

Table 3: Ablation study results across different k . Values are averaged over all samples. Success rates are in %.

k	Top-1 (%)	Top-5 (%)	L_2 Dist.	LPIPS	Overhead (s)
1	95.29	97.29	0.006070	0.000477	0.296
3	94.29	97.14	0.006478	0.000554	0.377
5	95.00	96.86	0.006567	0.000562	0.457
10	94.43	96.86	0.006556	0.000564	0.681
20	94.43	96.86	0.006565	0.000563	1.145
50	94.14	96.86	0.006566	0.000567	2.451

Table 4: Ablation study results across different λ . Values are averaged over all samples. Success rates are in %.

λ	Top-1 (%)	Top-5 (%)	L_2 Dist.	LPIPS	Overhead (s)
0.100	74.86	87.14	0.002854	0.000127	0.468
0.030	81.86	90.71	0.003879	0.000218	0.467
0.010	88.14	93.00	0.004803	0.000313	0.455
0.003	93.29	95.14	0.005749	0.000435	0.454
0.001	95.00	96.86	0.006567	0.000562	0.454
0.0003	96.71	98.14	0.007408	0.000721	0.453

APPENDIX

A ABLATION AND VISUALIZATION

In this section, we conduct an ablation study on the main hyperparameters and design choices to analyze the behavior of DOC in detail. All evaluations were performed on misclassified samples from ImageNet-1k, using success rate, distortion (L_2 , LPIPS), and computational overhead as metrics.

Top- k The success rate and distortion remain almost unchanged, while increasing k only increases the overhead.

λ Decreasing λ improves the success rate, but at the cost of increased distortion. This implies that a strong regularization term suppresses perturbations too much, making repair difficult.

Distance Function L_2 and Fisher-Rao show the best performance, while Cross-Entropy and Total Variation have unstable gradients, resulting in almost no successful repairs.

Steps Increasing the number of steps improves the success rate and reduces distortion. However, performance saturates around 8 to 16 steps, after which only the overhead increases.

Summary and Discussion The ablation results provide the following insights: (i) A small k is sufficient; a large k is unnecessary. (ii) A smaller λ leads to a higher success rate but increases distortion, making a proper trade-off important. (iii) For the distance function, L_2 and Fisher-Rao are effective, while others are unstable. (iv) The optimal number of steps is between 8 and 16; more steps decrease computational efficiency. These results suggest that a "small k , moderate λ , appropriate distance function, and a finite number of steps" is a well-balanced choice for designing DOC.

B TEST-TIME ADAPTATION RESULTS

In this section, we compare the performance of representative TTA methods (Tent, EATA, MEMO, SAR) on ResNet-50, a CNN baseline, and ViT-B. The evaluation metrics are repair success rate

Table 5: Ablation study results across different distance metrics (modes). Values are averaged over all samples. Success rates are in %.

Mode	Top-1 (%)	Top-5 (%)	L_2 Dist.	LPIPS	Overhead (s)
Cross-Entropy	0.00	0.29	213.053130	0.963402	0.498
Fisher-Rao	95.00	96.86	0.006567	0.000562	0.458
Hellinger	36.43	85.86	0.031668	0.019856	0.465
L_2	96.14	98.86	0.004426	0.000287	0.454
Total Variation	0.57	1.57	3.633467	0.915562	0.500

Table 6: Ablation study results across different numbers of steps. Values are averaged over all samples. Success rates are in %.

Steps	Top-1 (%)	Top-5 (%)	L_2 Dist.	LPIPS	Overhead (s)
2	86.43	95.57	0.010301	0.001543	0.117
4	93.29	96.43	0.007717	0.000809	0.235
8	95.00	96.86	0.006567	0.000562	0.458
12	94.86	97.43	0.006239	0.000502	0.693
16	95.57	97.43	0.006074	0.000476	0.931
24	94.86	97.29	0.005930	0.000454	1.369

(Top-1, Top-5), input distortion (L_2 , LPIPS), and computational overhead. The results are shown in Tables ?? and 8.

B.1 RESNET-50

On ResNet-50, EATA and MEMO slightly improved the repair success rate, with an effect of about 4% for Top-1. On the other hand, SAR was mostly unsuccessful, failing to provide stable repair behavior. We applied Tent to the CNN baseline, but its Top-1 success rate was also limited to about 4%. For all these methods, the input distortion (L_2 , LPIPS) was close to zero, indicating that they only make minor adjustments to the model’s output.

