A. Proof of Results of Section 3

We now provide the proofs of the main results in our paper.

Filtration for General System: We setup some notations before proceeding to the proofs. Let the underlying probability space on which the system is evolving be $(\Omega, \mathcal{F}, \mathbb{P})$. We recall from Sec.2 that the system evolves in stages, where each round $\tau \geq 1$ has four stages $st \in \{0,1,2,3\}$. In stage 0 the samples in the memory are matched with the classifiers. In stage 1 the classifiers label the alloted samples and the sample labels get updated. In stage 2 based on the updated labels samples either reenter the system or leave the system with a final label. The history at the beginning of stage st in round τ is denoted as H^τ_{st}. The sigma algebra generated by the histories H^τ_{st} are $\mathcal{F}^\tau_{st} = \sigma(H^\tau_{st})$ for $i = 0,1,2,3$ for all $\tau \geq 1$. These sigma algebras further forms a filtration $\mathcal{F}^\tau_0 \subseteq \mathcal{F}^\tau_1 \subseteq \mathcal{F}^\tau_2 \subseteq \mathcal{F}^\tau_3 \subseteq \mathcal{F}^{(\tau+1)}_0, \forall \tau \geq 1$.

There exists a sequence of r.v.s $\{U^\tau : \tau \geq 1\}$ which is independent of any other randomness in the system such that the following statements hold for any causal policy ϕ and for all rounds $\tau \geq 1$.[2]
(i) The matching decision $A^\tau = \phi_0(U^\tau, H^\tau_0)$.
(ii) The departure and the final labeling decisions $\{D^\tau, \{k^*_j : j \in D^\tau\} = \phi_2(U^\tau, H^\tau_2)$.

A.1. Independence of Routing and Belief

Proof of Lemma 3.4. Let us consider a causal policy ϕ, under which at slot $\tau \geq 1$ the state is S^τ. For any $\mathbf{t} \in T = [K]^{|S^\tau|}$, let $E(\mathbf{t}) = \{s_j[M] = t_j : j \in S^\tau\}$ denote that $\mathbf{t}$ is the true label for the samples in S^τ. From Bayes' rule the probability of event $E(\mathbf{t})$ conditioned on the past is

$$\mathbb{P}(E(\mathbf{t})|\mathcal{F}^\tau_0) = \frac{\mathbb{P}(H^\tau_0|E(\mathbf{t}))\mathbb{P}(E(\mathbf{t}))}{\sum_{\mathbf{t}' \in T} \mathbb{P}(H^\tau_0|E(\mathbf{t}'))\mathbb{P}(E'(\mathbf{t}'))}$$

We next quantify the probability of any specific history h^τ_0 given a particular event $E(\mathbf{t})$.[3]

$$\begin{aligned}
\mathbb{P}(h^\tau_0|E(\mathbf{t})) &= \mathbb{P}(h^{(\tau-1)}_3 \cap N_{ex}(\tau{-}1)|E(\mathbf{t})) \\
&\overset{(i)}{=} \mathbb{P}(h^{(\tau-1)}_0|E(\mathbf{t}))\mathbb{P}(N_{ex}(\tau-1)) \prod_{i=1}^{3} \mathbb{P}(h^{(\tau-1)}_i|h^{(\tau-1)}_{(i-1)}, E(\mathbf{t})) \\
&\overset{(ii)}{=} \mathbb{P}(h^{(\tau-1)}_0|E(\mathbf{t})) \prod_{i=1,3} \mathbb{P}(h^{(\tau-1)}_i|h^{(\tau-1)}_{(i-1)}) \times \dots \\
&\dots \times \mathbb{P}(h^{(\tau-1)}_2|h^{(\tau-1)}_1, E(\mathbf{t})) \\
&\overset{(iii)}{=} \mathbb{P}(h^{(\tau-1)}_0|E(\mathbf{t})) \prod_{i=1,3} \mathbb{P}(h^{(\tau-1)}_i|h^{(\tau-1)}_{(i-1)}) \times \dots \\
&\dots \times \left(\prod_{\substack{i \in [M]: j \neq \mathbf{e}, \\ j = A^{(\tau-1)}(i)}} \mathbb{P}(s_j[i] = \hat{s}^\tau_j[i]|s_j[M] = t_j) \right) \\
&\overset{(iv)}{=} \phi(h^\tau_0) \prod_{\tau' \geq 1}^{\tau-1} \prod_{\substack{i \in [M]: j \neq \mathbf{e}, \\ j = A^{(\tau-1)}(i)}} C_i(t_j, \hat{s}^{(\tau'+1)}_j[i]) \\
&\overset{(v)}{=} \phi(h^\tau_0) \prod_{j \in S^\tau} \prod_{i \in [M]} C_i(t_j, \hat{s}^\tau_j[i]). \text{ [with } C_i(k, \mathbf{e}) = 1]
\end{aligned}$$

In equality (iii) we use the notation

$$\phi(h^\tau_0) = \prod_{\tau' \geq 1}^{\tau-1} \prod_{i=1,2,4} \mathbb{P}(h_i(\tau')|h_{(i-1)}(\tau')).$$

The terms $\phi(h^\tau_0)$ denote the contribution of the policy and arrival process towards the probability $\mathbb{P}(h^\tau_0|E(\mathbf{t}))$. We explain the equalities as follows.

- In equality (i), we use the independence of the arrival process from any other processes.
- Under the causal policy ϕ the matching, departure and labeling decisions are independent of the true label, hence $E(\tau)$, given the history at the time of the decisions. This is true because $A^\tau = \phi_0(U^\tau, H^\tau_0)$ and $\{D^\tau, \{k^*_j : j \in D^\tau\} = \phi_2(U^\tau, H^\tau_2)$. Thus, equality (ii) follows.
- However, the labeling events are dependent on the true labels of the samples. Furthermore, due to independence of the label assignment process for the OnDS model we obtain the product form in equality (iii).
- The equality (iv) follows by iteratively carrying out the above steps. We also use the fact that due to Assumption 3.3, $\mathbb{P}(h_0(1)|E(\mathbf{t})) = 1$.
- The equality (v) follows from the observation that the terms present in the product corresponds to the labels of the samples in S^τ. This holds as once the sample is labeled by a classifier it may never change due to the *deterministic labeling of classifiers.*

Furthermore, we have for the OnDS model

$$\mathbb{P}(E(\mathbf{t})) = \prod_{j \in S^\tau} \mathbf{P}[s_j[M] = t_j] = \prod_{j \in S^\tau} \mathbf{p}_g(t_j).$$

Now by substituting the value of $\mathbb{P}(h^\tau_0|E(\mathbf{t}))$ and $\mathbb{P}(E(\mathbf{t}))$, we obtain the following.

$$\mathbb{P}(s_j[M] = t_j : j \in S^\tau|\mathcal{F}^\tau_0)$$

[2] The existence is guaranteed for our system which evolves in discrete time with countable decision sets at each stage.

[3] Here we slightly abuse notation to denote $\mathbb{P}(H^\tau_i = h^\tau_i|E(\mathbf{t}))$ as $\mathbb{P}(h^\tau_i|E(\mathbf{t}))$ for all $E(\mathbf{t})$, i and τ.

$$= \frac{\phi(H_0^\tau)\prod_{j\in S^\tau}\mathbf{p}_g(t_j)\prod_{i\in[M]}C_i(t_j,\hat{s}_j^\tau[i])}{\phi(H_0^\tau)\sum_{\mathbf{t}'\in T}\prod_{j\in S^\tau}\mathbf{p}_g(t_j)\prod_{i\in[M]}C_i(t_j,\hat{s}_j^\tau[i])}$$

$$\overset{(vi)}{=} \frac{\prod_{\ell\in[K]_e^M,k\in[K]}\left(\mathbf{p}_g(k)\prod_{i\in[M]}C_i(k,\ell[i])\right)^{n_{k\,\ell}^{(\tau,0)}(\mathbf{t})}}{\sum_{\mathbf{t}'\in T}\prod_{\ell,k}\left(\mathbf{p}_g(k)\prod_{i\in[M]}C_i(k,\ell[i])\right)^{n_{k\,\ell}^{(\tau,0)}(\mathbf{t}')}}$$

$$\overset{(vii)}{=} \frac{\prod_{\ell\in[K]_e^M,k\in[K]}P_\Psi(\ell,k)^{n_{k\,\ell}^{(\tau,0)}(\mathbf{t})}}{\sum_{\mathbf{t}'\in T}\prod_{\ell,k}P_\Psi(\ell,k)^{n_{k\,\ell}^{(\tau,0)}(\mathbf{t}')}}$$

Here, $n_{k\,\ell}^{(\tau,0)}(S,\mathbf{t})=|\{j\in S^\tau : t_j=k,\hat{\mathbf{s}}_j^\tau=\ell\}|$, $\forall k\in[K]$, $\ell\in[K]_e^M$. By rearranging all the j that has $t_j=k$ and the partial label as ℓ we obtain equality (vi). Finally, (vii) follows simply from definition of $P_\Psi(\ell,k)$. This concludes the proof for the part with $st=0$ in Lemma 3.4.

For the other stages, similar argument can be used to arrive at the conclusion presented in the lemma statement. □

A.2. Pareto Region Conserving Compression of History

We now prove the characterization of the capacity region. Before getting into the main proof we first present an outline for the proof.

Proof outline: The key to the proof of Theorem 3.6 is to show that the evolution of the random variables $Q_\ell(\tau)$ and $\inf_{j\in D_p(\tau)} Acc_j(\tau)$ under any causal policy ϕ can be matched (in a sense which we shortly make precise) with another causal policy, namely ${}_c\phi$, which is a function of a compressed history. This compress history is obtained from the history by *retaining only the partial labels* of the sample; while *discarding the sample ids*. This will imply that that the policy ${}_c\phi$ achieves the same throughput and threshold accuracy as the causal policy ϕ. Finally, we show as the decisions only depend on the partial labels, these can be mapped to the choice of hyper-edges in a network as function of $(\lambda_{ex},\theta_{th})$. Furthermore, we show the question of Pareto optimality can be reduced to feasibility of a network flow problem in the constructed network. As a final step, we show the feasibility of this network flow is equivalent to the feasibility of the network flow problem in Sec. 3.

Notation: In the following discussion, we use the notion of *multi-sets* where the elements can be repeated. Further, the sets are assumed to be sorted with respect to the entries. Thus, the sets are ordered. All the equalities involving random variables hold in *almost sure* sense, if not mentioned otherwise. To differentiate a quantity related to compressed causal policies from the same quantity for causal policies we add the *prescript c*, i.e. X in causal policy is denoted as ${}_cX$ in compressed causal policy.

