

## A Proofs

### A.1 Proof of Theorem 1

*Proof.* Suppose among the  $K$  classes, exactly  $k$  have the influential logit  $z^*$  and the remaining  $K - k$  have the common secondary (singular) logit  $Z$ , where  $\mathbf{z} = f_\theta(\mathbf{x})$ . The energy is defined by

$$E_\theta(\mathbf{z}) = -\log(ke^{z^*} + (K - k)e^Z), \quad (1)$$

so that  $e^{-E_\theta} = ke^{z^*} + (K - k)e^Z$ . Writing  $C(k) = (K - k)e^Z$  gives  $ke^{z^*} = e^{-E_\theta} - C(k)$ .

Under the Softmax function, the probability of each of the  $k$  classes with logit  $z^*$  is:

$$p^* = \frac{e^{z^*}}{ke^{z^*} + C(k)} = \frac{e^{z^*}}{e^{-E_\theta}} = \frac{e^{-E_\theta} - C(k)}{ke^{-E_\theta}} = \frac{1 - C(k)e^{E_\theta}}{k}, \quad (2)$$

and the probability of each of the  $K - k$  classes with logit  $Z$  is:

$$p_Z = \frac{e^Z}{ke^{z^*} + C(k)} = \frac{e^Z}{e^{-E_\theta}} = e^{Z+E_\theta}. \quad (3)$$

One checks immediately that  $kp^* + (K - k)p_Z = 1$ , so Softmax  $\mathbf{p}$  is a valid probability distribution.

Now, the non-negative Shannon entropy of  $\mathbf{p}$  is

$$H(\mathbf{p}) = -\sum_{j=1}^K p_j \log p_j = \underbrace{-kp^* \log p^*}_{\text{Primary term}} - \underbrace{(K - k)p_Z \log p_Z}_{\text{Secondary term}}. \quad (4)$$

Substituting the expressions above yields

$$\begin{aligned} \underbrace{-kp^* \log p^*}_{\text{Primary term}} &= -(1 - C(k)e^{E_\theta}) \log \left( \frac{1 - C(k)e^{E_\theta}}{k} \right), \\ \underbrace{-(K - k)p_Z \log p_Z}_{\text{Secondary term}} &= -C(k)e^{E_\theta} \log \left( \frac{e^Z}{e^{-E_\theta}} \right) \\ &= -C(k)e^{E_\theta} (Z + E_\theta). \end{aligned} \quad (5)$$

Combining these two contributions reproduces the asserted closed-form relationship,

$$H(E_\theta) = -(1 - C(k)e^{E_\theta}) \log \left( \frac{1 - C(k)e^{E_\theta}}{k} \right) - C(k)e^{E_\theta} (Z + E_\theta). \quad (6)$$

This completes the proof of Theorem 1.  $\square$

### A.2 Proof of Lemma 1

*Proof.* Let  $E_\theta(\cdot) : \mathbb{R}^D \mapsto \mathbb{R}$  be the energy function defined by  $E_\theta(\mathbf{z}) = -\log \sum_{k=1}^K \exp(z_k)$ , where  $\mathbf{z} = f_\theta(\mathbf{x})$ . Its concave conjugate  $E_\theta^*(\mathbf{g})$  is  $E_\theta^*(\mathbf{g}) = \min_{\mathbf{z}} \{\mathbf{g}^T \mathbf{z} - E_\theta(\mathbf{z})\}$ , subject to the domain constraints that arise from the definition of the conjugate.

Because  $E_\theta$  is differentiable and concave, we can introduce a Lagrange multiplier  $\eta$  to enforce the implicit affine constraint on  $\mathbf{g}$ . Recall that for the gradient of energy with respect to the logit  $\mathbf{z}$ ,  $\nabla_{\mathbf{z}} E_\theta(\mathbf{z}) = -\mathbf{p}(\mathbf{x})$ , so any admissible  $\mathbf{g}$  must satisfy  $\sum_{k=1}^K g_k = -1$  and  $g_k \leq 0$  for all  $k$ . We therefore introduce the Lagrangian:

$$L(\mathbf{z}, \eta) = \underbrace{\sum_{k=1}^K g_k z_k}_{\text{Product term}} + \underbrace{\log \sum_{k=1}^K \exp(z_k)}_{\text{Primal term}} - \underbrace{\eta \left( \sum_{k=1}^K g_k + 1 \right)}_{\text{Multiplier term}}, \quad (7)$$

Setting the partial derivative of  $L$  with respect to each  $z_k$  to zero yields the first-order condition:

$$\frac{\partial L}{\partial z_k} = g_k + \frac{\exp(z_k)}{\sum_{j=1}^K \exp(z_j)} = 0 \quad \Rightarrow \quad g_k = -\frac{\exp(z_k)}{\sum_{j=1}^K \exp(z_j)}. \quad (8)$$

Hence,  $\exp(z_k) = -g_k \sum_{k=1}^K \exp(z_k)$ , and by defining  $\mathcal{C} = \sum_{k=1}^K \exp(z_k)$ , we have

$$z_k = \log(-g_k) + \log \mathcal{C}, \quad \forall k. \quad (9)$$

Substitute  $z_k = \log(-g_k) + \log \mathcal{C}$  back into  $L$ . For the first term, we observe

$$\begin{aligned} \underbrace{\sum_{k=1}^K g_k z_k}_{\text{Product term}} &= \sum_{k=1}^K g_k [\log(-g_k) + \log \mathcal{C}] \\ &= \sum_{k=1}^K g_k \log(-g_k) + \left( \sum_{k=1}^K g_k \right) \log \mathcal{C} \\ &= \sum_{k=1}^K g_k \log(-g_k) - \log \mathcal{C}, \end{aligned} \quad (10)$$

because of  $\sum_{k=1}^K g_k = -1$ . For the second term, we observe

$$\begin{aligned} \underbrace{\log \sum_{k=1}^K \exp(z_k)}_{\text{Primal term}} &= \log \sum_{k=1}^K (-g_k) \mathcal{C} \\ &= \log \mathcal{C} + \log \sum_{k=1}^K (-g_k) \\ &= \log \mathcal{C}, \end{aligned} \quad (11)$$

because of  $\sum_{k=1}^K (-g_k) = 1$ . Finally, the Lagrange multiplier term vanishes at feasibility  $\eta(\sum_k g_k + 1) = 0$ . Therefore, we have

$$L(\mathbf{z}, \eta) = \sum_{k=1}^K g_k \log(-g_k) - \log \mathcal{C} + \log \mathcal{C} = \sum_{k=1}^K g_k \log(-g_k). \quad (12)$$

Thus, the minimum value of  $\mathbf{g}^T \mathbf{z} - E_\theta(\mathbf{z})$  is  $E_\theta^*(\mathbf{g}) = \sum_{k=1}^K g_k \log(-g_k)$ , subject to  $\sum_{k=1}^K g_k = -1$  and  $g_k \leq 0$ . Setting  $p(y = k|\mathbf{x}; \theta) = -g_k$  (so,  $p(y = k|\mathbf{x}; \theta) \geq 0$ ,  $\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) = 1$ ). Then, we can write:

$$\begin{aligned} E_\theta^*(\mathbf{g}) &= \sum_{k=1}^K (-p(y = k|\mathbf{x}; \theta)) \log p(y = k|\mathbf{x}; \theta) \\ &= - \sum_{k=1}^K p(y = k|\mathbf{x}; \theta) \log p(y = k|\mathbf{x}; \theta) \\ &= -\mathbf{p}(\mathbf{x})^T \log \mathbf{p}(\mathbf{x}) = H(\mathbf{p}), \end{aligned} \quad (13)$$

the negative entropy of Softmax  $\mathbf{p}$ . From the stationarity condition, we have  $\mathbf{g} = \nabla_{\mathbf{z}} E_\theta(\mathbf{z}) = -\mathbf{p}(\mathbf{x})$ . That is,  $\mathbf{g}$  is exactly the negative Softmax of the logits. This completes the proof of Lemma 1.  $\square$

### A.3 Proof of Lemma 2

*Proof.* Recall from Lemma 1 that for the energy  $E_\theta(\mathbf{z}) = -\log \sum_{k=1}^K \exp(z_k)$ , where  $\mathbf{z} = f_\theta(\mathbf{x})$ . We have its concave conjugate  $E_\theta^*(\mathbf{g}) = \sum_{k=1}^K g_k \log(-g_k)$ , subject to  $\sum_k g_k = -1$ ,  $g_k \leq 0$ .

Equivalently, setting  $p(y = k|\mathbf{x}; \theta) = -g_k$ , one sees  $E_\theta^*(\mathbf{g}) = -\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) \log p(y = k|\mathbf{x}; \theta)$ . The Fenchel-Moreau theorem (for closed, concave functions) then guarantees bi-duality:

$$E_\theta(\mathbf{z}) = \min_{\mathbf{g}} \{\mathbf{g}^T \mathbf{z} - E_\theta^*(\mathbf{g})\}, \quad (14)$$

with no duality gap, Substituting  $\mathbf{g} = -\mathbf{p}$  and  $E_\theta^*(\mathbf{g}) = H(\mathbf{p})$  gives exactly

$$E_\theta(\mathbf{z}) = \min_{\mathbf{p}} \{-\mathbf{p}^T \mathbf{z} - H(\mathbf{p})\}, \quad (15)$$

which is the claimed relation, Fenchel duality.

