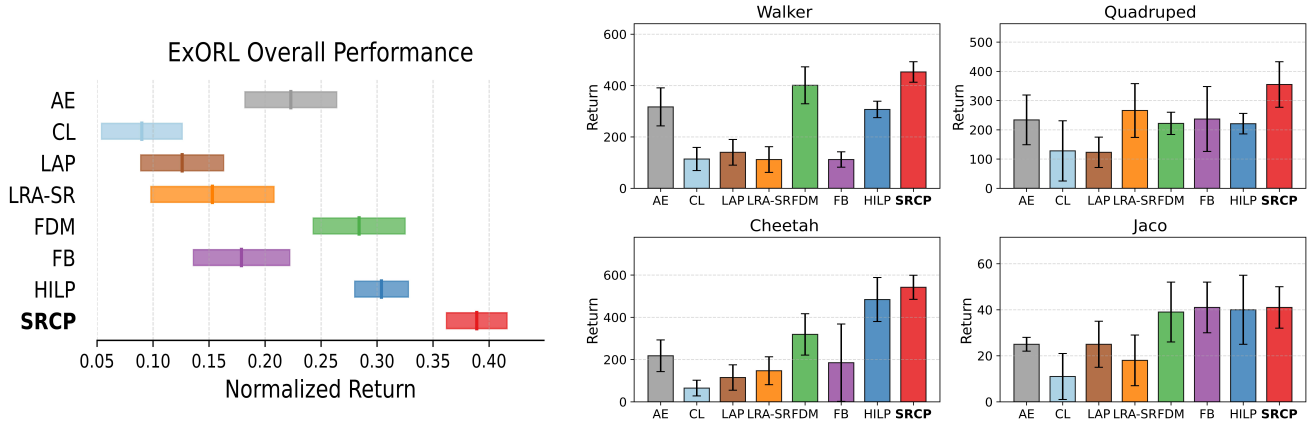
Appendix

A. Full results on the ExORL benchmark

We evaluate the generalization performance of SRCP and baseline methods across 16 visual continuous control tasks from 4 domains in the ExORL benchmark, using 4 different datasets for pretraining. All methods are pretrained from raw pixel observations for 500k steps. During evaluation, 10k transitions of task-specific visual-based data are provided to infer the skill vectors for downstream tasks, enabling zero-shot generalization. Fig.1 (a) summarizes the overall performance of SRCP and baseline methods across all 16 tasks from 4 domains with 4 datasets with 4 seeds. The results show that SRCP achieves SOTA overall performance, demonstrating superior zero-shot generalization ability. Fig.1 (b) provides a detailed breakdown of performance across 4 domains. The results highlight that SRCP achieves the best or near-best performance in each domains. Furthermore, the comparison between SRCP and HILP highlights the architectural improvements that SRCP brings to SR methods.



(a) Overall generalization performance

(b) Performance comparison across four domains

Figure 1. Zero-shot generalization performance on visual tasks. (a) Overall performance of each method evaluated on 4 datasets, 4 domains, and 4 tasks per domain using 4 random seeds (i.e., 256 values in total). (b) Performance in 4 domains.

Fig.2 illustrates the comparative performance of the SRCP method against baseline approaches across all tasks. Notably, SRCP consistently attains either superior or near-optimal performance across all tasks, highlighting its effectiveness across a diverse range of tasks. The complete results (unnormalized returns) are reported in Table 1, with all methods evaluated using 4 random seeds.