Compressed History: We now present the compressed history where the sample partial labels are kept but the sample ids are dropped. Recall, for round $\tau\geq 1$, the matching decision is A^τ and the departure and the labeling decisions are $\{D^\tau,\{k_j^*: j\in D^\tau\}\}$. The compressed memory is given as the multi-set ${}_cS^\tau=\{\hat{\mathbf{s}}_j^\tau : j\in S^\tau\}$. Let us denote the compressed matching decisions as ${}_cA^\tau$ where for all $i\in[M]$ ${}_cA^\tau(i)=\hat{\mathbf{s}}_j^\tau$ for $j=A^\tau(i)$. The compressed departure decisions are given as a multi-set ${}_cD^\tau=\{\hat{\mathbf{s}}_j^{(\tau+1)} : j\in D^\tau\}$. Similarly, the partial labels are ${}_cD_p^\tau=\{\hat{\mathbf{s}}_j^{(\tau+1)} : j\in D_p^\tau\}$. Finally, the labeling decisions are $\{k_\ell^* : \ell\in {}_cD^\tau\}$. Note that the classifier labels are unchanged as they do not contain sample ids. These labels are given as Cl^τ.

The compressed histories are iteratively defined as in general system with A^τ replaced with ${}_cA^\tau$ and $\{D^\tau,\{k_j^*: j\in D^\tau\}$ replaced with $\{{}_cD^\tau,\{k_\ell^*:\ell\in{}_cD^\tau\}$. For all $\tau\geq 1$ and $st\in\{0,1,2,3\}$. Let us call the compressed version of history H_{st}^τ as ${}_cH_{st}^\tau$. Similarly, we define the sigma algebras ${}_c\mathcal{F}_i^\tau$ and the filtrations with the random variables ${}_cH_i^\tau$ for $i=1,2,3$ and $\tau\geq 1$.

Therefore, we have deterministic maps mapping the decisions and histories with their compressed counterpart. The deterministic maps are $f_A(\cdot)$ for the matching decisions, and $f_D(\cdot)$ for the departure and labeling decisions. For each $i=0,1,2,3$, there exist deterministic maps $g_i(\cdot)$, which maps each realization of H_i^τ to a realization of ${}_cH_{st}^\tau$, for all τ. We also define their inverse maps $f_A^{-1}(\cdot)$, $f_D^{-1}(\cdot)$, and $g_i^{-1}(\cdot)$.[4] Note that these maps are same for all rounds $\tau\geq 1$.

Compressed Causal Policy: A policy is compressed causal if the decisions depend on the compressed histories. Specifically, for each $\tau\geq 1$ under a causal policy ${}_c\phi$, the matching ${}_cA^\tau$ is a random function of history ${}_cH_0^\tau$ and parameters $\mathcal{P}$, and the departure ${}_cD^\tau$ and and final labeling $\{k_\ell^*:\ell\in{}_cD^\tau\}$ is a random function of history ${}_cH_2^\tau$ and parameters $\mathcal{P}$. The class of *compressed causal policies* is denoted as ${}_c\mathcal{C}$.

We now prove in the equivalence of compressed causal policy class and causal policy class w.r.t. the queue length evolution and the accuracy of the departing samples.

Lemma A.1. *Given a system $\mathcal{P}$ and a causal policy ϕ, there exists a compressed causal policy ${}_c\phi$ satisfying:*
1) the queue lengths have the same distribution:

$$\forall\tau\geq 1;\ \mathbb{P}^{{}_c\phi}(\mathbf{Q}(\tau)|H_0^1)=\mathbb{P}^{\phi}(\mathbf{Q}(\tau)|H_0^1)\ a.s.,$$

2) the minimum accuracy of the departing partially labeled samples have the same distribution:

$$\forall\tau\geq 1,\ \mathbb{P}^{{}_c\phi}\left(\{\inf_{j\in{}_cD_p^\tau}Acc_j(\tau)<\theta_{th}\}|H_0^1\right) = \mathbb{P}^{\phi}\left(\{\inf_{j\in D_p^\tau}Acc_j(\tau)<\theta_{th}\}|H_0^1\right)\ a.s.$$

[4] For any map $f: X\to Y$, its inverse map $f^{-1}(\cdot)$ is a set valued function $f^{-1}(C)=\{x\in X : f(x)\in C\}$ for all $C\subseteq Y$.

We prove Lemma A.1 shortly. Given a causal policy ϕ, let us first construct a compressed causal policy ${}_c\phi = CC(\phi)$ which we use in the proof of Lemma A.1. The policy ${}_c\phi$ is given as follows. For all $\tau \geq 1$, matching decisions ${}_ca^\tau$, departure decisions ${}_cd^\tau$, and histories ${}_ch^\tau_{st}$ for stages $st \in \{0, 1, 2, 3\}$.

$$P1)\ \mathbb{P}^{{}_c\phi}\left({}_cA^\tau = {}_ca^\tau | {}_cH_1^\tau = {}_ch_1^\tau\right) = \mathbb{P}^{\phi}\left(A^\tau \in f_A^{-1}({}_ca^\tau) | H_1^\tau \in g_1^{-1}({}_ch_1^\tau)\right)$$

$$P2)\ \mathbb{P}^{\phi^Q}\left({}_cD^\tau = {}_cd^\tau | {}_cH_3^\tau = {}_ch_3^\tau\right) = \mathbb{P}^{\phi}\left(D^\tau \in f_D^{-1}({}_cd^\tau) | H_3^\tau \in g_3^{-1}({}_ch_3^\tau)\right)$$

The following lemma relates the evolution of compressed histories for policies ${}_c\phi$ and ϕ.

Lemma A.2. *For a given system $\mathcal{P}$ and a causal policy ϕ, the policy ${}_c\phi = CC(\phi)$ satisfies the following statements, for all round $\tau \geq 1$ and stages $st \in \{0, 1, 2, 3\}$, and for all feasible histories ${}_ch^\tau_{(st+1)}, {}_ch^\tau_{st}$ (where, $H_4^\tau = H_0^{(\tau+1)}$)*

$$\mathbb{P}^{\phi}\left(H^\tau_{(st+1)} \in g^{-1}_{(st+1)}({}_ch^\tau_{(st+1)}) | H^\tau_{st} \in g^{-1}_{st}({}_ch^\tau_{st})\right) = \mathbb{P}^{{}_c\phi}\left({}_cH^\tau_{(st+1)} = {}_ch^\tau_{(st+1)} | {}_cH^\tau_i = {}_ch^\tau_{st}\right).$$

The following corollary is an easy consequence of the above lemma.

Corollary A.3. *For a given system $\mathcal{P}$ and a causal policy ϕ, the policy ${}_c\phi = CC(\phi)$ satisfies the following equation for all round $\tau \geq 1$, stages $st \in \{0, 1, 2, 3\}$, and feasible histories h^τ_{st}.*

$$\mathbb{P}^{{}_c\phi}\left({}_cH^\tau_{st} = {}_ch^\tau_{st} | H_0^1\right) = \mathbb{P}^{\phi}\left(H^\tau_{st} \in g^{-1}_{st}({}_ch^\tau_{st}) | H_0^1\right) \quad (4)$$

Proof Sketch. From Lemma A.2 through repeated iteration over the stages the proof follows. □

Similar to the class of causal policies, the Pareto region of the compressed causal policy class is defined as: ${}_c\Lambda(\psi) = conv\left(\underline{{}_c\Lambda}(\psi)\right)$, where $\underline{{}_c\Lambda}(\psi) = \{(\lambda_{ex}, \theta_{th}) : \exists {}_c\phi \in {}_cC : (\lambda_{ex}, \theta_{th}) \in \Lambda^{{}_c\phi}(\Psi)\}$. Also, recall $\Lambda(\Psi)$ and $\underline{\Lambda}(\Psi)$ are the counterparts for the causal policy class.

Lemma A.4. *Given a OnDS model Ψ, the Pareto region of the compressed causal policy class is equal to the Pareto region of the class of causal policies ${}_c\Lambda(\psi) = \Lambda(\psi)$*

Proof. For each $(\lambda_{ex}, \theta_{th}) \in \underline{\Lambda}(\Psi)$, there exists a policy ϕ so that $(\lambda_{ex}, \theta_{th}) \in \Lambda^{\phi}(\Psi)$. But, Lemma A.1 implies that there exists a compressed causal policy ${}_c\phi$ such that $(\lambda_{ex}, \theta_{th}) \in \Lambda^{{}_c\phi}(\Psi)$. This further implies, $\underline{\Lambda}(\Psi) = \underline{{}_c\Lambda}(\Psi)$. Finally, taking the convex closure in both sides we obtain, $\Lambda(\Psi) = {}_c\Lambda(\Psi)$. □

Due to Lemma A.4, characterizing the Pareto region for compressed causal policies suffice to prove Theorem 3.6. To recap the compressed history only contain the partial labels of the samples and is oblivious to sample ids.

A.3. Pareto Region of Compressed Causal Policy Class

As the compressed history is oblivious to the sample ids, the evolution of the system is fully described by the evolution of the count of partial labels in the system, namely $Q_\ell(\tau)$ for $\tau \geq 1$ and $\ell \in [K]_e^M$. Further, the matching, the departure and labeling decisions can be mapped to network resource allocation decisions. In each round the decisions are taken in two stages. Such two staged decision problems arise in stochastic optimization literature (Gopalan et al., 2012). We now describe this network resource allocation problem.

Network of Virtual Queues: The network is formed by virtual queues $Q_\ell(\tau)$ for $\ell \in [K]_e^M$. Thus, the set of nodes is $[K]_e^M$.

Departure: The matching decisions are between the classifiers and partial labels. We can view a matching as an outgoing hyper-edge $\mathfrak{H}_o$. The set of all possible outgoing hyper-edges are $(\{\mathbf{e}\} \cup [K]_e^M)^M$. Here, $\mathfrak{H}_o(i) \neq \mathbf{e}$ implies a virtual packet exists node $\ell = \mathfrak{H}_o(i)$. Therefore, the service (*maximum possible* departure) for any label ℓ, for the choice of hyper-edge $\mathfrak{H}_o$ is $(\sum_{i\in[M]} \mathbb{1}(\mathfrak{H}_o(i) = \ell))$. The number of departures from node ℓ is $\min\{Q_\ell(\tau), \mathbb{1}(\mathfrak{H}_o(i) = \ell)\}$.

Here, we note that the departure from any label node ℓ in round τ is at most $Q_\ell(\tau)$. Thus even if $\mathfrak{H}_o(i) \neq \mathbf{e}$ the classifier i may remain idle if there are not enough samples of type ℓ.

Labeling: Once matched with samples, the classifiers provide the labels Cl^τ to the matched samples. Due to Lemma 3.4, if a classifier is allotted a sample with partial label ℓ, the labeling is i.i.d. across rounds. Specifically, given any $\mathfrak{H}_o$ the probability that the new labels are cl for some $cl \in (\{\mathbf{e}\} \cup [K])^M$ is given as a constant $P_\Psi(cl, \mathfrak{H}_o)$.

