DAMPER: RETURN-PRIORITIZED GRADIENT CONTROL FOR SMOOTH POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Actor–critic methods achieve strong performance in continuous control, but their policies can produce highly oscillatory actions. A common remedy is to add auxiliary smoothness losses. However, their contribution can be negligible when their gradients are small relative to the native actor gradient. Moreover, existing methods often combine multiple auxiliary losses, complicating loss balancing without necessarily improving the return–smoothness trade-off. We introduce DAMPER (Direction-Aware Magnitude-Controlled Projection with Explicit Return Priority), which combines the native actor gradient with a temporal-consistency gradient through conflict-conditioned projection and adaptive magnitude control. It removes the auxiliary component opposing the actor gradient and scales the retained temporal direction relative to the actor gradient norm, preserving positive alignment with the native actor gradient. Experiments with TD3 and SAC on six continuous-control tasks show reduced action oscillation relative to the native agents in all 12 task–backbone pairs and the best oscillation score among the compared methods in eight, with task-dependent return trade-offs. Our code is available at `https://anonymous.4open.science/r/DAMPER-CDA1`.

1 INTRODUCTION

Actor–critic algorithms are effective tools for continuous control, but high episodic return does not imply a well-behaved control signal. A learned policy may vary its action sharply between adjacent time steps even when the system state changes only slightly. Simulation often tolerates these oscillations; physical platforms do not. Rapid control changes can waste energy, amplify unmodeled dynamics, and increase actuator wear (de la Presilla et al., 2023). Learning policies that are both effective and temporally smooth is therefore an important requirement for deploying reinforcement learning on real systems.

Most existing approaches introduce smoothness through an auxiliary loss. They regularize policy outputs at consecutive or nearby states, constrain local policy sensitivity, or stabilize the critic gradients used by the actor (Mysore et al., 2021; Kobayashi, 2022; Lee et al., 2024; Kwak & Hwang, 2026; Lee et al., 2026). These methods can reduce action variation, but the smoothing they achieve is often limited. Moreover, many existing methods introduce two or more auxiliary losses to control smoothness (Mysore et al., 2021; Kobayashi, 2022; Kwak & Hwang, 2026; Lee et al., 2026), which complicates loss balancing without necessarily improving the return–smoothness trade-off, as illustrated by our comparison with a tuned single-loss baseline in Section 6.4.

We approach this problem as primary–auxiliary gradient optimization. The primary objective is the native actor loss of the underlying algorithm. The auxiliary objective is the established consecutive-state consistency penalty $\mathbb{E}_{(s,s')}[\|\pi_\theta(s') - \pi_\theta(s)\|_2^2]$, where π_θ is the deterministic action used for evaluation. Our contribution is not this temporal loss itself, but a rule for combining its gradient with the native actor gradient.

We introduce DAMPER, which combines the native actor and temporal gradients through two operations (Figure 1). First, when the temporal gradient conflicts with the native actor gradient, DAMPER removes only its return-opposing component. The native actor gradient is never projected, and an aligned temporal component is retained rather than discarded. Second, DAMPER controls the remaining direction relative to the native gradient norm. A bounded parameter $\eta \in [0, 1]$ interpolates between two endpoints: at $\eta = 0$, the temporal component is only capped at the native gradient norm,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

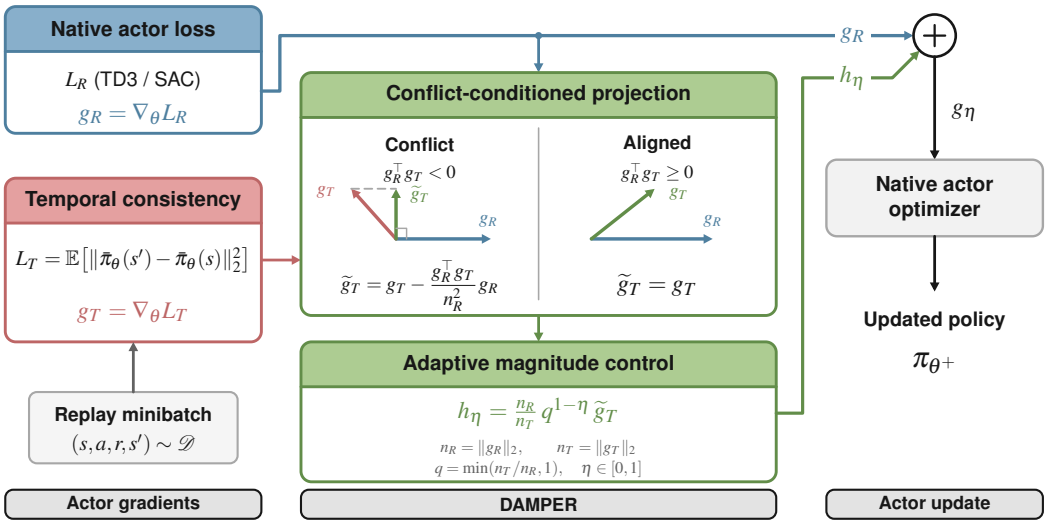

Figure 1: **Overview of DAMPER.**

and at $\eta = 1$, it is rescaled to that norm. Unlike a scalar regularization coefficient, η controls how the observed gradient magnitudes are reconciled after the objectives have been differentiated. Figure 1 provides an overview of these two operations.

Across six continuous-control tasks and two backbones, DAMPER improves the FFT-based oscillation score over the corresponding unregularized agent in all 12 task–backbone pairs and achieves the best score among the compared methods in 8. The improvement is more consistent for action stability than for return: several locomotion settings exhibit a clear smoothness–return trade-off. We report this limitation explicitly and isolate the role of conflict-conditioned projection in an ablation study.

Our contributions are as follows:

- We introduce DAMPER, a return-prioritized actor update that combines conflict-conditioned projection with norm-ratio interpolation.
- We establish finite-step descent for the fixed actor surrogate and characterize temporal progress at matched first-order primary progress.
- DAMPER reduces action oscillation relative to native TD3 and SAC on all six tasks, achieving the best score in eight of 12 task–backbone pairs. Projection ablations, interpolation sweeps, and fixed-weight scalarization comparisons assess the contributions of directional and magnitude control.

2 Related Work

2.1 Policy Smoothness in Reinforcement Learning

Policy-output regularization. CAPS combines temporal consistency between consecutive states with spatial consistency under state perturbations (Mysore et al., 2021). Closely related work directly penalizes consecutive deterministic policy outputs (De Cooman et al., 2021). We use the same form of temporal surrogate, but do not add it to the native actor loss with a fixed coefficient. Instead, we modify its gradient according to its alignment and scale relative to the native actor gradient.

Sensitivity and critic regularization. Grad-CAPS regularizes policy sensitivity (Lee et al., 2024), ASAP combines action alignment, action prediction, and second-order temporal regularization (Kwak & Hwang, 2026), and L2C2 constrains local changes in both the policy and value function (Kobayashi, 2022). PAVE instead regularizes the critic to stabilize the action-gradient field that drives policy learning (Lee et al., 2026). These approaches encode richer notions of regularity than our single temporal loss, but multi-component scalarized objectives introduce additional relative weights.

DAMPER is complementary: it focuses on how a temporal auxiliary gradient should enter the actor update when the native actor objective remains primary.

Structured smooth control. Smooth behavior can also arise from architectural or action-space structure. LipsNet controls policy sensitivity through an adaptive Lipschitz architecture (Song et al., 2023); generalized state-dependent exploration produces temporally correlated exploration noise (Raffin et al., 2022); and action-rate parameterizations smooth the executed control by construction (Chisari et al., 2021). In contrast, DAMPER changes only training-time actor gradients and leaves the policy architecture and execution interface unchanged.

2.2 GRADIENT COMBINATION AND OBJECTIVE PRIORITY

Multi-objective optimization. MGDA searches for a common descent direction (Désidéri, 2012; Sener & Koltun, 2018), GradNorm balances task-gradient magnitudes (Chen et al., 2018), PCGrad removes conflicting gradient components (Yu et al., 2020), and CAGrad constructs a conflict-averse common update (Liu et al., 2021). These methods typically treat objectives symmetrically. Our setting is hierarchical: the native actor objective is always primary, so its gradient is never altered to accommodate temporal consistency.

Primary–auxiliary learning. Gradient-similarity weighting (Du et al., 2018), Dynamic Barrier (Gong et al., 2021), and Bloop (Hsieh et al., 2024) prevent auxiliary objectives from degrading a primary task, while MetaBalance (He et al., 2022) adapts auxiliary-gradient magnitudes to a target gradient. DAMPER combines the two relevant ideas: it conditionally removes only the primary-opposing temporal component and separately controls its magnitude through a bounded norm-ratio interpolation.

