A More Ablation Tables

Table 3: Varying number of sampled objectives per-iteration.

Task	$\frac{3}{24}$ tasks	$\frac{6}{24}$ tasks
ACL-ARC	$72.11_{2.12}$	$\mathbf{73.26}_{1.32}$
SCIERC	$82.35_{1.76}$	$\mathbf{82.98}_{1.52}$
SE-2016-6	$\mathbf{72.46}_{1.65}$	$\mathbf{72.46}_{0.90}$
CHEMPROT	$\mathbf{83.91}_{0.32}$	$83.69_{0.98}$
H.PARTISAN	$\mathbf{98.46}_{0.0}$	$97.95_{0.73}$

B Dataset Details

Table 4: Specifications of datasets used to evaluate our methods.

Domain	Task	Label Type	Train Size	Dev Size	Test Size	Classes	Metric
BIOMED	CHEMPROT Kringelum et al. (2016)	relation classification	4169	2427	3469	13	Accuracy
CS	SCIERC Luan et al. (2018)	relation classification	3219	455	974	7	F1
STANCE	SE-2016-6 Mohammad et al. (2016)	stance detection	2497	417	1249	3	Accuracy
CS	ACL-ARC Jurgens et al. (2018)	citation intent	1688	114	139	6	F1
NEWS	H.PARTISAN Kiesel et al. (2019)	partisanship	515	65	65	2	Accuracy

C More Training Details

We run each hyper-parameter configuration across 3 seeds $\{0, 1, 2\}$. We use a batch size of 128 for all end-tasks tasks except H.PARTISAN where we use a batch size of 64. The auxiliary task batch-size, `aux_bsz`, is shared across all the n sub-sampled auxiliary objectives according to the objective's weight.

We use the AdamW optimizer (Loshchilov & Hutter, 2017), with weight decay of 0.01 for all experiments.

Table 5: **AANG**-TD specific Hyper-parameters

Hyper-parameter	Values	Description
aux_lr	1.0, 0.1	Learning rate for factor vectors - $\{W^{\text{All}}, W^{\mathcal{I}}, W^{\mathcal{T}}, W^{\mathcal{R}}, W^{\mathcal{O}}\}$
sopt_lr	0.1, 0.01	Learning rate for primary task weighting λ_e
nconf_subsamp	3, 6	Number of sub-sampled auxiliary tasks.
learning rate	1e-3, 1e-4	Learning rate used for further training of $\text{RoBERTa}_{\text{base}}$
aux_bsz	256	Batch size of for auxiliary objectives

Table 6: **AANG**-TD+ED specific Hyper-parameters

Hyper-parameter	Values	Description
aux_lr	1.0, 0.5, 0.1	Learning rate for factor vectors - $\{W^{\text{All}}, W^{\mathcal{I}}, W^{\mathcal{T}}, W^{\mathcal{R}}, W^{\mathcal{O}}\}$
sopt_lr	0.1	Learning rate for primary task weighting λ_e
nconf_subsamp	6, 12, 24	Number of sub-sampled auxiliary tasks.
learning rate	1e-4	Learning rate used for further training of $\text{RoBERTa}_{\text{base}}$
aux_bsz	1024	Batch size of for auxiliary objectives

META-TARTAN introduces a dev-head which is trained sporadically during training for estimating the meta-gradients. We use the following hyper-parameters for training this dev-head : we sample 32 examples (8 examples in the case of H.PARTISAN) and perform full batch gradient descent with

Table 7: **AANG** Hyper-parameters for single task auxiliary tasks

Hyper-parameter	Values	Description
sopt_lr	1.0, 0.1, 0.01	Learning rate for primary task weighting λ_e
learning rate	1e-3, 1e-4, 5e-5	Learning rate used for further training of $\text{RoBERTa}_{\text{base}}$

a learning rate of 1e-2 for 10 iterations. The dev-head is trained with the AdamW optimizer with weight decay set to 0.1.