Let $\mathcal{L}^{(cl,\mathfrak{H}_o)} \in (\{\mathbf{e}\} \cup [K]_e^M)^M$ denote the updated partial labels for the samples matched with the classifiers under $\mathfrak{h}_o$. This implies the updated label for the sample matched with classifier i is $\mathcal{L}^{(cl,\mathfrak{H}_o)}(i)$.

Arrival: The stage 2 decisions are that of departure and final labeling. The departure and final labeling decision is a function of the new labels cl and the matching decision $\mathfrak{H}_o$, denoted as $\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}$. For all cl and $\mathfrak{H}_o$, the set of stage 2 decisions is given as $([K] \cup \{\mathbf{e}\})^M$. Here, $\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}(i) = k \in [K]$ implies the sample matched with classifier i is given a final label k and exits the system. Otherwise, if $\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}(i) = \mathbf{e}$ then the sample matched with classifier i

re-enters the system.

The source node $\{\mathbf{e}\}^M$ has a constant arrival rate λ_{ex}. For the choice $\mathfrak{H}_o$, new labels cl, and the choice $\mathfrak{H}_{in}^{(\mathfrak{H}_o, cl)}$, the average number of arrival to the label node ℓ is

$$\sum_{i\in[M]} \mathbb{1}(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}(i) = \mathbf{e} \wedge \mathcal{L}^{(cl,\mathfrak{H}_o)}(i) = \ell).$$

Pareto Region: We now apply standard techniques in network optimization literature (Tassiulas & Ephremides, 1992; Stolyar, 2005), to obtain the Pareto region for compressed causal policies. A policy that choses the decisions independently according to a fixed distribution is known as *static-split policy* in the literature(Tassiulas & Ephremides, 1992). It is known that static-split policies can achieve the capacity region in the two staged network resource allocation problems (Gopalan et al., 2012).

In our setting, for each $(\lambda_{ex}, \theta_{th})$ in the Pareto region, it suffices to consider an appropriate probability distribution over the possible $(\mathfrak{H}_o, \mathfrak{H}_{in})$ (as function of $(\Psi, \lambda_{ex}, \theta_{th})$). In particular, this distribution should ensure two things. Firstly, to ensure stability the time average inflow and outflow for a non-destination node must be balanced. Thus, feasibility of flow balance equations implies stability of the system. Further, the distribution should satisfy a threshold accuracy condition. This feasibility can be described efficiently as a flow polytope. We next describe this flow polytope.

Let us denote the set of partial labels which have received labels from all classifiers as $\mathcal{L}_{cmp} \equiv \{\ell' : \forall i \in [M], \ell'(i) \neq \mathbf{e}\}$. The following flow polytope encodes the sustainability of the flow conditioned on meeting the threshold accuracy.

Remarks on Flow Polytope FP-II($\Psi, \lambda_{ex}, \theta_{th}$):

- In the flow balance equations, $x(\mathfrak{H}_o)$ denotes the fraction of time the stage 1 decision $\mathfrak{H}_o$ is chosen. Further, given $\mathfrak{H}_o$ is chosen and cl are the new labels, $y(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)})$ denotes the fraction of time the stage 2 decision $\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}$ is chosen.

- The condition (i) denotes the sample matched with classifier i departs the system. Further, (ii) and (iii) denotes that, due to random labeling, the departing sample has the partial label ℓ which is a) not complete and b) the given final label has accuracy less than threshold. Thus, (i)-(iii) together implies the threshold accuracy is violated for the choice of $\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}$, given the new labels cl and stage 1 decision $\mathfrak{H}_o$.

We now present our the characterization of the Pareto region of compressed causal policies.

Theorem A.5. *Given a OnDS model Ψ, a pair $(\lambda_{ex}, \theta_{th}) \in {}_c\Lambda(\Psi)$ if and only if flow polytope, FP-II($\Psi, \lambda_{ex}, \theta_{th}$) is feasible.*

Flow Polytope: FP-II($\Psi, \lambda_{ex}, \theta_{th}$)

Time-sharing Constraints

- $\forall \mathfrak{H}_o, x(\mathfrak{H}_o) \geq 0$, • $\sum_{\mathfrak{H}_o} x(\mathfrak{H}_o) = 1$.
- $\forall \mathfrak{H}_o, cl, \forall \mathfrak{H}_{in}^{\mathfrak{H}_o, cl}, y(\mathfrak{H}_{in}^{(\mathfrak{H}_o, cl)}) \geq 0$,
- $\forall \mathfrak{H}_o, cl, \sum_{\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}} y(\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}) = 1$.

Flow Balance Equations:

- $\lambda_{in}(\{\mathbf{e}\}^M) = \lambda_{ex}$, • $\forall \ell \in [K]_e^M, \lambda_{out}(\ell) \geq \lambda_{in}(\ell)$
- $\forall \ell \notin \mathcal{L}_{cmp}, \lambda_{out}(\ell) = \sum_{\mathfrak{H}_o} x(\mathfrak{H}_o) \sum_{i\in[M]} \mathbb{1}(H_o(i) = \ell)$
- $\forall \ell \in [K]_e^M, \lambda_{in}(\ell) = \sum_{cl,\mathfrak{H}_o} x(\mathfrak{H}_o) P_\Psi(cl, \mathfrak{H}_o) \times \dots$

$$\dots \times \sum_{\mathfrak{H}_{in}^{\mathfrak{H}_o,cl}} y(\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}) \sum_{i\in[M]} arr(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)})(i)$$

$$\left[arr(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)})(i) \equiv \mathbb{1}\left(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}(i) = \mathbf{e} \wedge \mathcal{L}^{(cl,\mathfrak{H}_o)}(i) = \ell\right)\right]$$

Threshold Accuracy Condition:

- $\forall \mathfrak{H}_o, cl, \forall \mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}, y(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}) = 0$,

  If $\exists i \in [M], \ell \in ([K]_e^M \setminus \mathcal{L}_{cmp}), k \in [K]$,

  such that (i)-(iii) hold

  $(i)\ \mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}(i) = k$, (Departure)

  $(iii)\ \mathcal{L}^{(cl,\mathfrak{H}_o)}(i) = \ell$, (Partial label ℓ at departure)

  $(ii)\ acc(\ell, k) < \theta_{th}$.(Accuracy below threshold)

A.4. Proof of Theorem 3.6

In this subsection we prove the main theorem in Sec 3. In the next lemma, we show that the FP-II($\Psi, \lambda_{ex}, \theta_{th}$) can be reduced to FP($\Psi, \lambda_{ex}, \theta_{th}$) which is used in the statement of Theorem 3.6. Lemma A.6 combined with Theorem A.5 and Lemma A.4 completes the proof of Theorem 3.6.

Lemma A.6. *For a given OnUEL instance $(\Psi, \lambda_{ex}, \theta_{th})$, FP-II($\Psi, \lambda_{ex}, \theta_{th}$) is feasible if and only if FP($\Psi, \lambda_{ex}, \theta_{th}$) (in Sec.3) is feasible.*

Proof. We first recall, $\mathcal{T}(\theta_{th}) = \{\ell : \ell \in [K]_e^M, acc^*(\ell) \geq \theta_{th}\} \cup \mathcal{L}_{cmp}$ and $\mathcal{T}^c(\theta_{th}) = [K]_e^M \setminus \mathcal{T}(\theta_{th})$.

FP feasible $\Rightarrow$ FP-II is feasible: Let the FP($\Psi, \lambda_{ex}, \theta_{th}$) be feasible. Then there exists at least one vector $\mathbf{x} = \{x_h : h \in \mathcal{T}^c(\theta_{th})\}$ such that all the conditions for FP($\Psi, \lambda_{ex}, \theta_{th}$) are met. In FP-II, we fix $\mathbf{x}$ as the probability distribution of the stage 1 decisions. Next, we fix the stage 2 probability $\mathbf{y}$. For each updated sample, the final label $k^*(\ell_i)$ is given if the accuracy is above the threshold,

otherwise the sample re-enters the system. Therefore, by construction of $\mathbf{y}$ the Threshold accuracy condition is satisfied. Formally, for each $\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}$,
1) $y(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}) = 1$ if (1a) and (1b) hold:
(using notation $\ell_i \equiv \mathcal{L}^{(cl,\mathfrak{H}_o)}(i)$ below)
(1a) $\forall i{\in}[M]$ if $acc^*(\ell_i){\geq}\theta_{th}$, $\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}(i){=}k^*(\ell_i)$;
(1b) for all $i{\in}[M]$ if $acc^*(\ell_i){<}\theta_{th}$, $\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}(i){=}\mathbf{e}$.

2) Otherwise, $y(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}) = 0$.

For the above, choice we have in FP-II: $\forall \ell \in \mathcal{T}(\theta_{th})$,

$$\lambda_{in}(\ell) = \sum_{cl,\mathfrak{H}_o} x(\mathfrak{H}_o) P_\Psi(cl,\mathfrak{H}_o) \sum_{i\in[M]} \mathbb{1}(\mathcal{L}^{(cl,\mathfrak{H}_o)}(i) = \ell)$$
$$= \sum_{\mathfrak{H}_o} x(\mathfrak{H}_o) \sum_{i\in[M]} \left(\sum_{cl} P_\Psi(cl,\mathfrak{H}_o) \mathbb{1}(\mathcal{L}^{(cl,\mathfrak{H}_o)}(i) = \ell) \right)$$
$$= \sum_{\mathfrak{H}_o} x(\mathfrak{H}_o) \sum_{i\in[M]} P_\Psi(\ell | i, \mathfrak{H}_o(i)).$$

Therefore, as we know that FP($\Psi, \lambda_{ex}, \theta_{th}$) is feasible for the choice $\mathbf{x}$, FP-II($\Psi, \lambda_{ex}, \theta_{th}$) flow balance equations are also feasible. Therefore, any feasible point in FP($\Psi, \lambda_{ex}, \theta_{th}$) is a feasible point in FP-II($\Psi, \lambda_{ex}, \theta_{th}$).

FP-II feasible $\Rightarrow$ FP is feasible: Let the FP-II($\Psi, \lambda_{ex}, \theta_{th}$) be feasible. Further, let $(\mathbf{x}, \mathbf{y}')$ is a feasible point in FP-II. We now consider the previous construction of $\mathbf{y}$. We claim that $(\mathbf{x}, \mathbf{y})$ is another feasible point in FP-II($\Psi, \lambda_{ex}, \theta_{th}$). Let $\boldsymbol{\lambda}'_{in}$ and $\boldsymbol{\lambda}_{in}$ denote the outflow vectors under $\mathbf{y}'$ and $\mathbf{y}$, respectively. We want to show that for all ℓ, $\lambda_{in}(\ell) \leq \lambda'_{in}(\ell)$. This will imply that under $(\mathbf{x}, \mathbf{y})$ the flow balance equations are satisfied. The Threshold condition is already satisfied by $(\mathbf{x}, \mathbf{y})$ due to our construction. Moreover, the feasibility of $(\mathbf{x}, \mathbf{y})$ will, in turn, imply (due to the first part) that the flow polytope FP($\Psi, \lambda_{ex}, \theta_{th}$) is also feasible for $\mathbf{x}$.