Priority-aware reinforcement learning. PEGrad prioritizes reward over energy minimization by orthogonalizing and capping the lower-priority gradient (Peri et al., 2025); GCR-PPO conditionally projects lower-priority reward components (Munn et al., 2026); and LPPG-RL considers lexicographically ordered policy-gradient objectives (Qiu et al., 2026). FCGrad addresses individual–collective return conflicts in mixed-motive multi-agent learning and changes priority according to current returns (Kim & Sycara, 2025). Our hierarchy is fixed, projection occurs only under conflict, and the retained temporal direction is scaled by an explicit interpolation between cap-only and norm-balanced updates. The contribution is this combination for temporal policy smoothing, rather than gradient projection in isolation.

3 PRELIMINARIES

Actor–critic learning. We consider a discounted Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, r, \gamma)$, with transition kernel $p(s' \mid s, a)$, reward $r(s, a)$, and discount factor $\gamma \in [0, 1)$. A policy π_θ seeks to maximize the expected return

$$J(\theta) := \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right].$$

Actor–critic methods use a learned action-value function $Q_\phi(s, a)$ to guide policy optimization. We adopt the loss-minimization convention and denote the underlying algorithm's native actor loss by $L_{\mathrm{R}}(\theta)$. Algorithm-specific definitions are provided in Appendix E.

Temporal policy consistency. Let $\pi_\theta(s)$ denote the differentiable deterministic action map associated with the current policy. For consecutive states (s, s') drawn from an observed transition distribution $\mathcal{D}$, we define the temporal-consistency loss

$$L_{\mathrm{T}}(\theta) := \mathbb{E}_{(s,s')\sim\mathcal{D}}\left[\|\pi_\theta(s') - \pi_\theta(s)\|_2^2\right]. \tag{1}$$

Both actions are computed using the same current policy. Minimizing this loss encourages similar actions at consecutive states, promoting temporal consistency along environment trajectories.

Combining objectives. We denote the actor-parameter gradients by

$$g_R := \nabla_\theta L_{\mathrm{R}}(\theta), \qquad g_T := \nabla_\theta L_{\mathrm{T}}(\theta). \tag{2}$$

The term "return gradient" refers to g_R, the critic-based actor-loss gradient, rather than a direct gradient of measured episodic return. A conventional auxiliary-loss formulation minimizes

$$\mathcal{L}_\lambda(\theta) := L_{\mathrm{R}}(\theta) + \lambda L_{\mathrm{T}}(\theta), \qquad \lambda > 0, \tag{3}$$

which yields the combined gradient $g_R + \lambda g_T$.

4 METHOD

We introduce DAMPER, an actor update rule designed to reduce action oscillation while prioritizing return maximization. It uses a temporal-consistency objective to encourage similar actions at consecutive states, with the native actor objective taking priority. To combine their gradients, DAMPER first removes any temporal component that opposes the native actor gradient, then controls the remaining contribution relative to the native gradient norm. These two operations address directional conflict and scale imbalance when incorporating temporal consistency into the actor update. Figure 1 summarizes the complete update.

4.1 CONFLICT-CONDITIONED PROJECTION

We use conflict-conditioned projection to prevent the temporal gradient from hindering first-order progress on the native actor objective. Let $g_R = \nabla_\theta L_{\mathrm{R}}$ and $g_T = \nabla_\theta L_{\mathrm{T}}$ denote the gradients of the objectives defined in Section 3. When $g_R^\top g_T < 0$, the temporal gradient contains a component that opposes the native actor gradient. We remove this component by projecting g_T onto the subspace orthogonal to g_R. Otherwise, we leave g_T unchanged. Denoting the resulting temporal gradient by $\widetilde{g}_T$, we obtain

$$\widetilde{g}_T := \begin{cases} g_T - \dfrac{g_R^\top g_T}{\|g_R\|_2^2} g_R, & g_R^\top g_T < 0, \\ g_T, & g_R^\top g_T \geq 0. \end{cases} \tag{4}$$

The native gradient g_R is never modified. The retained temporal gradient satisfies $g_R^\top \widetilde{g}_T \geq 0$, so adding a nonnegative multiple of $\widetilde{g}_T$ preserves the first-order decrease in the native actor loss provided by g_R. The Conflict and Aligned panels in Figure 1 illustrate the two cases.

4.2 ADAPTIVE MAGNITUDE CONTROL

Projection resolves directional conflict, but a temporal gradient that is small relative to the native actor gradient may have little influence on policy smoothness, whereas a much larger one may dominate the actor update. We therefore rescale the temporal gradient according to the relative norms of the original actor and temporal gradients.

For nonzero gradients, let $n_R := \|g_R\|_2$ and $n_T := \|g_T\|_2$. Before accounting for directional conflict, consider the target combined gradient

$$g_{\text{target}}(s) := g_R + s\frac{g_T}{n_T}, \qquad s > 0.$$

This target adds a temporal contribution of magnitude s along g_T to the unchanged actor gradient g_R.

The target may, however, reduce first-order progress on the native actor objective relative to using g_R alone. We therefore make the smallest adjustment, measured in Euclidean distance, to $g_{\text{target}}(s)$ that preserves this progress:

$$g(s) := \operatorname*{arg\,min}_{d : g_R^\top d \geq \|g_R\|_2^2} \|d - g_{\text{target}}(s)\|_2^2.$$

The constraint requires at least the first-order decrease in the native actor loss achieved by g_R. It can be shown (Appendix A) that the unique solution to the optimization problem is

$$g(s) = g_R + \frac{s}{n_T}\widetilde{g}_T,$$

Algorithm 1: DAMPER actor update

Input: Actor parameters θ, replay buffer $\mathcal{D}$, native actor loss L_{R}, interpolation $\eta \in [0, 1]$.

Sample a minibatch $\mathcal{B}$ from $\mathcal{D}$
Evaluate $L_{\mathrm{R}}(\theta)$ and $L_{\mathrm{T}}(\theta)$ in equation 1
$g_R \leftarrow \nabla_\theta L_{\mathrm{R}}$, $g_T \leftarrow \nabla_\theta L_{\mathrm{T}}$
$n_R \leftarrow \|g_R\|_2$, $n_T \leftarrow \|g_T\|_2$
$\widetilde{g}_T \leftarrow g_T$
if $g_R^\top g_T < 0$ **then**
    $\widetilde{g}_T \leftarrow g_T - \frac{g_R^\top g_T}{n_R^2} g_R$
$q \leftarrow \min(n_T/n_R, 1)$, $w_\eta \leftarrow q^{1-\eta}$
$g \leftarrow g_R + (n_R/n_T) w_\eta \widetilde{g}_T$
Assign g to the actor-gradient buffers and apply the native actor optimizer

where $\widetilde{g}_T$ is exactly the conflict-conditioned projection from the previous subsection.

We consider two reference target magnitudes. The *cap-only* choice $s_0 \coloneqq \min(n_T, n_R)$ caps the target temporal magnitude at the native actor gradient norm n_R, without amplifying weak temporal gradients. The *norm-matched* choice $s_1 \coloneqq n_R$ instead rescales the temporal contribution to n_R before projection, amplifying it when $n_T < n_R$.

To obtain intermediate behavior, we geometrically interpolate between these choices using a parameter $\eta \in [0, 1]$:

$$s_\eta \coloneqq s_0^{1-\eta} s_1^{\eta}, \qquad g_\eta \coloneqq g(s_\eta) = g_R + \frac{s_\eta}{n_T} \widetilde{g}_T. \tag{5}$$

Equivalently, defining $q \coloneqq \min(n_T/n_R, 1)$ and $w_\eta \coloneqq q^{1-\eta}$ gives $s_\eta = n_R w_\eta$. For $n_T < n_R$, increasing η strengthens the temporal gradient.

4.3 DAMPER ACTOR UPDATE

Algorithm 1 summarizes the core actor update for nonzero gradients. DAMPER computes the two gradients separately, applies conflict-conditioned projection and adaptive magnitude control, and passes the merged gradient to the native actor optimizer.

5 ANALYSIS OF THE PRIMARY–AUXILIARY UPDATE

We analyze a single Euclidean actor step using the direction constructed by Algorithm 1, with the critic and sampled transitions held fixed.

5.1 DESCENT ON THE NATIVE ACTOR LOSS

Consistent with our primary–auxiliary formuation, in which the native actor objective tasks priority over temporal consistency, we first show that DAMPER preserves descent on the native actor loss even when the two gradients conflicts.