We copy the end-task agnostic baseline results from (Dery et al., 2021b) when available. We use the hyper-parameters specified for TAPT in Gururangan et al. (2020) to train for the SE-2016-6 task.

All models were trained on one of two types of gpus: NVIDIA A100 or NVIDIA A6000. All models fit within a single gpu. We used gradient accumulation to expand the effective batch sizes used for our experiments.

D Generalization Error Bound for End-task Aware Training

D.1 Definitions

Definition D.1. *A function, $f : \Omega \to \mathbb{R}$ is L-Lipschitz if $\forall u, v \in \text{dom}(f)$:*

$$\|f(u) - f(v)\| \leq L\|u - v\|$$

Note that L-Lipschitz implies bounded gradients.

$$\|\nabla f(w)\| \leq L \quad \forall w$$

Definition D.2. *A function, $f : \Omega \to \mathbb{R}$ is β-smooth if $\forall u, v \in \Omega$:*

$$\|\nabla f(u) - \nabla f(v)\| \leq \beta\|u - v\|$$

Definition D.3. *An update rule, G is σ-bounded if :*

$$\sup_{w\in\Omega} \ \|w - G(w)\| \leq \sigma$$

Consider the following general setting. There is an unknown distribution $\mathcal{D}_e$ over examples from some space $\mathcal{Z}$. We receive a sample $S = (z_1, \ldots, z_{N_e})$ of N_e examples drawn i.i.d. from $\mathcal{D}_e$. Our goal is to find a model w, that parameterizes the function f_e, with small population risk defined as:

Definition D.4. ***Population Risk***

$$R[w] = \mathbf{E}_{z\sim\mathcal{D}_e} f_e(w; z)$$

Definition D.5. ***Empirical Risk***
Since we have a finite number of samples, we can only compute the empirical risk which is :

$$R_S[w] = \frac{1}{N_e}\sum_i f_e(w; z_i),$$

Let A be a potentially randomized algorithm (such as Stochastic Gradient Descent) that is a function of the S such that $w = A(S)$.

Definition D.6. ***Generalization Error*** $\epsilon_{gen}(A, N_e)$

$$\epsilon_{gen}(A, N_e) = \mathbf{E}_{S,A}\big[R_S[A(S)] - R[A(S)]\big]$$

Definition D.7. ***Uniform Stability***
A randomized algorithm A is ϵ-uniformly stable if for all data sets $S, S' \in \mathcal{Z},\ |S| = |S'| = N_e$ such that S and S' differ in at most one example, we have

$$\sup_z \ \mathbf{E}_A\big[f_e(A(S); z) - f_e(A(S'); z)\big] \leq \epsilon$$

Here, the expectation is taken only over the internal randomness of A. We will denote by $\epsilon_{\text{stab}}(A, N_e)$ the infimum over all ϵ for which the above holds.

D.2 RELEVANT THEOREMS

Theorem D.1 (Uniform Stability implies Generalization in expectation)**.** *Let Algorithm A be ϵ-uniformly stable. Then,*

$$\epsilon_{gen}(A, N_e) = \left| \mathbf{E}_{S,A}\big[R_S[A(S)] - R[A(S)]\big] \right| \leq \epsilon_{stab}(A, N_e)$$

For full proof see Theorem 2.2 of Hardt et al. (2016).

Theorem D.2 (Stochastic Gradient Method is stable)**.** *Assume that $f_e(;z) \in [0,1]$ is an L-Lipschitz and β_e-smooth loss function for every z. Suppose that we run SGM for T steps with monotonically non-increasing step sizes $\alpha_t \leq \frac{c}{t}$. Then, SGM has uniform stability with :*

$$\epsilon_{sgm} \leq \frac{1 + \frac{1}{q}}{N_e - 1}\big(2cL^2\big)^{\frac{1}{q+1}} T^{\frac{q}{q+1}}$$