To see this more concretely, consider the following minimization problem given by:

$$\begin{aligned} \min_{\mathbf{p} \in \mathbb{R}^K} F(\mathbf{p}) &= -\mathbf{p}^T \mathbf{z} + \sum_{k=1}^K p(y = k|\mathbf{x}; \theta) \log p(y = k|\mathbf{x}; \theta), \\ \text{s.t.} \quad &\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) = 1, \\ &p(y = k|\mathbf{x}; \theta) \geq 0. \end{aligned} \quad (16)$$

Introduce a Lagrange multiplier  $\eta$  for the simplex constraint. The Lagrangian is

$$L(\mathbf{p}, \eta) = -\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) z_k + \sum_{k=1}^K p(y = k|\mathbf{x}; \theta) \log p(y = k|\mathbf{x}; \theta) - \eta \left( \sum_{k=1}^K p(y = k|\mathbf{x}; \theta) - 1 \right), \quad (17)$$

Setting the partial derivative of  $L$  with respect to each  $p(y = k|\mathbf{x}; \theta)$  to zero yields the first-order condition:

$$\frac{\partial L}{\partial p(y = k|\mathbf{x}; \theta)} = -z_k + (1 + \log p(y = k|\mathbf{x}; \theta)) - \eta = 0 \Rightarrow \log p(y = k|\mathbf{x}; \theta) = z_k - (1 - \eta). \quad (18)$$

Then, exponentiation gives

$$p(y = k|\mathbf{x}; \theta) = \exp(z_k) \exp(-(1 - \eta)) \propto \exp(z_k). \quad (19)$$

Enforcing  $\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) = 1$  fixes the constant. Therefore, Eq. 19 recovers the Softmax:

$$p(y = k|\mathbf{x}; \theta) = \frac{\exp(z_k)}{\sum_{j=1}^K \exp(z_j)}. \quad (20)$$

Plug Eq. 20 into the objective Eq. 16:

$$F(\mathbf{p}) = \underbrace{-\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) z_k}_{\text{First term}} + \underbrace{\sum_{k=1}^K p(y = k|\mathbf{x}; \theta) \log p(y = k|\mathbf{x}; \theta)}_{\text{Second term}}. \quad (21)$$

Substituting  $p(y = k|\mathbf{x}; \theta) = \exp(z_k)/\mathcal{C}$ . Then, the first term is  $-\sum (\exp(z_k)/\mathcal{C}) z_k$  and the second term is  $\sum_k (\exp(z_k)/\mathcal{C}) (z_k - \log \mathcal{C})$ . Now, combine terms as follows:

$$\begin{aligned} F(\mathbf{p}) &= -\frac{1}{\mathcal{C}} \sum_{k=1}^K \exp(z_k) z_k + \frac{1}{\mathcal{C}} \sum_{k=1}^K \exp(z_k) z_k - \frac{\log \mathcal{C}}{\mathcal{C}} \sum_{k=1}^K \exp(z_k) \\ &= \left( -\frac{1}{\mathcal{C}} \sum_{k=1}^K \exp(z_k) z_k + \frac{1}{\mathcal{C}} \sum_{k=1}^K \exp(z_k) z_k \right) - \log \mathcal{C} \\ &= -\log \mathcal{C}. \end{aligned} \quad (22)$$

The two  $z_k$ -terms cancel, leaving  $-\log \mathcal{C}$ . Therefore, we have

$$-\log \mathcal{C} = -\log \sum_{k=1}^K \exp(z_k) = E_{\theta}(\mathbf{z}). \quad (23)$$

Thus, the minimization of  $F(\mathbf{p})$  yields  $E_{\theta}(\mathbf{z})$ . This completes the proof of Lemma 2.  $\square$

#### A.4 Proof of Lemma 3

*Proof.* We approximate the intractable term  $\mathbb{E}_{p_{\theta}}[\nabla_{\theta} E_{\theta}(\mathbf{x}_t)]$  by a single Langevin step starting from true samples  $\mathbf{x}_t \sim p_t$ . Recall that a one-step Langevin update is given by:

$$\mathbf{x}'_t = \mathbf{x}_t - \frac{\mu^2}{2} \nabla_{\mathbf{x}_t} E_{\theta}(\mathbf{x}_t) + \mu \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D). \quad (24)$$

Define  $\nabla_\theta E_\theta(\mathbf{x}'_t)$ . So, we have:

$$\mathbf{x}'_t = \mathbf{x}_t + \delta, \quad \delta = -\frac{\mu^2}{2} \nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t) + \mu \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_D) \quad (25)$$

where  $\delta \in \mathbb{R}^D$  is the deviation of  $\mathbf{x}_t$ . By the multivariate Taylor formula around  $\mathbf{x}_t$ :

$$\nabla_\theta E_\theta(\mathbf{x}_t + \delta) = \nabla_\theta E_\theta(\mathbf{x}_t) + \underbrace{[\nabla_{\mathbf{x}_t} \nabla_\theta E_\theta(\mathbf{x}_t)] \delta}_{\text{linear term}} + \underbrace{\frac{1}{2} \delta^T [\nabla_{\mathbf{x}_t}^2 \nabla_\theta E_\theta(\mathbf{x}_t)] \delta}_{\text{quadratic term}} + \dots \quad (26)$$

Taking the expectation over the Gaussian noise  $\epsilon$ , the linear term is proportional to  $\delta$ . Since  $\delta$  contains  $\mu\epsilon$ , taking expectation over  $\epsilon$  can eliminate that piece:

$$\mathbb{E}_\epsilon[\delta] = -\frac{\mu^2}{2} \nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t) + \mu \mathbb{E}[\epsilon] = -\frac{\mu^2}{2} \nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t), \quad (27)$$

and  $\mathbb{E}[\epsilon] = 0$ . But this drift term, when multiplied by  $\nabla_{\mathbf{x}_t} \nabla_\theta E_\theta(\mathbf{x}_t)$ , contributes an  $o(\mu^2)$  piece which we will combine with the quadratic term.

The quadratic term has two pieces inside  $\delta$ . The drift piece  $-\frac{\mu^2}{2} \nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)$  interacting with itself gives  $o(\mu^4)$ , which we neglect. The cross-term between the two drift/noise pieces is  $o(\mu^3)$ , also negligible. The noise-noise piece  $\frac{1}{2}(\mu\epsilon)^T [\nabla_{\mathbf{x}_t}^2 \nabla_\theta E_\theta(\mathbf{x}_t)] (\mu\epsilon)$  has expectation as follows:

$$\frac{1}{2}(\mu\epsilon)^T [\nabla_{\mathbf{x}_t}^2 \nabla_\theta E_\theta(\mathbf{x}_t)] (\mu\epsilon) = \frac{\mu^2}{2} \text{Tr}[\nabla_{\mathbf{x}_t}^2 \nabla_\theta E_\theta(\mathbf{x}_t)], \quad \text{where } \mathbb{E}[\epsilon\epsilon^T] = \mathbf{I}. \quad (28)$$

Putting the linear term and quadratic term together, after taking  $\mathbb{E}_\epsilon$ , we obtain

$$\mathbb{E}_\epsilon[\nabla_\theta E_\theta(\mathbf{x}'_t)] = \nabla_\theta E_\theta(\mathbf{x}_t) + \frac{\mu^2}{2} \text{Tr}[\nabla_{\mathbf{x}_t}^2 \nabla_\theta E_\theta(\mathbf{x}_t)] + o(\mu^2). \quad (29)$$

Notice that  $\text{Tr}[\nabla_{\mathbf{x}_t}^2 \nabla_\theta E_\theta(\mathbf{x}_t)] = \nabla_\theta (\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t))$ , where  $\Delta_{\mathbf{x}_t} = \sum_{i=1}^D \partial^2 / \partial x_i^2$  is Laplacian operator. Therefore, we have

$$\mathbb{E}_\epsilon[\nabla_\theta E_\theta(\mathbf{x}'_t)] = \nabla_\theta E_\theta(\mathbf{x}_t) + \frac{\mu^2}{2} \nabla_\theta (\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)) + o(\mu^2). \quad (30)$$

Now, we take the outer expectation over true samples  $\mathbf{x}_t \sim p_t$ :

$$\mathbb{E}_{p_t} \mathbb{E}_\epsilon[\nabla_\theta E_\theta(\mathbf{x}'_t)] = \mathbb{E}_{p_t}[\nabla_\theta E_\theta(\mathbf{x}_t)] + \frac{\mu^2}{2} \nabla_\theta \mathbb{E}_{p_t}[\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)] + o(\mu^2). \quad (31)$$