Let us fix an arbitary node ℓ. If ℓ is a terminal node, i.e. $\ell \in \mathcal{T}(\theta_{th})$, we have $\lambda_{in}(\ell) = 0 \leq \lambda'_{in}(\ell)$. If ℓ is a non-terminal node, i.e. $\ell \in \mathcal{T}^c(\theta_{th})$ then due to threshold accuracy condition all the flow re-enters the system. Specifically, we have the following where equality (i) is due to threshold accuracy condition.

$$\lambda'_{in}(\ell) = \sum_{cl,\mathfrak{H}_o} x(\mathfrak{H}_o) P_\Psi(cl,\mathfrak{H}_o) \sum_{\mathfrak{H}_{in}^{\mathfrak{H}_o,cl}} y'(\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}) \times \dots$$
$$\dots \times \sum_{i\in[M]} \mathbb{1}\left(\mathfrak{H}_{in}^{(cl,\mathfrak{H}_o)}(i){=}\mathbf{e} \wedge \mathcal{L}^{(cl,\mathfrak{H}_o)}(i){=}\ell\right)$$
$$\overset{(i)}{=} \sum_{cl,\mathfrak{H}_o} x(\mathfrak{H}_o) P_\Psi(cl,\mathfrak{H}_o) \sum_{\mathfrak{H}_{in}^{\mathfrak{H}_o,cl}} y'(\mathfrak{H}_{in}^{(\mathfrak{H}_o,cl)}) \times \dots$$
$$\dots \times \sum_{i\in[M]} \mathbb{1}\left(\mathcal{L}^{(cl,\mathfrak{H}_o)}(i){=}\ell\right)$$
$$= \sum_{cl,\mathfrak{H}_o} x(\mathfrak{H}_o) P_\Psi(cl,\mathfrak{H}_o) \sum_{i\in[M]} \mathbb{1}\left(\mathcal{L}^{(cl,\mathfrak{H}_o)}(i){=}\ell\right)$$
$$= \lambda_{in}(\ell)$$

This completes the proof. □

A.5. Proof of Technical Lemmas

Notations: We setup some notations to reduce notational complexity. Recall, that $h_i(\tau)$ and $h_i^Q(\tau)$ represents a realization of the history $H_i(\tau)$ and ${}_cH_i(\tau)$, resp., for all $\tau \geq 1$, $i = \{0,1,2,3\}$. Also,we denote $H_0(\tau+1) = H_4(\tau)$ for all $\tau \geq 1$. We abbreviate$\mathbb{P}(H_i(\cdot) = h_i(\cdot)|\cdot)$ as $\mathbb{P}(h_i(\cdot)|\cdot)$ for notational simplification.The term s $\mathbb{P}({}_ch_i(\cdot)|\cdot)$ are used similarly. When the round τ is clear from context, we may drop the time index τ, i.e. we may use h_i and ${}_ch_i$ instead of $h_i(\tau)$ and ${}_ch_i(\tau)$, respectively. We also use the arrow notation '$a \overset{e}{\to} b$' to denote that the event e changes system state a to system state b.

Proof of Lemma A.1. Consider the system (OnUEL instance) $\mathcal{P} = (\mathbf{p}_g, \{C_i :\in [M]\}, \lambda_{ex}, \theta_{th})$. Also consider a fixed causal policy ϕ. From ϕ we now construct a compressed causal policy ${}_c\phi$, that controls the network of queues through the decisions ${}_cA(\tau)$ and ${}_cD(\tau)$, for $\tau \geq 1$. We consider the first equality next.

$$\mathbb{P}^{{}_c\phi}(\mathbf{Q}(\tau)|H_0^1) = \sum_{{}_ch_0^\tau} \mathbb{P}^{{}_c\phi}(\mathbf{Q}(\tau)|{}_ch_0^\tau)\mathbb{P}^{{}_c\phi}({}_ch_0^\tau|H_0^1)$$
$$\overset{(i)}{=} \sum_{{}_ch_0^\tau} \mathbb{P}^{{}_c\phi}(\mathbf{Q}(\tau)|{}_ch_0^\tau)\mathbb{P}^{\phi}\left(H_0^\tau \in g_0^{-1}({}_ch_0^\tau)|H_0^1\right)$$
$$\overset{(ii)}{=} \sum_{{}_ch_0^\tau} \mathbb{P}^{\phi}(\mathbf{Q}(\tau)|g_0^{-1}({}_ch_0^\tau))\mathbb{P}^{\phi}\left(H_0^\tau \in g_0^{-1}({}_ch_0^\tau)|H_0^1\right)$$
$$= \mathbb{P}^{\phi}\left(\mathbf{Q}(\tau)|H_0^1\right)$$

Here, equality (i) is true from Eq. 4 in Corollary A.3 and equality (ii) is true because the queue length $\mathbf{Q}(\tau)$ is a function of the compressed history $g({}_cH_0^\tau)$.

Let for all $\tau \geq 1$, ${}_cAcc_{\text{min}}^\tau = \{\inf_{j\in {}_cD_p(\tau)} Acc_j(\tau){<}\theta_{th}\}$ and $Acc_{\text{min}}^\tau = \{\inf_{j\in D_p(\tau)} Acc_j(\tau){<}\theta_{th}\}$.

$$\mathbb{P}^{{}_c\phi}({}_cAcc_{\text{min}}^\tau|H_0^1) = \sum_{{}_ch_3^\tau} \mathbb{P}^{{}_c\phi}({}_cAcc_{\text{min}}^\tau|{}_ch_3^\tau)\mathbb{P}^{{}_c\phi}({}_ch_3^\tau|H_0^1)$$
$$\overset{(i)}{=} \sum_{{}_ch_3^\tau} \mathbb{P}^{{}_c\phi}({}_cAcc_{\text{min}}^\tau|{}_ch_3^\tau)\mathbb{P}^{\phi}\left(H_0^\tau \in g_3^{-1}({}_ch_3^\tau)|H_0^1\right)$$
$$\overset{(ii)}{=} \sum_{{}_ch_3^\tau} \mathbb{P}^{\phi}(Acc_{\text{min}}^\tau|g_3^{-1}({}_ch_3^\tau))\mathbb{P}^{\phi}\left(H_0^\tau \in g_3^{-1}({}_ch_3^\tau)|H_0^1\right)$$
$$= \mathbb{P}^{\phi}\left(Acc_{\text{min}}^\tau|H_0^1\right)$$

Equality (i) is true from Eq. 4 in Corollary A.3. We observe that, due to Lemma 3.4,

i) for ${}_c\phi$, $\inf_{j\in {}_cD_p^\tau} Acc_j(\tau) = \inf_{\ell\in {}_cD_p^\tau} acc(\ell, k_\ell^*)$, and
ii) for ϕ, $\inf_{j\in D_p^\tau} Acc_j(\tau) = \inf_{j\in D_p^\tau} acc(\hat{\mathbf{s}}_j^{(\tau+1)}, k_j^*)$.

It easily follows that $\inf_{\ell\in {}_cD_p^\tau} acc(\ell, k_\ell^*)$ is a function of ${}_cH_3^\tau$. Further, as $acc(\hat{\mathbf{s}}_j^{(\tau+1)}, k_j^*)$ does not depend on the sample id j given its labels, $\inf_{j\in D^{(p)}(\tau)} acc(\hat{\mathbf{s}}_j^{(\tau+1)}, k_j^*)$ is a function of $g_3({}_cH_3^\tau)$ under policy ϕ .

This completes the proof. □

Proof of Lemma A.2**.** **Outline:** The proof is divided into three main parts. In the first part we prove the lemma for stage $st = 3$, where the events under consideration are the arrivals in time slot τ. In the second part we prove the lemma for stages $st = 0, 2$, where we consider the scheduling and the departure events under the two policies. For this part we crucially use the properties of ${}_c\phi$. Finally, in the third part we prove the stage $st = 1$. In this case the classifiers provide new labels to the samples sent to them. The Lemma 3.4 plays the key role in showing the equivalence between causal and compressed-causal policies.

Full Proof: We fix an arbitrary timeslot $\tau \geq 1$ and proceed to the outlined case by case analysis for system $\mathcal{P}$, and policies ϕ, ${}_c\phi$. Recall, for all $\tau \geq 1$, that $h_4^\tau = h_0^{(\tau+1)}$ and the corresponding compression operator is $g_0(\cdot)$. Also, as we consider round τ we may drop the superscript τ.

Stage 3 (Arrival): Let ${}_ch_3$ and ${}_ch_4$ be compression of consequent histories h_3 and h_4. Under our arrival model, there exists *a unique* arrival event (set of samples) $arr(h_3, h_4)$ such that $h_3 \overset{arr(h_3,h_4)}{\rightarrow} h_4$. Further, for all $h_3 \in g_3^{-1}({}_ch_3)$ and $h_4 \in g_0^{-1}({}_ch_4)$, we have $|arr(h_3, h_4)| = {}_carr({}_ch_3, {}_ch_4)$, which is the number of arrivals that uniquely determines the transition ${}_ch_3 \rightarrow {}_ch_4$.

$$
\begin{aligned}
&\mathbb{P}^\phi\left(g_0^{-1}({}_ch_4)|g_3^{-1}({}_ch_3)\right)\\
&= \sum_{h_3\in g_3^{-1}({}_ch_3)} \mathbb{P}^\phi\left(g_0^{-1}({}_ch_4)|h_3\right)\mathbb{P}^\phi\left(h_3|g_3^{-1}({}_ch_3)\right)\\
&= \sum_{h_3\in g_3^{-1}({}_ch_3)} \mathbb{P}^\phi\left(\{arr(h_3,h_4) : h_4 \in g_0^{-1}({}_ch_4)\}|h_3\right)\times\\
&\qquad\qquad \times\mathbb{P}^\phi\left(h_3|g_3^{-1}({}_ch_3)\right)\\
&= \mathbb{P}\left({}_carr({}_ch_3, {}_ch_4)\right)\sum_{h_3\in g_3^{-1}({}_ch_3)} \mathbb{P}^\phi\left(h_3|g_0^{-1}({}_ch_3)\right)\\
&= \mathbb{P}^{{}_c\phi}\left({}_ch_4|{}_ch_3\right).
\end{aligned}
$$

Stage 1 (Matching) and 3 (Departure+Labeling): The cases where $i = 1$ correspond to decisions of matching and $i = 3$ to decisions of departures and final labeling. The proofs are identical for the two stages and uses the definition of ${}_c\phi = CC(\phi)$, crucially. We prove the statement for $st = 1$. We skip the proof of the case $st = 3$, which is similar.