$$\text{where } \; q = \beta_e c$$

We can simplify this to only terms involving T and N_e

$$\epsilon_{sgm} \lessapprox \frac{T^{1-\frac{1}{c\beta_e+1}}}{N_e} \tag{2}$$

Proof. For the full proof, see *Theorem 3.12* of Hardt et al. (2016) □

D.3 GROWTH FUNCTIONS

Lemma D.3 (Growth Recursion Under Dynamic Sampling)**.** *We consider the Stochastic Gradient update rule $G : \Omega \to \Omega$:*

$$G_f(w) = w - \alpha \nabla f(w)$$

Fix an arbitrary sequence of updates $G_{f_1}, \ldots, G_{f_T}$ and another $G'_{f_1}, \ldots, G'_{f_T}$. Let $w_0 = w'_0$ be a starting point in Ω given that $f : \Omega \to \mathbb{R}$ and define

$$\delta_t = \mathbb{E}_{f_1 \ldots f_t \sim \mathcal{P}_\lambda}\big[\|w_t - w'_t\|\big]$$

where w_t, w'_t are defined recursively through :

$$w_t = G_{f_t}(w_{t-1}) \quad w'_t = G'_{f_t}(w'_{t-1}) \quad t \geq 0$$

Then we have the recurrence relation :

$$\delta_0 = 0$$

$$\delta_{t+1} \leq \begin{cases} \min\big\{\big(1 + \alpha\lambda_1\beta_1\big)\delta_t + \alpha\lambda_2\big(\Delta + 2L\big),\ \big(1 + \alpha\big(\lambda_1\beta_1 + \lambda_2\beta_2\big)\big)\delta_t\big\} & G_{f_t} = G'_{f_t} \\ \delta_t + 2\sigma_t & G_{f_t}, G'_{f_t} \textit{ are } \sigma\textit{-bounded} \end{cases}$$

Note that $\mathcal{P}_f$ is a distribution over the support $\{f^1, f^2\}$ according to probabilities $\{\lambda_1, \lambda_2 \mid \lambda_1 + \lambda_2 = 1\}$. $\{f_1, f_2\}$ have smoothness β_1, β_2 respectively.

Proof. The second bound on δ_t is taken directly from Lemma 2.5 of Hardt et al. (2016). We now derive the first-half of the first bound

$$\begin{aligned}
\delta_{t+1} &= \mathbb{E}_{f_1 \dots f_{t+1} \sim \mathcal{P}_\lambda}\left[\|w_{t+1} - w'_{t+1}\|\right] \\
&= \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\lambda_1 \|G_{f^1}(w_t) - G'_{f^1}(w'_t)\| + \lambda_2 \|G_{f^2}(w_t) - G'_{f^2}(w'_t)\|\right] \\
&= \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\lambda_1 \|w_t - \alpha \nabla f^1(w_t) - w'_t + \alpha \nabla f^1(w'_t)\| + \lambda_2 \|w_t - \alpha \nabla f^2(w_t) - w'_t + \alpha \nabla f^2(w'_t)\|\right] \\
&\leq \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\|w_t - w'_t\|\right] + \alpha \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left(\lambda_1 \|\nabla f^1(w'_t) - \nabla f^1(w_t)\| + \lambda_2 \|\nabla f^2(w'_t) - \nabla f^2(w_t)\|\right) \\
&\quad \text{(Triangle Inequality used for above step)} \\
&= \delta_t + \alpha \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left(\lambda_1 \|\nabla f^1(w'_t) - \nabla f^1(w_t)\| + \lambda_2 \|\nabla f^2(w'_t) - \nabla f^2(w_t)\|\right) \\
&\quad \text{(Without Loss of Generality, let } \beta_1 \leq \beta_2\text{)} \\
&\leq \delta_t + \alpha \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\lambda_1 \beta_1 \|w_t - w'_t\| + \lambda_2 \|\nabla f^2(w'_t) - \nabla f^2(w_t)\|\right] \quad \text{(Smoothness)} \\
&= \delta_t + \alpha \lambda_1 \beta_1 \delta_t + \alpha \lambda_2 \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\|\nabla f^2(w'_t) - \nabla f^2(w_t)\|\right] \quad \text{(Triangle Inequality)} \\
&= \big(1 + \alpha \lambda_1 \beta_1\big)\delta_t + \alpha \lambda_2 \left\|\nabla f^2(w'_t) - \nabla f^1(w'_t) + \nabla f^1(w'_t) - \nabla f^2(w_t)\right\| \quad \text{(add zero)} \\
&\leq \big(1 + \alpha \lambda_1 \beta_1\big)\delta_t + \alpha \lambda_2 \Big(\|\nabla f^2(w'_t) - \nabla f^1(w'_t)\| + \|\nabla f^1(w'_t) - \nabla f^2(w_t)\|\Big) \quad \text{(Triangle Inequality)} \\
&\leq \big(1 + \alpha \lambda_1 \beta_1\big)\delta_t + \alpha \lambda_2 \Big(\Delta + \|\nabla f_1(w'_t) - \nabla f_2(w_t)\|\Big) \quad \text{Using Assumption A.1} \\
&\leq \big(1 + \alpha \lambda_1 \beta_1\big)\delta_t + \alpha \lambda_2 \Big(\Delta + \|\nabla f_1(w'_t)\| + \|\nabla f_2(w_t)\|\Big) \quad \text{Triangle Inequality} \\
&\leq \big(1 + \alpha \lambda_1 \beta_1\big)\delta_t + \alpha \lambda_2 \big(\Delta + 2L\big) \quad L\text{-Lipschitz function}
\end{aligned}$$