The derivative of the expected energy-based log-density  $\log p_\theta(\mathbf{x}_t)$  by substituting Eq. 31 with intractable term  $\mathbb{E}_{p_\theta}[\nabla_\theta E_\theta(\mathbf{x}_t)]$  is then given by:

$$\begin{aligned} \nabla_\theta \mathbb{E}_{p_t}[\log p_\theta(\mathbf{x}_t)] &= \mathbb{E}_{p_t}[\nabla_\theta E_\theta(\mathbf{x}_t)] + \frac{\mu^2}{2} \nabla_\theta \mathbb{E}_{p_t}[\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)] + o(\mu^2) - \mathbb{E}_{p_t}[\nabla_\theta E_\theta(\mathbf{x}_t)] \\ &= \frac{\mu^2}{2} \nabla_\theta \underbrace{\mathbb{E}_{p_t}[\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)]}_{(*)} + o(\mu^2) \end{aligned} \quad (32)$$

For the expectation of the Laplacian of the energy  $(*)$ , we want

$$\mathbb{E}_{p_t}[\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x})] = \int p_t(\mathbf{x}_t) \Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t) d\mathbf{x}_t, \quad (33)$$

Using an integration-by-parts argument by using the *divergence theorem* (and assuming boundary terms vanish), for any scalar function  $p_t(\mathbf{x}_t)$  and vector field  $\nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)$ , we then have

$$\int p_t(\mathbf{x}_t) \Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t) d\mathbf{x}_t = \int p_t(\mathbf{x}_t) \nabla \cdot (\nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)) d\mathbf{x}_t \quad (34)$$

$$= - \int (\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)) \cdot (\nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)) d\mathbf{x}_t \quad (35)$$

$$= - \int p_t(\mathbf{x}_t) [\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)] d\mathbf{x}_t \quad (36)$$

$$= - \int p_t(\mathbf{x}_t) [\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot (-\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t))] d\mathbf{x}_t \quad (37)$$

$$= \int p_t(\mathbf{x}_t) [\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)] d\mathbf{x}_t, \quad (38)$$

where  $\nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t) = p_t(\mathbf{x}_t) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$  and  $\nabla_{\mathbf{x}_t} E_\theta(\mathbf{x}_t) = -\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)$ . Equivalently,

$$\mathbb{E}_{p_t}[\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)] = \mathbb{E}_{p_t}[\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)]. \quad (39)$$

Now, to expand Eq. 39 to the Fisher divergence  $D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t))$ :

$$\begin{aligned} D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t)) &= \mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)\|^2 \\ &= \mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|^2 + \mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)\|^2 \\ &\quad - 2\mathbb{E}_{p_t} [\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)]. \end{aligned} \quad (40)$$

Rearrange to isolate the cross-term:

$$\begin{aligned} \mathbb{E}_{p_t} [\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)] &= \frac{1}{2} (\mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|^2 + \mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)\|^2 \\ &\quad - D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t))) \end{aligned} \quad (41)$$

Since  $\mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\|^2$  does not depend on  $\theta$ , its gradient vanishes. Therefore, when we differentiate both sides with respect to  $\theta$ , only the last two terms remain:

$$\nabla_\theta \mathbb{E}_{p_t} [\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)] = \frac{1}{2} \nabla_\theta \mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)\|^2 - \frac{1}{2} \nabla_\theta D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t)) \quad (42)$$

Here, a second integration-by-parts shows  $\nabla_\theta \mathbb{E}_{p_t} \|\nabla_{\mathbf{x}_t} \log p_\theta(\mathbf{x}_t)\|^2 = \nabla_\theta \mathbb{E}_{p_t} [\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)]$ . Therefore, using Eq. 39 together, we have

$$\nabla_\theta \mathbb{E}_{p_t} [\Delta_{\mathbf{x}_t} E_\theta(\mathbf{x}_t)] = -\nabla_\theta D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t)). \quad (43)$$

Now, Eq. 32 via this exact relation with Taylor expansion gives:

$$\nabla_\theta \mathbb{E}_{p_t} [\log p_\theta(\mathbf{x}_t)] = \frac{\mu^2}{2} (-\nabla_\theta D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t))) + o(\mu^2). \quad (44)$$

The minus is naturally absorbed in the Fisher divergence. Finally, we have

$$\nabla_\theta \mathbb{E}_{p_t} [\log p_\theta(\mathbf{x}_t)] \simeq \frac{\mu^2}{2} \nabla_\theta D_F(p_t(\mathbf{x}_t) \| p_\theta(\mathbf{x}_t)) + o(\mu^2), \quad (45)$$

where  $p_\theta(\mathbf{x}_t) = \exp(-E_\theta(\mathbf{x}_t))/Z(\theta)$ . Here, the derivatives of  $\log Z(\theta)$  cancel out because they involve  $\mathbb{E}_{p_\theta}[1]$  and thus do not affect the score. This completes the proof of Lemma 3.  $\square$

## B Experiment Details

We evaluate ReTTA using publicly available pretrained models: ResNet-50 with BatchNorm (BN) from `torchvision`, ResNet-50 with GroupNorm (GN), and ViT-Base with LayerNorm (LN) from the `timm` library. Below, we outline ReTTA’s configuration and key implementation choices.

**Optimization and hyperparameters.** Only the affine and shift parameters  $\theta_{\text{affine}} \subset \theta$  of each normalization layer are fine-tuned using Stochastic Gradient Descent (SGD) with momentum 0.9 and batch size 64. Learning rates are set to  $2.5 \times 10^{-4}$  for ResNet variants and  $1.0 \times 10^{-3}$  for ViT. We employ the DeYO TTA strategies: data samples whose entropy exceeds  $\tau_{\text{Ent}} = 0.5 \ln 1000$  are skipped, and an adaptive loss weight uses an entropy margin  $\text{Ent}_0 = 0.4 \ln 1000$ . To filter out unstable pseudo-labels, we also require the pseudo-label probability drop (PLPD) to exceed  $\tau_{\text{PLPD}} = 0.3$  (mild shifts) or 0.2 (online label shifts). The TCC regularization coefficient  $\lambda_2$  is fixed at 1. When integrating with EATA, we add a Fisher-penalty balance  $\gamma = 2000$  and enforce sufficient logit diversity via a cosine-similarity threshold  $\phi = 0.05$ , keeping  $\tau_{\text{Ent}} = \text{Ent}_0 = 0.4 \times \ln 1000$ . For combination with SAR, we introduce a weight-perturbation scale  $\rho = 0.05$  and reset the model if it falls below  $e_0 = 0.2$  for the moving average of entropy. We maintain  $\tau_{\text{Ent}} = 0.4 \times \ln 1000$ . The complete update routines are detailed in Section B.1.

**Reported Baselines.** For all EM-based TTA methods, we report the accuracies given in the original SAR and DeYO papers. For energy-based approaches, AEA’s ImageNet-C results are taken directly from its publication (*no official code release was available to reproduce the comparisons in Table 2 of Section 5.1*). In contrast, TEA is evaluated by running its publicly released implementation<sup>1</sup> under our ImageNet-C evaluation protocol.

