY	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP, MOMENTUM }
# TUNING TRIALS	150

Table 31. **Hyperparameter Search Space of LSEL in Figure 2 Top.** The best hyperparameter combination is highlighted in cyan.

LEARNING RATE	{ 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6} }
WEIGHT DECAY	{ 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} }
OPTIMIZER	{ ADAM, RMSPROP, MOMENTUM }
# TUNING TRIALS	150

Table 32. **Hyperparameter Search Space of Logistic Loss in Figure 2 Bottom.** The best hyperparameter combination is highlighted in cyan.

ORDER	{ 0, 25, 50, 75, 99 }
LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^{2}, 10^{3}}
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	300

Table 33. **Hyperparameter Search Space of LSEL in Figure 2 Bottom.** The best hyperparameter combination is highlighted in cyan.

ORDER	{ 0, 25, 50, 75, 99 }
LEARNING RATE	{ 10^{-2}, 10^{-3}, 10^{-4} }
WEIGHT DECAY	{ 10^{-3}, 10^{-4}, 10^{-5} }
γ	{ 10^{-1}, 1, 10, 10^{2}, 10^{3}}
OPTIMIZER	{ ADAM, RMSPROP }
# TUNING TRIALS	300

I.10. Exact Error Rates in Figures 5 and 9

Tables 34–41 show the averaged per-class error rates plotted in the figures in Section 4.

Table 34. **Averaged Per-Class Error Rates (%) and SEM of Figure 2 (Top: NMNIST-H).** Blanks mean N/A.

Time	LSIF	LSIFwC	DSKL	BARR
1.00	83.822 ± 2.142	81.585 ± 2.908		66.579 ± 0.108
2.00	73.330 ± 2.142	67.749 ± 2.908	45.145 ± 0.170	45.902 ± 0.108
3.00	70.492 ± 2.142	63.237 ± 2.908	30.719 ± 0.170	30.698 ± 0.108
4.00	68.258 ± 2.142	60.061 ± 2.908	20.409 ± 0.170	20.172 ± 0.108
5.00	66.485 ± 2.142	57.860 ± 2.908	18.102 ± 0.170	12.990 ± 0.108
6.00	64.975 ± 2.142	55.939 ± 2.908	17.899 ± 0.170	8.374 ± 0.108
7.00	63.549 ± 2.142	54.317 ± 2.908	14.442 ± 0.170	5.529 ± 0.108
8.00	62.137 ± 2.142	52.635 ± 2.908	12.942 ± 0.170	3.993 ± 0.108
9.00	60.551 ± 2.142	50.886 ± 2.908	11.681 ± 0.170	3.202 ± 0.108
10.00	58.657 ± 2.142	48.947 ± 2.908	10.368 ± 0.170	2.787 ± 0.108
11.00	56.716 ± 2.142	46.501 ± 2.908	9.858 ± 0.170	2.563 ± 0.108
12.00	54.976 ± 2.142	44.114 ± 2.908	8.493 ± 0.170	2.457 ± 0.108
13.00	52.800 ± 2.142	41.901 ± 2.908	7.909 ± 0.170	2.404 ± 0.108
14.00	49.940 ± 2.142	40.073 ± 2.908	7.151 ± 0.170	2.376 ± 0.108
15.00	47.293 ± 2.142	38.990 ± 2.908	6.216 ± 0.170	2.359 ± 0.108
16.00	45.203 ± 2.142	38.013 ± 2.908	5.821 ± 0.170	2.352 ± 0.108
17.00	42.664 ± 2.142	36.799 ± 2.908	4.632 ± 0.170	2.348 ± 0.108
18.00	38.406 ± 2.142	34.540 ± 2.908	3.971 ± 0.170	2.345 ± 0.108
19.00	32.666 ± 2.142	30.737 ± 2.908	3.484 ± 0.170	2.343 ± 0.108
20.00	30.511 ± 2.142	29.880 ± 2.908	2.088 ± 0.170	2.343 ± 0.108
Trials	40	40	60	40

Time	LLLR	Logistic	LSEL
1.00	63.930 ± 0.026		63.956 ± 0.020
2.00	44.250 ± 0.026	43.873 ± 0.097	43.918 ± 0.020
3.00	30.693 ± 0.026	29.058 ± 0.097	28.957 ± 0.020
4.00	21.370 ± 0.026	18.973 ± 0.097	18.865 ± 0.020
5.00	14.834 ± 0.026	11.992 ± 0.097	11.910 ± 0.020
6.00	10.386 ± 0.026	7.522 ± 0.097	7.444 ± 0.020
7.00	7.516 ± 0.026	4.970 ± 0.097	4.877 ± 0.020
8.00	5.693 ± 0.026	3.485 ± 0.097	3.403 ± 0.020
9.00	4.563 ± 0.026	2.702 ± 0.097	2.647 ± 0.020
10.00	3.818 ± 0.026	2.326 ± 0.097	2.288 ± 0.020
11.00	3.316 ± 0.026	2.133 ± 0.097	2.103 ± 0.020
12.00	2.961 ± 0.026	2.030 ± 0.097	2.010 ± 0.020
13.00	2.705 ± 0.026	1.985 ± 0.097	1.977 ± 0.020
14.00	2.516 ± 0.026	1.964 ± 0.097	1.956 ± 0.020
15.00	2.365 ± 0.026	1.951 ± 0.097	1.941 ± 0.020
16.00	2.254 ± 0.026	1.943 ± 0.097	1.934 ± 0.020
17.00	2.163 ± 0.026	1.938 ± 0.097	1.932 ± 0.020
18.00	2.094 ± 0.026	1.937 ± 0.097	1.932 ± 0.020
19.00	2.036 ± 0.026	1.937 ± 0.097	1.932 ± 0.020
20.00	1.968 ± 0.026	1.937 ± 0.097	1.932 ± 0.020
Trials	80	80	80

Table 35. **Averaged Per-Class Error Rates (%) and SEM of Figure 2 (Bottom: NMNIST-100f). Frames 1–50.** Blanks mean N/A.