To obtain the second half of the first bound :

$$\begin{aligned}
\delta_{t+1} &= \mathbb{E}_{f_1 \dots f_{t+1} \sim \mathcal{P}_\lambda}\left[\|w_{t+1} - w'_{t+1}\|\right] \\
&= \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\lambda_1 \|G_{f^1}(w_t) - G'_{f^1}(w'_t)\| + \lambda_2 \|G_{f^2}(w_t) - G'_{f^2}(w'_t)\|\right] \\
&= \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\lambda_1 \|w_t - \alpha \nabla f^1(w_t) - w'_t + \alpha \nabla f^1(w'_t)\| + \lambda_2 \|w_t - \alpha \nabla f^2(w_t) - w'_t + \alpha \nabla f^2(w'_t)\|\right] \\
&\leq \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\|w_t - w'_t\|\right] + \alpha \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left(\lambda_1 \|\nabla f^1(w'_t) - \nabla f^1(w_t)\| + \lambda_2 \|\nabla f^2(w'_t) - \nabla f^2(w_t)\|\right) \\
&\quad \text{(Triangle Inequality used for above step)} \\
&\leq \delta_t + \alpha \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\lambda_1 \beta_1 \|w_t - w'_t\| + \lambda_2 \beta_2 \|w_t - w'_t\|\right] \quad \text{(Smoothness)} \\
&= \delta_t + \alpha \lambda_1 \beta_1 \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\|w_t - w'_t\|\right] + \alpha \lambda_2 \beta_2 \mathbb{E}_{f_1 \dots f_t \sim \mathcal{P}_\lambda}\left[\|w_t - w'_t\|\right] \\
&= \delta_t + \alpha(\lambda_1 \beta_1 + \lambda_2 \beta_2)\delta_t \\
&= (1 + \alpha(\lambda_1 \beta_1 + \lambda_2 \beta_2))\delta_t
\end{aligned}$$

□

D.4 Stability of Dynamic Sampling

We repeat the description of our Auxiliary Learning with Dynamic Sampling Setting here for ease of access.

Setting : We are given an auxiliary objective $f_a(\cdot; z) \in [0, 1]$ with N_a samples $S_a = (z_1, \ldots, z_{N_a})$ from the distribution $\mathcal{D}_a$. At any iteration of SGD, we sample a choice of either the end-task function f_e or the auxiliary objective f_a according to the probabilities λ_e, $\lambda_a \mid \lambda_e + \lambda_a = 1$. Given the chosen objective, we sample a data-point and perform stochastic gradient descent (SGD) based on the sampled data-point.