<sup>1</sup><https://github.com/yuanyige/tea>

---

**Algorithm 1** ReTTA based on EATA

---

**Input:** Test mini-batch  $\mathcal{B}_t = \{\mathbf{x}_t^j\}_{j=1}^M$ ; model  $f_\theta(\cdot)$  with trainable parameters  $\theta_{\text{affine}} \subset \theta$ ; step size  $\eta > 0$ ; TCC loss coefficient  $\lambda_2 > 0$ ;  $\text{Ent}_0, \tau_{\text{Ent}}, \phi, \gamma > 0$ .  
**Given:** Initialize  $\theta_0 = \theta_{\text{affine}}$  and  $\mathbf{m} = \mathbf{0}$ .  
**for**  $\mathbf{x}_t \sim \mathcal{B}_t$  **do**  
    Compute entropy  $H(\mathbf{p}_t)$  and predict  $\tilde{y}_t \leftarrow \arg\max_{k \in [1, K]} f_\theta(\mathbf{x}_t)$   $\triangleright \mathbf{p}_t = \text{Softmax}(f_\theta(\mathbf{x}_t))$   
    **if**  $H(\mathbf{p}_t) > \tau_{\text{Ent}}$  **then**  
        **continue**  
    **end if**  
    **if**  $\cos(f_\theta(\mathbf{x}_t), \mathbf{m}) < \phi$  **then**  
         $\mathbf{m} \leftarrow 0.9 \times \mathbf{m} + 0.1 \times \text{Softmax}(f_\theta(\mathbf{x}_t))$  **if**  $\mathbf{m} \neq \mathbf{0}$  **else**  $\text{Softmax}(f_\theta(\mathbf{x}_t))$   
        **continue**  
    **end if**  
    Compute gradients of entropy:  $\mathbf{g}_H = \nabla_{\theta_{\text{affine}}} H(\mathbf{p}_t)$   
    Compute loss weight  $\beta_\theta(\mathbf{x}_t; \text{Ent}_0)$   $\triangleright$  From Eq. 3 in [3]  
    **// Our implementation starts //**  
    Compute gradients of SSM:  $\mathbf{g}_{\text{SSM}} = \nabla_{\theta_{\text{affine}}} \ell_{\text{SSM}}(\theta; \mathbf{x}_t)$   $\triangleright$  From Eq. 15  
    Compute gradients of TCC:  $\mathbf{g}_{\text{TCC}} = \nabla_{\theta_{\text{affine}}} \ell_{\text{TCC}}(\theta; \mathbf{x}_t)$   $\triangleright$  From Eq. 16  
    Compute coefficient  $\lambda_1(\alpha) = \max(\min((1 - \alpha)/\alpha, 1), 0)$   $\triangleright \alpha = \min_{\alpha \in [0, 1]} \|\alpha \mathbf{g}_H + (1 - \alpha) \mathbf{g}_{\text{SSM}}\|_2^2$   
    **// Our implementation ends //**  
    Compute gradients of ReTTA:  $\mathbf{g}_{\text{ReTTA}} = \beta_\theta(\mathbf{x}_t^j; \text{Ent}_0) \cdot \mathbf{g}_H + \lambda_1(\alpha) \cdot \mathbf{g}_{\text{SSM}} + \lambda_2 \cdot \mathbf{g}_{\text{TCC}}$   
    Compute gradients of Fisher regularization:  $\mathbf{g}_{\text{FR}} = \nabla_{\theta_{\text{affine}}} \mathcal{R}(\theta_{\text{affine}}, \theta_0)$   $\triangleright$  From Eq. 7 in [3]  
    Update  $\theta_{\text{affine}} \leftarrow \theta_{\text{affine}} - \eta(\mathbf{g}_{\text{ReTTA}} + \gamma \cdot \mathbf{g}_{\text{FR}})$   
**end for**  
**Output:** Predicted labels  $\{\tilde{y}_t^j\}_{j=1}^M$

---

---

**Algorithm 2** ReTTA based on SAR

---

**Input:** Test mini-batch  $\mathcal{B}_t = \{\mathbf{x}_t^j\}_{j=1}^M$ ; model  $f_\theta(\cdot)$  with trainable parameters  $\theta_{\text{affine}} \subset \theta$ ; step size  $\eta > 0$ ; perturbation size  $\rho > 0$ ;  $\tau_{\text{Ent}} > 0$ ;  $e_0 > 0$  for model recovery; TCC loss coefficient  $\lambda_2 > 0$ .  
**Given:** Initialize  $\theta_0 = \theta_{\text{affine}}$  and moving average of entropy  $e_m = 0$ .  
**for**  $\mathbf{x}_t \sim \mathcal{B}_t$  **do**  
    Compute entropy  $H(\mathbf{p}_t)$  and predict  $\tilde{y}_t \leftarrow \arg\max_{k \in [1, K]} f_\theta(\mathbf{x}_t)$   $\triangleright \mathbf{p}_t = \text{Softmax}(f_\theta(\mathbf{x}_t))$   
    **if**  $H(\mathbf{p}_t) > \tau_{\text{Ent}}$  **then**  
        **continue**  
    **end if**  
    Compute gradient  $\nabla_{\theta_{\text{affine}}} H(\mathbf{p}_t)$   
    Compute  $\epsilon(\theta_{\text{affine}})$  from the solution to a dual norm problem.  $\triangleright$  From Eq. 4 in [4]  
    Compute gradients of entropy:  $\mathbf{g}_H = \nabla_{\theta_{\text{affine}}} H(\mathbf{p}_t)|_{\theta_{\text{affine}} + \epsilon(\theta_{\text{affine}})}$   
    **// Our implementation starts //**  
    Compute gradients of SSM:  $\mathbf{g}_{\text{SSM}} = \nabla_{\theta_{\text{affine}}} \ell_{\text{SSM}}(\theta; \mathbf{x}_t)|_{\theta_{\text{affine}} + \epsilon(\theta_{\text{affine}})}$   $\triangleright$  From Eq. 15  
    Compute gradients of TCC:  $\mathbf{g}_{\text{TCC}} = \nabla_{\theta_{\text{affine}}} \ell_{\text{TCC}}(\theta; \mathbf{x}_t)|_{\theta_{\text{affine}} + \epsilon(\theta_{\text{affine}})}$   $\triangleright$  From Eq. 16  
    Compute coefficient  $\lambda_1(\alpha) = \max(\min((1 - \alpha)/\alpha, 1), 0)$   $\triangleright \alpha = \min_{\alpha \in [0, 1]} \|\alpha \mathbf{g}_H + (1 - \alpha) \mathbf{g}_{\text{SSM}}\|_2^2$   
    **// Our implementation ends //**  
    Compute gradients of ReTTA:  $\mathbf{g}_{\text{ReTTA}} = \mathbf{g}_H + \lambda_1(\alpha) \cdot \mathbf{g}_{\text{SSM}} + \lambda_2 \cdot \mathbf{g}_{\text{TCC}}$   
    Update  $\theta_{\text{affine}} \leftarrow \theta_{\text{affine}} - \eta \mathbf{g}_{\text{ReTTA}}$   
     $e_m = 0.9 \times e_m + 0.1 \times H(\mathbf{p}_t)|_{\theta_{\text{affine}} + \epsilon(\theta_{\text{affine}})}$  **if**  $e_m \neq 0$  **else**  $H(\mathbf{p}_t^j)|_{\theta_{\text{affine}} + \epsilon(\theta_{\text{affine}})}$   
    **if**  $e_m < e_0$  **then**  
        Recover model weights  $\theta_{\text{affine}} \leftarrow \theta_0$   
    **end if**  
**end for**  
**Output:** Predicted labels  $\{\tilde{y}_t^j\}_{j=1}^M$

---

## B.1 Algorithms for ReTTA

**ReTTA based on EATA [3].** We extend the EATA framework that filters out high-entropy samples and enforces a minimal logit-diversity check via a moving-average cosine similarity. For each surviving sample, it computes the entropy-based gradient, SSM gradient, and the TCC gradient. An interpolation weight  $\lambda_1(\alpha)$  is analytically found, and an adaptive loss weight scales the entropy term. A Fisher-regularization gradient is added with a balance factor. The combined update ReTTA with

---

**Algorithm 3** ReTTA based on DeYO

---

**Input:** Test mini-batch  $\mathcal{B}_t = \{\mathbf{x}_t^j\}_{j=1}^M$ ; model  $f_\theta(\cdot)$  with trainable parameters  $\theta_{\text{affine}} \subset \theta$ ; step size  $\eta > 0$ ; an object-destructive transformation  $\mathcal{T}$ ; TCC loss coefficient  $\lambda_2 > 0$ ;  $\text{Ent}_0, \tau_{\text{Ent}}, \tau_{\text{PLPD}} > 0$ .

**Given:** Initialize  $\theta_0 = \theta_{\text{affine}}$ .

```
for  $\mathbf{x}_t \sim \mathcal{B}_t$  do
  Compute entropy  $H^j = H(\mathbf{p}_t^j)$  and predict  $\tilde{y}_t \leftarrow \arg\max_{k \in [1, K]} f_\theta(\mathbf{x}_t)$   $\triangleright \mathbf{p}_t^j = \text{Softmax}(f_\theta(\mathbf{x}_t))$ 
  if  $H^j > \tau_{\text{Ent}}$  then
    continue
  end if
  Obtain  $\hat{\mathbf{x}}_t = \mathcal{T}(\mathbf{x}_t)$ 
  Compute  $\text{PLPD}_\theta(\mathbf{x}_t, \hat{\mathbf{x}}_t)$   $\triangleright$  From Eq. 9 in [2]
  if  $\text{PLPD}_\theta(\mathbf{x}_t, \hat{\mathbf{x}}_t) < \tau_{\text{PLPD}}$  then
    continue
  end if
  Compute gradients of entropy:  $\mathbf{g}_H = \nabla_{\theta_{\text{affine}}} H(\mathbf{p}_t)$ 
  Compute loss weight  $\beta_\theta(\mathbf{x}_t^j; \text{Ent}_0)$   $\triangleright$  From Eq. 10 in [2]
  // Our implementation starts //
  Compute gradients of SSM:  $\mathbf{g}_{\text{SSM}} = \nabla_{\theta_{\text{affine}}} \ell_{\text{SSM}}(\theta; \mathbf{x}_t)$   $\triangleright$  From Eq. 15
  Compute gradients of TCC:  $\mathbf{g}_{\text{TCC}} = \nabla_{\theta_{\text{affine}}} \ell_{\text{TCC}}(\theta; \mathbf{x}_t)$   $\triangleright$  From Eq. 16
  Compute coefficient  $\lambda_1(\alpha) = \max(\min((1 - \alpha)/\alpha, 1), 0)$   $\triangleright \alpha = \min_{\alpha \in [0, 1]} \|\alpha \mathbf{g}_H + (1 - \alpha) \mathbf{g}_{\text{SSM}}\|_2^2$ 
  // Our implementation ends //
  Compute gradients of ReTTA:  $\mathbf{g}_{\text{ReTTA}} = \beta_\theta(\mathbf{x}_t; \text{Ent}_0) \cdot \mathbf{g}_H + \lambda_1(\alpha) \cdot \mathbf{g}_{\text{SSM}} + \lambda_2 \cdot \mathbf{g}_{\text{TCC}}$ 
  Update  $\theta_{\text{affine}} \leftarrow \theta_{\text{affine}} - \eta \mathbf{g}_{\text{ReTTA}}$ 
end for
Output: Predicted labels  $\{\tilde{y}_t^j\}_{j=1}^M$ 
```

---

Fisher regularization is applied to the affine parameters via SGD. See Algorithm 1, used in Table 4 in Section 5.2 and Table 2 in Section C.