We first fix an arbitrary pair of realizations ${}_ch_0$ and ${}_ch_1$. We denote by $a(h_0, h_1)$ the unique scheduling decision that takes h_0 to h_1 under ϕ. Similarly, ${}_ca({}_ch_1, {}_ch_2)$ denotes the unique scheduling decision that takes ${}_ch_1$ to ${}_ch_2$ under ${}_c\phi$. From the connection of the ${}_cA(\tau)$ and $A(\tau)$ we have

$$
f_A^{-1}({}_ca({}_ch_1, {}_ch_2)) = \{a(h_1, h_2) : h_i \in g_{st}^{-1}({}_ch_i), st = 0, 1\}.
$$

We now use the above to show the equality in the lemma statement for $st = 1$.

$$
\begin{aligned}
&\mathbb{P}^{{}_c\phi}\left({}_ch_1|{}_ch_0\right) = \mathbb{P}^{{}_c\phi}\left({}_ca({}_ch_0, {}_ch_1)|{}_ch_0\right)\\
&= \mathbb{P}^\phi\left(f_A^{-1}({}_ca({}_ch_0, {}_ch_1))|g_0^{-1}({}_ch_0)\right)\text{ [property of } {}_c\phi]\\
&= \sum_{h_0\in g_0^{-1}({}_ch_0)} \mathbb{P}^\phi\left(f_A^{-1}({}_ca({}_ch_0, {}_ch_1))|h_0\right)\mathbb{P}^\phi\left(h_0|g_0^{-1}({}_ch_0)\right)\\
&= \sum_{\substack{h_0\in g_0^{-1}({}_ch_0)\\ h_1\in g_1^{-1}({}_ch_1)}} \mathbb{P}^\phi\left(A(h_0, h_1)|h_0\right)\mathbb{P}^\phi\left(h_0|g_0^{-1}({}_ch_0)\right)\\
&= \sum_{\substack{h_0\in g_0^{-1}({}_ch_0)\\ h_1\in g_1^{-1}({}_ch_1)}} \mathbb{P}^\phi\left(h_1|h_0\right)\mathbb{P}^\phi\left(h_0|g_0^{-1}({}_ch_0)\right)\\
&= \mathbb{P}^\phi\left(g_1^{-1}({}_ch_1)|g_0^{-1}({}_ch_0)\right)
\end{aligned}
$$

Stage 2 (Classifier Labels): This scenario the systems makes transition due to classification events.

We consider the general system first. For the history h_1, let $a(h_1)$ be the matching between samples and the classifiers at time τ. Here, $a(h_1)(i)$denotes the sample received by classifier i, if $a(h_1)(i) \neq \mathbf{e}$. Otherwise, the classifier i is idle. Let us consider the event that under $a(h_1)$ the true label of the sample allotted to classifier i is t_i. We denote this event as $E_{cl}(a(h_1), \mathbf{t}) = \{s_j[M] = t_i : j = a(h_1)(i), i \in [M]\}$. In this classification stage a *unique set* $Cl(h_1, h_2) \in [K]_e^M$ (i.e. the new labels in round τ) maps the history h_1 to history h_2. Given $E_{cl}(a(h_1), \mathbf{t})$ and the matching $a(h_1)$, the labeling events $Cl(h_1, h_2)$ are independent of the policy ϕ in the general system (due to property of OnDS model).

The dynamics of the compressed causal policy is described next. We denote by ${}_ca({}_ch_1)$ the matching between queues and classifiers (at time τ) under policy ${}_c\phi$ for history ${}_ch_1$. Here, ${}_ca({}_ch_1)$ corresponds to the set of matching events $\cup_{h_1\in g_1^{-1}({}_ch_1)} a(h_1)$ in under the (original) policy ϕ. [5]

For any fixed pair ${}_ch_1$, ${}_ch_2$, and for all $h_1 \in g_1^{-1}({}_ch_1)$, $h_2 \in g_2^{-1}({}_ch_2)$, the following statements are true.

[5] Also, ${}_ca({}_ch_1) = \cup_{{}_ch_0}{}_ca({}_ch_0, {}_ch_1)$ and $f_A^{-1}({}_ca({}_ch_1)) = \cup_{{}_ch_0} f_A^{-1}({}_ca({}_ch_0, {}_ch_1)) = \cup_{h_1\in g_1^{-1}({}_ch_1)} a(h_1)$.

(2a) The classifications events are identical, i.e. $Cl(h_1, h_2) := Cl({}_ch_1, {}_ch_2)$.

(2b) The probabilities $\mathbb{P}(Cl(h_1, h_2)|E_{cl}(a(h_1), \mathbf{t}))$ admit the same value for any fixed $\mathbf{t}$.

(2c) The probabilities $\mathbb{P}(E_{cl}(a(h_1), \mathbf{t})|h_1)$ admit the same value for any fixed $\mathbf{t}$.

From (2a),(2b), and (2c) it follows that for any fixed pair ${}_ch_1$, ${}_ch_2$, and for all $h_1 \in g_1^{-1}({}_ch_1)$, $\mathbb{P}^\phi\left(Cl({}_ch_1, {}_ch_2)|h_1\right) = \mathbb{P}^{c\phi}\left(Cl({}_ch_1, {}_ch_2)|{}_ch_1\right)$. This equality essentially says that given an assignment between classifiers and partial labels, the labels given by the classifiers are identically distributed.

We now proceed with the rest of the proof. We fix a pair of compressed histories ${}_ch_1$ and ${}_ch2$.

$$
\begin{aligned}
&\mathbb{P}^\phi\left(g_2^{-1}({}_ch_2)|g_1^{-1}({}_ch_1)\right)\\
&= \sum_{h_1 \in g_1^{-1}({}_ch_1)} \mathbb{P}^\phi\left(Cl({}_ch_1, {}_ch_2)|h_1\right)\mathbb{P}^\phi\left(h_1|g_1^{-1}({}_ch_1)\right)\\
&= \sum_{h_1 \in g_1^{-1}({}_ch_1)} \mathbb{P}^{c\phi}\left(Cl({}_ch_1, {}_ch_2)|{}_ch_1\right)\mathbb{P}^\phi\left(h_1|g_1^{-1}({}_ch_1)\right)\\
&= \mathbb{P}^{c\phi}\left(Cl({}_ch_1, {}_ch_2)|{}_ch_1\right) \sum_{h_1 \in g_1^{-1}({}_ch_1)} \mathbb{P}^\phi\left(h_1|g_1^{-1}({}_ch_1)\right)\\
&= \mathbb{P}^{c\phi}\left({}_ch_2|{}_ch_1\right)
\end{aligned}
$$

Validity of (2a),(2b), and (2C): Due to the compression property we have (2a). To see the validity of (2b), we first observe that due to the compression property, for all such $a(h_1)$, the partial label for the sample matched with classifier i is the same. Formally, $\forall h_1 \in g_1^{-1}({}_ch_1), i \in [M], \hat{\mathbf{s}}_{a(h_1)(i)} = {}_ca({}_ch_1)(i)$. Next, due to Lemma 3.4, the true label distribution of any set of samples (through marginalization over the entire set) is independent of any history given the partial labels. Therefore, $\mathbb{P}(Cl(h_1, h_2)|E_{cl}(a(h_1), \mathbf{t}))$ is a function of the partial labels $\{\hat{\mathbf{s}}_{a(h_1)(i)} : i \in [M]\}$ and $\mathbf{t}$. This implies, due to the first observation, $\mathbb{P}(Cl(h_1, h_2)|E_{cl}(a(h_1), \mathbf{t}))$ is identical for all $\forall h_1 \in g_1^{-1}({}_ch_1)$ and all $\mathbf{t}$. Therefore, (2b) is valid. Similarly, due to Lemma 3.4, we have (2c). This completes the proof of the lemma. □

B. Proof of Results in Section 4

We use Lyapunov optimization techniques to provide the delay guarantees for our algorithm. Specifically, we show that the term $L(\tau) = \sum_{\ell \in [K]_e^M} Q_\ell^2(\tau)$ has an expected negative drift. The Lemma B.1 (below) shows that in each step the drift is negative for large enough $Q_\ell^2(\tau)$ and round τ.

Lemma B.1. *Given an OnUEL instance $\mathcal{P}=(\Psi, \lambda_{ex}, \theta_{th})$ such that $(\lambda_{ex}, \theta_{th})(1+\epsilon_g) \in \Lambda(\Psi)$ for some $\epsilon_g > 0$, and an (α, β) oracle with $\alpha, \beta > 0$; under Algorithm 1 with parameters $\epsilon_s(\tau) = \frac{\log(\tau)}{\tau}$ and $\epsilon_\theta(\tau) = \frac{2}{\log(\tau)^\alpha}$;*

$$
\begin{aligned}
&\mathbb{E}\left[L(\tau+1) - L(\tau)|\mathbf{Q}(\tau)\right]\\
&\leq -\tfrac{\epsilon_g\lambda_{min}}{(1+\epsilon_g)} \sum_{\ell \in \mathcal{T}^c(\theta_{th}(1+\epsilon_g))} Q_\ell(\tau) + MC_1\mathbb{1}(\tau < C_1)\\
&+ C_0 + M^2 + O\left(\tfrac{1}{\log(\tau)^{\min(\alpha,\beta)}}\right)Q_\ell(\tau),
\end{aligned}
\tag{5}
$$

where $C_0 := M^2 + \lambda_{max}^2$, $C_1 := O(exp(\epsilon_g^{-1/\alpha}))$, and $\lambda_{min} := \left(\min_{\ell \in \mathcal{T}^c(\theta_{th}(1+\epsilon_g))} \sum_k P_\Psi(\ell, k)\right)\lambda_{ex}$.

The proof has a two key steps. Firstly, we need to make use of Lemma 3.4, to show that the max-weight scheduling maximizes the expected drift conditioned on the queue length. Secondly, we need to show that with $\log(\tau)/\tau$ exploration rate and $\epsilon_\theta(\tau)$ cushion for accuracy the effect of oracle error can be nullified. In this step, we note that the underlying network changes as the terminal sets changes as a function of the oracle output. We first present a brief review of the necessary parts for the proof of this Lemma. We then prove the lemma. Finally, we provide a sketch of how this lemma leads to the Theorem 4.1.

B.1. Preliminaries

Recalling Necessary parts in Sec 3: Let us recall some necessary notations from Sec 3. The terminal set is given as $\mathcal{T}(\theta_{th})$ refers to the terminal labels nodes. For each of these nodes, either (i) all classifiers are used or (ii) its accuracy $acc(\ell, k) \geq \theta_{th}$.

