A Proofs

A.1 Main Proofs

Proposition 1. *For any ground-truth pair* $(\mathbf{x}^*, \mathbf{y}^*)$*,* P_{Q_R} *and* Q_R *satisfy the following marginal match condition and terminal condition:*

$$\prod_{t=1}^{|\mathbf{y}|} P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}) = P_R(\mathbf{y} \mid \mathbf{x}^*) \quad \forall \mathbf{y} \in \mathcal{Y} \tag{21}$$

$$Q_R(\hat{\mathbf{y}}, \textit{eos}; \mathbf{y}^*) = R(\hat{\mathbf{y}} + \textit{eos}; \mathbf{y}^*) - R(\hat{\mathbf{y}}; \mathbf{y}^*) \quad \forall \hat{\mathbf{y}} \in \mathcal{Y}^- \tag{22}$$

if and only if for any $\mathbf{y} \in \mathcal{Y}$*,*

$$Q_R(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = \begin{cases} R(\mathbf{y}_1^t; \mathbf{y}^*) - R(\mathbf{y}_1^{t-1}; \mathbf{y}^*) + \ \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w; \mathbf{y}^*)/\tau\right), & t < |\mathbf{y}| \\ R(\mathbf{y}_1^t; \mathbf{y}^*) - R(\mathbf{y}_1^{t-1}; \mathbf{y}^*), & t = |\mathbf{y}| \end{cases} \tag{23}$$

Proof. To avoid clutter, we drop the dependency on $\mathbf{x}^*$ and $\mathbf{y}^*$. The following proof holds for each possible pair of $(\mathbf{x}^*, \mathbf{y}^*)$.

Firstly, it is easy to see that the terminal condition in Eqn. (22) exactly corresponds to the $t = |\mathbf{y}|$ case of Eqn. (23), since $y_t = \texttt{eos}$ for $y \in \mathcal{Y}$. So, we will focus on the non-terminal case next.

Sufficiency For convenience, define $V_R(\mathbf{y}_1^t) = \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w)/\tau\right)$. Suppose Eqn. (23) is true. Then for any $\mathbf{y} \in \mathcal{Y}$,

$$\begin{aligned} P_{Q_R}(\mathbf{y}) &= \prod_{t=1}^{|\mathbf{y}|} P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}) \\ &= \exp\left(\frac{\sum_{t=1}^{|\mathbf{y}|} Q_R(\mathbf{y}_1^{t-1}, y_t) - V_R(\mathbf{y}_1^{t-1})}{\tau}\right) \\ &= \exp\left(\frac{\sum_{t=1}^{|\mathbf{y}|} \left[R(\mathbf{y}_1^t) - R(\mathbf{y}_1^{t-1})\right] + \sum_{t=1}^{|\mathbf{y}|-1} V_R(\mathbf{y}_1^t) - \sum_{t=1}^{|\mathbf{y}|} V_R(\mathbf{y}_1^{t-1})}{\tau}\right) \\ &= \exp\left(\frac{R(\mathbf{y}) - V_R(\emptyset)}{\tau}\right) \end{aligned}$$

where $V_R(\emptyset)$ denotes $V_R(\mathbf{y}_1^t)$ when $t = 0$ and $\mathbf{y}_1^t$ is an empty set. Since $P_{Q_R}(\mathbf{y})$ is a valid distribution by construction, we have

$$V_R(\emptyset) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp\left(\frac{R(\mathbf{y})}{\tau}\right)$$

Hence,

$$P_{Q_R}(\mathbf{y}) = \frac{R(\mathbf{y})/\tau}{\sum_{\mathbf{y}' \in \mathcal{Y}} R(\mathbf{y}')/\tau} = P_R(\mathbf{y}),$$

which satisfies the marginal match requirement.

Necessity Now, we show that the specific formulation of Q_R (Eqn. (23)) is also a necessary condition of the marginal match condition (Eqn. (21)).