TIME	LOGISTIC (WITH M-TANDEM)	LSEL (WITH M-TANDEM)
1.00	56.273 ± 0.045	56.196 ± 0.044
2.00	37.772 ± 0.045	37.492 ± 0.044
3.00	27.922 ± 0.045	27.520 ± 0.044
4.00	22.282 ± 0.045	21.889 ± 0.044
5.00	18.725 ± 0.045	18.369 ± 0.044
6.00	16.251 ± 0.045	15.901 ± 0.044
7.00	14.383 ± 0.045	14.035 ± 0.044
8.00	12.898 ± 0.045	12.590 ± 0.044
9.00	11.730 ± 0.045	11.421 ± 0.044
10.00	10.758 ± 0.045	10.468 ± 0.044
11.00	9.943 ± 0.045	9.659 ± 0.044
12.00	9.245 ± 0.045	8.992 ± 0.044
13.00	8.663 ± 0.045	8.410 ± 0.044
14.00	8.158 ± 0.045	7.907 ± 0.044
15.00	7.721 ± 0.045	7.472 ± 0.044
16.00	7.341 ± 0.045	7.096 ± 0.044
17.00	7.006 ± 0.045	6.776 ± 0.044
18.00	6.717 ± 0.045	6.498 ± 0.044
19.00	6.476 ± 0.045	6.257 ± 0.044
20.00	6.258 ± 0.045	6.041 ± 0.044
21.00	6.069 ± 0.045	5.851 ± 0.044
22.00	5.900 ± 0.045	5.686 ± 0.044
23.00	5.748 ± 0.045	5.545 ± 0.044
24.00	5.615 ± 0.045	5.423 ± 0.044
25.00	5.502 ± 0.045	5.317 ± 0.044
26.00	5.406 ± 0.045	5.221 ± 0.044
27.00	5.318 ± 0.045	5.136 ± 0.044
28.00	5.246 ± 0.045	5.064 ± 0.044
29.00	5.175 ± 0.045	5.001 ± 0.044
30.00	5.114 ± 0.045	4.947 ± 0.044
31.00	5.057 ± 0.045	4.898 ± 0.044
32.00	5.011 ± 0.045	4.854 ± 0.044
33.00	4.974 ± 0.045	4.816 ± 0.044
34.00	4.939 ± 0.045	4.784 ± 0.044
35.00	4.908 ± 0.045	4.757 ± 0.044
36.00	4.882 ± 0.045	4.733 ± 0.044
37.00	4.858 ± 0.045	4.713 ± 0.044
38.00	4.837 ± 0.045	4.694 ± 0.044
39.00	4.818 ± 0.045	4.676 ± 0.044
40.00	4.801 ± 0.045	4.661 ± 0.044
41.00	4.786 ± 0.045	4.649 ± 0.044
42.00	4.773 ± 0.045	4.639 ± 0.044
43.00	4.761 ± 0.045	4.630 ± 0.044
44.00	4.751 ± 0.045	4.623 ± 0.044
45.00	4.742 ± 0.045	4.615 ± 0.044
46.00	4.735 ± 0.045	4.608 ± 0.044
47.00	4.727 ± 0.045	4.604 ± 0.044
48.00	4.722 ± 0.045	4.598 ± 0.044
49.00	4.716 ± 0.045	4.594 ± 0.044
50.00	4.711 ± 0.045	4.590 ± 0.044
TRIALS	150	150

Table 36. **Averaged Per-Class Error Rates (%) and SEM of Figure 2 (Bottom: NMNIST-100f). Frames 51–100.** Blanks mean N/A.

TIME	LOGISTIC (WITH M-TANDEM)	LSEL (WITH M-TANDEM)
51.00	4.706 ± 0.045	4.587 ± 0.044
52.00	4.703 ± 0.045	4.586 ± 0.044
53.00	4.700 ± 0.045	4.584 ± 0.044
54.00	4.697 ± 0.045	4.580 ± 0.044
55.00	4.694 ± 0.045	4.577 ± 0.044
56.00	4.692 ± 0.045	4.576 ± 0.044
57.00	4.690 ± 0.045	4.575 ± 0.044
58.00	4.688 ± 0.045	4.573 ± 0.044
59.00	4.687 ± 0.045	4.572 ± 0.044
60.00	4.686 ± 0.045	4.572 ± 0.044
61.00	4.685 ± 0.045	4.571 ± 0.044
62.00	4.684 ± 0.045	4.570 ± 0.044
63.00	4.683 ± 0.045	4.569 ± 0.044
64.00	4.683 ± 0.045	4.569 ± 0.044
65.00	4.682 ± 0.045	4.569 ± 0.044
66.00	4.681 ± 0.045	4.568 ± 0.044
67.00	4.681 ± 0.045	4.568 ± 0.044
68.00	4.680 ± 0.045	4.567 ± 0.044
69.00	4.679 ± 0.045	4.567 ± 0.044
70.00	4.678 ± 0.045	4.567 ± 0.044
71.00	4.678 ± 0.045	4.566 ± 0.044
72.00	4.678 ± 0.045	4.566 ± 0.044
73.00	4.678 ± 0.045	4.565 ± 0.044
74.00	4.678 ± 0.045	4.565 ± 0.044
75.00	4.678 ± 0.045	4.565 ± 0.044
76.00	4.678 ± 0.045	4.565 ± 0.044
77.00	4.678 ± 0.045	4.565 ± 0.044
78.00	4.677 ± 0.045	4.565 ± 0.044
79.00	4.677 ± 0.045	4.564 ± 0.044
80.00	4.676 ± 0.045	4.565 ± 0.044
81.00	4.676 ± 0.045	4.564 ± 0.044
82.00	4.676 ± 0.045	4.564 ± 0.044
83.00	4.676 ± 0.045	4.564 ± 0.044
84.00	4.676 ± 0.045	4.564 ± 0.044
85.00	4.676 ± 0.045	4.564 ± 0.044
86.00	4.676 ± 0.045	4.564 ± 0.044
87.00	4.676 ± 0.045	4.564 ± 0.044
88.00	4.676 ± 0.045	4.564 ± 0.044
89.00	4.676 ± 0.045	4.564 ± 0.044
90.00	4.676 ± 0.045	4.564 ± 0.044
91.00	4.676 ± 0.045	4.564 ± 0.044
92.00	4.676 ± 0.045	4.564 ± 0.044
93.00	4.676 ± 0.045	4.564 ± 0.044
94.00	4.676 ± 0.045	4.564 ± 0.044
95.00	4.676 ± 0.045	4.564 ± 0.044
96.00	4.676 ± 0.045	4.564 ± 0.044
97.00	4.676 ± 0.045	4.564 ± 0.044
98.00	4.676 ± 0.045	4.564 ± 0.044
99.00	4.676 ± 0.045	4.564 ± 0.044
100.00	4.676 ± 0.045	4.564 ± 0.044
TRIALS	150	150

Table 37. **Averaged Per-Class Error Rates (%) and SEM of Figure 5 (NMNIST-H).** Blanks mean N/A.

TIME	MSPRT-TANDEM	LSTM-S	LSTM-M	EARLIEST 10^{-2}	EARLIEST 10^{2}
1.00		64.003 ± 0.010	64.009 ± 0.011		
1.34					57.973 ± 0.066
2.00	44.172 ± 0.046	46.676 ± 0.008	46.697 ± 0.008		
3.00	29.190 ± 0.046	35.517 ± 0.007	35.536 ± 0.008		
4.00	18.964 ± 0.046	25.936 ± 0.007	25.940 ± 0.007		
5.00	11.994 ± 0.046	20.510 ± 0.007	20.512 ± 0.006		
6.00	7.449 ± 0.046	15.393 ± 0.007	15.404 ± 0.006		
7.00	4.870 ± 0.046	12.603 ± 0.005	12.614 ± 0.005		
8.00	3.383 ± 0.046	10.025 ± 0.005	10.018 ± 0.005		
9.00	2.639 ± 0.046	8.036 ± 0.005	8.033 ± 0.005		
10.00	2.281 ± 0.046	6.963 ± 0.005	6.958 ± 0.005		
11.00	2.087 ± 0.046	5.788 ± 0.005	5.801 ± 0.005		
12.00	1.996 ± 0.046	4.886 ± 0.003	4.892 ± 0.004		
13.00	1.963 ± 0.046	4.398 ± 0.003	4.400 ± 0.003		
13.41				3.162 ± 0.027	
14.00	1.942 ± 0.046	3.735 ± 0.003	3.737 ± 0.003		
15.00	1.926 ± 0.046	3.190 ± 0.003	3.198 ± 0.003		
16.00	1.919 ± 0.046	2.841 ± 0.003	2.850 ± 0.003		
17.00	1.917 ± 0.046	2.576 ± 0.003	2.576 ± 0.003		
18.00	1.916 ± 0.046	2.376 ± 0.003	2.374 ± 0.003		
19.00	1.916 ± 0.046	2.118 ± 0.003	2.119 ± 0.003		
20.00	1.916 ± 0.046	1.923 ± 0.003	1.921 ± 0.003		
TRIALS	300	300	300	50	45

Table 38. **Averaged Per-Class Error Rates (%) and SEM of Figure 5 (NMNIST-100f). Frames 1–50.** Blanks mean N/A.