An equivalent way to instantiate this procedure to create S_A by drawing $N' = N_e + N_a$ total samples from the end-task and auxiliary task according to $\mathcal{P}_\lambda$. S'_A is then created by replacing 1 end-task sample in S_A. At each step, a sample is drawn from a distribution : $z_i, z'_i \sim P_{S_A}, P_{S'_A}$ and a gradient step is taken on the function corresponding to the set the sample was drawn from.

Lemma D.4 (Stability of dynamic sampling)**.** *We denote the outputs of T steps of SGM on S_A and S'_A with the dynamically sampled functions, as w_T and w'_T respectively. Then, for every $z_e \in Z_e$ and every $t_0 > 0$, under both the random update rule and the random permutation rule, we have :*

$$\mathbb{E}\big|f_e(w_T; z) - f_e(w'_T; z)\big| \leq \frac{\gamma t_0}{N'} \sup_{w, z_e} f_e(w; z_e) + L\mathbb{E}[\delta_T | \delta_{t_0} = 0]$$

Where $N' = N_e + N_a$ and $\gamma = \frac{\lambda_e \cdot N'}{N_e} = \frac{\lambda_e}{\lambda^r}$.

Proof. Let $\mathcal{E} = \mathbf{1}[\delta_{t_0} = 0]$ denote the event that $\delta_{t_0} = 0$. We have

$$\begin{aligned}
\mathbb{E}\big|f_e(w_T; z) - f_e(w'_T; z)\big| &= P\{\mathcal{E}\}\mathbb{E}\big[\big|f_e(w_T; z) - f_e(w'_T; z)\big|\big|\mathcal{E}\big] \\
&\quad + P\{\mathcal{E}^c\}\mathbb{E}\big[\big|f_e(w_T; z) - f_e(w'_T; z)\big|\big|\mathcal{E}^c\big] \\
&\leq \mathbb{E}\big[\big|f_e(w_T; z) - f_e(w'_T; z)\big|\big|\mathcal{E}\big] + P\{\mathcal{E}^c\} \cdot \sup_{w, z_e} f_e(w; z_e) \\
&\quad \text{because } f_e \text{ is non-negative} \\
&\leq L\mathbb{E}\big[\|w_T - w'_T\|\big|\mathcal{E}\big] + P\{\mathcal{E}^c\} \cdot \sup_{w, z_e} f_e(w; z_e) \\
&\quad \text{because } f_e \text{ is } L\text{-Lipschitz}
\end{aligned} \tag{3}$$

We now proceed to bound $P\{\mathcal{E}^c\}$. Let $i_* \in [N']$ denote the position in which S_A, S'_A differ and consider the random variable I assuming the index of the first time step in which SGM uses the example $z_e^{i_*}$. Note that when $I > t_0$, then we must have that $\delta_{t_0} = 0$ since the two samples are identical up until this point.

$$P\{\mathcal{E}^c\} = P\{\delta_0 \neq 0\} \leq P\{I \leq t_0\}$$

Using the selection rule specified above (sample either f_e, f_a according to the probabilities λ_e, λ_a and then sample uniformly from the selected task data) we have that :

$$P\{I \leq t_0\} = \sum_{t=1}^{t_0} P\{I = t_0\} = \sum_{t=1}^{t_0} \big(\lambda_e \cdot \frac{1}{N_e}\big) = \frac{\lambda_e t_0}{N_e} = \frac{\gamma t_0}{N'}$$