Table 7: TTA performance on ResNet-50

Model	Method	Top-1 Success	Top-5 Success	L_2	LPIPS	Overhead (s)
ResNet-50	EATA	0.0376	0.1912	0.0000	0.0000	0.0526
ResNet-50	MEMO	0.0408	0.1992	0.0000	0.0000	0.1945
ResNet-50	SAR	0.0032	0.0052	0.0000	0.0000	0.0543
ResNet-50	Tent	0.0392	0.1970	0.0000	0.0000	0.1159

B.2 ViT-B

For ViT-B, we evaluated EATA, MEMO, and SAR. The results show that while the Top-5 success rate remained high at about 79%, the Top-1 success rate was close to zero. This indicates that TTA methods contribute to local adaptation of the distribution but have limited ability to cross the decision boundary to repair misclassifications. The computational overhead was relatively low for EATA and SAR, while MEMO required slightly more cost.

C EXPERIMENTAL SETUP

In this study, we compared perturbation-based methods and test-time adaptation (TTA) methods on ResNet-50, a CNN baseline, and ViT-B. The evaluation metrics were repair success rate (Top-1, Top-5), input distortion (L_2 , LPIPS), Jensen - Shannon divergence, and computational overhead. The Fisher - Rao distance was used as the default Lyapunov function.

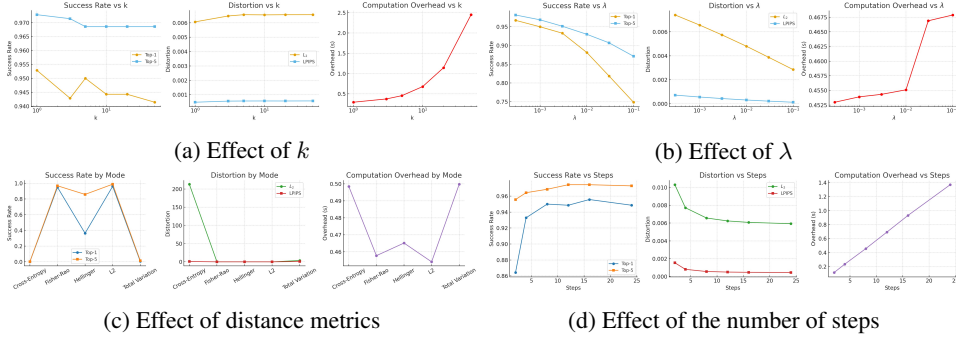


Figure 6: Ablation study visualization results.

Table 8: TTA performance on ViT-B

Method	Top-1 Success	Top-5 Success	L_2	LPIPS	Overhead (s)
EATA	0.0000	0.7866	0.0000	0.0000	0.0691
MEMO	0.0062	0.7888	0.0000	0.0000	0.1779
SAR	0.0000	0.7874	0.0000	0.0000	0.0582

C.1 PERTURBATION-BASED METHODS

First, we compared methods that directly add perturbations to the input space to repair misclassifications. We implemented Projected Gradient Descent (PGD) and DeepFool as representative methods and unified the computational cost by setting the same number of iterations. The settings are shown in Table 9.

Table 9: Hyperparameter settings for perturbation-based methods

Method	Iterations	Step Size / Overshoot	Radius ϵ
PGD	10	$\alpha = 0.003$	$\epsilon = 0.01$
DeepFool	5	Overshoot = 0.02	—

C.2 TEST-TIME ADAPTATION METHODS

Next, we evaluated TTA methods that update the normalization layer parameters online: Tent, EATA, MEMO, and SAR. To prevent overfitting, each method updated only the affine parameters of Batch Normalization or Layer Normalization. The settings are shown in Table 10.

C.3 IMPLEMENTATION NOTES

All implementations were based on PyTorch, and an AlexNet backbone was used for LPIPS evaluation. Input images were processed after normalization and, if necessary, converted to a $[-1, 1]$ scale. The experiments were run on an NVIDIA GPU environment, and the overhead for each method was measured as the processing time per sample.

D PROOFS OF THEOREMS

D.1 T1: LEMMA AND PROOF FOR LYAPUNOV DECREASE

Assumption 1. f is C^1 , V is C^1 , and J is locally bounded. J^\dagger represents the standard Moore–Penrose pseudoinverse.

Table 10: Hyperparameter settings for TTA methods

Method	Iterations	Learning Rate	Additional Settings
Tent	10	1×10^{-3}	Momentum = 0.9
EATA	10	5×10^{-4}	Conf. Thr = 0.7, Fisher scale = 1×10^{-3}
MEMO	10	1×10^{-3}	$k = 4$ augmentations
SAR	10	1×10^{-3}	Rho = 0.05, Ent. Thr = 1.5

Lemma D.1 (Projected Form of Output Dynamics). *For the DOC dynamics*

$$\dot{x} = -J^\dagger \nabla_y V(y) \quad (\text{D.1})$$

the time evolution of the output is

$$\dot{y} = -P_{\text{Im } J} \nabla_y V(y), \quad P_{\text{Im } J} := JJ^\dagger \quad (\text{D.2})$$

$P_{\text{Im } J}$ is the orthogonal projection onto $\text{Im } J$, and is symmetric and idempotent.