TIME	MSPRT-TANDEM	LSTM-S	LSTM-M	EARLIEST LAM1E-2	EARLIEST LAM1E-4
1.00		54.677 ± 0.016	54.696 ± 0.016		
2.00	35.942 ± 0.148	38.748 ± 0.014	38.677 ± 0.015		
3.00	25.203 ± 0.148	29.348 ± 0.014	29.350 ± 0.014		
4.00	19.167 ± 0.148	24.048 ± 0.014	23.981 ± 0.015		
5.00	15.450 ± 0.148	20.490 ± 0.014	20.422 ± 0.014		
6.00	12.997 ± 0.148	18.065 ± 0.014	17.959 ± 0.015		
7.00	11.224 ± 0.148	15.715 ± 0.014	15.606 ± 0.015		
8.00	9.912 ± 0.148	14.306 ± 0.014	14.215 ± 0.014		
9.00	8.880 ± 0.148	13.124 ± 0.014	13.046 ± 0.013		
10.00	8.068 ± 0.148	12.129 ± 0.012	12.038 ± 0.012		
11.00	7.405 ± 0.148	11.386 ± 0.013	11.302 ± 0.013		
12.00	6.862 ± 0.148	10.838 ± 0.012	10.767 ± 0.013		
13.00	6.409 ± 0.148	10.121 ± 0.012	10.034 ± 0.012		
14.00	6.032 ± 0.148	9.626 ± 0.013	9.555 ± 0.013		
15.00	5.715 ± 0.148	9.102 ± 0.012	9.007 ± 0.012		
16.00	5.444 ± 0.148	8.710 ± 0.013	8.625 ± 0.013		
17.00	5.212 ± 0.148	8.268 ± 0.011	8.235 ± 0.012		
18.00	5.025 ± 0.148	7.903 ± 0.012	7.852 ± 0.012		
19.00	4.865 ± 0.148	7.613 ± 0.012	7.586 ± 0.011		
19.46				20.044 ± 0.229	
20.00	4.730 ± 0.148	7.366 ± 0.012	7.354 ± 0.011		
21.00	4.623 ± 0.148	7.150 ± 0.012	7.150 ± 0.011		
22.00	4.536 ± 0.148	6.998 ± 0.011	7.006 ± 0.011		
23.00	4.464 ± 0.148	6.816 ± 0.011	6.821 ± 0.011		
24.00	4.401 ± 0.148	6.674 ± 0.011	6.655 ± 0.011		
25.00	4.353 ± 0.148	6.491 ± 0.012	6.502 ± 0.013		
26.00	4.310 ± 0.148	6.351 ± 0.011	6.373 ± 0.012		
27.00	4.275 ± 0.148	6.200 ± 0.011	6.200 ± 0.013		
28.00	4.246 ± 0.148	6.061 ± 0.011	6.077 ± 0.013		
29.00	4.222 ± 0.148	5.935 ± 0.011	5.948 ± 0.012		
30.00	4.203 ± 0.148	5.876 ± 0.012	5.870 ± 0.011		
31.00	4.186 ± 0.148	5.811 ± 0.011	5.823 ± 0.012		
32.00	4.171 ± 0.148	5.658 ± 0.011	5.676 ± 0.012		
33.00	4.160 ± 0.148	5.552 ± 0.011	5.566 ± 0.012		
34.00	4.147 ± 0.148	5.456 ± 0.012	5.474 ± 0.012		
35.00	4.139 ± 0.148	5.395 ± 0.011	5.412 ± 0.012		
36.00	4.133 ± 0.148	5.307 ± 0.011	5.319 ± 0.012		
37.00	4.126 ± 0.148	5.231 ± 0.011	5.249 ± 0.012		
38.00	4.121 ± 0.148	5.183 ± 0.011	5.211 ± 0.012		
39.00	4.118 ± 0.148	5.148 ± 0.011	5.182 ± 0.011		
40.00	4.116 ± 0.148	5.121 ± 0.011	5.168 ± 0.012		
41.00	4.113 ± 0.148	5.055 ± 0.011	5.098 ± 0.012		
42.00	4.112 ± 0.148	5.057 ± 0.012	5.091 ± 0.013		
43.00	4.110 ± 0.148	4.980 ± 0.012	5.017 ± 0.012		
44.00	4.109 ± 0.148	4.940 ± 0.011	4.976 ± 0.013		
45.00	4.109 ± 0.148	4.892 ± 0.012	4.921 ± 0.013		
46.00	4.108 ± 0.148	4.825 ± 0.012	4.865 ± 0.013		
47.00	4.108 ± 0.148	4.780 ± 0.012	4.795 ± 0.013		
48.00	4.108 ± 0.148	4.730 ± 0.012	4.767 ± 0.013		
49.00	4.107 ± 0.148	4.709 ± 0.012	4.742 ± 0.013		
50.00	4.107 ± 0.148	4.690 ± 0.012	4.731 ± 0.013		
TRIALS	150	150	150	50	60

Table 39. **Averaged Per-Class Error Rates (%) and SEM of Figure 5 (NMNIST-100f). Frames 51–100.** Blanks mean N/A.