□

Theorem D.5 (Stability Bound on Dynamic Sampling)**.** *Assume that $f_e(; z_e), f_a(; z_a) \in [0, 1]$ are L-Lipschitz and β_e and β_a-smooth loss functions. Consider that we have $N' = N_e + N_a$ total samples where f_e and f_a have N_e and N_a samples respectively. Suppose that we run SGM for T steps with monotonically non-increasing step sizes $\alpha_t \leq \frac{c}{t}$ by dynamically sampling the tasks according to λ_e and λ_a. Then, with respect to f_e, SGM has uniform stability with :*

$$\epsilon_{\text{stab}} \leq \left(1 + \frac{1}{c\bar{\beta}}\right)\left(\frac{2\gamma L^2 c}{N' - \gamma} + \rho L c\right)^{\frac{1}{c\bar{\beta}+1}} \left(\frac{\gamma T}{N'}\right)^{\frac{c\bar{\beta}}{1+c\bar{\beta}}}$$

$$\textit{Where} \quad \gamma = \frac{\lambda_e N'}{N_e}$$

Given that $\beta^* = \min\{\beta_e, \beta_a\}$ and λ^* is the corresponding weighting of the function with smaller smoothness.

Depending on which one gives a tighter bound the pair $(\bar{\beta}, \rho)$ can be :

$$(\bar{\beta}, \rho)_1 = (\lambda^* \beta^*,\ (1 - \lambda^*)\big(\Delta + 2L\big))$$

or

$$(\bar{\beta}, \rho)_2 = (\lambda_e \beta_e + \lambda_a \beta_a,\ 0)$$

When $(\bar{\beta}, \rho)_1$ gives the tighter bound, we can simplify to :

$$\epsilon_{\text{gen}} \lessapprox (\Delta)^{\frac{1}{1+c\lambda^*\beta^*}} \left(\frac{\gamma T}{N'}\right)^{1-\frac{1}{c\lambda^*\beta^*+1}}$$

As presented in Section 4.

Proof. Let S_A, S'_A be two sample of size $N' = N_e + N_a$ as described in lemma D.4. Consider the gradient updates $G_{f_1}, \ldots, G_{f_T}$ and $G'_{f_1}, \ldots, G'_{f_T}$ induced by running SGM on samples S_A and S'_A respectively. Let w_T and w'_T denote the corresponding outputs of SGM. By lemma D.4 we have :

$$\mathbb{E}\big|f_e(w_T; z) - f_e(w'_T; z)\big| \leq \frac{\gamma t_0}{N'} \sup_{w, z_e} f_e(w; z_e) + L\mathbb{E}[\delta_T | \delta_{t_0} = 0] \tag{4}$$

Let $\Psi_T = \mathbb{E}[\delta_T | \delta_{t_0} = 0]$. We will bound Ψ_T as function of t_0 and then minimize for t_0. Note the following :

- At any step t, with probability $\big(1 - \frac{\gamma}{N'}\big)$, the sample selected is the same in both S_A and S'_A. In this case $G_{f_t} = G'_{f_t}$ and we use the corresponding expansivity rule from lemma D.4. This gives :

  $$\delta_{t+1} \leq \min\Big\{\big(1 + \alpha_t\lambda^*\beta^*\big)\delta_t + \alpha_t(1-\lambda^*)\big(\Delta + 2L\big),\ \big(1 + \alpha_t\big(\lambda_e\beta_e + \lambda_a\beta_a\big)\big)\delta_t\Big\}$$

  Where $\beta^* = \min\{\beta_e, \beta_a\}$ and λ^* is the corresponding weighting of the function with smaller smoothness. To avoid deriving the bound independently for each case, we perform a variable substituation that captures the two cases :

  $$\delta_{t+1} \leq \big(1 + \alpha_t\bar{\beta}\big)\delta_t + \alpha_t\rho$$

  $\bar{\beta} = \big\{\lambda^*\beta^*,\ \lambda_e\beta_e + \lambda_a\beta_a\big\}$ and $\rho = \big\{(1-\lambda^*)\big(\Delta + 2L\big), 0\big\}$. We can present the final bound in terns of these variables which can be substituted depending on the minimizer.

- With probability $\frac{\gamma}{N'}$ the selected example is different. Note that in this case, we know that we are evaluating the end-task function f_e. We use that both G_{f_t} and G'_{f_t} are $(\sigma_t = \alpha_t L)$-bounded according to lemma D.3 since f_e is L-Lipschitz.