The token-level target distribution can be simplified as

$$P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1}) = \frac{\exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^{t-1}, w)/\tau\right)} = \exp\left(\frac{Q_R(\mathbf{y}_1^{t-1}, y_t) - V_R(\mathbf{y}_1^{t-1})}{\tau}\right).$$

Suppose Eqn. (21) is true. For any $\mathbf{y} \in \mathcal{Y}^-$ and $t \le |\mathbf{y}|$ and define $\mathbf{y}' = \mathbf{y}_1^t + \texttt{eos}$ and $\mathbf{y}'' = \mathbf{y}_1^{t-1} + \texttt{eos}$. Obviously, it follows $\mathbf{y}', \mathbf{y}'' \in \mathcal{Y}$. Also, by definition,

$$\begin{aligned} P_R(\mathbf{y}') &= P_R(\texttt{eos} \mid \mathbf{y}_1^t) \times P_R(y_t \mid \mathbf{y}_1^{t-1}) \times P_R(\mathbf{y}_1^{t-1}) \\ P_R(\mathbf{y}'') &= P_R(\texttt{eos} \mid \mathbf{y}_1^{t-1}) \times P_R(\mathbf{y}_1^{t-1}) \end{aligned}$$

Then, consider the ratio

$$\begin{aligned} \frac{P_R(\mathbf{y}')}{P_R(\mathbf{y}'')} &= \frac{P_R(\texttt{eos} \mid \mathbf{y}_1^t) \times P_R(y_t \mid \mathbf{y}_1^{t-1}) \times \cancel{P_R(\mathbf{y}_1^{t-1})}}{P_R(\texttt{eos} \mid \mathbf{y}_1^{t-1}) \times \cancel{P_R(\mathbf{y}_1^{t-1})}} \\ \exp\left(\frac{R(\mathbf{y}') - R(\mathbf{y}'')}{\tau}\right) &= \exp\left(\frac{Q_R(\mathbf{y}_1^t, \texttt{eos}) - V_R(\mathbf{y}_1^t)}{\tau}\right) \times \exp\left(\frac{Q_R(\mathbf{y}_1^{t-1}, y_t) - \cancel{V_R(\mathbf{y}_1^{t-1})}}{\tau}\right) \\ &\Big/ \exp\left(\frac{Q_R(\mathbf{y}_1^{t-1}, \texttt{eos}) - \cancel{V_R(\mathbf{y}_1^{t-1})}}{\tau}\right) \\ R(\mathbf{y}') - R(\mathbf{y}'') &= Q_R(\mathbf{y}_1^t, \texttt{eos}) - Q_R(\mathbf{y}_1^{t-1}, \texttt{eos}) - V_R(\mathbf{y}_1^t) + Q_R(\mathbf{y}_1^{t-1}, y_t). \end{aligned}$$

Now, by the terminal condition (Eqn. (22)), we essentially have

$$\begin{aligned} Q_R(\mathbf{y}_1^t, \texttt{eos}) &= R(\mathbf{y}_1^t + \texttt{eos}) - R(\mathbf{y}_1^t) = 0 \\ Q_R(\mathbf{y}_1^{t-1}, \texttt{eos}) &= R(\mathbf{y}_1^{t-1} + \texttt{eos}) - R(\mathbf{y}_1^{t-1}) = 0 \end{aligned}$$

Thus, it follows

$$\begin{aligned} & R(\mathbf{y}') - R(\mathbf{y}'') = Q_R(\mathbf{y}_1^{t-1}, y_t) - V_R(\mathbf{y}_1^t) \\ \iff & Q_R(\mathbf{y}_1^{t-1}, y_t) = R(\mathbf{y}_1^t) - R(\mathbf{y}_1^{t-1}) + \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^t, w)/\tau\right), \end{aligned}$$

which completes the proof. □

Corollary 1. *Please refer to §3.2 for the Corollary.*

Proof. Similarly, we drop the dependency on $\mathbf{x}^*$ and $\mathbf{y}^*$ to avoid clutter. We first prove the equivalence of $Q^*(\mathbf{y}_1^{t-1}, y_t)$ with $Q_R(\mathbf{y}_1^{t-1}, y_t)$ by induction.