**ReTTA based on SAR [4].** We extend the SAR test-time adaptation that filters out high-entropy samples. Then, we compute entropy and SSM gradients at perturbed weights  $\theta + \epsilon$ . These gradients are interpolated with the TCC. We reset the model’s initial affine weights if the moving-average entropy  $e_m$  drops below  $e_0$ . Refer to Algorithm 2, whose results appear in Table 4 and Figure 1 within Section 3.2 as well as Table 2 in Section C.

**ReTTA based on DeYO [2].** Incorporates DeYO’s object-destructive transformation and test of pseudo-label probability drop (PLPD) to discard unreliable samples. For the remaining data, it computes entropy, SSM, and TCC gradients and blends them using an adaptive weighting  $\beta(\text{Ent}_0)$ . The full procedure is detailed in Algorithm 3, with performance reported in Tables 1 to 4 and Figures 2 to 4 of the manuscript, as well as Table 1 in Section C.

## C Supplementary Experiment

**Quantitative visualization to support Theorem 1.** Theorem 1 assumes that, as the number of dominant classes  $k$  varies, the average of the top- $k$  logits ( $z^*$ ) shifts. In contrast, the mean of the remaining  $K - k$  logits ( $Z$ ) remains singular and comparatively small, independent of  $k$ . Figure 1 confirms this on ImageNet-C (severity 5) using a ResNet-50 (BN): although  $z^*$  decreases with  $k$ ,  $Z$  stays almost constant, justifying its use as a fixed hyperparameter. Using the empirical stability of  $Z$ , we plot the analytic energy-entropy relationship for two values of  $Z$  near its observed mean (e.g.,  $Z = 0.0$  and  $Z = -0.01$ ) with  $K = 1000$ . Figure 2 (which also includes the case  $Z = 0.5$  from Figure 1(d) in Section 3.2) presents that small shifts in  $Z$  around its average produce almost identical  $H(E_\theta)$  curves. This confirms the generality and solid derivation of our closed-form equation.

**Impact of adopting CD for energy loss into ReTTA.** Following Figure 4(b) in Section 5.2, we test another energy loss to replace our proxy, SSM, with Contrastive Divergence (CD), defined by Eq. 8 in Section 4.1 (which SSM was derived to avoid generative sampling). As shown in Table 1, CD yields substantially lower performance, likely because it depends on unstable generative sampling [7].

**Effects of SSM and TCC within EM methods under online label shifts.** In the extreme scenario (label imbalance ratio= $\infty$ ), Table 2 demonstrates that combining EM loss with SSM and TCC

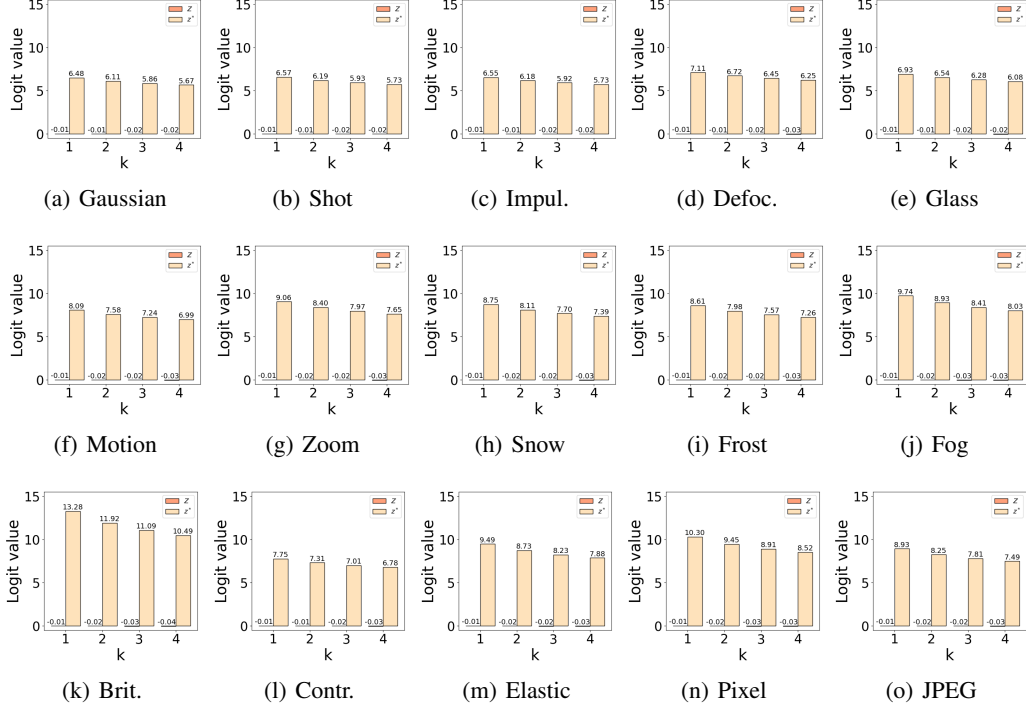


Figure 1: Quantitative visualization of average logits as  $k$  changes. For each  $k$ , the primary logit  $z^*$  is computed as the mean of the top- $k$  logits, and the secondary logit  $Z$  as the mean of the remaining  $K - k$  logits. Results are shown for ResNet-50 (BN) evaluated on ImageNet-C at severity level 5.

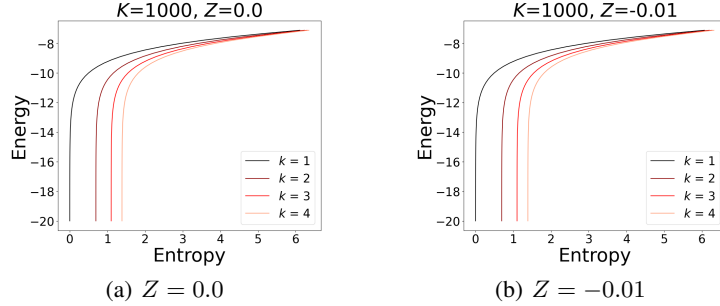


Figure 2: Energy–entropy curves for  $K = 1000$  classes, shown for two fixed values of the secondary logit  $Z$ . Each curve illustrates how the entropy varies as the energy changes under a specified  $Z$ .

Table 1: Comparisons among alternative losses for SSM in energy adaptation on ImageNet-C at severity level 5 under mild scenario in terms of accuracy (%). CD represents Contrastive Divergence.

Mild	Noise			Blur				Weather			Digital					Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
CD [7]	35.3	37.2	36.6	31.2	32.2	47.7	51.4	51.6	45.2	59.9	67.0	42.8	57.6	60.4	54.6	47.5
SM	36.5	38.3	37.9	33.8	33.3	48.6	52.9	52.9	<b>46.2</b>	60.5	68.2	45.5	58.5	61.3	55.8	48.7
SSM-VR	37.2	39.3	<b>39.1</b>	34.3	34.0	49.0	52.9	<b>52.9</b>	45.8	60.7	68.2	46.8	58.6	61.5	55.8	49.1
SSM	<b>37.3</b>	<b>39.7</b>	38.9	<b>34.5</b>	<b>34.1</b>	<b>49.3</b>	<b>53.1</b>	52.7	46.1	<b>60.7</b>	<b>68.2</b>	<b>47.6</b>	<b>58.6</b>	<b>61.5</b>	<b>56.0</b>	<b>49.2</b>

improves overall accuracy. For ResNet-50 (GN), DeYO gains most on Noise corruptions (+1.1% on average), SAR improves across nearly all corruption types except Defocus and Glass, and EATA sees its largest boost in the Noise category. VitBase (LN) benefits even more: SSM + TCC raises EATA’s averaged accuracy by +2.1%, SAR’s by +1.6%, and DeYO’s by +2.5%. Although enhancements on certain Blur corruptions (Defocus, Glass) are smaller, the joint energy- and entropy-based adaptation reliably strengthens robustness across diverse distribution shifts.