TIME	MSPRT-TANDEM	LSTM-S	LSTM-M	EARLIEST 10^{-2}	EARLIEST 10^{-4}
51.00	4.107 ± 0.148	4.620 ± 0.012	4.661 ± 0.014		
52.00	4.107 ± 0.148	4.588 ± 0.012	4.637 ± 0.014		
53.00	4.107 ± 0.148	4.595 ± 0.011	4.634 ± 0.013		
54.00	4.106 ± 0.148	4.538 ± 0.011	4.580 ± 0.013		
55.00	4.107 ± 0.148	4.526 ± 0.010	4.566 ± 0.013		
56.00	4.107 ± 0.148	4.531 ± 0.012	4.555 ± 0.013		
57.00	4.107 ± 0.148	4.470 ± 0.011	4.509 ± 0.012		
58.00	4.107 ± 0.148	4.444 ± 0.011	4.489 ± 0.012		
59.00	4.108 ± 0.148	4.449 ± 0.011	4.496 ± 0.012		
60.00	4.108 ± 0.148	4.447 ± 0.011	4.494 ± 0.013		
61.00	4.108 ± 0.148	4.459 ± 0.011	4.484 ± 0.012		
62.00	4.108 ± 0.148	4.408 ± 0.011	4.447 ± 0.012		
63.00	4.108 ± 0.148	4.391 ± 0.011	4.429 ± 0.013		
64.00	4.108 ± 0.148	4.377 ± 0.011	4.412 ± 0.013		
65.00	4.108 ± 0.148	4.370 ± 0.011	4.397 ± 0.013		
66.00	4.109 ± 0.148	4.378 ± 0.011	4.405 ± 0.013		
67.00	4.109 ± 0.148	4.394 ± 0.011	4.422 ± 0.012		
68.00	4.109 ± 0.148	4.360 ± 0.011	4.386 ± 0.012		
69.00	4.109 ± 0.148	4.307 ± 0.012	4.324 ± 0.012		
70.00	4.109 ± 0.148	4.277 ± 0.011	4.315 ± 0.012		
71.00	4.109 ± 0.148	4.268 ± 0.012	4.312 ± 0.012		
72.00	4.109 ± 0.148	4.252 ± 0.012	4.286 ± 0.011		
73.00	4.109 ± 0.148	4.247 ± 0.011	4.278 ± 0.012		
74.00	4.109 ± 0.148	4.203 ± 0.011	4.221 ± 0.012		
75.00	4.109 ± 0.148	4.207 ± 0.011	4.234 ± 0.012		
76.00	4.109 ± 0.148	4.203 ± 0.011	4.217 ± 0.011		
77.00	4.109 ± 0.148	4.171 ± 0.011	4.191 ± 0.012		
78.00	4.109 ± 0.148	4.154 ± 0.011	4.182 ± 0.012		
79.00	4.109 ± 0.148	4.155 ± 0.010	4.179 ± 0.012		
80.00	4.109 ± 0.148	4.129 ± 0.010	4.144 ± 0.012		
81.00	4.109 ± 0.148	4.134 ± 0.011	4.138 ± 0.012		
82.00	4.109 ± 0.148	4.113 ± 0.011	4.126 ± 0.013		
83.00	4.109 ± 0.148	4.117 ± 0.011	4.135 ± 0.013		
84.00	4.109 ± 0.148	4.093 ± 0.011	4.105 ± 0.012		
85.00	4.109 ± 0.148	4.096 ± 0.012	4.113 ± 0.012		
86.00	4.109 ± 0.148	4.106 ± 0.012	4.112 ± 0.012		
87.00	4.109 ± 0.148	4.099 ± 0.011	4.110 ± 0.013		
88.00	4.109 ± 0.148	4.093 ± 0.011	4.101 ± 0.012		
89.00	4.109 ± 0.148	4.072 ± 0.011	4.094 ± 0.013		
90.00	4.109 ± 0.148	4.077 ± 0.011	4.095 ± 0.013		
91.00	4.109 ± 0.148	4.063 ± 0.011	4.086 ± 0.013		
92.00	4.109 ± 0.148	4.069 ± 0.011	4.083 ± 0.013		
93.00	4.109 ± 0.148	4.073 ± 0.011	4.096 ± 0.013		
94.00	4.109 ± 0.148	4.069 ± 0.011	4.105 ± 0.013		
95.00	4.109 ± 0.148	4.062 ± 0.011	4.101 ± 0.013		
96.00	4.109 ± 0.148	4.076 ± 0.012	4.109 ± 0.013		
97.00	4.109 ± 0.148	4.056 ± 0.011	4.098 ± 0.012		
98.00	4.109 ± 0.148	4.063 ± 0.011	4.104 ± 0.012		
99.00	4.109 ± 0.148	4.066 ± 0.011	4.100 ± 0.012		
99.99					4.586 ± 0.020
100.00	4.109 ± 0.148	4.057 ± 0.011	4.097 ± 0.012		
TRIALS	150	150	150	50	60

Table 40. **Averaged Per-Class Error Rates (%) and SEM of Figure 9. Frames 1–50.** Blanks mean N/A.

TIME	TANDEM LSEL	TANDEM MULT	TANDEM LSEL+MULT
1.00	56.196 ± 0.044	55.705 ± 0.036	
2.00	37.742 ± 0.044	36.650 ± 0.036	35.942 ± 0.148
3.00	27.721 ± 0.044	26.072 ± 0.036	25.203 ± 0.148
4.00	22.037 ± 0.044	20.019 ± 0.036	19.167 ± 0.148
5.00	18.486 ± 0.044	16.320 ± 0.036	15.450 ± 0.148
6.00	16.015 ± 0.044	13.842 ± 0.036	12.997 ± 0.148
7.00	14.127 ± 0.044	12.028 ± 0.036	11.224 ± 0.148
8.00	12.664 ± 0.044	10.636 ± 0.036	9.912 ± 0.148
9.00	11.487 ± 0.044	9.564 ± 0.036	8.880 ± 0.148
10.00	10.526 ± 0.044	8.692 ± 0.036	8.068 ± 0.148
11.00	9.711 ± 0.044	7.989 ± 0.036	7.405 ± 0.148
12.00	9.042 ± 0.044	7.393 ± 0.036	6.862 ± 0.148
13.00	8.454 ± 0.044	6.897 ± 0.036	6.409 ± 0.148
14.00	7.945 ± 0.044	6.477 ± 0.036	6.032 ± 0.148
15.00	7.510 ± 0.044	6.119 ± 0.036	5.715 ± 0.148
16.00	7.124 ± 0.044	5.830 ± 0.036	5.444 ± 0.148
17.00	6.803 ± 0.044	5.579 ± 0.036	5.212 ± 0.148
18.00	6.519 ± 0.044	5.370 ± 0.036	5.025 ± 0.148
19.00	6.276 ± 0.044	5.188 ± 0.036	4.865 ± 0.148
20.00	6.060 ± 0.044	5.032 ± 0.036	4.730 ± 0.148
21.00	5.869 ± 0.044	4.899 ± 0.036	4.623 ± 0.148
22.00	5.700 ± 0.044	4.785 ± 0.036	4.536 ± 0.148
23.00	5.556 ± 0.044	4.687 ± 0.036	4.464 ± 0.148
24.00	5.434 ± 0.044	4.604 ± 0.036	4.401 ± 0.148
25.00	5.327 ± 0.044	4.537 ± 0.036	4.353 ± 0.148
26.00	5.229 ± 0.044	4.477 ± 0.036	4.310 ± 0.148
27.00	5.143 ± 0.044	4.426 ± 0.036	4.275 ± 0.148
28.00	5.070 ± 0.044	4.383 ± 0.036	4.246 ± 0.148
29.00	5.006 ± 0.044	4.353 ± 0.036	4.222 ± 0.148
30.00	4.952 ± 0.044	4.325 ± 0.036	4.203 ± 0.148
31.00	4.903 ± 0.044	4.299 ± 0.036	4.186 ± 0.148
32.00	4.858 ± 0.044	4.279 ± 0.036	4.171 ± 0.148
33.00	4.820 ± 0.044	4.261 ± 0.036	4.160 ± 0.148
34.00	4.787 ± 0.044	4.246 ± 0.036	4.147 ± 0.148
35.00	4.760 ± 0.044	4.234 ± 0.036	4.139 ± 0.148
36.00	4.736 ± 0.044	4.222 ± 0.036	4.133 ± 0.148
37.00	4.715 ± 0.044	4.211 ± 0.036	4.126 ± 0.148
38.00	4.697 ± 0.044	4.204 ± 0.036	4.121 ± 0.148
39.00	4.679 ± 0.044	4.197 ± 0.036	4.118 ± 0.148
40.00	4.662 ± 0.044	4.193 ± 0.036	4.116 ± 0.148
41.00	4.651 ± 0.044	4.188 ± 0.036	4.113 ± 0.148
42.00	4.640 ± 0.044	4.183 ± 0.036	4.112 ± 0.148
43.00	4.631 ± 0.044	4.180 ± 0.036	4.110 ± 0.148
44.00	4.624 ± 0.044	4.176 ± 0.036	4.109 ± 0.148
45.00	4.615 ± 0.044	4.174 ± 0.036	4.109 ± 0.148
46.00	4.609 ± 0.044	4.172 ± 0.036	4.108 ± 0.148
47.00	4.604 ± 0.044	4.170 ± 0.036	4.108 ± 0.148
48.00	4.599 ± 0.044	4.168 ± 0.036	4.108 ± 0.148
49.00	4.594 ± 0.044	4.167 ± 0.036	4.107 ± 0.148
50.00	4.591 ± 0.044	4.166 ± 0.036	4.107 ± 0.148
TRIALS	150	150	150

Table 41. **Averaged Per-Class Error Rates (%) and SEM of Figure 9. Frames 51–100.** Blanks mean N/A.