- **Base case**: When $t = T$, for any $\mathbf{y} \in \mathcal{Y}$, y_T can only be $\texttt{eos}$. So, by definition, we have

  $$\begin{aligned} & V^*(\mathbf{y}_1^{T-1}) = Q^*(\mathbf{y}_1^{T-1}, \texttt{eos}) \\ \iff & \tau \log \sum_{a \in \mathcal{W}} \exp\left(Q^*(\mathbf{y}_1^{T-1}, a)/\tau\right) = Q^*(\mathbf{y}_1^{T-1}, \texttt{eos}) \\ \implies & Q^*(\mathbf{y}_1^{T-1}, a) = -\infty, \forall a \neq \texttt{eos}. \end{aligned}$$

  Hence,

  $$Q^*(\mathbf{y}_1^{T-1}, y_T) = \begin{cases} r(\mathbf{y}_1^{T-1}, \texttt{eos}), & \text{if } y_T = \texttt{eos} \\ -\infty, & \text{otherwise} \end{cases}$$

  For the first case, it directly follows

  $$Q^*(\mathbf{y}_1^{T-1}, \texttt{eos}) = r(\mathbf{y}_1^{T-1}, \texttt{eos}) = R(\mathbf{y}_1^{T-1} + \texttt{eos}) - R(\mathbf{y}_1^{T-1}) = Q_R(\mathbf{y}_1^{T-1}, \texttt{eos}).$$

  For the second case, since only $\texttt{eos}$ is allowed to be generated, the target distribution P_{Q_R} should be a single-point distribution at $\texttt{eos}$. This is equivalent to define

  $$Q_R(\mathbf{y}_1^{T-1}, a) = -\infty, \forall a \neq \texttt{eos},$$

  which proves the second case. Combining the two cases, it concludes

  $$Q^*(\mathbf{y}_1^{T-1}, a) = Q_R(\mathbf{y}_1^{T-1}, a), \forall \mathbf{y} \in \mathcal{Y}, a \in \mathcal{W}.$$

- **Induction step**: When $0 < t < T$, assume the equivalence holds when $k > t$, i.e.,

$$Q^*(\mathbf{y}_1^{k-1}, w) = Q_R(\mathbf{y}_1^{k-1}, w), \forall k > t, w \in \mathcal{W}.$$

  Then,

$$\begin{aligned}
Q^*(\mathbf{y}_1^{t-1}, y_t) &= r(\mathbf{y}_1^{t-1}, y_t) + \gamma \mathop{\mathbb{E}}_{s' \sim \rho_s}[\alpha \log \sum_{a \in \mathcal{A}} \exp\left(Q^*(s', a)/\alpha\right)] \\
&= r(\mathbf{y}_1^{t-1}, y_t) + \tau \log \sum_{a \in \mathcal{W}} \exp\left(Q^*(\mathbf{y}_1^{t}, a)/\tau\right) && (\alpha = \tau, \mathcal{A} = \mathcal{W}) \\
&= r(\mathbf{y}_1^{t-1}, y_t) + \tau \log \sum_{a \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^{t}, a)/\tau\right) && (Q^*(\mathbf{y}_1^{k}, a) = Q_R(\mathbf{y}_1^{k}, a) \text{ for } k \geq t) \\
&= Q_R(\mathbf{y}_1^{t-1}, y_t).
\end{aligned}$$

Thus, $Q^*(\mathbf{y}_1^{t-1}, y_t) = Q_R(\mathbf{y}_1^{t-1}, y_t)$ holds for $t \in [1, T]$.

With the equivalence between Q_R and Q^*, we can easily prove $V^* = V_R$ and $\pi^* = P_{Q_R}$,

$$\begin{aligned}
V^*(\mathbf{y}_1^{t-1}) &= \alpha \log \sum_{a \in \mathcal{A}} \exp\left(Q^*(\mathbf{y}_1^{t-1}, a)/\alpha\right) \\
&= \tau \log \sum_{a \in \mathcal{W}} \exp\left(Q^*(\mathbf{y}_1^{t-1}, a)/\tau\right) && (\alpha = \tau, \mathcal{A} = \mathcal{W}) \\
&= V_R(\mathbf{y}_1^{t-1}) \\
\pi^*(y_t \mid \mathbf{y}_1^{t-1}) &= \frac{\exp\left(Q^*(\mathbf{y}_1^{t-1}, y_t)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q^*(\mathbf{y}_1^{t-1}, y_t)/\tau\right)} \\
&= \frac{\exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t)/\tau\right)}{\sum_{w \in \mathcal{W}} \exp\left(Q_R(\mathbf{y}_1^{t-1}, y_t)/\tau\right)} \\
&= P_{Q_R}(y_t \mid \mathbf{y}_1^{t-1})
\end{aligned}$$