Theorem 1 (Finite-step descent for the fixed actor surrogate)**.** *Assume that L_{R} is differentiable and globally β_R-smooth for $\beta_R > 0$, and that $g_R = \nabla_\theta L_{\mathrm{R}}(\theta) \neq 0$. Let g be the direction returned by Algorithm 1. Then every step*

$$\theta^+ = \theta - \alpha g, \qquad 0 < \alpha < \frac{1}{\beta_R}, \tag{6}$$

strictly decreases the fixed native actor loss: $L_{\mathrm{R}}(\theta^+) < L_{\mathrm{R}}(\theta)$.

Thus, for sufficiently small steps, DAMPER incorporates the temporal gradient while preserving descent on the native actor loss. The proof is in Appendix B.1.

5.2 Temporal Progress at Matched Primary Progress

We next compare the temporal progress of DAMPER and fixed-weight scalarization. For this comparison, we rescale each update to give the same predicted decrease in the native actor loss, using a first-order approximation. This holds primary improvement fixed, allowing us to compare the temporal benefit of the update directions independently of their overall scale. For any direction d with $g_R^\top d > 0$, define

$$\mathcal{G}(d) := \frac{g_T^\top d}{g_R^\top d} - \frac{g_T^\top g_R}{\|g_R\|_2^2}. \tag{7}$$

The first term measures the predicted temporal decrease per unit of predicted native-actor-loss decrease; subtracting the corresponding quantity for the native-gradient update gives the additional temporal gain.

Using this measure, the next result shows that, for noncollinear gradients and fixed η, DAMPER achieves greater additional temporal gain than scalarization with any fixed positive weight when the temporal gradient is sufficiently small relative to the native actor gradient.

Theorem 2 (Temporal gain in the weak-gradient regime)**.** *Let $g_R = \nabla_\theta L_R(\theta)$ and $g_T = \nabla_\theta L_T(\theta)$ be nonzero, and define*

$$q := \min\left\{\frac{\|g_T\|_2}{\|g_R\|_2}, 1\right\}, \qquad c := \frac{g_R^\top g_T}{\|g_R\|_2\|g_T\|_2}.$$

Fix $\eta \in [0,1]$ and any $\lambda > 0$. Assume $0 < q < 1$ and $\lambda q < 1 - \delta$ for some fixed $\delta \in (0,1)$. For the analytic DAMPER *direction g_η in equation 5 and $g_\lambda = g_R + \lambda g_T$, the matched temporal gains satisfy*

$$\mathcal{G}(g_\eta) = \Theta\big((1-c^2)q^{2-\eta}\big), \qquad \mathcal{G}(g_\lambda) = \Theta\big((1-c^2)q^2\big). \tag{8}$$

Here, Θ denotes two-sided bounds with positive constants independent of q and c, but possibly dependent on λ and δ.

The result highlights the role of magnitude control in the weak-temporal regime, where the temporal gradient is small relative to the native actor gradient. For fixed $\eta > 0$ and $\lambda > 0$, DAMPER provides greater additional temporal gain than scalarization and sufficiently small q, when the first-order primary progress is matched. Appendix B.2 gives the proof.

6 Experiments

6.1 Experimental Setup

In this subsection, we describe the tasks, baselines, and evaluation protocol used to assess DAMPER.

Tasks and backbones. We consider six continuous-control tasks from Gymnasium (Towers et al., 2024) and MuJoCo (Todorov et al., 2012): LunarLander, Pendulum, Reacher, Ant, Hopper, and Walker. Each method is evaluated with both TD3 and SAC. Our implementation and protocol follow the PAVE codebase, and all neural networks use SiLU activations.

Baselines. We compare with the unregularized TD3 and SAC agents and five smooth-control methods: CAPS, Grad-CAPS, ASAP, L2C2, and PAVE. The comparison therefore spans policy-output regularization, policy-sensitivity regularization, joint policy–value constraints, and critic-gradient stabilization.

Evaluation protocol. Each method is trained with five independent random seeds. For each seed, the learned policy is evaluated for ten episodes; tables report the mean and standard deviation across seeds. We measure cumulative episodic return (re, higher is better) and action oscillation (sm, lower is better). The latter is a spectral smoothness metric computed using the fast Fourier transform (FFT), following Mysore et al. (2021); Christmann et al. (2024).

Table 1: **TD3 results.** Entries are mean (standard deviation) across five seeds. Higher return (re) and lower oscillation score (sm) are better. The best value in each column is highlighted in **bold**.

	LunarLander		Pendulum		Reacher		Ant		Hopper		Walker	
Method	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$
TD3	205.5 (98.5)	1.815 (0.955)	−157.6 (84.0)	2.308 (1.422)	**−3.34** (1.45)	0.050 (0.015)	4749 (1226)	2.044 (0.307)	**3604** (150)	3.061 (0.421)	4596 (328)	1.937 (0.139)
CAPS	246.1 (81.6)	0.634 (0.233)	−159.4 (81.7)	0.454 (0.140)	**−3.34** (1.47)	0.046 (0.015)	**5370** (275)	2.064 (0.101)	2928 (1025)	1.730 (0.341)	5032 (241)	1.865 (0.176)
Grad-CAPS	225.5 (61.4)	0.905 (0.333)	**−153.4** (77.9)	1.011 (0.503)	**−3.34** (1.49)	**0.040** (0.011)	4201 (1859)	1.713 (0.408)	3371 (619)	1.287 (0.128)	**5046** (460)	1.412 (0.291)
ASAP	245.5 (58.6)	1.345 (0.616)	−155.3 (78.3)	2.013 (1.047)	−3.36 (1.44)	0.048 (0.015)	4925 (1210)	2.041 (0.343)	2581 (1448)	1.866 (0.883)	4880 (745)	1.615 (0.288)
L2C2	218.5 (83.3)	1.626 (0.727)	−156.0 (79.1)	3.293 (1.034)	−3.38 (1.52)	0.048 (0.014)	4078 (1658)	1.974 (0.402)	2807 (1371)	2.427 (1.087)	4960 (1048)	1.707 (0.320)
PAVE	238.1 (71.5)	0.841 (0.601)	−154.5 (80.4)	**0.410** (0.121)	−3.55 (1.51)	0.046 (0.016)	5064 (1245)	1.865 (0.237)	3328 (541)	1.022 (0.148)	4996 (700)	1.991 (0.410)
DAMPER (ours)	**266.4** (35.8)	**0.313** (0.036)	−154.6 (78.7)	0.463 (0.131)	−3.63 (1.44)	0.048 (0.016)	5281 (887)	**1.261** (0.147)	3078 (599)	**0.240** (0.035)	4275 (322)	**0.346** (0.088)

6.2 Overall Performance

In this subsection, we show that DAMPER consistently reduces action oscillation across TD3 and SAC, with task-dependent effects on return. It lowers sm relative to the unregularized agents in all 12 task–backbone pairs and achieves the lowest score among the compared methods in eight (Tables 1 and 2). Figure 2 also shows smaller consecutive action changes on Walker for both backbones.

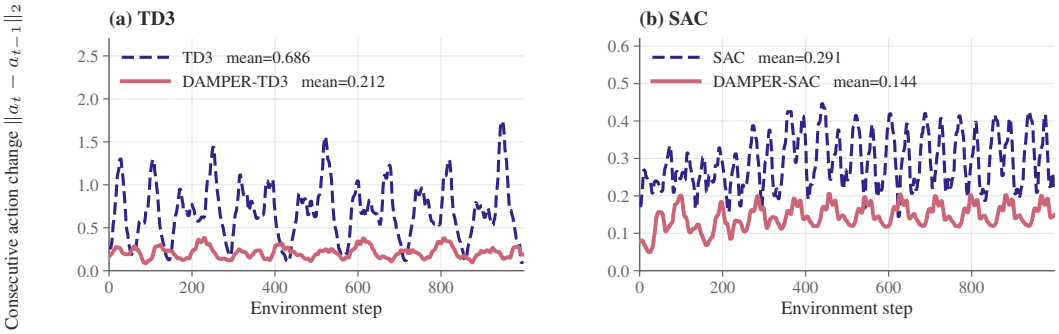

Figure 2: DAMPER reduces consecutive action variation on Walker under both actor–critic backbones. We plot $\|a_t - a_{t-1}\|_2$ over a 1,000-step rollout.

TD3 results. DAMPER achieves the lowest oscillation score on LunarLander, Ant, Hopper, and Walker (Table 1). On LunarLander, it improves both metrics, increasing return from 205.5 to 266.4 and reducing sm from 1.815 to 0.313. Hopper and Walker also show large reductions in sm, from 3.061 to 0.240 and from 1.937 to 0.346, respectively, accompanied by lower returns than unregularized TD3.

SAC results. DAMPER achieves the lowest oscillation score on LunarLander, Reacher, Ant, and Walker (Table 2). It also obtains the highest return on LunarLander and Hopper; on Hopper, sm decreases from 0.708 to 0.571 relative to unregularized SAC. On Ant and Walker, the lowest oscillation scores are accompanied by lower returns than the unregularized agent, again showing that the return–smoothness balance depends on the task.