We are given a OnUEL instance $\mathcal{P} = (\Psi, \lambda_{ex}, \theta_{th})$. Additionally, for some $\epsilon_g > 0$ we know that $(\lambda_{ex}, \theta_{th})(1+\epsilon_g) \in \Lambda(\Psi)$. Therefore, in view of Theorem 3.6, there exists a probability distribution over the hyper-edge, namely $\mathbf{x}^{SS}$, such that FP$(\Psi, \lambda_{ex}(1+\epsilon_g), \theta_{th}(1+\epsilon_g))$ is feasible. Therefore, the following equations are true.

- $\lambda_{in}(\{\mathbf{e}\}^M) = \lambda_{ex}(1+\epsilon_g)$
- $\forall \ell, \lambda_{out}(\ell) \geq \lambda_{in}(\ell)$
- $\forall \ell \neq \{\mathbf{e}\}^M, \lambda_{in}(\ell) = \sum_h x_h \sum_{i \in [M]} P_\Psi(\ell|i, h(i))$
- $\forall \ell \notin \mathcal{T}(\theta_{th}(1+\epsilon_g)), \lambda_{out}(\ell) = \sum_h x_h \sum_{i \in [M]} \mathbb{1}(h(i) = \ell)$

Recalling Dynamics of Algorithm 1: Let us recall that the hyper-edges h in the graph define matchings in our system. The max-weight algorithm in any round $\tau \geq 1$, selects the matching/hyper-edge that maximizes the function

$$
\sum_{\ell \in \widehat{\mathcal{T}}^c(\tau-1)} \sum_{i \in [M]} Q_\ell(\tau)\left(\mathbb{1}(h(i) = \ell) - \widehat{P}_\Psi(\ell|i, h(i))\right). \tag{opt1}
$$

Let us call the selected matching as h^τ in round τ, for all $\tau \geq 1$. Further, this maximization involves the estimation error. Let in round τ, for all $\tau \geq 1$, $h^{*\tau}$ be the optimal matching without any error, i.e. $h^{*\tau}$ maximizes (note $\widehat{P}_\Psi$ is replaced with P_Ψ)

$$\sum_{\ell \in \widehat{\mathcal{T}}^c(\tau-1)} \sum_{i \in [M]} Q_\ell(\tau)\left(\mathbb{1}(h(i)=\ell) - P_\Psi(\ell|i,h(i))\right). \quad \text{(opt2)}$$

Further, after the partial label $h^\tau(i)$ is matched with i, under Algorithm 1, we denote the event that the new partial label is ℓ', as $\mathbb{1}(\ell'|i,h^\tau(i))$. Therefore, the decrease in $Q_\ell(\tau)$ due to matching, $R_\ell(\tau) = \sum_{i\in[M]} \mathbb{1}(h^\tau(i)=\ell)$ and the increase in $Q_\ell(\tau)$ due to subsequent label update is $N_\ell(\tau) = \sum_{i\in[M]} \mathbb{1}(\ell'|i,h^\tau(i))$.

Finally, we recall that the oracle in consideration is an (α,β) oracle which has been defined as follows (repeated here for easy reference).

$(\boldsymbol{\epsilon(\sigma)}, \boldsymbol{\delta(\sigma)})$ Oracle: An oracle is an'(α, β) oracle' if and only if, for all (i) partial labels $\ell, \ell' \in [K]$, (ii) classifiers $i \in [M]$, and (iii) classes $k \in [K]$, the oracle with access to σ *exploration samples* satisfies,
(1) $|acc(\ell,k) - \widehat{acc}(\ell,k)| \leq \sigma^{-\alpha}$ w.p. $\geq (1-\sigma^{-\beta})$,
(2) $|P_\Psi(\ell'|i,\ell) - \widehat{P}_\Psi(\ell'|i,\ell)| \leq \sigma^{-\alpha}$ w.p. $\geq (1-\sigma^{-\beta})$.

Dynamic Terminal Sets: The terminal nodes are fixed for a given $theta_{th}$. However, when estimation error is there, the set of terminal nodes change across rounds. We need to define the terminal sets at different rounds. The complements in the following definition are w.r.t. the set $[K]_e^M$.

For round $\tau \geq 1$, let us define
(i) the terminal set estimated by oracle inputs $\widehat{acc}^\tau(\ell,k)$ as
$\widehat{\mathcal{T}}(\tau) = \mathcal{L}_{cmp} \cup \{\ell : \widehat{acc}^{*\tau}(\ell) \geq \theta_{th} + \epsilon_\theta(\tau)\}$;
(ii) the intersection between the estimated non-terminal sets in round $(\tau-1)$ and τ as $\widehat{\mathcal{T}}^c_\cap(\tau) := \widehat{\mathcal{T}}^c(\tau) \cap \widehat{\mathcal{T}}^c(\tau-1)$;
(iii) the parts which are not in the intersection $\widehat{\mathcal{T}}^c_\cap(\tau)$ are
$\widehat{\mathcal{T}}^c_\setminus(\tau') := \widehat{\mathcal{T}}^c(\tau') \setminus \widehat{\mathcal{T}}^c_\cap(\tau)$, for $\tau' = \tau, \tau-1$;
(iv) the labels in the estimated terminal set that are not in the original terminal set are denoted as
$\widehat{\mathcal{T}}^c_+(\tau) := \widehat{\mathcal{T}}^c(\tau) \setminus \mathcal{T}^c(\theta_{th}(1+\epsilon_g))$;
(v) the labels in the original terminal set that are not in the estimated terminal set are denoted as
$\widehat{\mathcal{T}}^c_-(\tau) := \mathcal{T}^c(\theta_{th}(1+\epsilon_g)) \setminus \widehat{\mathcal{T}}^c(\tau)$.

B.2. Proof of Lemma B.1

One of key aspect of the designed algorithm is that there are two separate stages of decisions involved in the dynamics of the queues. We need to account for the effect of such dynamic changes in terminal sets in our analysis. The idea is to separate out the effect of estimation error and exploration events. Let us denote the estimation in round τ as $\widehat{acc}^\tau(\cdot)$ and $\widehat{P}^\tau_\Psi(\cdot)$

$$L(\tau+1) - L(\tau) = \tfrac{1}{2} \sum_{\ell \in [K]_e^M} \left(Q_\ell^2(\tau+1) - Q_\ell^2(\tau)\right)$$

$$= \tfrac{1}{2}\left(\sum_{\ell\in\widehat{\mathcal{T}}^c(\tau)} Q_\ell^2(\tau+1) - \sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} Q_\ell^2(\tau)\right)$$
$$= \tfrac{1}{2}\sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} \left(Q_\ell^2(\tau+1) - Q_\ell^2(\tau)\right) + \tfrac{1}{2}\sum_{\ell\in\widehat{\mathcal{T}}^c_\setminus(\tau)} Q_\ell^2(\tau+1)$$
$$\overset{(*1)}{=} \tfrac{1}{2}\sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} \left(Q_\ell^2(\tau+1) - Q_\ell^2(\tau)\right) + M^2.$$

Validity of (*1): The term $\frac{1}{2}\sum_{\ell\in\widehat{\mathcal{T}}^c_\setminus(\tau)} Q_\ell^2(\tau+1)$ admits the upper bound M^2. This is true because for all $\ell \in \widehat{\mathcal{T}}^c_\setminus(\tau)$, $Q_\ell(\tau)=0$. Therefore, $Q_\ell(\tau+1)$ for $\ell \in \widehat{\mathcal{T}}^c_\setminus(\tau)$ can only increase through arrivals in round τ. Further, there are at most M arrivals in each round, which leads to the upper bound. Finally, for all $\ell \in \widehat{\mathcal{T}}^c_\setminus(\tau)$, $Q_\ell(\tau) = 0$ as at the end of round $(\tau-1)$ the terminal set is $\widehat{\mathcal{T}}(\tau-1)$. Recall, all samples in the terminal set departs under Algorithm 1.

We now proceed with bounding the first term in the final equality $(*1)$ above as,

$$\tfrac{1}{2}\sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} \left(Q_\ell^2(\tau+1) - Q_\ell^2(\tau)\right)$$
$$\leq \sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} Q_\ell(\tau)(R_\ell(\tau) - N_\ell(\tau)) + \tfrac{1}{2}\sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} (N_\ell^2 + R_\ell^2)$$
$$\leq \sum_{\ell\in\widehat{\mathcal{T}}^c(\tau-1)} Q_\ell(\tau)(R_\ell(\tau) - N_\ell(\tau)) + C_0$$
$$\overset{(*2)}{=} \sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(\mathbb{1}(\ell|i,h^\tau(i)) - \mathbb{1}(h^\tau(i)=\ell)\right)$$
$$+ C_0 + Q_{\{\mathbf{e}\}^M}(\tau) N_{ex}(\tau). \text{ [Here, } C_0 := (M^2 + 0.5\lambda_{max}^2)]$$

Next our objective is to bound the expectation of such event conditioned on the current queue length vector $\mathbf{Q}(\tau)$. For this purpose, we need to consider three separate events in round $\tau \geq 1$.

- The first event $\mathcal{E}_1^\tau$ is exploration which happens with probability $\epsilon_s(\tau) = \frac{\log(\tau)}{\tau}$, by construction of our algorithm.
- The second event $\mathcal{E}_2^\tau$ is that the error in estimating parameter $\widehat{acc}(\cdot)$ and $\widehat{P}_\Psi(\cdot)$ is greater than $\log(\tau)^{-\alpha}$. Using Chernoff bounds, the number of exploration samples up to time τ is $\Omega(\log(\tau))$ with probability at least $(1 - 1/\tau^{\log(\tau)})$. Therefore, using union bound and the property of an (α,β) oracle, the probability of $\mathcal{E}_2^\tau$ is upper bounded as $O(\log(\tau)^{-\beta})$.
- The third event $\mathcal{E}_3^\tau$ is the 'good event', where the Algo-

rithm 1 exploits and the oracle ensures that the estimation error is less than $\log(\tau)^{-\alpha}$.