□

A.2 Other Proofs

We derive the equivalence between the VAML's objective (Eqn. (17)) and the RAML's objective (Eqn. (2)).

$$\begin{aligned}
&\operatorname{CE}\left(P_{Q_\phi} \| P_\theta\right) \\
=& - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \log P_\theta(\mathbf{y}) \\
=& - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \sum_{t=1}^{|\mathbf{y}|} \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) \\
=& - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t} \sim P_{Q_\phi}(Y_1^t)} \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) && (T \text{ is longest possible length}) \\
=& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \left[- \mathop{\mathbb{E}}_{y_t \sim P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1})} \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) \right] \\
=& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \operatorname{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \\
=& \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \sum_{y_t \in \mathcal{W}} P_{Q_\phi}(y_t \mid \mathbf{y}_1^{t-1}) \underbrace{\operatorname{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right)}_{\text{const. w.r.t. } y_t}
\end{aligned}$$

$$= \sum_{t=1}^{T} \underbrace{\mathop{\mathbb{E}}_{\mathbf{y}_1^{t-1} \sim P_{Q_\phi}(\mathbf{Y}_1^{t-1})} \mathop{\mathbb{E}}_{y_t \in P_{Q_\phi}(W|\mathbf{y}_1^{t-1})}}_{\mathbb{E}_{\mathbf{y}_1^t \sim P_{Q_\phi}(\mathbf{Y}_1^t)}} \left[\mathrm{CE} \left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1}) \right) \right]$$

$$= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{y}_1^t \sim P_{Q_\phi}(\mathbf{Y}_1^t)} \left[\mathrm{CE} \left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1}) \right) \right]$$

$$= \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}(\mathbf{Y})} \sum_{t=1}^{|\mathbf{y}|} \mathrm{CE} \left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1}) \right)$$

B Implementation Details

B.1 RAML

In RAML, we want to optimize the cross entropy $\mathrm{CE}\left(P_R(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right)$. As discussed in §2.1, directly sampling from the exponentiated pay-off distribution $P_R(Y \mid x^*)$ is impractical. Hence, normalized importance sampling has been exploited in previous work (Norouzi et al., 2016; Ma et al., 2017). Define the proposal distribution to be $P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$. Then, the objective can be rewritten as

$$\begin{aligned}
\mathrm{CE}\left(P_R(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right) &= - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_S(\mathbf{Y}|\mathbf{x}^*,\mathbf{y}^*)} \frac{P_R(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*)}{P_S(\mathbf{y} \mid \mathbf{x}^*, \mathbf{y}^*)} \log P_\theta(\mathbf{y} \mid \mathbf{x}^*) \\
&= - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_S(\mathbf{Y}|\mathbf{x}^*,\mathbf{y}^*)} \frac{\frac{\exp(R(\mathbf{y},\mathbf{y}^*)/\tau)}{\tilde{P}_S(\mathbf{y}|\mathbf{x}^*,\mathbf{y}^*)}}{\mathbb{E}_{\mathbf{y}' \sim P_S(\mathbf{Y}|\mathbf{x}^*,\mathbf{y}^*)} \frac{\exp(R(\mathbf{y}',\mathbf{y}^*)/\tau)}{\tilde{P}_S(\mathbf{y}'|\mathbf{x}^*,\mathbf{y}^*)}} \log P_\theta(\mathbf{y} \mid \mathbf{x}^*) \\
&= - \mathop{\mathbb{E}}_{\mathbf{y} \sim P_S(\mathbf{Y}|\mathbf{x}^*,\mathbf{y}^*)} \frac{w(\mathbf{y}, \mathbf{y}^*)}{\mathbb{E}_{\mathbf{y}' \sim P_S(\mathbf{Y}|\mathbf{x}^*,\mathbf{y}^*)} w(\mathbf{y}', \mathbf{y}^*)} \log P_\theta(\mathbf{y} \mid \mathbf{x}^*) \\
&\approx - \sum_{i=1}^{M} \frac{w(\mathbf{y}^{(i)}, \mathbf{y}^*)}{\sum_{i=1}^{M} w(\mathbf{y}^{(i)}, \mathbf{y}^*)} \log P_\theta(\mathbf{y}^{(i)} \mid \mathbf{x}^*),
\end{aligned}$$