6.3 Ablation Studies

Projection Ablation. We show that conflict-conditioned projection achieves the highest mean return among the evaluated update rules on three TD3 tasks: LunarLander, Hopper, and Walker. We compare DAMPER with *Unconditional Sum*, which always adds the scaled temporal direction, and

Table 2: **SAC results.** Entries are mean (standard deviation) across five seeds. Higher return (re) and lower oscillation score (sm) are better. The best value in each column is highlighted in **bold**.

	LunarLander		Pendulum		Reacher		Ant		Hopper		Walker	
Method	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$
SAC	162.6 (139.9)	0.402 (0.143)	−149.9 (76.9)	0.479 (0.128)	−3.49 (1.43)	0.053 (0.016)	5001 (1145)	1.935 (0.234)	3059 (825)	0.708 (0.074)	4771 (292)	0.747 (0.081)
CAPS	120.0 (171.6)	0.279 (0.065)	−152.7 (80.1)	**0.325** (0.107)	−3.48 (1.39)	0.048 (0.014)	5439 (707)	1.931 (0.170)	3222 (548)	0.653 (0.058)	4955 (296)	0.745 (0.164)
Grad-CAPS	269.0 (27.6)	0.277 (0.064)	**−148.9** (75.7)	0.379 (0.119)	−3.51 (1.41)	0.046 (0.013)	**5675** (960)	1.797 (0.248)	3144 (653)	**0.494** (0.061)	**5060** (262)	0.585 (0.067)
ASAP	270.1 (19.6)	0.303 (0.080)	−149.2 (76.0)	0.505 (0.138)	−3.46 (1.40)	0.053 (0.016)	4564 (2104)	1.727 (0.326)	2831 (908)	0.637 (0.104)	5035 (408)	0.760 (0.155)
L2C2	239.4 (58.2)	0.407 (0.168)	−149.3 (76.2)	0.510 (0.145)	**−3.45** (1.38)	0.052 (0.016)	4947 (1565)	1.810 (0.415)	3101 (580)	0.732 (0.106)	4834 (277)	0.901 (0.221)
PAVE	263.6 (24.2)	0.203 (0.034)	−151.6 (78.7)	0.334 (0.114)	−3.51 (1.49)	0.051 (0.015)	5543 (1007)	1.741 (0.218)	2403 (862)	0.584 (0.097)	4544 (646)	0.713 (0.090)
DAMPER (ours)	**281.5** (24.4)	**0.102** (0.037)	−156.6 (78.5)	0.336 (0.164)	−3.57 (1.57)	**0.038** (0.018)	4564 (1577)	**1.207** (0.253)	**3384** (273)	0.571 (0.022)	4402 (1086)	**0.268** (0.044)

Unconditional Projection, which orthogonalizes it at every update. The former retains components that oppose the native actor gradient, while the latter removes aligned components as well.

Table 3: Ablation of conflict-conditioned projection with TD3. Entries are mean (standard deviation); each variant uses five training seeds and ten evaluation episodes per seed. The numerically best mean for each metric is shown in bold.

	LunarLander		Hopper		Walker	
Method	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$	$re\uparrow$	$sm\downarrow$
DAMPER	**266.4** (35.8)	0.313 (0.036)	**3078** (599)	0.240 (0.035)	**4275** (322)	0.346 (0.088)
Unconditional projection	206.9 (122.7)	0.370 (0.223)	3050 (587)	0.249 (0.040)	3525 (1576)	0.314 (0.072)
Unconditional sum	263.5 (68.8)	**0.279** (0.113)	317 (341)	**0.053** (0.106)	3575 (514)	**0.268** (0.088)

On Hopper, unconditional summation reduces mean return from 3078 to 317, despite achieving a lower oscillation score (Table 3). This result highlights the need to assess smoothness together with task performance. Unconditional projection yields a similar Hopper return but lowers Walker return from 4275 to 3525, with greater variability. On LunarLander, unconditional summation remains close to DAMPER in mean return, whereas unconditional projection has a lower mean and greater variability. These results support resolving conflicting components while retaining aligned temporal information.

Comparison with a PEGrad-Based Baseline. We compare DAMPER with a PEGrad-based baseline (Peri et al., 2025). This baseline doesn't use conditional projection and limits the temporal gradient's magnitude, but lacks a parameter for strengthening weak temporal gradients relative to the native actor gradient. DAMPER can amplify weak temporal gradients through $w_\eta = q^{1-\eta}$. Across the three tasks, DAMPER lowers the mean oscillation score by 49–62% (Tables 3 and 4), while achieving higher mean returns on LunarLander and Hopper and a slightly lower mean return on Walker. These results are consistent with the benefit of magnitude interpolation alongside conflict-conditioned projection.

Table 4: **PEGrad-based baseline with TD3.** Mean (standard deviation); five training seeds and ten evaluation episodes per seed.

Task	$re\uparrow$	$sm\downarrow$
LunarLander	198.8 (123.5)	0.619 (0.353)
Hopper	2760 (835)	0.636 (0.046)
Walker	4361 (176)	0.712 (0.142)

6.4 Comparison with Fixed-Weight Scalarization

We evaluate fixed-weight scalarization on all six TD3 tasks. For each of the 20 weights $\lambda \in \Lambda := \{0.05, 0.10, \ldots, 1.0\}$, we train agents using

$$\mathcal{L}_\lambda(\theta) := L_{\mathrm{R}}(\theta) + \lambda L_{\mathrm{T}}(\theta), \qquad \lambda \in \Lambda, \tag{9}$$

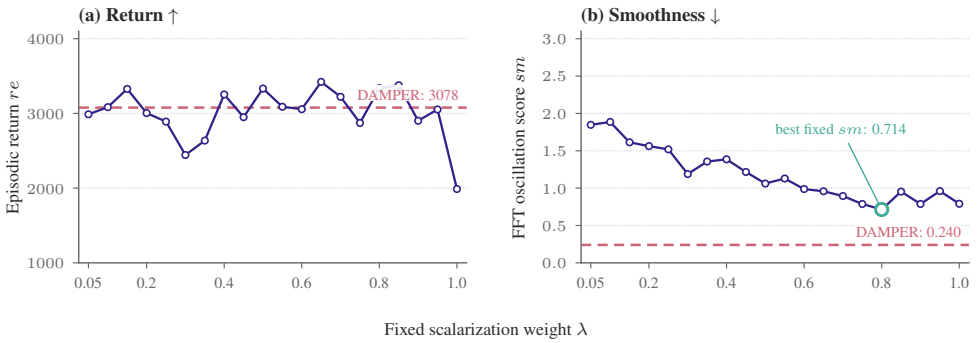

Figure 3: Fixed-weight scalarization on TD3 Hopper. Points show means for $\lambda = 0.05, 0.10, \ldots, 1.0$; dashed lines mark DAMPER.

with λ fixed throughout training. Figure 3 shows the Hopper comparison with DAMPER at $\eta = 1$.

Across the six tasks, DAMPER achieves a lower mean oscillation score than every evaluated fixed weight on five tasks, with Reacher as the exception. On Hopper, even the smoothest scalarized policy, obtained at $\lambda = 0.8$, has $sm = 0.714$, compared with 0.240 for DAMPER. These results are consistent with the temporal-progress analysis in Theorem 2.

Tuned scalarization nevertheless achieves competitive returns and lower oscillation than several specialized smoothness baselines. On LunarLander, Ant, and Hopper, selected weights improve both mean metrics over multiple such baselines, suggesting limited additional benefit from their more elaborate mechanisms over a well-tuned single temporal loss in these settings (Appendix C).

6.5 Effect of the Interpolation Parameter

We sweep $\eta \in \mathcal{H} = \{0, 0.2, \ldots, 1.0\}$ on LunarLander, Hopper, and Walker using TD3. For each task e, we normalize the mean oscillation score over the sweep as

$$\widetilde{sm}_e(\eta) = \frac{sm_e(\eta) - m_e}{M_e - m_e}, \qquad (10)$$

where m_e and M_e are the minimum and maximum mean scores over $\mathcal{H}$, respectively. Zero therefore denotes the smoothest tested setting for each task; absolute scores remain task-dependent.

Figure 4 shows that oscillation decreases monotonically on LunarLander and Hopper, while Walker exhibits a temporary increase at $\eta = 0.6$. All three tasks attain their lowest mean score at $\eta = 1$. This trend is consistent with Theorem 2: increasing η makes $\mathcal{G}(g_\eta)$ nondecreasing at matched first-order primary progress.