Table 2: Adaptivity of components in ReTTA applied to state-of-the-art EM methods (EATA, SAR, and DeYO). Each denotes accuracy (%) on ImageNet-C (severity 5) under online label shifts.

Label Shifts	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet-50 (GN) + EM ([3])	25.7	28.6	24.8	18.5	19.6	24.1	28.4	35.3	33.0	41.2	65.2	33.3	28.0	42.4	43.1	32.7
+SSM+TCC	27.8	32.7	29.2	21.0	16.9	25.6	29.1	33.9	31.5	42.5	68.1	31.4	27.5	43.5	46.2	33.8
ResNet-50 (GN) + EM ([4])	33.7	36.9	35.3	19.3	20.3	33.8	29.8	21.9	44.7	34.9	71.9	46.7	6.6	52.3	56.2	36.3
+SSM+TCC	33.8	37.0	35.3	8.8	17.0	34.9	27.7	20.6	47.9	50.1	72.0	44.8	9.3	52.8	56.3	36.6
ResNet-50 (GN) + EM ([2])	42.5	44.9	43.8	22.2	16.3	41.0	13.2	52.2	51.5	39.7	73.4	52.6	46.9	59.3	59.3	43.9
+SSM+TCC	42.7	45.1	44.2	29.4	22.9	41.1	34.4	52.8	51.1	58.5	73.5	49.8	48.4	59.8	59.3	47.5
VitBase (LN) + EM ([3])	36.2	34.7	35.5	43.4	44.3	49.3	48.5	53.2	53.5	62.3	72.7	18.8	58.0	64.7	62.8	49.2
+SSM+TCC	45.6	45.2	46.9	40.7	46.4	47.2	49.6	53.2	51.8	59.8	71.7	27.3	57.6	65.3	61.0	51.3
VitBase (LN) + EM ([4])	42.3	34.9	44.1	50.0	50.5	55.6	53.1	59.7	47.2	66.2	75.2	50.3	60.1	67.3	65.0	54.8
+SSM+TCC	53.6	54.0	54.9	55.0	53.3	58.9	53.6	6.9	37.4	69.9	75.5	66.9	62.0	69.4	67.0	55.9
VitBase (LN) + EM ([2])	53.5	36.0	54.6	57.6	58.7	63.7	46.2	67.6	66.0	73.2	77.9	66.7	69.0	73.5	70.3	62.3
+SSM+TCC	54.0	55.0	55.2	57.8	58.7	64.7	58.5	69.0	67.1	71.2	77.9	67.6	69.8	74.1	71.6	64.8

Table 3: Effect of components in ReTTA. Each denotes accuracy (%) on ImageNet-C (severity 5) under the mild scenario, with DeYO as the baseline EM method.

Mild	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet-50 (BN)+EM [2]	35.6	37.9	37.1	33.8	34.1	48.5	52.8	52.7	46.4	60.6	68.0	46.1	58.4	61.5	55.7	48.6
+SSM	37.0	39.6	38.6	34.3	34.3	48.8	52.9	52.6	45.7	60.6	68.2	46.9	58.5	61.4	55.8	49.0
+TCC	36.0	38.9	38.1	32.6	33.6	48.3	52.8	52.5	45.2	60.5	68.2	46.1	58.6	61.3	55.6	48.6
+SSM+TCC	37.3	39.7	38.9	34.5	34.1	49.3	53.1	52.7	46.1	60.7	68.2	47.6	58.6	61.5	56.0	49.2

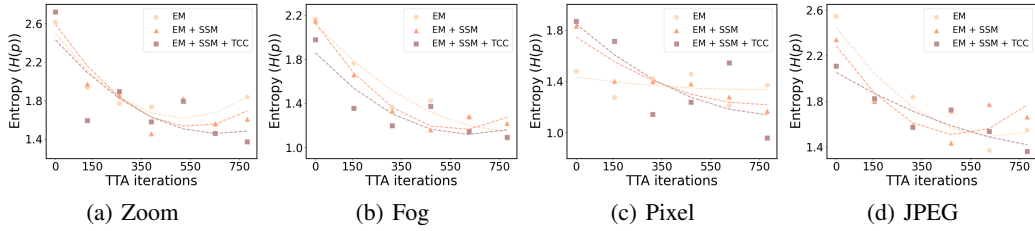


Figure 3: Effects of TCC under Zoom, Fog, Pixel, and JPEG corruption of ImageNet-C (severity 5).

**Component analysis on mild scenario.** Table 3 demonstrates that SSM alone shows widespread improvements, boosting average accuracy from 48.6% to 49.0% (e.g., +1.4% on Gaussian noise, +0.8% on Frost). TCC yields smaller, more focused gains (e.g., +0.8% on Shot noise) but without trade-offs. Combining SSM and TCC yields the strongest results across every corruption type, notably +1.7% on Gaussian noise and +0.9% on Brightness, raising the overall average to 49.2%.

**Effects of TCC under additional corruptions.** Figure 3 compares the entropy trajectories throughout TTA iterations across a range of ImageNet-C corruptions, with and without the TCC regularizer. While standard EM often shows entropy rebound, indicating an inconsistency with its intended minimization direction, adding TCC yields a steady decrease in the entropy at test time. When combined with SSM, this steady entropy reduction demonstrates synergistic gains in accuracy, as reflected in the performance improvements summarized in Table 1 within Section 5.1.

**Visual comparisons ReTTA with EM only.** To confirm that ReTTA encourages simultaneous decreases in energy and entropy, Figure 4 plots each test sample’s position in the energy–entropy plane before and after adaptation under EM alone [4] versus ReTTA. Because TTA operates without labels, overall accuracy gains are modest—about a 1 percent point increase in top-1 accuracy on average (see Table 4 in Section 5.2). In line with this, approximately 1% of samples shift noticeably under ReTTA, moving downward (lower energy) and leftward (lower entropy). Although small, this consistent migration demonstrates that ReTTA more reliably pursues the dual objective of reducing both entropy and energy compared to EM alone.

**Robustness to mild scenarios using ViTBase.** To ensure experimental fairness with Table 1 of the main paper, we measure all severity 5 ImageNet-C evaluations using the official DeYO repository and adopt the ViTBase (LN) as the target model. The full results are summarized in Table 4. Even under these mild conditions, ReTTA achieves strong performance, ranking first on 10 of the 15 corruptions and second on the remaining five. On average, ReTTA outperforms all baseline TTA methods, highlighting that, across different model architectures, its combined effects of energy adaptation and discriminability remain essential for sustaining high accuracy under severe distribution shifts.

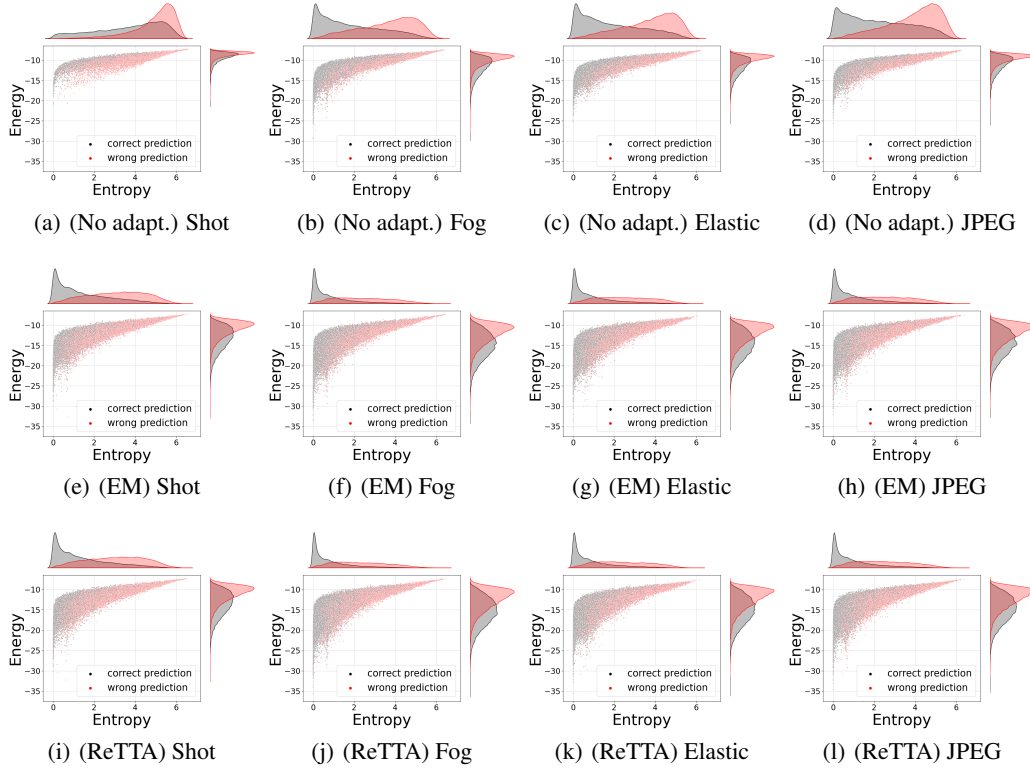


Figure 4: Distribution of test data based on energy and entropy values. **(Top)** No adapt.: inference without TTA. **(Middle)** EM: results based on the state-of-the-art EM method [4]. **(Bottom)** ReTTA (ours): results based on Algorithm 2. We use ResNet-50 (BN) on ImageNet-C (severity level 5).