where $w(\mathbf{y}, \mathbf{y}^*) = \frac{\exp(R(\mathbf{y},\mathbf{y}^*)/\tau)}{\tilde{P}_S(\mathbf{y}|\mathbf{x}^*,\mathbf{y}^*)}$ is the unnormalized importance weight, $\tilde{P}_S$ denotes the unnormalized probability of $P_S = \frac{\tilde{P}_S}{Z}$, M is the number of samples used, and $\mathbf{y}^{(i)}$ is the i-th sample drawn from the proposal distribution $P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$.

With importance sampling, the problem turns to what proposal distribution we should use. In the original work (Norouzi et al., 2016), the proposal distribution is defined by the hamming distance as used. Ma et al. (2017) find that it suffices to perform N-gram replacement of the reference sentence. Specifically, $P_S(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$ can be a uniform distribution defined on set $\mathcal{Y}_{\text{ngram}}$ where $\mathcal{Y}_{\text{ngram}}$ is obtained by randomly replacing an n-gram of $\mathbf{y}^*$ ($n \leq 4$).

In this work, we adapt the simple n-gram replacement distribution, denoted as $P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$, which simplifies the RAML objective into

$$\min_\theta - \sum_{i=1}^{M} \frac{\exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M} \exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)} \log P_\theta(\mathbf{y}^{(i)} \mid \mathbf{x}^*)$$

Following Ma et al. (2017), we make sure the reference sequence is always among the M samples used.

B.2 VAML

As discussed in §4, the VAML training consists of two phases:

- In the first phase, Soft Q-Learning is used to train Q_ϕ based on Eqn. (16). Since Soft Q-Learning accepts off-policy trajectories, in this work, we use two types of off-policy sequences:
  1. The first type is simply the ground-truth sequence, which provides strong learning signals.

2. The second type of sequences is actually drawn from the same n-gram replacement distribution discussed above. The reason is that in the second training phase, such n-gram replaced trajectories will be used. Since the learned Q_ϕ won't be perfect, we hope the exposing Q_ϕ with these trajectories can improve its accuracy on them, making the second phase of training easier.

Algorithm 1 summarizes the first phase.

Algorithm 1 VAML Phase 1: Soft Q-Learning to approximate Q^*

Require: A Q-function approximator Q_ϕ with parameter ϕ, and the hyper-parameters τ, M.

1: **while** Not Converged **do**

2: Receive a random example $(\mathbf{x}^*, \mathbf{y}^*)$.

3: Sample $M-1$ sequences $\{\mathbf{y}^{(i)}\}_{i=1}^{M-1}$ from $P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$ and let $\mathbf{y}^{(M)} = \mathbf{y}^*$.

4: Compute all the rewards $r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ for each $\mathbf{y} \in \{\mathbf{y}^{(i)}\}_{i=1}^{M}$ and $t = 1, \ldots, |\mathbf{y}|$.

5: Compute the target Q-values for each $\mathbf{y} \in \{\mathbf{y}^{(i)}\}_{i=1}^{M}$ and $t = 1, \ldots, |\mathbf{y}|$

$$\hat{Q}_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) + \tau \log \sum_{w \in \mathcal{W}} \exp\left(Q_\phi(\mathbf{y}_1^{t}, w; \mathbf{y}^*)/\tau\right).$$

6: Compute the Soft-Q Learning loss

$$\mathcal{L}_{\text{SoftQ}} = \frac{1}{M} \sum_{i=1}^{M} \sum_{t=1}^{|\mathbf{y}^{(i)}|} \left\| Q_\phi({\mathbf{y}^{(i)}}_1^{t-1}, y_t^{(i)}; \mathbf{y}^*) - \hat{Q}_\phi({\mathbf{y}^{(i)}}_1^{t-1}, y_t^{(i)}; \mathbf{y}^*) \right\|_2^2.$$

7: Update Q_ϕ according to the loss $\mathcal{L}_{\text{SoftQ}}$.