TIME	TANDEM LSEL	TANDEM MULT	TANDEM LSEL+MULT
51.00	4.588 ± 0.044	4.165 ± 0.036	4.107 ± 0.148
52.00	4.586 ± 0.044	4.164 ± 0.036	4.107 ± 0.148
53.00	4.584 ± 0.044	4.164 ± 0.036	4.107 ± 0.148
54.00	4.580 ± 0.044	4.163 ± 0.036	4.106 ± 0.148
55.00	4.577 ± 0.044	4.163 ± 0.036	4.107 ± 0.148
56.00	4.576 ± 0.044	4.163 ± 0.036	4.107 ± 0.148
57.00	4.575 ± 0.044	4.163 ± 0.036	4.107 ± 0.148
58.00	4.573 ± 0.044	4.162 ± 0.036	4.107 ± 0.148
59.00	4.572 ± 0.044	4.163 ± 0.036	4.108 ± 0.148
60.00	4.572 ± 0.044	4.163 ± 0.036	4.108 ± 0.148
61.00	4.571 ± 0.044	4.163 ± 0.036	4.108 ± 0.148
62.00	4.570 ± 0.044	4.164 ± 0.036	4.108 ± 0.148
63.00	4.570 ± 0.044	4.163 ± 0.036	4.108 ± 0.148
64.00	4.569 ± 0.044	4.163 ± 0.036	4.108 ± 0.148
65.00	4.569 ± 0.044	4.163 ± 0.036	4.108 ± 0.148
66.00	4.568 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
67.00	4.568 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
68.00	4.567 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
69.00	4.567 ± 0.044	4.163 ± 0.036	4.109 ± 0.148
70.00	4.567 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
71.00	4.566 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
72.00	4.566 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
73.00	4.565 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
74.00	4.565 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
75.00	4.565 ± 0.044	4.164 ± 0.036	4.109 ± 0.148
76.00	4.565 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
77.00	4.565 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
78.00	4.565 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
79.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
80.00	4.565 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
81.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
82.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
83.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
84.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
85.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
86.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
87.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
88.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
89.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
90.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
91.00	4.564 ± 0.044	4.165 ± 0.036	4.109 ± 0.148
92.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
93.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
94.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
95.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
96.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
97.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
98.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
99.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
100.00	4.564 ± 0.044	4.166 ± 0.036	4.109 ± 0.148
TRIALS	150	150	150

J. LLR Trajectories

Figures 10 and 11 show the LLR trajectories of NMNIST-H, NMNIST-100f, UCF101, and HMDB51. The base models are the same as those in Figure 5.

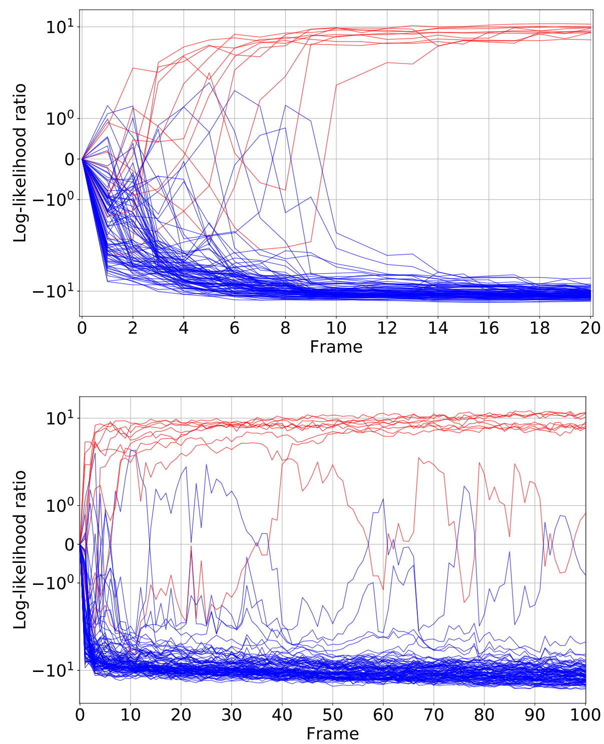

Figure 10. **Examples of LLR trajectories. Top: NMNIST-H. Bottom: NMNIST-100f.** The red curves represent $\min_l\{\hat{\lambda}_{y_i l}\}$, while the blue curves represent $\min_l\{\hat{\lambda}_{kl}\}$ ($k \neq y_i$). If the red curve reaches the threshold (necessarily positive), then the prediction is correct, while if the blue curve reaches the threshold, then the prediction is wrong. We plot ten different i's randomly selected from the validation set. The red and blue curves are gradually separated as more frames are observed: evidence accumulation.

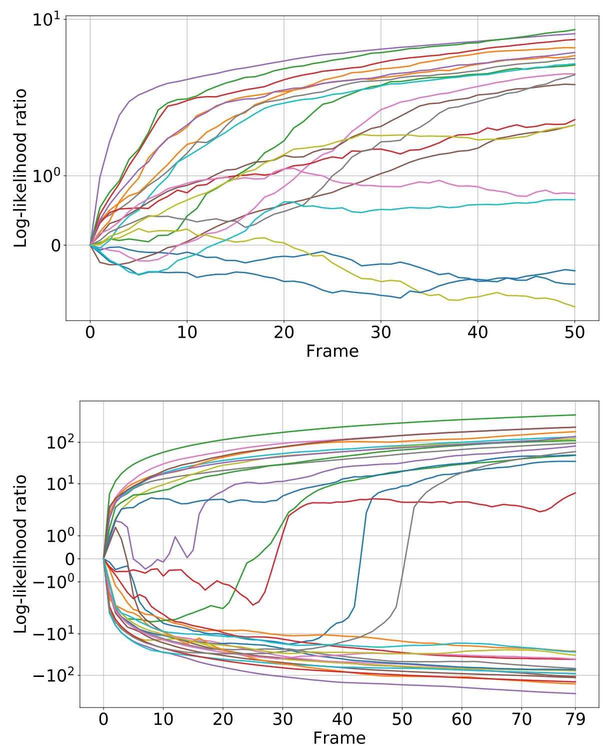

Figure 11. **Examples of LLR trajectories. Top: UCF101. Bottom: HMDB51.** All curves are $\min_l\{\hat{\lambda}_{y_i l}\}$, and the negative rows $\min_l\{\hat{\lambda}_{kl}\}$ ($k \neq y_i$) are omitted for clarity. Therefore, all the curves should be in the upper half-plane (positive LLRs). We plot 20 and 30 different i's randomly selected from the validation sets of UCF101 and HMDB51, respectively. Several curves gradually go upwards as more frames are observed: evidence accumulation.