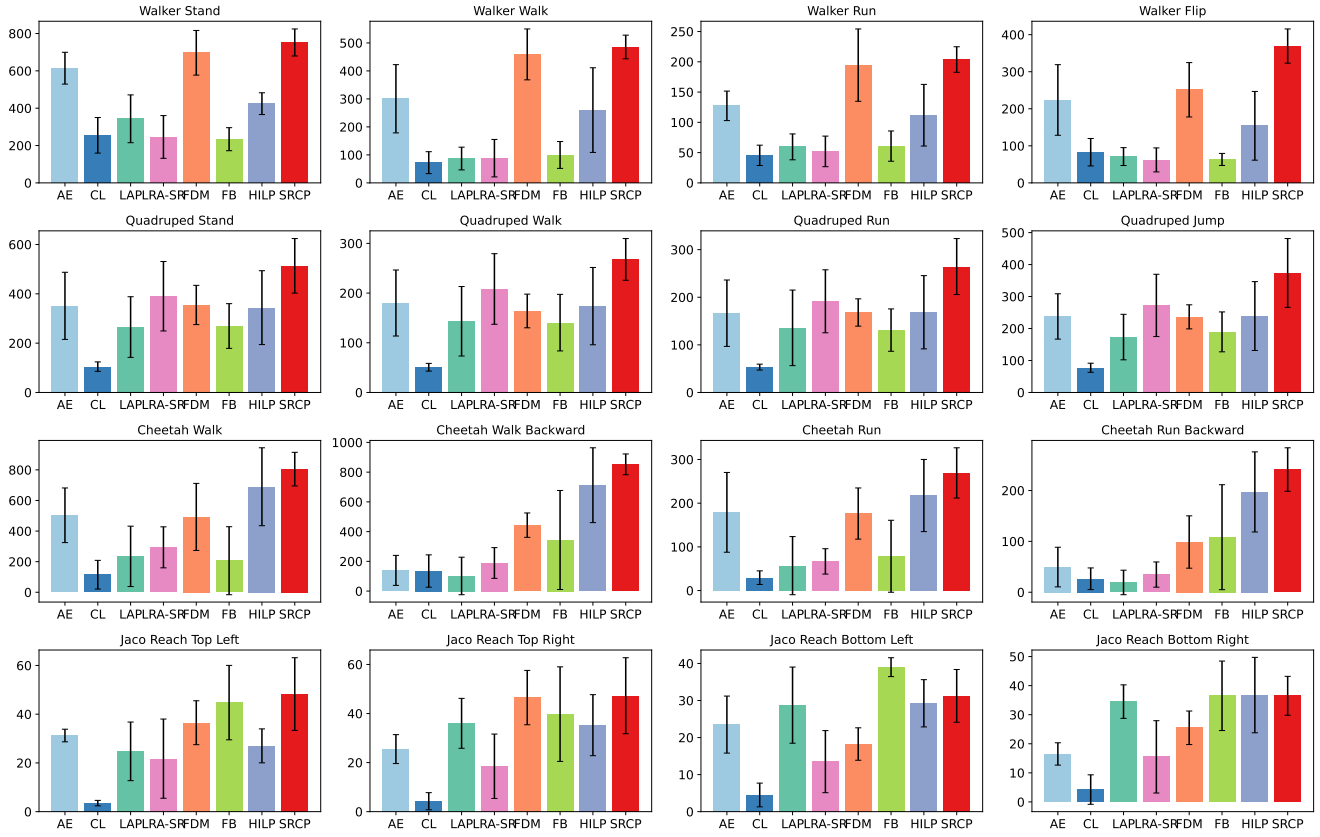


Figure 2. Experimental results on the pixel-based ExORL benchmark for each task, aggregated over four datasets and four random seeds (i.e., 16 runs in total).