8: **end while**

- Once the Q_ϕ is well trained in the first phase, the second phase is to minimize the cross entropy $\text{CE}\left(P_{Q_\phi}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*) \| P_\theta(\mathbf{Y} \mid \mathbf{x}^*)\right)$ based on Eqn. (17), i.e.,

$$\min_\theta \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \left[\sum_{t=1}^{|\mathbf{y}|} \text{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \right].$$

Ideally, we would like to directly sample from P_{Q_ϕ}, and perform the optimization. However, we find samples from P_{Q_ϕ} are quite similar to each other. We conjecture this results from both the imperfect training in the first phase, and the intrinsic difficulty of getting diverse samples from an exponentially large space when the distribution is high concentrated.

Nevertheless, for this work, we fall back to the same importance sampling method as used in RAML and use the n-gram replacement distribution as the proposal. Hence, the objective becomes

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{y} \sim P_{Q_\phi}} \left[\sum_{t=1}^{|\mathbf{y}|} \text{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \right] \\
=& \mathop{\mathbb{E}}_{\mathbf{y} \sim P_{\text{ngram}}} \left[\frac{w(\mathbf{y}, \mathbf{y}^*)}{\mathbb{E}_{\mathbf{y}' \sim P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)}\, w(\mathbf{y}', \mathbf{y}^*)} \sum_{t=1}^{|\mathbf{y}|} \text{CE}\left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1})\right) \right] \\
\approx& \sum_{i=1}^{M} \frac{\exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M} \exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)} \left[\sum_{t=1}^{|\mathbf{y}^{(i)}|} \text{CE}\left(P_{Q_\phi}(Y_t \mid {\mathbf{y}^{(i)}}_1^{t-1}) \| P_\theta(Y_t \mid {\mathbf{y}^{(i)}}_1^{t-1})\right) \right].
\end{aligned}$$

However, we found directly using this objective does not yield improved performance compared to RAML, mostly likely due to some erratic estimations of Q_ϕ. Thus, we only use this objective for some step with certain probability $\kappa \in (0, 1)$, leaving others trained by MLE. Formally, define

$$\mathcal{J}_\kappa(\mathbf{y}_1^t) = \mathop{\mathbb{E}}_{z \sim \text{Bernoulli}(\kappa)} \left[z \text{CE} \left(P_{Q_\phi}(Y_t \mid \mathbf{y}_1^{t-1}) \| P_\theta(Y_t \mid \mathbf{y}_1^{t-1}) \right) - (1 - z) \log P_\theta(y_t \mid \mathbf{y}_1^{t-1}) \right],$$

the VAML objective practically used is

$$\min_\theta \sum_{i=1}^{M} \frac{\exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M} \exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)} \left[\sum_{t=1}^{|\mathbf{y}^{(i)}|} \mathcal{J}_\kappa({\mathbf{y}^{(i)}}_1^t) \right].$$

Algorithm 2 summarizes the second phase.

Algorithm 2 VAML Phase 2: Sequence model training with token-level target

Require: A sequence prediction model P_θ with parameter θ, a pre-trained Q-function approximator Q_ϕ, and hyper-parameters τ, M, κ