K. NMNIST-H and NMNIST-100f

(Ebihara et al., 2021) propose an MNIST-based sequential dataset for early classification of time series: *NMNIST*; however, NMNIST is so simple that accuracy tends to saturate immediately and the performance comparison of models is difficult, especially in the early stage of sequential predictions. We thus create a more complex dataset, NMNIST-H, with higher noise density than NMNIST. Figure 12 is an example video of NMNIST-H. It is hard, if not impossible, for humans to classify the video within 10 frames.

NMNIST-100f is a more challenging dataset than NMNIST-H. Each video consists of 100 frames, which is 5 times longer than in NMNIST and NMNSIT-H. Figure 13 is an example video of NMNIST-100f. Because of the dense noise, classification is unrealistic for humans, while MSPRT-TANDEM attains approximately 90 % accuracy with only 8 frames (see Figure 5).

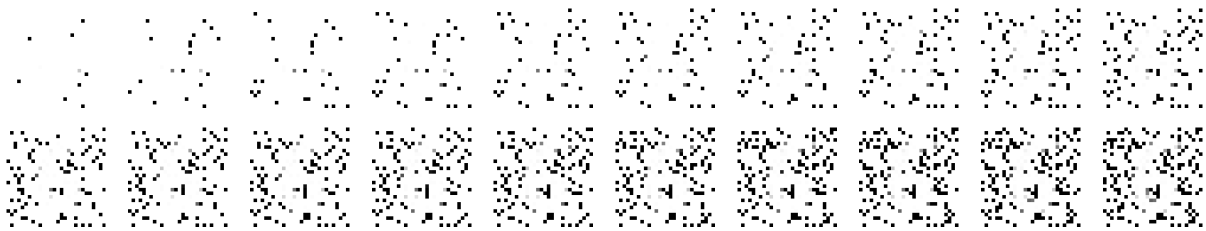

Figure 12. **NMNIST-H.** The first frame is at the top left, and the last frame is at the bottom right. The original MNIST image (28×28 pixels) is gradually revealed (10 pixels per frame). The label of this example is 6. The mean image is given in Figure 14 (left).

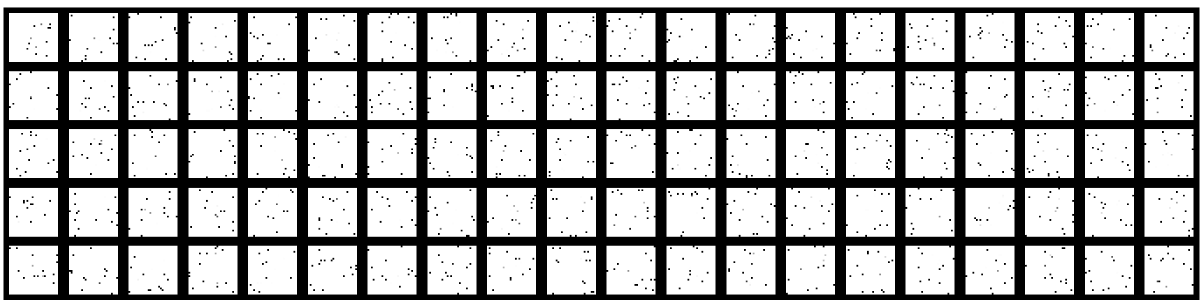

Figure 13. **NMNIST-100f**. The first frame is at the top left, and the last frame is at the bottom right. An MNIST image (28×28 pixels) is filled with white pixels except for 15 randomly selected pixels. Unlike NMNIST-H, the number of original pixels (15) is fixed throughout all frames. The label of this example is 3. The mean image is given in Figure 14 (right).

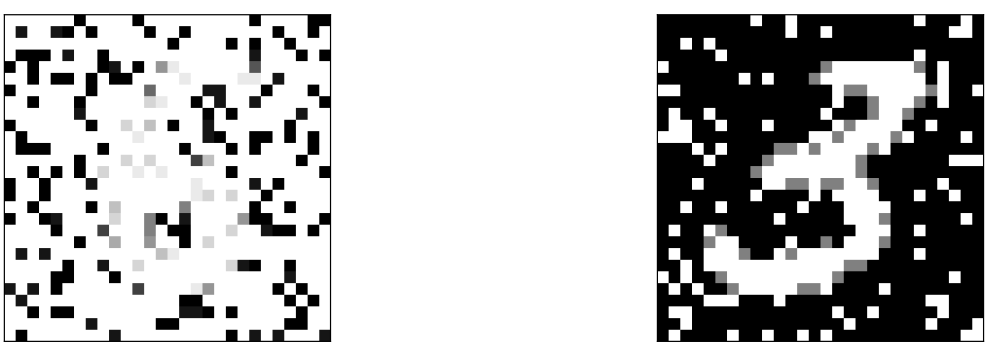

Figure 14. **Mean images of Figures 12 (left) and 13 (right).**

L. Details of Statistical Tests

L.1. Model Comparison: Figure 5

For an objective comparison, we conduct statistical tests: two-way ANOVA (Fisher, 1925) followed by Tukey-Kramer multi-comparison test (Tukey, 1949; Kramer, 1956). In the tests, a small number of trials reduces test statistics, making it difficult to claim significance because the test statistic of the Tukey-Kramer test is proportional to $1/\sqrt{(1/n + 1/m)}$, where n and m are trial numbers of two models to be compared. These statistical tests are standard, e.g., in biological science, in which variable trial numbers are inevitable in experiments. All the statistical tests are executed with a customized (MATLAB, 2017) script.

In the two-way ANOVA, the two factors are defined as the phase (early and late stages of the SAT curve) and model. The actual numbers of frames shown in Table 42 are chosen so that the compared models can use as similar frames as possible and thus depend only on the dataset. Note that EARLIEST cannot flexibly change the mean hitting time; thus, we include the results of EARLIEST to the groups with as close to the number of frames as possible.

The p-values are summarized in Tables 43 to 47. The p-values with asterisks are statistically significant: one, two and three asterisks show $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively. Our results, especially in the late phase, are statistically significantly better than those in the early phase, confirming that accumulating evidence leads to better performance.

L.2. Ablation Study: Figure 9

We also test the three conditions in the ablation study. We select one phase at the 20th frame to conduct the one-way ANOVA with one model factor: LSEL + Multiplet loss, LSEL only, and Multiplet only. The p-values are summarized in Table 48. The result shows that using both the LSEL and the multiplet loss is statistically significantly better than using either of the two losses.

Table 42. Definition of the phases (in the number of frames).

	Early phase		Late phase	
	EARLIEST	All but EARLIEST	EARLIEST	All but EARLIEST
NMNIST-H	1.34	2	13.41	13
NMNIST-100f	13.41	19	99.99	100
UCF101	1.36	1	49.93	50
HMDB51	1.43	1	36.20	36

Table 43. p-values from the Tukey-Kramer multi-comparison test conducted on NMNIST-100f.(Figure 2)

		Logistic		LSEL
		early	late	early
Logistic	late	***1E-07		
LSEL	early	***1E-09	***1E-09	
	late	***1E-09	*2E-3	***1E-09

Table 44. p-values from the Tukey-Kramer multi-comparison test conducted on NMNIST-H (Figure 5).