Proof. By definition,

$$\dot{y} = J\dot{x} = -JJ^\dagger \nabla_y V(y) = -(JJ^\dagger) \nabla_y V(y). \quad (\text{D.3})$$

The conclusion follows from the fact that JJ^\dagger is an orthogonal projection. \square

Proof of Theorem T1. From the chain rule and Lemma D.1,

$$\dot{V}(y) = \langle \nabla_y V, \dot{y} \rangle = -\langle \nabla_y V, P_{\text{Im } J} \nabla_y V \rangle = -\|P_{\text{Im } J}^{1/2} \nabla_y V\|_2^2. \quad (\text{D.4})$$

Thus, $\dot{V}(y) \leq 0$. Equality holds only when $\nabla_y V \in \ker J^\top$. \square

D.2 T2: LEMMA AND PROOF FOR MINIMUM-NORM CONTROL

Lemma D.2 (Minimum-Norm Solution for Linear Constraints). *For $J \in \mathbb{R}^{k \times d}$ and $v \in \text{Im } J$, the solution to the equation $Ju = v$ can be expressed as*

$$u = J^\dagger v + (I - J^\dagger J)z, \quad z \in \mathbb{R}^d \quad (\text{D.5})$$

In particular, $u_\star = J^\dagger v$ is the unique minimum-norm solution.

Proof. Using the property $JJ^\dagger v = v$, we have

$$Ju = JJ^\dagger v + J(I - J^\dagger J)z = v. \quad (\text{D.6})$$

Also, since $J^\dagger v \perp \ker J$,

$$\|u\|_2^2 = \|J^\dagger v\|_2^2 + \|(I - J^\dagger J)z\|_2^2 \geq \|J^\dagger v\|_2^2. \quad (\text{D.7})$$

Equality holds only when $z = 0$. \square

Proof of Theorem T2. Let $v = -P_{\text{Im } J} g \in \text{Im } J$. From Lemma D.2, the minimum-norm solution is

$$u = J^\dagger v. \quad (\text{D.8})$$

Since $J^\dagger P_{\text{Im } J} = J^\dagger$, it follows that

$$u = -J^\dagger g. \quad (\text{D.9})$$

\square

D.3 T3: LEMMA AND PROOF FOR NATURAL GRADIENT EQUIVALENCE

Assumption 2 (Neighborhood of Softmax+CE). *The final layer is softmax $y = \text{softmax}(z)$, and the objective function is $V(y) = D_{\text{KL}}(y^* \| y)$. The Jacobian of softmax $S(y) = \text{diag}(y) - yy^\top$ has $\text{rank}(S) = k - 1$, and $\text{Im } S = T_y \Delta^{k-1}$. Furthermore, since $J = \nabla_x y = S(y) \nabla_x z$, locally $\text{Im } J = T_y \Delta^{k-1}$ holds.*

Lemma D.3 (Fisher Metric on the Simplex). *$F(y) = \text{diag}(y) - yy^\top$ is positive definite on $T_y \Delta^{k-1}$, and the natural gradient is*

$$\text{grad}^F V(y) = F^+ \nabla_y V(y). \quad (\text{D.10})$$

Lemma D.4 (Relation between Euclidean Projection and Fisher Projection). *For any $w \in \mathbb{R}^k$,*

$$P_T w = F F^+ w, \quad (\text{D.11})$$

where P_T is the orthogonal projection onto $T = T_y \Delta^{k-1}$.

Proof of Theorem T3. By assumption, $\text{Im } J = T$. Therefore,

$$\dot{y} = -P_T \nabla_y V = -F F^+ \nabla_y V = -F \text{grad}^F V. \quad (\text{D.12})$$

Since F is positive definite on T , it has the same trajectory as the natural gradient flow up to a time reparameterization. \square

D.4 T4: IRREPARABILITY DIAGNOSIS (THEORETICAL LOWER BOUND)

Theorem D.5 (Irreparability Condition). *Let $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ be differentiable in a neighborhood of x . Let $\|\cdot\|$ be any norm, and $\|A\|$ its induced operator norm. For a target set $\mathcal{T} \subset \mathbb{R}^m$, define*

$$\gamma := \text{dist}(f(x), \mathcal{T}) = \inf_{y \in \mathcal{T}} \|y - f(x)\|. \quad (\text{D.13})$$

Let $J := Df(x)$. Under the input constraint $\|\delta x\| \leq \varepsilon$, if

$$\varepsilon < \frac{\gamma}{\|J\|} \quad (\text{D.14})$$

holds, then (within first-order approximation) $f(x + \delta x) \notin \mathcal{T}$, i.e., repair is impossible.