The Event $\mathcal{E}\tau_3$: We consider the good event first and continue bounding the queue dependent term in equation $(*\mathbf{2})$, To avoid clutter let $\mathbb{E}_3^\tau[\cdot] = \mathbb{E}[\cdot|\mathbf{Q}(\tau), \mathcal{E}_3^\tau]$.

$$
\begin{aligned}
&\mathbb{E}_3^\tau\left[\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(\mathbb{1}(\ell|i,h^\tau(i)) - \mathbb{1}(h^\tau(i))\right)\right]\\
&\overset{(i)}{=} \mathbb{E}_3^\tau\left[\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(P_\Psi(\ell|i,h^\tau(i)) - \mathbb{1}(h^\tau(i))\right)\right]\\
&\leq \mathbb{E}_3^\tau\left[\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(\widehat{P}_\Psi(\ell|i,h^\tau(i)) - \mathbb{1}(h^\tau(i)=\ell)\right)\right]\\
&+ \sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\mathbb{E}_3^\tau\left[|\widehat{P}(\ell|i,h^\tau(i)) - \widehat{P}_\Psi(\ell|i,h^\tau(i))|\right]\\
&\overset{(ii)}{\leq} \mathbb{E}_3^\tau\left[\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(\widehat{P}_\Psi(\ell|i,h^\tau(i)) - \mathbb{1}(h^\tau(i)=\ell)\right)\right]\\
&+ \sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&\overset{(iii)}{\leq} \mathbb{E}_3^\tau\left[\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(\widehat{P}_\Psi(\ell|i,h^{*\tau}(i)) - \mathbb{1}(h^{*\tau}(i)=\ell)\right)\right]\\
&+ \sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&\overset{(iv)}{\leq} \mathbb{E}_3^\tau\left[\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(P_\Psi(\ell|i,h^{*\tau}(i)) - \mathbb{1}(h^{*\tau}(i)=\ell)\right)\right]\\
&+ 2\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&\overset{(v)}{\leq} \mathbb{E}_3^\tau\left[\sum_h \tilde{x}_h \sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\left(P_\Psi(\ell|i,h) - \mathbb{1}(h(i)=\ell)\right)\right]\\
&+ 2\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&\overset{(vi)}{\leq} \sum_h \tilde{x}_h \sum_{\substack{\ell\in\mathcal{T}^c(\theta_{th}(1+\epsilon_g))\\ i\in[M]}} Q_\ell(\tau)\left(P_\Psi(\ell|i,h) - \mathbb{1}(h(i)=\ell)\right)\\
&+ M\sum_{\ell\in\widehat{\mathcal{T}}_+^c(\tau)} Q_\ell(\tau) + 2\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&\overset{(vii)}{\leq} \sum_h \tilde{x}_h \sum_{\substack{\ell\in\mathcal{T}^c(\theta_{th}(1+\epsilon_g))\\ i\in[M]}} Q_\ell(\tau)\left(P_\Psi(\ell|i,h) - \mathbb{1}(h(i)=\ell)\right)\\
&+ MC_1\mathbb{1}(\tau < C_1) + 2\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&\overset{(*3)}{\leq} -\tfrac{\epsilon_g}{(1+\epsilon_g)} \sum_{\ell\in\mathcal{T}^c(\theta_{th}(1+\epsilon_g))\setminus\{\mathbf{e}\}^M} Q_\ell(\tau)\tilde{\lambda}_{in}(\ell)\\
&- (1+\epsilon_g)\lambda_{ex}Q_{\{\mathbf{e}\}^M}(\tau) + 2\sum_{\substack{\ell\in\widehat{\mathcal{T}}^c(\tau-1)\\ i\in[M]}} Q_\ell(\tau)\log(\tau)^{-\alpha}\\
&+ MC_1\mathbb{1}(\tau < C_1)
\end{aligned}
$$

Validity of the Relations (i)-(vii):

Case (i): To obtain the expectation we average over the classifier labeling events. Due to Lemma3.4 the equality follows.
Case (ii), and Case (iv): The validity is due to the bounded error under event $\mathcal{E}_3$.
Case (iii): The validity is due to the fact that h^τ optimizes the term in equation (opt1).
Case (v): This follows because $h^{*\tau}$ optimizes the term in equation (opt2).
Case(vi): Firstly, the inner term is a function of $\mathbf{Q}(\tau)$, therefore the expectation $\mathbb{E}_3^\tau[\cdot]$ can be removed. Secondly, $Q_\ell(\tau) = 0$ for all $\ell \notin \widehat{\mathcal{T}}(\tau-1)$. Therefore, we only need to consider the queue lengths for $\ell \in \widehat{\mathcal{T}}_+^c(\tau)$.
Case(vii): Furthermore, due to the bounded error property of the oracle, we know that there exists a $C_1 \equiv \tau_0 \leq O(exp(\epsilon_g^{-1/\alpha}))$, such that conditioned on $\mathcal{E}_3^\tau$ for all $\tau \geq \tau_0$ $\widehat{acc}^{*,\tau}(\ell,k) + \epsilon_\theta < \theta_{th} + \epsilon_g$. Thus for all $\tau \geq \tau_0$ we have under $\widehat{\mathcal{E}}_3^\tau$, $\ell \in \widehat{\mathcal{T}}(\tau-1) \setminus \mathcal{T}^c(\theta_{th}(1+\epsilon_g))$, which implies $\widehat{\mathcal{T}}_+^c(\tau) = \emptyset$.

Validity of the Relation (*3): The validity of (*3) is standard in network optimization literature, at least dating back to (Tassiulas & Ephremides, 1992). As a proof sketch, we may view that $\epsilon_g\lambda_{ex}$ fictitious flow is passing through the network . Under this flow, the unused capacity for each ℓ is $\frac{\epsilon_g}{(1+\epsilon_g)}$ flow of the input flow $\lambda_{in}(\ell)$. Thus the gap of $\frac{\epsilon_g}{(1+\epsilon_g)}\tilde{\lambda}_{in}(\ell)$ is present between the output flow $\tilde{\lambda}_{out}(\ell)$

and the input flow $\tilde{\lambda}_{in}(\ell)$. The distribution over h is given as $\mathbf{x}^{SS}$, which then consists of two parts $\tilde{\mathbf{x}} \prec \mathbf{x}^{SS}$ that corresponds to the real part of the flow. The reaming part $(\mathbf{x}^{SS} - \tilde{\mathbf{x}})$ corresponds to the fictitious part of the flow.

Furthermore, due to Bayesian routing it can be argued that at label node $\ell \in \mathcal{T}^c(\theta_{th}(1+\epsilon_g))$, the flow $\tilde{\lambda}_{in}(\ell)$ is at least $\sum_k P_\Psi(\ell,k)\lambda_{ex}$. Therefore, the uniform bound over all ℓ for a fixed λ_{ex}, θ_{th}, and ϵ_g is $\lambda_{min} := \left(\min_{\ell \in \mathcal{T}^c(\theta_{th}(1+\epsilon_g))} \sum_k P_\Psi(\ell,k)\right)\lambda_{ex}$.

The Event $\mathcal{E}_1^\tau$ and $\mathcal{E}_2^\tau$: We next consider the other two events next. In both these events, we have the trivial upper bound of $2M\sum_{\ell \in \mathcal{T}^c(\theta_{th}(1+\epsilon_g))} Q_\ell(\tau)$.

The Drift Inequality: We now provide the drift bound. Putting the cases together along with their respective probabilities, and accounting for the other terms in (1*) and (2*), we obtain the following.

$$\begin{aligned}
&\mathbb{E}\left[L(\tau+1) - L(\tau)|\mathbf{Q}(\tau)\right] \\
&\leq -\tfrac{\epsilon_g\lambda_{min}}{(1+\epsilon_g)} \sum_{\ell \in \mathcal{T}^c(\theta_{th}(1+\epsilon_g))} Q_\ell(\tau) + MC_1\mathbb{1}(\tau < C_1) \\
&+ C_0 + M^2 + O\left(\tfrac{1}{\log(\tau)^{\min(\alpha,\beta)}}\right) Q_\ell(\tau). \qquad (6)
\end{aligned}$$

B.3. Proof of Theorem 4.1

Proof Sketch. Following standard arguments in Lyapunov stability (Neely et al., 2005) we can argue that $\mathcal{Q}_{sum}^{MW} \leq O\left(\frac{\max(\lambda_{max}^2, M^2)}{\epsilon_g\lambda_{min}}\right)$. This completes the first part of Theorem 4.1.

The second part of the proof follows from the Bayesian departure and the properties of the oracle. Specifically, for the choice of the parameter $\epsilon_\theta = 2\log(\tau)^{-\alpha}$, given the event $\mathcal{E}_3^\tau$, the terminal set at round τ contains the set $\mathcal{T}(\theta_{th})$, i.e. $\widehat{\mathcal{T}}(\tau) \leq \mathcal{T}(\theta_{th})$. This is true because, under$\mathcal{E}_3^\tau$ the error in accuracy $\widehat{acc}(\ell,k) - acc(\ell,k)| \leq \log(\tau)^{-\alpha}$. Therefore, due to Lemma 3.4 this means that under $\mathcal{E}_3^\tau$, $\{\inf_{j\in D_p^\tau} Acc(j)\} \geq \theta_{th}$. Furthermore, $\mathcal{E}_3^\tau$ happens with probability at least $\tau^{-log(\tau)}$. Thus, we obtain $\mathbb{P}(\{\inf_{j\in D_p^\tau} Acc(j)\} < \theta_{th}) \leq \tau^{-log(\tau)}$. Averaging these quantities over time completes the proof. □

C. Proof of Section 5

In this section we provide proof and elaborations on the results presented in Section 5.

Inefficiency of the Previous Oracle: The algorithm designed in (Zhang et al., 2014) uses a robust tensor decomposition step (line 2b in Algo.1 therein) which is the most time consuming part. In order to achieve the desired accuracy we require performing tensor power updates starting from $\mathcal{O}(K\log(K)\log(1/\delta_U))$ randomly initialized points. For an (α,β) oracle in each round $\mathcal{O}(\beta\log(\sigma))$ time is required once σ exploration samples are collected. As $\sigma \to \infty$ the time required per round also grows unbounded.

Proof Sketch of Lemma 5.2. We provide a proof sketch of Lemma 5.2 where we ignore the dependence on constant terms such as K, M, and the minimum and maximum eigenvalues of the tensor. Our focus is solely on the dependence on τ.

To recall, instead of reinitializing the eigenvectors, we reuse the recovered eigenvectors from past rounds. In particular, we keep track of the eigenvectors recovered from stage $(\tau{-}1)$ and check whether it satisfies the stopping criterion (i.e. Eq. 3 in Algorithm 2) developed in (Anandkumar et al., 2015). If the condition is satisfied we do not require any initialization.

Rare Initialization: As long as the noise in the estimated tensor do not change, the set of eigenvectors continues to satisfy the condition by Lemma 5.1 and Lemma C.2 (Anandkumar et al., 2015), conditioned on the event that the tensor error is bounded by a constant value. At round $\tau \geq 1$, with probability at least $(1-\log(\tau)^{-\gamma})$ this event holds for large enough τ. We assume this event holds for the following paragraph for any $\gamma < 2$.

Fast Decay: If such a restart happens, at time τ, due to quadratic decay of robust tensor power method (Lemma 5.1 in (Anandkumar et al., 2015)), we achieve an error less than $\log(\tau)^{-\gamma'}$ within a $O(\log\log\log(\tau))$ time frame deterministically, for any constant $\gamma' > 0$, given the tensor estimation error is of the same order. Due to Lemma 1 in Appendix A in (Zhang et al., 2014), we know that the tensor error after round τ will be of the order of $\log(\tau)^{-1}$. Thus we may take any $\gamma' < 1$ for this step.