1: **while** Not Converged **do**
2: Receive a random example $(\mathbf{x}^*, \mathbf{y}^*)$.
3: Sample $M - 1$ sequences $\{\mathbf{y}^{(i)}\}_{i=1}^{M-1}$ from $P_{\text{ngram}}(\mathbf{Y} \mid \mathbf{x}^*, \mathbf{y}^*)$ and let $\mathbf{y}^{(M)} = \mathbf{y}^*$.
4: Compute the VAML loss using

$$\mathcal{L}_{\text{VAML}} = \sum_{i=1}^{M} \frac{\exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)}{\sum_{i=1}^{M} \exp\left(R(\mathbf{y}^{(i)}, \mathbf{y}^*)/\tau\right)} \left[\sum_{t=1}^{|\mathbf{y}^{(i)}|} \mathcal{J}_\kappa({\mathbf{y}^{(i)}}_1^t) \right].$$

5: Update P_θ according to the loss $\mathcal{L}_{\text{VAML}}$.
6: **end while**

B.3 ERAC

Following Bahdanau et al. (2016), we first pre-train the actor, then train the critic with the fixed actor and finally fine-tune them together. The specific procedure for training ERAC is

- Pre-training the actor using maximum likelihood training
- Pre-training the critic using Algorithm 3 with the actor fixed
- Fine-tuning both the actor and critic with Algorithm 3

B.4 Hyper-parameters

RAML & VAML The hyper-parameters for RAML and VAML training are summarized in Tab. 5. We set the gradient clipping value to 5.0 for both the Q-function approximator Q_ϕ and the sequence prediction model P_θ, except for the sequence prediction model in the captioning task where the gradient clipping value is set to 1.0.

AC & ERAC As described in §B.3, the training using AC and ERAC involves three phases. The hyper-parameters used for ERAC training in each phase are summarized in Table 6. In all phases, the learning rate is halved when there is no improvement on the validation set. We use the same hyper-parameters for AC training, except that the entropy regularization coefficient τ is 0. Similar to the VAML case, the gradient clipping value is set to 5.0 for both the actor and the critic, except that we set the gradient clipping value to 1.0 for the actor in the captioning task.

Algorithm 3 ERAC Algorithm

Require: A critic $Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ and an actor $\pi_\theta(w \mid \mathbf{y}_1^t)$ with weights ϕ and θ respectively, and hyper-parameters τ, β, λ_{var}, λ_{mle}

Initialize delayed target critic $Q_{\bar{\phi}}$ with the same weights: $\bar{\phi} = \phi$.

while Not Converged **do**

  Receive a random example $(\mathbf{x}^*, \mathbf{y}^*)$.

  Generate a sequence $\mathbf{y}$ from π_θ.

  Compute the rewards $r(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*)$ for $t = 1, \ldots, |\mathbf{y}|$.

  Compute targets for the critic

$$\hat{Q}_{\bar{\phi}}(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) = r(\mathbf{y}_1^{t-1}, y_t) + \tau\, \mathcal{H}(\pi_\theta(\cdot \mid \mathbf{y}_1^t)) + \sum_{w \in \mathcal{W}} \pi_\theta(w \mid \mathbf{y}_1^t) Q_{\bar{\phi}}(\mathbf{y}_1^t, w; \mathbf{y}^*).$$

  Compute loss for critic

$$\mathcal{L}_{\text{critic}} = \sum_{t=1}^{|\mathbf{y}|} \left[Q_\phi(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) - \hat{Q}_{\bar{\phi}}(\mathbf{y}_1^{t-1}, y_t; \mathbf{y}^*) \right]^2 + \lambda_{\text{var}} \sum_{w \in \mathcal{W}} \left[Q_\phi(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*) - \bar{Q}_\phi(\mathbf{y}_1^{t-1}; \mathbf{y}^*) \right]^2,$$

$$\text{where} \quad \bar{Q}_\phi(\mathbf{y}_1^{t-1}; \mathbf{y}^*) = \frac{1}{|\mathcal{W}|} \sum_{w' \in \mathcal{W}} Q_\phi(\mathbf{y}_1^{t-1}, w'; \mathbf{y}^*)$$

  Compute loss for actor

$$\mathcal{L}_{\text{actor}} = -\left[\sum_{t=1}^{|\mathbf{y}|} \sum_{w \in \mathcal{W}} \pi_\theta(w \mid \mathbf{y}_1^{t-1}) Q_\phi(\mathbf{y}_1^{t-1}, w; \mathbf{y}^*) + \tau \mathcal{H}(\pi_\theta(\cdot \mid \mathbf{y}_1^{t-1})) + \lambda_{\text{mle}} \sum_{t=1}^{|\mathbf{y}^*|} \log \pi_\theta(y_t^* \mid {\mathbf{y}^*}_1^{t-1}) \right]$$

  Update critic according to the loss $\mathcal{L}_{\text{critic}}$.