Proof. Since f is differentiable at x , the expansion

$$f(x + \delta x) = f(x) + J \delta x + r(\delta x), \quad \frac{\|r(\delta x)\|}{\|\delta x\|} \rightarrow 0 \quad (\text{D.15})$$

holds. For any δx ,

$$\|f(x + \delta x) - f(x)\| \leq \|J\| \|\delta x\| + \|r(\delta x)\|. \quad (\text{D.16})$$

Thus, for sufficiently small $\varepsilon_0 > 0$, for all $0 < \|\delta x\| \leq \varepsilon_0$,

$$\|r(\delta x)\| \leq \frac{1}{2} \|J\| \|\delta x\|. \quad (\text{D.17})$$

Then,

$$\|f(x + \delta x) - f(x)\| \leq \frac{3}{2} \|J\| \|\delta x\|. \quad (\text{D.18})$$

If $\varepsilon \leq \min\{\varepsilon_0, \gamma/\|J\|\}$ and $\varepsilon < \gamma/\|J\|$, then for any $\|\delta x\| \leq \varepsilon$,

$$\|f(x + \delta x) - f(x)\| < \|J\| \varepsilon \leq \gamma. \quad (\text{D.19})$$

But by definition of γ , this contradicts $f(x + \delta x) \in \mathcal{T}$. Hence repair is impossible. \square

Algorithm 1 Direct Output Control (DOC): N -step repair at inference time

Require: Pre-trained f , input x_0 , target y^* , number of steps N , step sizes $\{\alpha_t\}$, damping λ , active set size k

- 1: **for** $t = 0$ **to** $N - 1$ **do**
- 2: $y_t \leftarrow f(x_t)$, $\psi_t \leftarrow \sqrt{y_t}$, $\theta_t \leftarrow \arccos(\langle \psi_t, \sqrt{y^*} \rangle)$
- 3: $v_{y_t} \leftarrow \Pi_{y_t} \nabla_y (\theta_t^2)$
- 4: $S_t \leftarrow \text{TopK}(z(x_t), k) \cup \{\arg \max y^*\}$
- 5: $J_S \leftarrow \frac{\partial(y_t)_{S_t}}{\partial x}$ (autograd)
- 6: $u_t \leftarrow -J_S^\top (J_S J_S^\top + \lambda I)^{-1} (v_{y_t})_{S_t}$
- 7: $x_{t+1} \leftarrow x_t + \alpha_t u_t$
- 8: **end for**
- 9: **return** x_N

D.5 T5: FULL STATEMENT ON DISCRETIZATION CONVERGENCE

Assumption 3 (Smoothness and Armijo Condition (Revised)). V is L -smooth, and $P_k = J(x_k)J(x_k)^\dagger$ is an orthogonal projection. The regularized version $P_k = J(x_k)J(x_k)_\lambda^\dagger$ is semi-definite, not necessarily idempotent.

Lemma D.6 (Sufficient Decrease and Lower Bound on Step Size).

$$V(y_{k+1}) \leq V(y_k) - c\alpha_k \|P_k g_k\|^2, \quad (\text{D.20})$$

and furthermore

$$\alpha_k \geq (1 - \beta)/L \quad (\text{D.21})$$

holds.

Theorem D.7 (Finite Iteration Convergence). (i) $V(y_{k+1}) \leq V(y_k)$ holds for all k . (ii) $\sum_k \alpha_k \|P_k g_k\|^2 < \infty$, so $\|P_k g_k\| \rightarrow 0$.

Furthermore, if the PL inequality

$$\frac{1}{2\mu} \|Pg\|^2 \geq V(y) - V^* \quad (\text{D.22})$$

holds, then

$$V(y_k) - V^* \leq (1 - c\alpha_{\min}\mu)^k (V(y_0) - V^*). \quad (\text{D.23})$$

Proof. This follows from L -smoothness and Armijo conditions. Since P_k is a projection, $\|P_k g_k\| \leq \|g_k\|$. Linear convergence follows from the PL inequality. \square

Remark. For $\alpha \rightarrow 0$, this converges to the ODE $\dot{y} = -P_{\text{Im } J} \nabla_y V$.

E ODE STABILITY AND PSEUDOCODE

This section provides pseudocode to clarify the implementation procedure of DOC. The implementation is designed to be easily described within a standard automatic differentiation framework.

E.1 PSEUDOCODE

Algorithm 1 takes an input x_0 and a target distribution y^* and provides a procedure to sequentially repair the input by using the distance on the output distribution as a Lyapunov function. Each step performs the following operations: (i) Calculate the angular distance θ_t from the output y_t and the target y^* . (ii) Project its gradient to obtain the direction vector v_{y_t} . (iii) Obtain the Jacobian J_S for the top k classes using automatic differentiation. (iv) Determine the input-side update direction u_t through regularized pseudoinverse calculation. (v) Update the input according to the learning rate α_t .