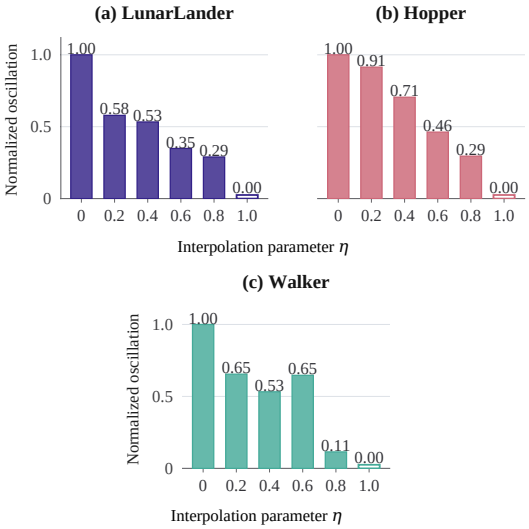

Figure 4: Effect of η on task-normalized action oscillation with TD3.

7 Conclusion

We introduced DAMPER, a primary–auxiliary actor update that combines conflict-conditioned temporal-gradient projection with norm-ratio magnitude control. The update leaves the native actor gradient unchanged, admits a local descent guarantee for a fixed critic and minibatch, and reduces the reported action-oscillation score across all 12 evaluated task–backbone pairs. Return is not uniformly preserved, however, and our theoretical analysis is limited to a single actor update rather than the full training process. Evaluation on physical systems is an important next step for understanding when smoother control can be obtained without sacrificing task performance.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and coding. For writing, AI tools were used to edit and polish author-written text for grammar, clarity, and readability, and to help format LaTeX (e.g., tables and references). For coding, AI tools were used to assist in creating and editing software code, including debugging, refactoring, and writing experiment and plotting scripts.

Generative AI tools were also used to assist with checking and refining mathematical derivations and proofs, including identifying gaps and assessing consistency between theorem statements, assumptions, and supporting arguments. These tools served as supplementary aids, with the authors retaining responsibility for the validity of all theoretical claims and proofs. AI tools were not used to originate research hypotheses or design the research methodology or experiments. Synthetic dataset generation, translation, dataset cleaning, and qualitative data analysis are not applicable to this work.

We have reviewed all AI-assisted work. All AI-edited text was checked by the authors to ensure that it accurately reflects our intended meaning and claims. All AI-assisted code was reviewed, tested, and verified for correctness by the authors before being used to produce the reported results. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 794–803. PMLR, 2018. URL https://proceedings.mlr.press/v80/chen18a.html.

Eugenio Chisari, Alexander Liniger, Alisa Rupenyan, Luc Van Gool, and John Lygeros. Learning from simulation, racing in reality. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8046–8052, 2021. doi: 10.1109/ICRA48506.2021.9562079.

Guilherme Christmann, Ying-Sheng Luo, Hanjaya Mandala, and Wei-Chao Chen. Benchmarking smoothness and reducing high-frequency oscillations in continuous control policies. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 627–634, 2024. doi: 10.1109/IROS58592.2024.10802057.

Bram De Cooman, Johan A. K. Suykens, and Andreas Ortseifen. Improving temporal smoothness of deterministic reinforcement learning policies with continuous actions. In *Proceedings of BNAIC/BENELEARN 2021*, pp. 217–240, 2021. URL https://lirias.kuleuven.be/3635727.

Román de la Presilla, Sebastian Wandel, Matthias Stammler, Markus Grebe, Gerhard Poll, and Sergei Glavatskih. Oscillating rolling element bearings: A review of tribotesting and analysis approaches. *Tribology International*, 188:108805, 2023. ISSN 0301-679X. doi: 10.1016/j.triboint.2023.108805.

Jean-Antoine Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus Mathématique*, 350(5–6):313–318, 2012. doi: 10.1016/j.crma.2012.03.014.

Yunshu Du, Wojciech M. Czarnecki, Siddhant M. Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Chengyue Gong, Xingchao Liu, and Qiang Liu. Automatic and harmless regularization with constrained and lexicographic optimization: A dynamic barrier approach. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29630–29642, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Yun He, Xue Feng, Cheng Cheng, Geng Ji, Yunsong Guo, and James Caverlee. Metabalance: Improving multi-task recommendations via adapting gradient magnitudes of auxiliary tasks. In *Proceedings of the ACM Web Conference 2022*, pp. 2205–2215, 2022. doi: 10.1145/3485447.3512093.

Yu-Guan Hsieh, James Thornton, Eugene Ndiaye, Michal Klein, Marco Cuturi, and Pierre Ablin. Careful with that scalpel: Improving gradient surgery with an EMA. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19085–19100. PMLR, 2024. URL https://proceedings.mlr.press/v235/hsieh24a.html.

Woojun Kim and Katia Sycara. Fair cooperation in mixed-motive games via conflict-aware gradient adjustment. *arXiv preprint arXiv:2508.17696*, 2025.

Taisuke Kobayashi. L2C2: Locally lipschitz continuous constraint towards stable and smooth reinforcement learning. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4032–4039, 2022. doi: 10.1109/IROS47612.2022.9981812.

Kyoleen Kwak and Hyoseok Hwang. Enhancing control policy smoothness by aligning actions with predictions from preceding states. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(27):22707–22715, 2026. doi: 10.1609/aaai.v40i27.39432.

I Lee, Hoang-Giang Cao, Cong-Tinh Dao, Yu-Cheng Chen, and I-Chen Wu. Gradient-based regularization for action smoothness in robotic control with reinforcement learning. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 603–610, 2024. doi: 10.1109/IROS58592.2024.10801464.

Jeong Woon Lee, Kyoleen Kwak, Daeho Kim, and Hyoseok Hwang. Stabilizing the Q-gradient field for policy smoothness in actor-critic. *arXiv preprint arXiv:2601.22970*, 2026.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18878–18890, 2021.

Humphrey Munn, Brendan Tidd, Peter Böhm, Marcus Gallagher, and David Howard. Scalable multi-objective robot reinforcement learning through gradient conflict resolution. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Siddharth Mysore, Bassel Mabsout, Renato Mancuso, and Kate Saenko. Regularizing action policies for smooth control with reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1810–1816, 2021. doi: 10.1109/ICRA48506.2021.9561138.

Skand Peri, Akhil Perincherry, Bikram Pandit, and Stefan Lee. Non-conflicting energy minimization in reinforcement learning based robot control. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 221–237. PMLR, 2025. URL https://proceedings.mlr.press/v305/peri25a.html.

Ruiyu Qiu, Rui Wang, Guanghui Yang, Xiang Li, and Zhijiang Shao. LPPG-RL: Lexicographically projected policy gradient reinforcement learning with subproblem exploration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(30):25009–25017, 2026. doi: 10.1609/aaai.v40i30.39689.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1634–1644. PMLR, 2022. URL https://proceedings.mlr.press/v164/raffin22a.html.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Xujie Song, Jingliang Duan, Wenxuan Wang, Shengbo Eben Li, Chen Chen, Bo Cheng, Bo Zhang, Junqing Wei, and Xiaoming Simon Wang. Lipsnet: A smooth and robust neural network with adaptive lipschitz constant for high accuracy optimal control. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32253–32272. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23b.html.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5824–5836, 2020.

A Optimization Characterization of the Combined Gradient

The following lemma establishes the closed-form solution to the optimization problem in Section 4.2.

Lemma 1 (Closest primary-preserving combined gradient)**.** *Let $g_R, g_T \in \mathbb{R}^p$ be nonzero gradients, with $n_R := \|g_R\|_2$ and $n_T := \|g_T\|_2$. For any $s > 0$, define the target combined gradient*

$$g_{\text{target}}(s) := g_R + s\frac{g_T}{n_T}.$$

Then the optimization problem

$$g(s) := \operatorname*{arg\,min}_{d \in \mathbb{R}^p:\, g_R^\top d \geq n_R^2} \|d - g_{\text{target}}(s)\|_2^2 \tag{11}$$

has the unique solution

$$g(s) = g_R + \frac{s}{n_T}\widetilde{g}_T, \qquad \widetilde{g}_T := g_T - \frac{\min\{g_R^\top g_T, 0\}}{n_R^2} g_R. \tag{12}$$

Here, $\widetilde{g}_T$ is the conflict-conditioned projection defined in Section 4.1.

Proof. Write $t := g_{\text{target}}(s)$ and $b := g_R^\top g_T$.

If $b \geq 0$, then

$$g_R^\top t = n_R^2 + \frac{s}{n_T} b \geq n_R^2.$$

Thus, t is feasible and attains an objective value of zero, making it the unique minimizer. Since $\widetilde{g}_T = g_T$ in this case, the claimed solution follows.