  If actor is not fixed, update actor according to the loss $\mathcal{L}_{\text{actor}}$

  Update delayed target critic: $\bar{\phi} = \beta\phi + (1 - \beta)\bar{\phi}$

end while

	Machine Translation			Image Captioning		
Hyper-parameters	VAML-1	VAML-2	RAML	VAML-1	VAML-2	RAML
optimizer	Adam	SGD	SGD	Adam	SGD	SGD
learning rate	0.001	0.6	0.6	0.001	0.5	0.5
batch size	50	42	42	32×5	32×5	32×5
M	5	5	5	2	6	6
τ	0.4	0.4	0.4	0.7	0.7	0.7
κ	N.A.	0.2	N.A.	N.A.	0.1	N.A.

Table 5: Optimization related hyper-parameters of RAML and VAML for two tasks. "VAML-1" and "VAML-2" indicate the phase 1 and phase 2 of VAML training respectively. "N.A." means not applicable. "32×5" indicates using 32 images each with 5 reference captions.

C Comparison with Previous Work

The detailed comparison with previous work in shown in Table 7. Under different comparable architectures (1 layer or 2 layers), ERAC outperforms previous algorithms with a clear margin.

Hyper-parameters	**MT w/ input feeding**	**MT w/o input feeding**	**Image Captioning**
	Pre-train Actor		
optimizer	SGD	SGD	SGD
learning rate	0.6	0.6	0.5
batch size	50	50	32×5
	Pre-train Critic		
optimizer	Adam	Adam	Adam
learning rate	0.001	0.001	0.001
batch size	50	50	32×5
τ (entropy regularization)	0.045	0.04	0.01
β (target net speed)	0.001	0.001	0.001
λ_{var} (smoothness)	0.001	0.001	0.001
	Joint Training		
optimizer	Adam	Adam	Adam
learning rate	0.0001	0.0001	0.0001
batch size	50	50	32×5
τ (entropy regularization)	0.045	0.04	0.01
β (target net speed)	0.001	0.001	0.001
λ_{var} (smoothness)	0.001	0.001	0.001
λ_{MLE}	0.1	0.1	0.1

Table 6: Hyper-parameters for ERAC training

Algorithm	**Encoder**		**Decoder**				**BLEU**
	NN Type	Size	NN Type	Size	Attention	Input Feed	
MIXER (Ranzato et al., 2015)	1-layer CNN	256	1-layer LSTM	256	Dot-Prod	N	20.73
BSO (Wiseman and Rush, 2016)	1-layer BiLSTM	128×2	1-layer LSTM	256	Dot-Prod	Y	27.9
Q(BLEU) (Li et al., 2017)	1-layer BiLSTM	128×2	1-layer LSTM	256	Dot-Prod	Y	28.3
AC (Bahdanau et al., 2016)	1-layer BiGRU	256×2	1-layer GRU	256	MLP	Y	28.53
RAML (Ma et al., 2017)	1-layer BiLSTM	256×2	1-layer LSTM	256	Dot-Prod	Y	28.77
VAML	1-layer BiLSTM	128×2	1-layer LSTM	256	Dot-Prod	Y	28.94
ERAC	1-layer BiLSTM	128×2	1-layer LSTM	256	Dot-Prod	Y	**29.36**
NPMT (Huang et al., 2017)	2-layer BiGRU	256×2	2-layer LSTM	512	N.A.	N.A.	29.92
NPMT+LM (Huang et al., 2017)	2-layer BiGRU	256×2	2-layer LSTM	512	N.A.	N.A.	30.08
ERAC	2-layer BiLSTM	256×2	2-layer LSTM	512	Dot-Prod	Y	**30.85**

Table 7: Comparison of algorithms with detailed architecture information on the IWSTL 2014 dataset for MT.