Table 1. Zero-shot Performance across Multiple Domains

Dataset	Domain	Task	AE	CL	LAP	LRA-SR	FDM	FB	HILP	SRCP
RND	Walker	Stand	546 ± 30	144 ± 4	296 ± 35	213 ± 49	557 ± 99	135 ± 12	386 ± 98	671 ± 51
		Walk	358 ± 12	36 ± 1	49 ± 5	147 ± 16	452 ± 52	48 ± 9	326 ± 31	520 ± 32
		Run	125 ± 7	25 ± 2	43 ± 2	82 ± 3	146 ± 60	31 ± 10	121 ± 20	194 ± 38
		Flip	265 ± 14	50 ± 3	35 ± 2	101 ± 14	282 ± 52	36 ± 15	92 ± 35	369 ± 25
		Average	324	64	106	147	359	63	231	**439**
	Quadruped	Stand	467 ± 31	114 ± 12	446 ± 18	476 ± 37	374 ± 85	257 ± 42	455 ± 84	703 ± 22
		Walk	248 ± 21	58 ± 4	252 ± 26	274 ± 4	199 ± 63	104 ± 12	254 ± 31	339 ± 27
		Run	228 ± 30	59 ± 7	252 ± 35	239 ± 18	192 ± 29	105 ± 36	241 ± 12	361 ± 24
		Jump	274 ± 55	88 ± 12	270 ± 11	340 ± 31	273 ± 66	164 ± 22	269 ± 28	535 ± 14
		Average	304	80	305	332	260	158	305	**485**
	Cheetah	Walk	658 ± 98	275 ± 10	181 ± 15	445 ± 95	470 ± 182	559 ± 102	895 ± 33	859 ± 34
		Walk Backward	5 ± 2	100 ± 6	25 ± 9	196 ± 8	441 ± 107	803 ± 58	927 ± 35	934 ± 45
		Run	258 ± 16	51 ± 8	40 ± 7	106 ± 33	178 ± 41	211 ± 34	276 ± 46	322 ± 44
		Run Backward	3 ± 3	19 ± 3	5 ± 1	24 ± 2	126 ± 9	243 ± 18	297 ± 46	293 ± 42
		Average	231	111	63	193	304	454	599	**602**
	Jaco	Reach Top Left	28 ± 3	5 ± 3	40 ± 8	38 ± 4	40 ± 16	45 ± 14	23 ± 9	53 ± 9
		Reach Top Right	20 ± 7	10 ± 4	24 ± 5	20 ± 4	38 ± 21	70 ± 10	39 ± 5	60 ± 7
		Reach Bottom Left	22 ± 4	10 ± 2	43 ± 4	21 ± 11	20 ± 14	63 ± 51	28 ± 6	43 ± 8
		Reach Bottom Right	22 ± 4	13 ± 3	35 ± 5	33 ± 9	16 ± 8	51 ± 9	47 ± 5	42 ± 8
		Average	23	10	36	28	29	**52**	34	50
Proto	Walker	Stand	677 ± 18	316 ± 3	389 ± 7	156 ± 5	854 ± 46	241 ± 32	365 ± 33	830 ± 39
		Walk	216 ± 38	125 ± 2	78 ± 3	28 ± 2	563 ± 268	92 ± 7	117 ± 12	504 ± 66
		Run	122 ± 5	60 ± 1	67 ± 1	26 ± 1	267 ± 44	48 ± 6	61 ± 8	227 ± 16
		Flip	155 ± 27	132 ± 4	75 ± 2	29 ± 2	212 ± 65	74 ± 6	80 ± 15	428 ± 75
		Average	293	158	152	60	474	114	156	**497**
	Quadruped	Stand	120 ± 7	112 ± 9	138 ± 10	163 ± 18	226 ± 65	193 ± 14	86 ± 8	455 ± 21
		Walk	70 ± 6	56 ± 3	68 ± 6	91 ± 9	108 ± 39	110 ± 11	45 ± 9	234 ± 3
		Run	48 ± 5	54 ± 6	52 ± 4	80 ± 12	120 ± 21	108 ± 12	39 ± 5	230 ± 18
		Jump	116 ± 8	71 ± 3	89 ± 10	104 ± 11	182 ± 22	129 ± 16	67 ± 11	238 ± 68
		Average	89	73	87	110	159	135	59	**289**
	Cheetah	Walk	206 ± 49	46 ± 4	74 ± 41	109 ± 28	150 ± 126	20 ± 21	351 ± 67	624 ± 44