If $b < 0$, define

$$d^\star := t - \frac{sb}{n_T n_R^2} g_R.$$

Then $g_R^\top d^\star = n_R^2$, so $d^\star$ is feasible. Moreover, $d^\star - t$ is a positive multiple of g_R. For any feasible d, we have $g_R^\top (d - d^\star) \geq 0$, and hence

$$\begin{aligned} \|d - t\|_2^2 - \|d^\star - t\|_2^2 &= \|d - d^\star\|_2^2 + 2(d - d^\star)^\top (d^\star - t) \\ &\geq \|d - d^\star\|_2^2. \end{aligned}$$

The right-hand side is strictly positive unless $d = d^\star$. Therefore, $d^\star$ is the unique minimizer. Substituting the definition of t gives

$$d^\star = g_R + \frac{s}{n_T}\left(g_T - \frac{b}{n_R^2} g_R\right) = g_R + \frac{s}{n_T}\widetilde{g}_T,$$

as required. □

Connection to magnitude control. The cap-only and norm-matched target magnitudes are $s_0 := \min(n_T, n_R)$ and $s_1 := n_R$, respectively. For $\eta \in [0, 1]$, define

$$s_\eta := s_0^{1-\eta} s_1^\eta, \qquad q := \min\left(\frac{n_T}{n_R}, 1\right), \qquad w_\eta := q^{1-\eta}.$$

Since $s_0 = n_R q$ and $s_1 = n_R$, we obtain $s_\eta = n_R w_\eta$. Lemma 1 therefore yields

$$g_\eta = g(s_\eta) = g_R + \frac{s_\eta}{n_T}\widetilde{g}_T = g_R + \frac{n_R}{n_T} w_\eta \widetilde{g}_T, \tag{13}$$

recovering the norm-ratio combination rule.

Because $\|\widetilde{g}_T\|_2 \leq n_T$ and $s_\eta \leq n_R$, the temporal contribution satisfies

$$\|g_\eta - g_R\|_2 = s_\eta \frac{\|\widetilde{g}_T\|_2}{n_T} \leq s_\eta \leq n_R. \tag{14}$$

Thus, s_η specifies the temporal magnitude before projection, not necessarily the magnitude of the final temporal contribution. Under conflict, $\|\widetilde{g}_T\|_2 < n_T$, so this contribution is strictly smaller than s_η. Using the original norm n_T in the denominator preserves this attenuation rather than amplifying the projected gradient back to the target magnitude.

B Proofs for the Primary–Auxiliary Analysis

B.1 Proof of Theorem 1

Proof. Write $r := g_R$ and $n_R := \|r\|_2 > 0$. If Algorithm 1 activates its numerical guard, then $g = r$. The descent lemma gives

$$\begin{aligned} L_{\mathrm{R}}(\theta - \alpha r) &\le L_{\mathrm{R}}(\theta) - \alpha \|r\|_2^2 + \frac{\beta_R \alpha^2}{2} \|r\|_2^2 \\ &= L_{\mathrm{R}}(\theta) - \alpha \left(1 - \frac{\beta_R \alpha}{2}\right) n_R^2. \end{aligned} \tag{15}$$

This is strictly smaller than $L_{\mathrm{R}}(\theta)$ whenever $0 < \alpha < 2/\beta_R$, and hence under the stated condition $0 < \alpha < 1/\beta_R$.

Otherwise, $g = g_\eta$. Define $u_R := g_R/n_R$ and $\widetilde{u}_T := \widetilde{g}_T/n_T$, where $n_T := \|g_T\|_2$. Let

$$c_+ := \max\{c, 0\}, \qquad c_- := \max\{-c, 0\}. \tag{16}$$

From equation 4,

$$u_R^\top \widetilde{u}_T = c_+, \qquad \|\widetilde{u}_T\|_2^2 = 1 - c_-^2. \tag{17}$$

Using $r = n_R u_R$ and equation 5,

$$\begin{aligned} r^\top g_\eta &= n_R^2 u_R^\top (u_R + w_\eta \widetilde{u}_T) \\ &= n_R^2 (1 + w_\eta c_+) > 0. \end{aligned} \tag{18}$$

Moreover,

$$\begin{aligned} \|g_\eta\|_2^2 &= n_R^2 \|u_R + w_\eta \widetilde{u}_T\|_2^2 \\ &= n_R^2 \left[1 + 2 w_\eta c_+ + w_\eta^2 (1 - c_-^2)\right]. \end{aligned} \tag{19}$$

The descent lemma therefore yields

$$\begin{aligned} L_{\mathrm{R}}(\theta - \alpha g_\eta) - L_{\mathrm{R}}(\theta) \le & -\alpha n_R^2 (1 + w_\eta c_+) \\ & + \frac{\beta_R \alpha^2 n_R^2}{2} \left[1 + 2 w_\eta c_+ + w_\eta^2 (1 - c_-^2)\right]. \end{aligned} \tag{20}$$

The right-hand side is strictly negative when

$$0 < \alpha < \frac{2(1 + w_\eta c_+)}{\beta_R [1 + 2 w_\eta c_+ + w_\eta^2 (1 - c_-^2)]}. \tag{21}$$

Because the numerical guard is inactive, $q \in (0, 1]$, and hence $w_\eta = q^{1-\eta} \in (0, 1]$. Consequently,

$$\begin{aligned} 1 + 2 w_\eta c_+ + w_\eta^2 (1 - c_-^2) &\le 2 + 2 w_\eta c_+ \\ &= 2(1 + w_\eta c_+). \end{aligned} \tag{22}$$

The upper bound in equation 21 is therefore at least $1/\beta_R$, proving the theorem. □

B.2 Proof of Theorem 2

Matched-progress interpretation. For $g_R^\top d > 0$, define

$$\widehat{d} := \frac{\|g_R\|_2^2}{g_R^\top d}\, d.$$

Then $g_R^\top \widehat{d} = \|g_R\|_2^2$, so $\widehat{d}$ and g_R yield the same first-order decrease in the native actor loss. As $\alpha \to 0$, a Taylor expansion gives

$$\begin{aligned} & L_T(\theta - \alpha g_R) - L_T(\theta - \alpha \widehat{d}) \\ &\quad = \alpha g_T^\top (\widehat{d} - g_R) + o(\alpha) \\ &\quad = \alpha \|g_R\|_2^2 \left[\frac{g_T^\top d}{g_R^\top d} - \frac{g_T^\top g_R}{\|g_R\|_2^2}\right] + o(\alpha) \\ &\quad = \alpha \|g_R\|_2^2 \mathcal{G}(d) + o(\alpha). \end{aligned} \tag{23}$$

Hence, $\mathcal{G}(d)$ measures the additional first-order temporal decrease per unit primary decrease relative to the native-gradient update.

Proof. Write $w := q^{1-\eta}$ and $c_+ := \max\{c, 0\}$. Since $0 < q < 1$, the cap is inactive, $q = \|g_T\|_2/\|g_R\|_2$, and $0 < w \leq 1$. The projection rule gives

$$g_\eta = \big(1 - w \min\{c, 0\}\big) g_R + q^{-\eta} g_T.$$

Consequently,

$$g_R^\top g_\eta = \|g_R\|_2^2 (1 + w c_+) > 0.$$

Substitution into equation 7 yields

$$\mathcal{G}(g_\eta) = \frac{q^{2-\eta}(1 - c^2)}{1 + w c_+}. \tag{24}$$

For scalarization, $g_\lambda = g_R + \lambda g_T$, and

$$g_R^\top g_\lambda = \|g_R\|_2^2 (1 + \lambda q c).$$

Since $\lambda q < 1 - \delta$ and $|c| \leq 1$,

$$\delta < 1 + \lambda q c < 2 - \delta. \tag{25}$$

Thus, the scalarized direction also has positive primary progress. Direct substitution gives

$$\begin{aligned} \mathcal{G}(g_\lambda) &= q \frac{c + \lambda q}{1 + \lambda q c} - q c \\ &= \frac{\lambda q^2 (1 - c^2)}{1 + \lambda q c}. \end{aligned} \tag{26}$$

Since $1 \leq 1 + w c_+ \leq 2$, equation 24 gives

$$\frac{1}{2}(1 - c^2) q^{2-\eta} \leq \mathcal{G}(g_\eta) \leq (1 - c^2) q^{2-\eta}. \tag{27}$$

Similarly, combining equation 26 with equation 25 yields

$$\frac{\lambda}{2 - \delta}(1 - c^2) q^2 \leq \mathcal{G}(g_\lambda) \leq \frac{\lambda}{\delta}(1 - c^2) q^2. \tag{28}$$

Because $\lambda > 0$ and $\delta \in (0, 1)$ are fixed, these bounds establish equation 8, with positive constants independent of q and c. They also cover $|c| = 1$, where both gains vanish.