Combining the above we have :

$$\begin{aligned}
\Psi_{t+1} &\leq \big(1 - \frac{\gamma}{N'}\big)\Big(\big(1 + \alpha_t\bar{\beta}\big)\Psi_t + \alpha_t\rho\Big) + \frac{\gamma}{N'}\big(\Psi_t + 2\alpha_t L\big) \\
&= \Big(\frac{\gamma}{N'} + \big(1 - \frac{\gamma}{N'}\big)\big(1 + \alpha_t\bar{\beta}\big)\Big)\Psi_t + \frac{2\gamma\alpha_t L}{N'} + \alpha_t\big(1 - \frac{\gamma}{N'}\big)\rho \\
&= \Big(1 + \big(1 - \frac{\gamma}{N'}\big)\alpha_t\bar{\beta}\Big)\Psi_t + \frac{\alpha_t\big(2\gamma L + (N' - \gamma)\rho\big)}{N'} \\
&\leq \Big(1 + \big(1 - \frac{\gamma}{N'}\big)\frac{c}{t}\bar{\beta}\Big)\Psi_t + \frac{c\big(2\gamma L + (N' - \gamma)\rho\big)}{tN'} \\
&\leq \exp\Big(\big(1 - \frac{\gamma}{N'}\big)\frac{c}{t}\bar{\beta}\Big)\Psi_t + \frac{c\big(2\gamma L + (N' - \gamma)\rho\big)}{tN'} \\
&\quad \text{We use } 1 + x \leq \exp(x)\ \forall x \\
&\leq \exp\Big(\big(1 - \frac{\gamma}{N'}\big)\frac{c}{t}\bar{\beta}\Big)\Psi_t + \frac{c\bar{\rho}}{tN'} \\
&\quad \text{Where } \bar{\rho} = \big(2\gamma L + (N' - \gamma)\rho\big)
\end{aligned} \tag{5}$$

We can unwind the recurrence until $\Psi_{t_0} = 0$.

$$\begin{aligned}
\Psi_T &\leq \sum_{t=t_0+1}^{T} \left(\prod_{k=t+1}^{T} \exp\left((1-\frac{\gamma}{N'})\frac{c\bar{\beta}}{k}\right)\right)\left(\frac{c\bar{\rho}}{tN'}\right) \\
&= \sum_{t=t_0+1}^{T} \left(\frac{c\bar{\rho}}{tN'}\right) \exp\left((1-\frac{\gamma}{N'})c\bar{\beta}\sum_{k=t+1}^{T}\frac{1}{k}\right) \\
&\leq \sum_{t=t_0+1}^{T} \left(\frac{c\bar{\rho}}{tN'}\right) \exp\left((1-\frac{\gamma}{N'})c\bar{\beta}\log\left(\frac{T}{t}\right)\right) \\
&= \frac{c\bar{\rho}T^{c\bar{\beta}(1-\frac{\gamma}{N'})}}{N'} \sum_{t=t_0+1}^{T} t^{-c\bar{\beta}(1-\frac{\gamma}{N'})-1} \qquad (6)
\end{aligned}$$

We can upper bound the sum over t with an integral + drop negative terms

$$\begin{aligned}
&\leq \frac{c\bar{\rho}}{N'c\bar{\beta}(1-\frac{\gamma}{N'})}\left(\frac{T}{t_0}\right)^{c\bar{\beta}(1-\frac{\gamma}{N'})} \\
&= \frac{\bar{\rho}}{\bar{\beta}(N'-\gamma)}\left(\frac{T}{t_0}\right)^{c\bar{\beta}(1-\frac{\gamma}{N'})} \\
&\leq \frac{\bar{\rho}}{\bar{\beta}(N'-\gamma)}\left(\frac{T}{t_0}\right)^{c\bar{\beta}}
\end{aligned}$$