		MSPRT-TANDEM		NP test		LSTM-s		LSTM-m		EARLIEST
		early	late	early	late	early	late	early	late	early
MSPRT-TANDEM	late	***1E-07								
NP test	early	***1E-07	***1E-07							
	late	***1E-07	***1E-07	***1E-07						
LSTM-s	early	***1E-07	***1E-07	1.0	***1E-07					
	late	***1E-07	***1E-07	***1E-07	1.0	***1E-07				
LSTM-m	early	***1E-07	***1E-07	1.0	***1E-07	1.0	***1E-07			
	late	***1E-07	***1E-07	***1E-07	1.0	***1E-07	1.0	***1E-07		
EARLIEST	early	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	
	late	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07

Table 45. p-values from the Tukey-Kramer multi-comparison test conducted on NMNIST-100f (Figure 5).

		MSPRT-TANDEM		NP test		LSTM-s		LSTM-m		EARLIEST
		early	late	early	late	early	late	early	late	early
MSPRT-TANDEM	late	***1E-07								
NP test	early	***1E-07	***1E-07							
	late	***1E-07	1.0	***1E-07						
LSTM-s	early	***1E-07	***1E-07	***1E-07	***1E-07					
	late	***1E-07	1.0	***1E-07	1.0	***1E-07				
LSTM-m	early	***1E-07	***1E-07	***1E-07	***1E-07	1.0	***1E-07			
	late	***1E-07	1.0	***1E-07	1.0	***1E-07	1.0	***1E-07		
EARLIEST	early	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	
	late	***7E-05	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07

Table 46. p-values from the Tukey-Kramer multi-comparison test conducted on UCF101 (Figure 5).

		MSPRT-TANDEM		NP test		LSTM-s		LSTM-m		EARLIEST
		early	late	early	late	early	late	early	late	early
MSPRT-TANDEM	late	***1E-07								
NP test	early	1.0	***1E-07							
	late	***1E-07	1.0	***1E-07						
LSTM-s	early	***1E-07	***1E-07	***1E-07	***1E-07					
	late	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07				
LSTM-m	early	***1E-07	***1E-07	***1E-07	***1E-07	1.0	***1E-07			
	late	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	*3E-03	***1E-07		
EARLIEST	early	***1E-07	***1E-07	***1E-07	***1E-07	1E-01	***1E-07	*6E-03	***1E-07	
	late	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07

Table 47. p-values from the Tukey-Kramer multi-comparison test conducted on HMDB51 (Figure 5).

		MSPRT-TANDEM		NP test		LSTM-s		LSTM-m		EARLIEST
		early	late	early	late	early	late	early	late	early
MSPRT-TANDEM	late	***1E-07								
NP test	early	1.0	***1E-07							
	late	***1E-07	0.2	***1E-07						
LSTM-s	early	1.0	***1E-07	1.0	***1E-07					
	late	***1E-07	***2E-07	***1E-07	***2E-02	***1E-07				
LSTM-m	early	1.0	***1E-07	1.0	***1E-07	1.0	***1E-07			
	late	***1E-07	***1E-07	***1E-07	***1E-04	***1E-07	1.0	***1E-07		
EARLIEST	early	1.0	***1E-07	1.0	***1E-07	1.0	***1E-07	1.0	***1E-07	
	late	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***1E-07	***7E-07	***1E-07

Table 48. Figure 9: p-values from the Tukey-Kramer multi-comparison test conducted on the ablation test.

	MSPRT-TANDEM	
	LSEL+Multiplet	LSEL only
LSEL only	***1E-09	
Multiplet only	***2E-07	***1E-09

M. Supplementary Discussion

Interpretability. Interpretability of classification results is one of the important interests in early classification of time series (Xing et al., 2011; Ghalwash & Obradovic, 2012; Ghalwash et al., 2013; Ghalwash et al., 2014; Karim et al., 2019). MSPRT-TANDEM can use the LLR trajectory to visualize the prediction process (see Figures 2 (Bottom) and 10); a large gap of LLRs between two timestamps means that these timestamps are decisive.

Threshold matrix. In our experiment, we use single-valued threshold matrices for simplicity. General threshold matrices may enhance performance, especially when the dataset is class-imbalanced (Longadge & Dongre, 2013; Ali et al., 2015; Hong et al., 2016). Tuning the threshold after training is referred to as thresholding, or threshold-moving (Richard & Lippmann, 1991; Buda et al., 2018). MSPRT-TANDEM has multiple thresholds in the matrix form, and thus it is an interesting future work to exploit such a matrix structure to attain higher accuracy.

How to determine threshold. A user can choose a threshold by evaluating the mean hitting time and accuracy on a dataset at hand (possibly the validation dataset, training dataset, or both). As mentioned in Section 3.4, we do not have to retrain the model (mentioned in Section 3.4). We can modify the threshold even after deployment, if necessary, to address domain shift. This flexibility is a huge advantage compared with other models that require additional training every time the user wants to control the speed-accuracy tradeoff (Cai et al., 2020).

Supplementary references

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. License: Apache License 2.0. Software available from tensorflow.org.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, pp. 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. License: MIT License.

Ali, A., Shamsuddin, S. M., Ralescu, A. L., et al. Classification with class imbalance problem: a review. *Int. J. Advance Soft Compu. Appl*, 7(3):176–204, 2015.

Arrow, K. J., Blackwell, D., and Girshick, M. A. Bayes and minimax solutions of sequential decision problems. *Econometrica, Journal of the Econometric Society*, pp. 213–244, 1949.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. Algorithms for hyper-parameter optimization. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 24, pp. 2546–2554. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

Buda, M., Maki, A., and Mazurowski, M. A. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

Cai, H., Gan, C., Wang, T., Zhang, Z., and Han, S. Once-for-all: Train one network and specialize it for efficient deployment. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HylxE1HKwS.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, L. Universal transformers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyzdRiR9Y7.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Dragalin, V. P., Tartakovsky, A. G., and Veeravalli, V. V. Multihypothesis sequential probability ratio tests. i. asymptotic optimality. *IEEE Transactions on Information Theory*, 45(7):2448–2461, November 1999. doi: 10.1109/18.796383.

Ebihara, A. F., Miyagawa, T., Sakurai, K., and Imaoka, H. Sequential density ratio estimation for simultaneous optimization of speed and accuracy. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Rhsu5qD36cL.

Ferguson, T. S. *Mathematical statistics: A decision theoretic approach*, volume 1. Academic press, 2014.

Fisher, R. *Statistical methods for research workers*. Edinburgh Oliver & Boyd, 1925.

Gangopadhyay, T., Ramanan, V., Akintayo, A., Boor, P. K., Sarkar, S., Chakravarthy, S. R., and Sarkar, S. 3d convolutional selective autoencoder for instability detection in combustion systems, 2021.

Gers, F. A. and Schmidhuber, J. Recurrent nets that time and count. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, 3:189–194 vol.3, 2000.

Ghalwash, M. F. and Obradovic, Z. Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC bioinformatics*, 13(1):195, 2012.

Ghalwash, M. F., Radosavljevic, V., and Obradovic, Z. Extraction of interpretable multivariate patterns for early diagnostics. In *2013 IEEE 13th International Conference on Data Mining*, pp. 201–210, 2013. doi: 10.1109/ICDM.2013.19.