Finally, if $|c| < 1$, both gains are positive, and their common factor $1 - c^2$ cancels:

$$\frac{\mathcal{G}(g_\eta)}{\mathcal{G}(g_\lambda)} = \frac{q^{-\eta}}{\lambda} \frac{1 + \lambda q c}{1 + w c_+}. \tag{29}$$

Combining $1 \leq 1 + w c_+ \leq 2$ with equation 25 gives

$$\frac{\delta}{2\lambda} q^{-\eta} \leq \frac{\mathcal{G}(g_\eta)}{\mathcal{G}(g_\lambda)} \leq \frac{2 - \delta}{\lambda} q^{-\eta}. \tag{30}$$

Thus,

$$\frac{\mathcal{G}(g_\eta)}{\mathcal{G}(g_\lambda)} = \Theta(q^{-\eta}),$$

with constants independent of q and c. For every fixed $\eta > 0$ and $\lambda > 0$, the lower bound diverges as $q \to 0$, establishing the claimed asymptotic advantage for noncollinear gradients. □

C Additional Fixed-Weight Scalarization Results

We report the five TD3 task sweeps omitted from Section 6.4: LunarLander, Pendulum, Reacher, Ant, and Walker. Each sweep uses the 20 fixed weights $\Lambda = \{0.05, 0.10, \ldots, 1.0\}$, with five training seeds and ten evaluation episodes per seed. Points show mean return and FFT-based oscillation score; dashed lines show the reported TD3 means of DAMPER. The highlighted point has the lowest mean oscillation score within each sweep. Axis ranges vary by task.

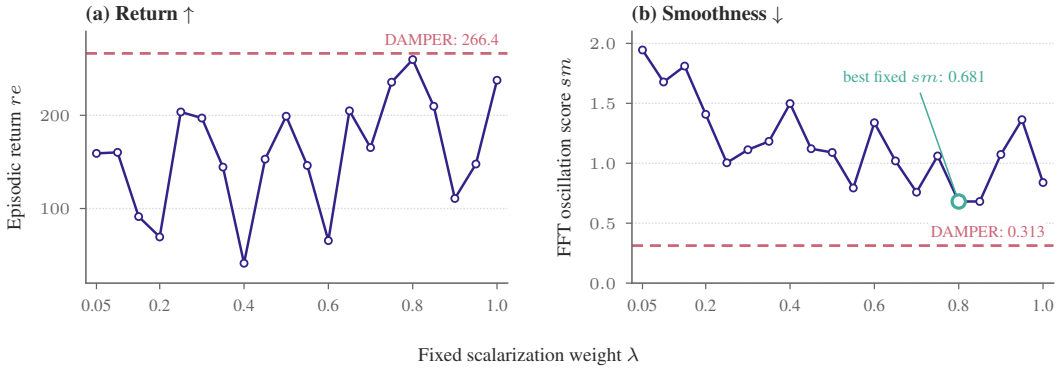

Figure 5: Fixed-weight scalarization on TD3 LunarLander. The lowest scalarization score is $sm = 0.681$ at $\lambda = 0.80$, compared with 0.313 for DAMPER.

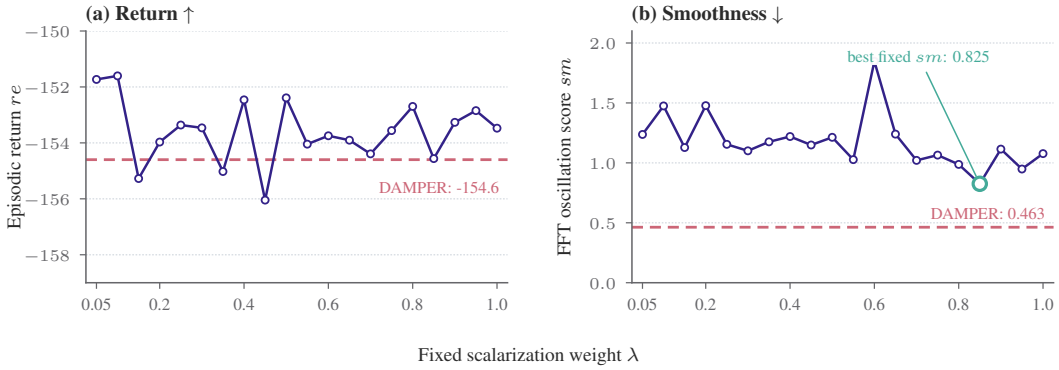

Figure 6: Fixed-weight scalarization on TD3 Pendulum. The lowest scalarization score is $sm = 0.825$ at $\lambda = 0.85$, compared with 0.463 for DAMPER.

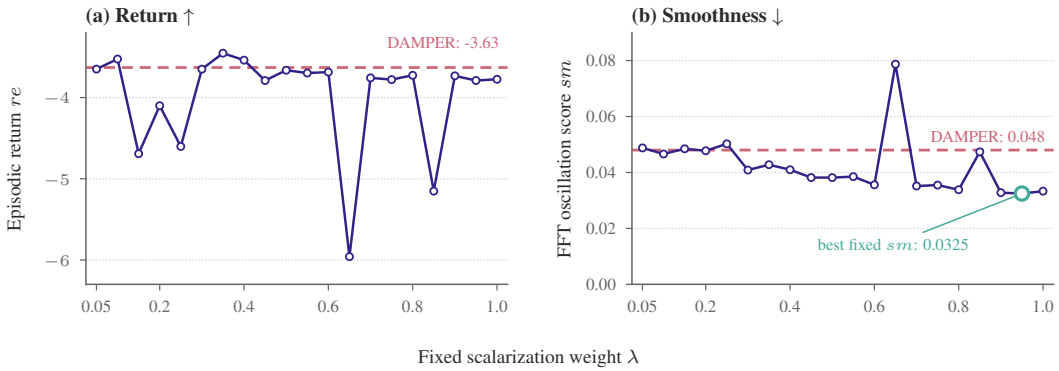

Figure 7: Fixed-weight scalarization on TD3 Reacher. Reacher is the exception: scalarization reaches $sm = 0.0325$ at $\lambda = 0.95$, below the reported 0.048 for DAMPER.

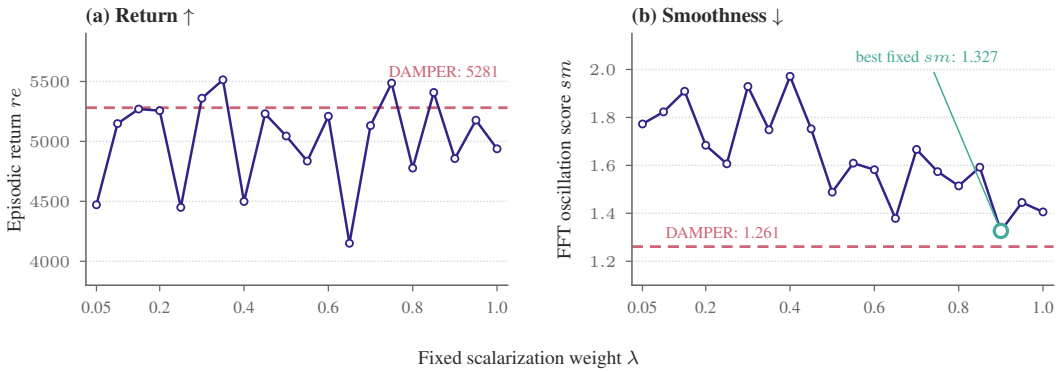

Figure 8: Fixed-weight scalarization on TD3 Ant. The lowest scalarization score is $sm = 1.327$ at $\lambda = 0.90$, compared with 1.261 for DAMPER.

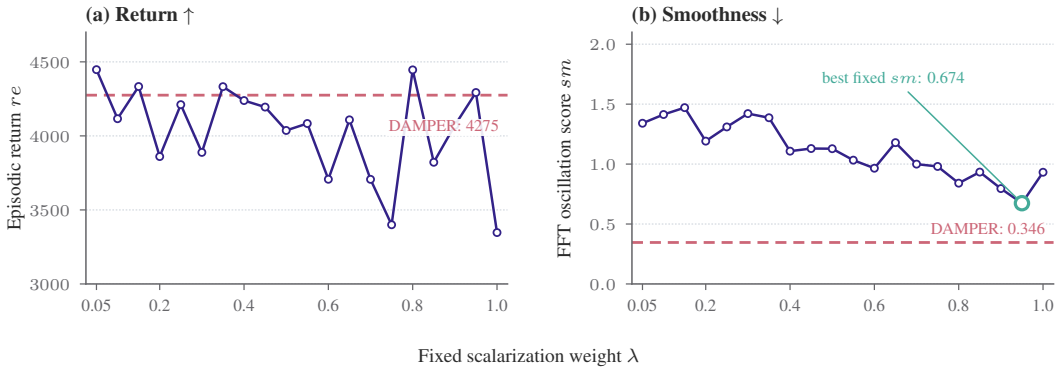

Figure 9: Fixed-weight scalarization on TD3 Walker. The lowest scalarization score is $sm = 0.674$ at $\lambda = 0.95$, compared with 0.346 for DAMPER.