Table 4: Comparisons with baseline TTA methods on ImageNet-C at severity level 5 under mild scenario in terms of accuracy (%). \* TEA was not publicly reported and was tested directly.

Mild	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
VitBase (LN)	35.1	32.2	35.9	31.4	25.3	39.4	31.6	24.5	30.1	54.7	64.5	49.0	34.2	53.2	56.5	39.8
Tent	51.9	51.5	53.1	51.8	47.8	56.3	49.4	10.6	18.3	67.1	73.3	66.7	51.5	65.2	64.5	51.9
EATA	<b>55.9</b>	<b>56.4</b>	<b>57.1</b>	53.3	53.5	58.2	<b>58.5</b>	61.9	60.1	71.4	75.3	<b>68.5</b>	62.4	69.5	66.6	61.9
SAR	51.8	51.7	52.9	50.8	48.9	55.6	49.1	12.8	51.0	65.8	73.1	66.1	51.9	64.0	63.3	53.9
DeYO	54.5	55.1	55.8	53.8	54.6	62.4	58.0	64.0	63.3	71.6	77.3	67.3	65.6	71.5	68.3	62.9
TEA*	8.8	20.2	11.8	1.9	4.2	14.7	4.4	1.2	3.6	8.9	73.4	62.8	3.0	67.3	64.2	23.3
<b>ReTTA (ours)</b>	<b>55.1<math>\pm</math>0.1</b>	<b>55.9<math>\pm</math>0.0</b>	<b>56.5<math>\pm</math>0.0</b>	<b>56.4<math>\pm</math>0.1</b>	<b>55.9<math>\pm</math>0.1</b>	<b>62.5<math>\pm</math>0.1</b>	<b>57.9<math>\pm</math>0.0</b>	<b>64.6<math>\pm</math>0.0</b>	<b>63.5<math>\pm</math>0.1</b>	<b>72.2<math>\pm</math>0.0</b>	<b>77.4<math>\pm</math>0.0</b>	<b>67.4<math>\pm</math>0.1</b>	<b>65.7<math>\pm</math>0.1</b>	<b>71.6<math>\pm</math>0.0</b>	<b>68.6<math>\pm</math>0.0</b>	<b>63.4<math>\pm</math>0.0</b>

Table 5: Comparisons with baseline TTA methods on ImageNet-C at severity level 1 under mild scenario in terms of accuracy (%). \* TEA was not publicly reported and was tested directly.

Mild	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet-50 (BN)	59.6	57.8	48.4	59.0	53.7	64.5	52.2	54.3	61.1	61.4	73.8	64.4	66.4	63.9	65.9	60.4
Tent	68.0	67.6	64.3	66.4	67.3	70.5	65.4	65.1	66.4	70.2	<b>74.6</b>	72.0	69.9	72.0	70.1	68.7
EATA	68.7	68.4	65.6	67.0	68.0	70.6	66.2	66.5	66.9	70.8	74.4	72.3	70.2	72.3	70.4	69.2
SAR	67.9	67.5	64.3	66.3	67.3	70.4	65.4	64.9	63.3	70.1	74.5	71.9	69.9	72.0	70.1	68.6
DeYO	68.9	68.6	66.0	67.2	68.0	70.5	66.3	<b>67.0</b>	67.1	70.8	74.1	72.2	70.1	<b>72.3</b>	70.2	69.3
TEA*	59.4	58.9	54.0	56.8	58.5	61.6	54.7	56.9	57.0	62.4	66.8	63.6	60.3	64.0	61.2	59.7
<b>ReTTA (ours)</b>	<b>68.9<math>\pm</math>0.1</b>	<b>68.7<math>\pm</math>0.1</b>	<b>66.0<math>\pm</math>0.0</b>	<b>67.4<math>\pm</math>0.1</b>	<b>68.3<math>\pm</math>0.2</b>	<b>70.7<math>\pm</math>0.1</b>	<b>66.3<math>\pm</math>0.2</b>	<b>66.9<math>\pm</math>0.0</b>	<b>67.1<math>\pm</math>0.0</b>	<b>70.8<math>\pm</math>0.2</b>	<b>74.1<math>\pm</math>0.0</b>	<b>72.4<math>\pm</math>0.1</b>	<b>70.3<math>\pm</math>0.1</b>	<b>72.2<math>\pm</math>0.0</b>	<b>70.4<math>\pm</math>0.0</b>	<b>69.4<math>\pm</math>0.0</b>

**Robustness to corruption levels of test data.** Following Table 1 of the main paper, we evaluate model robustness under lower corruption severity levels (i.e., 1 and 3) on ImageNet-C. At severity 1, Table 5 shows that all TTA methods achieve comparable performance, although ReTTA reaches the highest average accuracy (69.4%). However, at severity 3, the performance gap widens as corruption strength increases (see Table 6). While previous methods such as Tent and EATA drop to 59.2% and 61.0%, respectively, ReTTA maintains 61.9%, resulting in a +2.7% improvement over Tent and +0.9% over EATA. These results indicate that joint optimization of energy and entropy becomes

Table 6: Comparisons with baseline TTA methods on ImageNet-C at severity level 3 under mild scenario in terms of accuracy (%). \* TEA was not publicly reported and was tested directly.

Mild	Noise			Blur				Weather				Digital				Avg.
	Gauss.	Shot	Impul.	Defoc.	Glass	Motion	Zoom	Snow	Frost	Fog	Brit.	Contr.	Elastic	Pixel	JPEG	
ResNet-50 (BN)	27.6	25.0	25.2	37.9	16.9	37.7	35.2	35.2	32.1	46.7	69.6	46.0	55.6	46.2	59.3	39.7
Tent	54.8	54.3	53.7	49.2	46.5	58.8	57.7	55.9	48.6	65.8	72.1	67.1	69.4	67.5	65.9	59.2
EATA	57.0	56.8	56.2	52.4	50.2	61.0	59.7	58.6	51.3	67.1	72.2	68.2	69.9	68.1	66.7	61.0
SAR	54.6	54.1	53.5	49.3	46.3	58.6	57.6	55.6	48.6	65.6	72.0	67.1	69.3	67.3	65.7	59.0
DeYO	58.1	58.0	57.1	53.2	51.2	61.9	59.8	59.6	51.9	67.5	72.0	68.5	69.6	68.6	66.6	61.6
TEA*	24.8	22.0	24.5	20.1	14.9	45.3	32.0	35.0	17.6	57.8	64.2	50.9	60.4	59.0	55.5	38.9
<b>ReTTA (ours)</b>	<b>58.6<math>\pm</math>0.2</b>	<b>58.7<math>\pm</math>0.1</b>	<b>58.0<math>\pm</math>0.1</b>	<b>53.2<math>\pm</math>0.0</b>	<b>51.5<math>\pm</math>0.2</b>	<b>61.9<math>\pm</math>0.1</b>	<b>59.9<math>\pm</math>0.1</b>	<b>59.7<math>\pm</math>0.1</b>	<b>52.1<math>\pm</math>0.1</b>	<b>67.5<math>\pm</math>0.1</b>	<b>72.3<math>\pm</math>0.1</b>	<b>68.7<math>\pm</math>0.2</b>	<b>69.9<math>\pm</math>0.0</b>	<b>68.8<math>\pm</math>0.1</b>	<b>66.9<math>\pm</math>0.1</b>	<b>61.9<math>\pm</math>0.0</b>

Table 7: Results of sentiment classification on the Amazon Polarity (Books  $\rightarrow$  Electronics) dataset under a cross-domain TTA using DistilBERT. Each value reports accuracy (%) in the target domain (Electronics). The entropy threshold for data filtering is set to 0.2.

Dataset	Method	EM	Data filtering	SSM	TCC	Accuracy (%)
Source (Books)						97.21
Target (Electronics)	No adapt.					<b>92.99</b>
	Tent	✓				87.73
		✓	✓			87.75
		✓	✓	✓		<b>92.31</b>
		✓	✓		✓	87.76
	ReTTA	✓	✓	✓	✓	<b>92.28</b>

increasingly beneficial under stronger corruptions, improving the adaptation effect compared with entropy-only objectives in the mild scenarios.