Decay completion before restart: Let $p_{rst}(\tau)$ be the probability of such a restart at time τ. Then for by union bound we have the probability of failure events as at most $p_{rst}O(\log\log\log(\tau))$, at round τ. Finally, we note that the tensor is updated 'only' during the exploration instances. Therefore, at time τ the probability of restart is $\log(\tau)/\tau$. Therefore, $p_{rst}O(\log\log\log(\tau)) = \tilde{O}(\log(\tau)/t)$.

Bounding the Bad events: Therefore, it follows through union bound over the two bad events— (i) tensor error is less than a given constant and (ii) the last failure events happened within $O(\log\log\log(\tau))$ rounds, that with probability at least $(1-\log(\tau)^{-\gamma})$ we have the error in time τ bounded from above by $\log(\tau)^{-\gamma'}$.

Furthermore, we know that with extremely high probability $(1-\tau^{-\log(\tau)})\,\sigma(\tau)$, i.e. the number of exploration samples at time τ, is of the order $\Theta(\log^{-2}(\tau))$. Therefore, we have for Algorithm 2 error in $\widehat{acc}(\cdot)$ and $\widehat{P}_\Psi(\cdot)$ is bounded by $\sigma^{-\gamma'/2}$ with probability $(1-\sigma^{-\gamma/2})$. This completes the proof of the above lemma, with any $\alpha < 1/2$ and with any $\beta < 1$.

Time Complexity: Finally, the time taken by the algorithm comprises of the time spent in the while loop (in line 8) and in the remaining parts. In the remaining parts the time complexity is constant (depending on M and K). To complete the proof we consider the amortized time complexity in the while loop. As discussed earlier, the while loop is not entered if there is no exploration events, thus the rate at which the while loop is entered at time τ is $\log(\tau)/\tau$. Furthermore, at time τ, we spent $\log\log(\log(\tau))$ iterations in the while loop with high probability. Therefore, the amortized time complexity of computations inside the while loop is vanishing as $\tilde{O}(\log(\tau)/\tau)$. □

D. Ensemble Property

The first ensemble comprises of three Alexnet, one VGG-19, and two Resnet-18 neural networks.(Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; He et al., 2016). The confusion matrices for this matrices are given in Fig.4.

$$\begin{bmatrix} 0.632 & 0.165 & 0.203 \\ 0.185 & 0.655 & 0.160 \\ 0.272 & 0.131 & 0.597 \end{bmatrix}$$

(a) First AlexNet

$$\begin{bmatrix} 0.610 & 0.154 & 0.236 \\ 0.230 & 0.633 & 0.137 \\ 0.296 & 0.159 & 0.545 \end{bmatrix}$$

(b) Second AlexNet

$$\begin{bmatrix} 0.635 & 0.123 & 0.241 \\ 0.226 & 0.583 & 0.191 \\ 0.279 & 0.098 & 0.623 \end{bmatrix}$$

(c) Third AlexNet

$$\begin{bmatrix} 0.888 & 0.041 & 0.071 \\ 0.052 & 0.916 & 0.032 \\ 0.123 & 0.043 & 0.834 \end{bmatrix}$$

(d) Second VGG-19

$$\begin{bmatrix} 0.897 & 0.044 & 0.059 \\ 0.057 & 0.914 & 0.029 \\ 0.131 & 0.049 & 0.820 \end{bmatrix}$$

(e) First Resnet-18

$$\begin{bmatrix} 0.925 & 0.022 & 0.053 \\ 0.057 & 0.917 & 0.026 \\ 0.134 & 0.035 & 0.831 \end{bmatrix}$$

(f) Second Resnet-18

Figure 3: Confusion Matrices of Ensemble-1

The second ensemble comprises of one VGG-11, one VGG-16, two VGG-19, and two Resnet-18 neural networks.(Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; He et al., 2016). The confusion matrices for this matrices are given in Fig.4.

$$\begin{bmatrix} 0.81 & 0.015 & 0.045 & 0.091 & 0.031 \\ 0.012 & 0.913 & 0.018 & 0.030 & 0.026 \\ 0.048 & 0.019 & 0.857 & 0.037 & 0.039 \\ 0.082 & 0.033 & 0.030 & 0.820 & 0.035 \\ 0.027 & 0.038 & 0.034 & 0.035 & 0.866 \end{bmatrix}$$

(a) VGG-11

$$\begin{bmatrix} 0.799 & 0.018 & 0.066 & 0.076 & 0.041 \\ 0.009 & 0.903 & 0.018 & 0.025 & 0.043 \\ 0.064 & 0.028 & 0.824 & 0.024 & 0.061 \\ 0.087 & 0.037 & 0.043 & 0.797 & 0.035 \\ 0.015 & 0.033 & 0.030 & 0.025 & 0.898 \end{bmatrix}$$

(b) VGG-16

$$\begin{bmatrix} 0.822 & 0.009 & 0.060 & 0.090 & 0.019 \\ 0.011 & 0.919 & 0.022 & 0.018 & 0.030 \\ 0.053 & 0.019 & 0.864 & 0.025 & 0.038 \\ 0.081 & 0.036 & 0.057 & 0.794 & 0.032 \\ 0.018 & 0.028 & 0.036 & 0.034 & 0.884 \end{bmatrix}$$

(c) First VGG-19

$$\begin{bmatrix} 0.808 & 0.009 & 0.052 & 0.099 & 0.032 \\ 0.015 & 0.888 & 0.017 & 0.046 & 0.033 \\ 0.058 & 0.021 & 0.838 & 0.040 & 0.044 \\ 0.079 & 0.025 & 0.038 & 0.820 & 0.037 \\ 0.022 & 0.032 & 0.041 & 0.030 & 0.875 \end{bmatrix}$$

(d) Second VGG-19

$$\begin{bmatrix} 0.850 & 0.017 & 0.049 & 0.067 & 0.018 \\ 0.019 & 0.925 & 0.015 & 0.024 & 0.016 \\ 0.050 & 0.017 & 0.881 & 0.019 & 0.032 \\ 0.084 & 0.028 & 0.035 & 0.827 & 0.026 \\ 0.021 & 0.027 & 0.027 & 0.025 & 0.900 \end{bmatrix}$$

(e) First Resnet-18

$$\begin{bmatrix} 0.825 & 0.019 & 0.057 & 0.067 & 0.032 \\ 0.011 & 0.924 & 0.018 & 0.019 & 0.027 \\ 0.054 & 0.024 & 0.859 & 0.015 & 0.048 \\ 0.101 & 0.041 & 0.035 & 0.792 & 0.032 \\ 0.019 & 0.038 & 0.026 & 0.024 & 0.892 \end{bmatrix}$$

(f) Second Resnet-18

Figure 4: Confusion Matrices of Ensemble-2

E. Performance Comparison

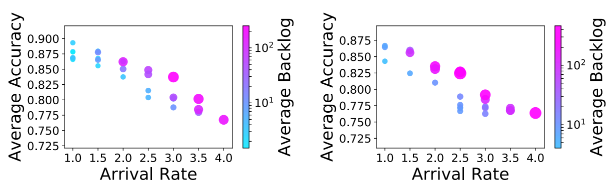
In this section we show that our proposed algorithm performs better than a random scheduler.

Random Scheduler: The random scheduler is a parameterized by the number of classifiers per sample, $n_{clf} \in (0, M)$, where M is the number of classifiers. In each time-slot, a we schedule a random matching between the samples and the classifiers. Each sample leaves the system after either receiving $\lceil n_{clf} \rceil$ number of labels w.p. $(n_{clf} - \lfloor n_{clf} \rfloor)$, or $\lfloor n_{clf} \rfloor$ number of labels. The samples are returned to the memory until it collects the required number of labels. Upon leaving the system, the maximum *a posteriori prediction is used as the sample label.*

Pareto Region of Random Scheduler: *As we increase* n_{clf}, *the* average accuracy *of the random scheduler increases, whereas the maximum arrival rate supported , which is equal to* M/n_{clf}, *decreases as the number of classifiers are fixed. The random scheduler is not adaptive to the labels collected by any sample as the number of classifiers are a-priori determined. Due to this reason,* the random scheduler can not support threshold accuracy larger than $\|\mathbf{p}_g\|_\infty$ for $n_{clf} < M$ *(i.e. each sample is sent to all the classifiers). Therefore, the random scheduler has a bi-modal Pareto region* $(\lambda, \theta_{th}) \in [0, \infty) \times [0, \|\mathbf{p}_g\|_\infty] \cup [0, 1) \times (\|\mathbf{p}_g\|_\infty, 1]$.

Empirical Evaluation: *We use the same experimental setup as Section 6. Recall that in the first the first setting, the dataset has three groups: (airplanes, ships, trucks, cars), (birds, frogs cats), and (dogs, deer, horses) and the ensemble has 6 classifiers - three Alexnet, one VGG-19, and two ResNet-18 neural nets (Krizhevsky et al., 2012; Simonyan & Zisserman, 2014; He et al., 2016).*

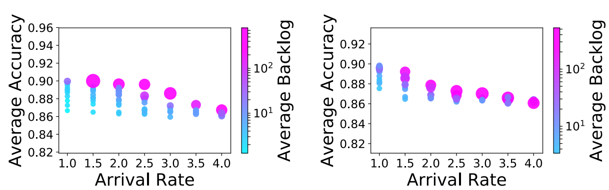
In the second, we have the dataset with five groups- (airplanes, ships), (trucks, cars), (birds, frogs) (cats, dogs) and (deer, horses), and the ensemble with 6 classifiers- one VGG-11 , one VGG-16, two VGG-19, and two ResNet-18 neural nets.



(a) Max-weight Scheduler and Bayesian Labeling

(b) Random Scheduler and Bayesian Labeling

Figure 5: Comparison for Ensemble 1



(a) Max-weight Scheduler and Bayesian Labeling

(b) Random Scheduler and Bayesian Labeling

Figure 6: Comparison for Ensemble 2

We compare the performance of our algorithm against the Random Scheduler, for the two above mentioned settings by showing the average accuracy vs arrival rate plots. In the two figures, the circles' sizes (also the varying color) depict the logarithm of the average queue length. In Fig. 5, for the first setting we observe that for arrival rate between 1 *to* 3 *samples/slot the difference in average accuracy is* 2.5% *to* 5% *larger for the proposed algorithm. Similarly, in Fig. 6, for the second setting we observe that for arrival rate between* 2 *to* 3 *samples/slot the difference in average accuracy is* 1.5% *to* 2% *larger for the proposed algorithm. In summary, over the range of arrival rates we examined, our proposed algorithm performs better than, or equivalently to random scheduling.*

Theoretically and empirically comparing the proposed Pareto optimal algorithm with other simple algorithms is an interesting future work.