Ghalwash, M. F., Radosavljevic, V., and Obradovic, Z. Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 402–411, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623694. URL https://doi.org/10.1145/2623330.2623694.

Ghodrati, A., Bejnordi, B. E., and Habibian, A. FrameExit: Conditional early exiting for efficient video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

Golubev, G. and Khas'minskii, R. Sequential testing for several signals in gaussian white noise. *Theory of Probability & Its Applications*, 28(3):573–584, 1984.

Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Gupta, A., Pal, R., Mishra, R., Gupta, H. P., Dutta, T., and Hirani, P. Game theory based early classification of rivers using time series data. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 686–691, 2019. doi: 10.1109/WF-IoT.2019.8767251.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *The journal of machine learning research*, 13(1):307–361, 2012.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., Del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with NumPy. *Nature*, 585(7825):357–362, 09 2020. License: BSD 3-Clause "New" or "Revised" License.

Hartvigsen, T., Sen, C., Kong, X., and Rundensteiner, E. Adaptive-halting policy network for early classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 101–110, New York, NY, USA, 2019. ACM.

Hatami, N. and Chira, C. Classifiers with a reject option for early time-series classification. *2013 IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL)*, pp. 9–16, 2013.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pp. 630–645, 2016b.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

Hong, C., Ghosh, R., and Srinivasan, S. Dealing with class imbalance using thresholding. *arXiv preprint arXiv:1607.02705*, 2016.

Huang, Z., Ye, Z., Li, S., and Pan, R. Length adaptive recurrent model for text classification. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, pp. 1019–1027, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3132947. URL https://doi.org/10.1145/3132847.3132947.

Kanamori, T., Hido, S., and Sugiyama, M. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009.

Karim, F., Darabi, H., Harford, S., Chen, S., and Sharabiani, A. A framework for accurate time series classification based on partial observation. In *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 634–639, 2019. doi: 10.1109/COASE.2019.8843256.

Kaya, Y., Hong, S., and Dumitras, T. Shallow-Deep Networks: Understanding and mitigating network overthinking. In *International Conference on Machine Learning*, pp. 3301–3310. PMLR, 2019.

Khan, H., Marcuse, L., and Yener, B. Deep density ratio estimation for change point detection. *arXiv preprint arXiv:1905.09876*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kramer, C. Y. Extension of multiple range tests to group means with unequal numbers of replications. *Biometrics*, 12(3):307–310, 1956.

Kullback, S. and Leibler, R. A. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010. License: Creative Commons Attribution-Share Alike 3.0 license.

Lenyk, Z. and Park, J. Microsoft Vision Model ResNet-50. URL https://pypi.org/project/microsoftvision/. Accessed: May 14, 2021. License: Unknown.

Longadge, R. and Dongre, S. Class imbalance problem in data mining review. *arXiv preprint arXiv:1305.1707*, 2013.

López-Monroy, A. P., González, F. A., Montes, M., Escalante, H. J., and Solorio, T. Early text classification using multi-resolution concept representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1216–1225, 2018.

Lorden, G. Nearly-optimal sequential tests for finitely many parameter values. *Annals of Statistics*, 5:1–21, 01 1977. doi: 10.1214/aos/1176343737.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ma, S., Sigal, L., and Sclaroff, S. Learning activity progression in lstms for activity detection and early detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1942–1950, 2016.

Martinez, C., Ramasso, E., Perrin, G., and Rombaut, M. Adaptive early classification of temporal sequences using deep reinforcement learning. *Knowledge-Based Systems*, 190:105290, February 2020. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2019.105290. URL http://www.sciencedirect.com/science/article/pii/S0950705119305829.

MATLAB. *version 9.3.0 (R2017b)*. The MathWorks Inc., Natick, Massachusetts, 2017.

McGovern, A., Rosendahl, D. H., Brown, R. A., and Droegemeier, K. K. Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction. *Data Mining and Knowledge Discovery*, 22(1-2): 232–258, 2011.

Menon, A. K., Jayasumana, S., Rawat, A. S., Jain, H., Veit, A., and Kumar, S. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=37nvvqkCo5.

Mori, U., Mendiburu, A., Keogh, E. J., and Lozano, J. A. Reliable early classification of time series based on discriminating the classes over time. *Data Mining and Knowledge Discovery*, 31:233–263, 04 2016. doi: 10.1007/s10618-016-0462-1.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. License: https://github.com/pytorch/pytorch/blob/master/LICENSE.

Richard, M. D. and Lippmann, R. P. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Rußwurm, M., Lefèvre, S., Courty, N., Emonet, R., Körner, M., and Tavenard, R. End-to-end learning for early classification of time series. *CoRR*, abs/1901.10681, 2019. URL http://arxiv.org/abs/1901.10681.

Schäfer, P. and Leser, U. TEASER: Early and accurate time series classification. *Data Mining and Knowledge Discovery*, 34(5):1336–1362, 2020.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

Suzuki, T., Kataoka, H., Aoki, Y., and Satoh, Y. Anticipating traffic accidents with adaptive loss and large-scale incident db. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3521–3529, 2018.

Tartakovskij, A. G. Sequential composite hypothesis testing with dependent nonstationary observations. *Probl. Inf. Transm.*, 17:18–28, 1981. ISSN 0032-9460; 1608-3253/e.

Tartakovsky, A. Sequential methods in the theory of information systems, 1991.

Tartakovsky, A. Asymptotically optimal sequential tests for nonhomogeneous processes. *Sequential analysis*, 17(1): 33–61, 1998a.

Tartakovsky, A., Nikiforov, I., and Basseville, M. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. Chapman & Hall/CRC, 1st edition, 2014.

Tartakovsky, A. G. Asymptotic optimality of certain multihypothesis sequential tests: Non-i.i.d. case. *Statistical Inference for Stochastic Processes*, 1(3):265–295, 1998b.

Tukey, J. W. Comparing individual means in the analysis of variance. *Biometrics*, 5 2:99–114, 1949.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Verdenskaya, N. and Tartakovskii, A. Asymptotically optimal sequential testing of multiple hypotheses for nonhomogeneous gaussian processes in asymmetric case. *Theory of Probability & Its Applications*, 36(3):536–547, 1992.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . . SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. License: BSD 3-Clause "New" or "Revised" License.

Wald, A. Sequential tests of statistical hypotheses. *Ann. Math. Statist.*, 16(2):117–186, 06 1945.

Wang, B., Huang, L., and Hoai, M. Active vision for early recognition of human actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Wang, W., Chen, C., Wang, W., Rai, P., and Carin, L. Earliness-aware deep convolutional networks for early time series classification. *arXiv preprint arXiv:1611.04578*, 2016.

Xing, Z., Pei, J., Yu, P. S., and Wang, K. Extracting interpretable features for early classification on time series. In *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, pp. 247–258. SIAM, 2011.

Xing, Z., Pei, J., and Yu, P. S. Early classification on time series. *Knowledge and Information Systems*, 31(1):105–127, April 2012.

Zhou, W., Xu, C., Ge, T., McAuley, J., Xu, K., and Wei, F. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems*, 2020. URL https://papers.nips.cc/paper/2020/file/d4dd111a4fd973394238aca5c05bebe3-Paper.pdf.