Plugging this bound back into Equation 4 and using the fact that $f_e \in [0,1]$:

$$\mathbb{E}\big|f_e(w_T;z) - f_e(w'_T;z)\big| \leq \frac{\gamma t_0}{N'} + \frac{L\bar{\rho}}{\bar{\beta}(N'-\gamma)}\left(\frac{T}{t_0}\right)^{c\bar{\beta}} \qquad (7)$$

We let $q^* = c\bar{\beta}$, we can minimize the R.H.S by setting :

$$t_0 = \left(\frac{N'Lc\bar{\rho}}{\gamma(N'-\gamma)}\right)^{\frac{1}{q^*+1}} T^{\frac{q^*}{q^*+1}}$$

Plugging this in gives us :

$$\begin{aligned}
\mathbb{E}\big|f_e(w_T;z) - f_e(w'_T;z)\big| &\leq \left(\frac{(1+\frac{1}{c\bar{\beta}})}{N'}\right)\left(\frac{N'Lc\big(2\gamma L + (N'-\gamma)\rho\big)}{(N'-\gamma)}\right)^{\frac{1}{c\bar{\beta}+1}} \big(\gamma T\big)^{\frac{c\bar{\beta}}{1+c\bar{\beta}}} \\
&= \left(1+\frac{1}{c\bar{\beta}}\right)\left(\frac{2\gamma L^2 c}{N'-\gamma} + \rho Lc\right)^{\frac{1}{c\bar{\beta}+1}} \left(\frac{\gamma T}{N'}\right)^{\frac{c\bar{\beta}}{1+c\bar{\beta}}} \qquad (8)
\end{aligned}$$

Recall that :

$$\bar{\beta} = \big\{\lambda^*\beta^*,\ \lambda_e\beta_e + \lambda_a\beta_a\big\}$$

$$\rho = \big\{(1-\lambda^*)\big(\Delta + 2L\big), 0\big\}$$

We can choose whichever of the pairs for $\bar{\beta}, \rho$ that minimizes the bound : □

E Discussion of Generalization Error Bounds

E.1 What Does Theorem D.5 Say.

We consider the setting where

$$\bar{\beta} = \lambda^*\beta^*$$

$$\rho = (1-\lambda^*)\big(\Delta + 2L\big)$$

Assuming the ρ term dominates Equation 8 in this setting is :

$$\begin{aligned}\epsilon_{\text{gen}}^{\text{auxdyn}} \leq \epsilon_{\text{stab}}^{\text{auxdyn}}\big|_{(\bar{\beta},\rho)_1} &\lessapprox \sqrt[1+c\bar{\beta}]{(1-\lambda^*)(\Delta+2L)}\left(\frac{\gamma T}{N'}\right)^{\frac{c\bar{\beta}}{1+c\bar{\beta}}} \\ &\lessapprox (\Delta)^{\frac{1}{1+c\lambda^*\beta^*}}\left(\frac{\gamma T}{N'}\right)^{1-\frac{1}{c\lambda^*\beta^*+1}} \quad \text{This is Equation 1 from Section 4}\end{aligned} \tag{9}$$

In going from the first line to the second we consider the setting where $\Delta \gg 2L$. This is a case where the auxiliary task is sufficiently different from the primary task. Some observations about this setting:

1. Smaller Δ implies auxiliary task is similar to main task and leads to improving the bound.
2. Dependence of the bound on N' is a bit more nuanced. **Note that increasing N' increases γ unless we reduce λ_e appropriately**. Remember that λ_e is the rate at which we sample the primary task. Thus, if we add more auxiliary data but still sample the primary task at the original rate, then we are effectively ignoring the extra auxiliary data.
3. It might be tempting to assume that we can get arbitrary improvements in this setting by setting $\lambda_e = 0$. However, **note that whilst this might reduce the generalization error**, it means that we are seeing none of the end-task which would result in large **increase in the training error**
4. Note that $(\bar{\beta} = \lambda^*\beta^* \leq \beta_e)$ always. So we get improvements on the dependence on T compared to Theorem D.2.
5. We can optimize λ_e, λ_a to minimize $\epsilon_{\text{stab}}^{\text{auxdyn}}$.