Comparison with smoothing baselines. We compare tuned scalarization with the five smoothing baselines in Table 1, excluding DAMPER. On Ant, the weight $\lambda = 0.75$ gives $(re, sm) = (5486, 1.574)$. Its return exceeds the highest baseline return (5370, CAPS), while its oscillation score is 8.1% below the lowest baseline score (1.713, Grad-CAPS). Thus, it improves both metrics over all five baselines. On LunarLander, $\lambda = 0.80$ gives $(259.8, 0.681)$, improving both metrics over Grad-CAPS, ASAP, L2C2, and PAVE; PAVE, for example, obtains $(238.1, 0.841)$. On Hopper, $\lambda = 0.80$ gives $(3338, 0.714)$, improving both metrics over CAPS, ASAP, L2C2, and PAVE. Its oscillation score is 30.1% below the best baseline score (1.022, PAVE), with a similar return (3338 versus 3328). On Pendulum, $\lambda = 0.85$ gives $(-154.6, 0.825)$, improving on both ASAP $(-155.3, 2.013)$ and L2C2 $(-156.0, 3.293)$ in both metrics.

These gains are task dependent. CAPS remains smoother on LunarLander, and CAPS and PAVE remain smoother on Pendulum. The lowest scalarization scores on Reacher (0.0325) and Walker (0.674) improve on all five baselines, but their returns are lower: -3.79 versus -3.55 to -3.34 on Reacher, and 4293 versus 4880 to 5046 on Walker. These results make a tuned temporal-consistency loss a strong baseline: additional smoothing mechanisms do not consistently improve the return–smoothness trade-off in these experiments. Comparisons use reported means and weights selected from the sweep; they do not establish statistical significance.

D EXPERIMENTAL DETAILS

Baseline configurations. We follow the baseline hyperparameter settings reported in the PAVE appendix (Lee et al., 2026). Table 5 lists the settings for CAPS, Grad-CAPS, ASAP, and PAVE from Appendix E.1 (Tables 10 and 11) of that work. Grad-CAPS is denoted as GRAD in the original tables. CAPS, Grad-CAPS, and ASAP use the same settings under TD3 and SAC; PAVE uses backbone-specific loss weights.

Table 5: Baseline hyperparameters adopted from PAVE. Settings apply to both TD3 and SAC unless a backbone is specified. Parameter notation follows each baseline.

Method	Parameter	LunarLander	Pendulum	Reacher	Ant	Hopper	Walker
CAPS	λ_T	0.1	1.0	0.1	0.1	0.1	0.1
	λ_S	0.5	5.0	0.5	0.5	0.5	0.5
	σ	0.2	0.2	0.2	0.2	0.2	0.2
Grad-CAPS	λ_T	1.0	1.0	1.0	1.0	1.0	1.0
ASAP	λ_T	0.005	0.005	0.1	0.05	0.07	0.05
	λ_S	0.03	0.03	0.1	0.3	0.3	0.3
	λ_P	2.0	2.0	2.0	2.0	2.0	2.0
PAVE (TD3)	λ_1	0.1	2.0	0.1	0.1	0.1	0.1
	λ_2	0.1	0.005	0.1	0.005	0.005	0.1
	λ_3	0.01	2.0	0.01	0.5	0.5	0.01
PAVE (SAC)	λ_1	0.1	0.1	0.1	0.1	2.0	2.0
	λ_2	0.5	0.005	0.0005	0.0005	0.0005	0.005
	λ_3	0.05	0.5	1.0	1.0	3.0	2.0

For PAVE, λ_1, λ_2, and λ_3 weight mixed-partial regularization, vector-field consistency, and curvature preservation, respectively. Its perturbation scale $\sigma = 0.01$ and curvature floor $\delta = 1.0$ are fixed across all tasks and both backbones.

L2C2. For both TD3 and SAC, we use $\sigma = 1.0$, $\underline{\lambda} = 0.01$, $\overline{\lambda} = 1.0$, and $\beta = 0.1$ on all six tasks.

DAMPER. Table 6 lists the interpolation parameter η used for each task–backbone pair in our main experiments. The parameter sweeps used for ablation are reported separately.

Table 6: Interpolation parameter η for DAMPER.

Backbone	LunarLander	Pendulum	Reacher	Ant	Hopper	Walker
TD3	0.8	1.0	1.0	0.0	1.0	1.0
SAC	1.0	1.0	0.8	0.0	0.0	1.0

Hardware. Experiments were conducted using NVIDIA GeForce RTX 2080 and RTX 5060 Ti GPUs.

E NATIVE ACTOR OBJECTIVES

We specify the native actor losses and deterministic action maps used in our TD3 and SAC implementations. Expectations over states use the state marginal of the replay transition distribution $\mathcal{D}$. The critic parameters are held fixed during actor optimization, while gradients pass through the critic's action input.

TD3. For TD3 (Fujimoto et al., 2018), the deterministic actor μ_θ is optimized using

$$L_{\mathrm{R}}^{\mathrm{TD3}}(\theta) = -\mathbb{E}_{s\sim\mathcal{D}}\left[Q_{\phi_1}(s, \mu_\theta(s))\right]. \tag{31}$$

The temporal-consistency loss uses $\bar{\mu}_\theta(s) = \mu_\theta(s)$.

SAC. For SAC (Haarnoja et al., 2018), let $a_\theta(s, \epsilon)$ denote a reparameterized action with $\epsilon \sim \mathcal{N}(0, I)$, and let α_{ent} denote the entropy temperature. The native actor loss is

$$\begin{aligned} L_{\mathrm{R}}^{\mathrm{SAC}}(\theta) = \mathbb{E}_{s\sim\mathcal{D},\epsilon}\big[&\alpha_{\mathrm{ent}} \log \pi_\theta(a_\theta(s,\epsilon) \mid s) \\ &- \min_{j\in\{1,2\}} Q_{\phi_j}(s, a_\theta(s,\epsilon))\big]. \end{aligned} \tag{32}$$

Table 7: Training budgets and network sizes following the PAVE protocol (Lee et al., 2026).

Setting	TD3	SAC
Training iterations		
LunarLander	500,000	500,000
Pendulum	100,000	100,000
Reacher	500,000	500,000
Ant	1,000,000	1,000,000
Hopper	1,000,000	1,000,000
Walker2d	1,000,000	1,000,000
Hidden-layer widths	$(400, 300)$	$(256, 256)$

Table 8: Average update time on Hopper-v5. Relative time is the ratio to the corresponding unregularized backbone. Lower is better. Each entry averages 1,000 timed update iterations.

	TD3		SAC	
Method	Time (ms)	Relative ($\times$)	Time (ms)	Relative ($\times$)
Unregularized	4.54	1.00	10.87	1.00
CAPS	7.32	1.61	18.78	1.73
Grad-CAPS	6.35	1.40	18.60	1.71
ASAP	9.85	2.17	23.38	2.15
L2C2	12.71	2.80	48.36	4.45
PAVE	24.07	5.30	32.22	2.96
DAMPER	6.57	1.45	14.26	1.31

The temporal-consistency loss uses the squashed mean action $\pi_\theta(s) = a_\theta(s, 0)$, including any action-bound rescaling used by the policy.

F COMPUTATIONAL COST

We measure the per-iteration update time of TD3 and SAC and their smoothness variants on an NVIDIA GeForce RTX 5060 Ti GPU using Hopper-v5 transitions collected under random actions. Each configuration starts from freshly initialized networks and uses a batch size of 256. We perform 50 warmup iterations followed by 1,000 timed iterations. The measured operations include forward computation, loss evaluation, gradient computation, and optimizer steps; logging, diagnostics, and target-network copy operations are excluded. During the timed interval, TD3 performs 500 actor updates and SAC performs 1,000. We therefore report time per update iteration and normalize it to the unregularized baseline within each backbone.

Table 8 shows that DAMPER with $\eta = 0$ requires 6.57 ms per iteration with TD3 and 14.26 ms with SAC, corresponding to overheads of 44.8% and 31.3%, respectively, over the unregularized backbones. On TD3, DAMPER is slightly slower than Grad-CAPS but faster than CAPS, ASAP, L2C2, and PAVE. On SAC, it has the lowest measured runtime among the compared smoothness methods. These measurements characterize update computation under the tested configuration and do not represent end-to-end training time.