## D Case Study: Applicability to Text Modality

**Experimental settings.** We evaluate ReTTA on a cross-domain sentimental classification scenario using the Amazon Polarity Reviews dataset. The source domain is Books, and the target domain is Electronics. Each review contains a 1-5 star rating: we binarize these into negative (ratings 1-2, label 0) and positive (ratings 4-5, label 1), discarding neutral (3-star) reviews. At test time, we deploy a Books-pretrained model to classify Electronics reviews under the TTA paradigm, using DistilBERT [5] from HuggingFace. We train the model with AdamW, using a base learning rate scaled linearly by batch size, specifically  $2 \times 10^{-5} \times (\text{batch size}/32)$ , and a batch size of 256. Training runs for 3 epochs. For the Books domain, we fine-tune all transformer layers and the classifier head, while keeping the embedding layer frozen. For Electronics TTA, we update only the LayerNorm affine parameters in transformer layers. All input sequences are truncated or padded to 128 tokens. Books are split into 90% training and 10% validation sets. Because we address a single source-to-target shift, the self-adjusting coefficient is not used; both  $\lambda_1$  and  $\lambda_2$  are fixed to 1.

**Discussion.** To examine ReTTA’s adaptability to non-visual modalities, we extend the SSM formulation to the embedding space of DistilBERT. Each input sequence is represented as a token-embedding matrix  $\mathbf{s}_t \in \mathbb{R}^{l \times 768}$ , where  $l$  denotes the sequence length produced by DistilBERT’s encoder. For every  $\mathbf{s}_t$ , we compute the Jacobian and Hessian of the energy-based model with respect to  $\mathbf{s}_t$ , denoted as  $p(\mathbf{s}_t) := \exp(-E_\theta(\mathbf{s}_t))/Z(\theta)$ . We then sample random projection vectors  $\mathbf{v} \in \mathbb{R}^{l \times 768}$  and minimize the sliced Fisher divergence defined as:

$$D_{SF}(p_t(\mathbf{s}_t)||p_\theta(\mathbf{s}_t)) = \mathbb{E}_{p_t, p(\mathbf{v})}[\mathbf{v}^T \nabla_{\mathbf{s}_t}^2 \log p_\theta(\mathbf{s}_t) \mathbf{v} + \frac{1}{2} ||\mathbf{v}^T \nabla_{\mathbf{s}_t} \log p_\theta(\mathbf{s}_t)||^2]. \quad (46)$$

We evaluate TTA exploiting Eq. 46 on the Amazon Polarity Reviews task (Books  $\rightarrow$  Electronics). As shown in Table 7, the model pretrained on Books achieves 97.21% accuracy in its source domain but drops to 92.99% on Electronics, reflecting a moderate domain shift.

When applied alone, EM (Tent) substantially degrades performance (87.73%), indicating that naive EM can be counterproductive under small or highly overlapping domain shifts. Filtering low-entropy samples yields only a marginal gain (87.75%), whereas SSM recovers most of the gap, improving the target accuracy to 92.31%. The full ReTTA (EM+SSM+TCC) framework achieves 92.28%, closely matching the SSM-only result. These observations suggest that, unlike in image-corruption settings, where large shifts benefit EM, text domains with smaller distribution gaps are more vulnerable to overconfident entropy updates.

Table 8: Computational overhead of various test-time adaptation methods on ResNet-50 (BN) using 50,000 ImageNet-C samples corrupted by Gaussian noise at severity 5. For each method, we report whether source data is required, whether updates occur online, the number of forward and backward passes, and additional computations. “No adaptation” performs one forward pass per sample.

Method	Need source data?	Online update?	#Forward	#Backward	Other computation
No adaptation	$\times$	$\times$	50,000	-	n/a
MEMO	$\times$	$\times$	$50,000 \times 65$	$50,000 \times 64$	AugMix
Tent	$\times$	$\checkmark$	50,000	50,000	n/a
EATA	$\checkmark$	$\checkmark$	50,000	19,085	regularizer
SAR	$\times$	$\checkmark$	$50,000 + 18,608$	$18,608 + 12,491$	Additional model updates
DeYO	$\times$	$\checkmark$	$50,000 + 33,943$	25,836	Compute distance (Eq.9 in [2])
TEA	$\times$	$\checkmark$	$50,000 + 50,000$	$50,000 + 50,000$	Generative process
AEA	$\times$	$\checkmark$	50,000	50,000	Compute distance (Eq.5 in [1])
ReTTA	$\times$	$\checkmark$	$50,000 + 25,841$	25,841	Eq. 18

Table 9: Comparison of ReTTA defined by Algorithm 3 with existing TTA methods on ImageNet-C at severity level 5 under batch-size-1 adaptation settings in terms of accuracy (%).

Model	No adapt.	MEMO	Tent	EATA	SAR	DeYO	ReTTA (ours)
ResNet-50 (GN)	30.6	31.2	24.7	36.4	34.5	44.4	<b>28.0</b>
VitBase (LN)	29.9	39.1	48.9	46.3	56.6	64.4	<b>8.8</b>

In such scenarios, minimizing energy through SSM enhances local observability around token embeddings and mitigates the over-confidence penalties typical of EM-based methods. Hence, energy adaptation provides a principled way to counteract uncertainty amplification and sustain effective adaptation when source and target distributions overlap.

## E Adaptation Overhead Analysis

As Table 5 shows, ReTTA requires one forward pass over the 50,000 test images with an additional 25,841 forward operations for the first entropy-based filtering and 25,841 backward passes for the adaptation, substantially fewer than MEMO (over 3.2 million forwards/backwards) or DeYO (84,000+ extra forwards and 25,836 backwards). Compared to SAR, which divides its 18,608+12,491 backward updates across two stages, ReTTA consolidates all 25,841 updates into one. Unlike generative approaches such as TEA that double both forward and backward passes, ReTTA’s closed-form, sampling-free update balances performance and computational overhead. *Discrepancies in hardware infrastructure (e.g., clock frequency, memory subsystems) and code-level optimizations (e.g., loop unrolling, parallelism) can skew runtime measurements.* Hence, we skip timing results.

## F Limitation

Because ReTTA still relies on an unsupervised cross-entropy fit to its own pseudo-labels, it tends to over-commit to one-hot targets when used alongside methods that perform no data filtering (e.g., Tent [6]). In those cases, uncertain or incorrect pseudo-labels can push the affine parameters in the wrong direction and degrade accuracy. In addition, the SSM proxy was derived to approximate an expected log-density objective; when executed with a batch size of one (the extreme “streaming” setting), it literally maximizes the log-density of a single sample, leading to a mode collapse, as shown in Table 9. In practice, this can be mitigated by gradually down-weighting the SSM term when updating per sample, as the batch size is predictable, deterministic, and controllable during inference. Finally, while one could imagine engineering ReTTA to adapt to every incoming data, we believe that in a purely unsupervised scenario, it is both natural and effective to first filter for a small set of high-confidence data before adapting—extending adaptation to all samples would add complexity without clear benefits.

Additionally, ReTTA’s performance tends to diminish when the source and target distributions overlap, as observed in Table 7. This limitation reflects a broader challenge in unsupervised adaptation for semantically similar domains, where energy-based regularization provides limited leverage once prediction uncertainty is already low. A promising direction for addressing this issue is to incorporate cross-domain contrastive objectives, which could help ReTTA better discriminate subtle yet systematic shifts across overlapping domains. We leave these directions for future work.

## G Broader Impacts

This study advances test-time adaptation (TTA) by jointly optimizing energy- and entropy-based objectives, yielding models that maintain reliable confidence and likelihood estimates under real-world distribution shifts. This has positive implications for applications where robustness is critical, such as autonomous vehicles navigating new weather conditions, medical imaging systems encountering unexpected scanner artifacts, and surveillance algorithms operating in dynamic lighting. By reshaping the energy landscape and enforcing discriminability at test time, ReTTA can reduce catastrophic failures, improve safety, and increase trust in AI-driven decisions.

However, online adaptation can also introduce new vulnerabilities. An adversary might craft inputs designed to mislead the energy-entropy balance, causing the model to adapt in harmful ways or amplify biases. Moreover, reliance on energy-based proxies for likelihood does not fully certify calibration across all subpopulations; systematic evaluation is needed to prevent unintentional fairness violations. We therefore recommend deploying ReTTA with additional safeguards, such as anomaly detection on energy traces and conservative adaptation techniques, to ensure that the benefits of robust adaptation are realized equitably and safely.

## References

- [1] Wonjeong Choi, Do-Yeon Kim, Jungwuk Park, Jungmoon Lee, Younhyun Park, Dong-Jun Han, and Jackyun Moon. Adaptive energy alignment for accelerating test-time adaptation. In *International Conference on Learning Representations*, 2025.
- [2] Jonghyun Lee, Dahuin Jung, Saehyung Lee, Junsung Park, Juhyeon Shin, Uiwon Hwang, and Sungroh Yoon. Entropy is not enough for test-time adaptation: From the perspective of disentangled factors. In *International Conference on Learning Representations*, 2024.
- [3] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofu Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International Conference on Machine Learning*, 2022.
- [4] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiqian Wen, Yaofu Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.
- [5] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- [6] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021.
- [7] Yige Yuan, Bingbing Xu, Liang Hou, Fei Sun, Huawei Shen, and Xueqi Cheng. Tea: Test-time energy adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.