		Walk Backward	117 ± 24	56 ± 3	25 ± 9	59 ± 11	384 ± 232	8 ± 5	692 ± 26	856 ± 44
		Run	30 ± 6	7 ± 1	5 ± 1	24 ± 3	87 ± 67	6 ± 5	157 ± 22	174 ± 43
		Run Backward	21 ± 2	10 ± 1	5 ± 2	10 ± 2	41 ± 22	7 ± 3	100 ± 15	208 ± 12
		Average	94	30	27	51	166	10	325	**466**
	Jaco	Reach Top Left	30 ± 9	4 ± 2	32 ± 8	5 ± 2	43 ± 25	47 ± 12	39 ± 5	43 ± 11
		Reach Top Right	34 ± 6	2 ± 1	50 ± 10	4 ± 1	48 ± 21	43 ± 9	44 ± 6	57 ± 5
		Reach Bottom Left	33 ± 8	3 ± 3	34 ± 6	2 ± 1	24 ± 22	38 ± 11	24 ± 5	30 ± 10
		Reach Bottom Right	18 ± 7	1 ± 1	40 ± 4	1 ± 1	26 ± 21	43 ± 4	29 ± 4	44 ± 19
		Average	29	3	39	3	35	43	34	**44**
APS	Walker	Stand	516 ± 12	377 ± 17	519 ± 38	441 ± 13	608 ± 90	304 ± 22	429 ± 26	689 ± 24
		Walk	158 ± 13	95 ± 1	155 ± 20	100 ± 4	317 ± 156	81 ± 4	120 ± 19	413 ± 27
		Run	97 ± 5	64 ± 3	91 ± 5	72 ± 1	126 ± 26	51 ± 13	74 ± 14	174 ± 18
		Flip	116 ± 8	105 ± 4	103 ± 10	87 ± 5	158 ± 25	66 ± 8	133 ± 12	299 ± 39
		Average	222	160	217	175	303	126	189	**394**
	Quadruped	Stand	426 ± 18	72 ± 5	310 ± 80	533 ± 37	444 ± 16	421 ± 16	415 ± 44	471 ± 27
		Walk	206 ± 14	38 ± 5	154 ± 38	254 ± 11	176 ± 25	239 ± 12	197 ± 18	257 ± 14
		Run	195 ± 12	43 ± 2	165 ± 57	243 ± 27	172 ± 27	208 ± 14	193 ± 19	260 ± 35
		Jump	268 ± 17	57 ± 5	210 ± 26	323 ± 20	221 ± 23	294 ± 9	256 ± 23	389 ± 61
		Average	274	53	210	338	253	318	265	**344**
	Cheetah	Walk	625 ± 44	52 ± 7	570	226 ± 9	613 ± 77	12 ± 2	273 ± 383	821 ± 122
		Walk Backward	288 ± 126	63 ± 7	37	155 ± 39	371 ± 307	48 ± 7	967 ± 8	742 ± 51
		Run	179 ± 31	30 ± 5	170	68 ± 3	189 ± 41	13 ± 5	118 ± 113	275 ± 11
		Run Backward	75 ± 7	14 ± 1	6	29 ± 4	59 ± 73	9 ± 4	248 ± 46	191 ± 61
		Average	292	40	196	120	308	21	402	**507**
	Jaco	Reach Top Left	35 ± 6	2 ± 1	9 ± 1	6 ± 2	21 ± 8	22 ± 6	22 ± 9	28 ± 12
		Reach Top Right	28 ± 15	4 ± 2	29 ± 5	11 ± 1	36 ± 13	23 ± 4	14 ± 5	21 ± 5
		Reach Bottom Left	27 ± 4	2 ± 1	20 ± 4	22 ± 5	12 ± 2	36 ± 13	40 ± 7	28 ± 2
		Reach Bottom Right	12 ± 5	2 ± 1	38 ± 1	7 ± 2	29 ± 18	19 ± 6	63 ± 11	32 ± 11
		Average	26	3	24	12	25	25	**35**	27
APT	Walker	Stand	718 ± 78	182 ± 14	169 ± 8	174 ± 6	768 ± 110	256 ± 48	517 ± 16	818 ± 37
		Walk	470 ± 20	33 ± 1	66 ± 3	32 ± 1	504 ± 79	178 ± 37	477 ± 21	505 ± 91
		Run	165 ± 15	33 ± 3	37 ± 3	28 ± 2	239 ± 34	67 ± 6	191 ± 14	220 ± 29
		Flip	359 ± 17	44 ± 4	71 ± 4	31 ± 2	353 ± 37	77 ± 8	311 ± 11	381 ± 55
		Average	428	73	86	66	466	145	374	**481**
	Quadruped	Stand	392 ± 34	121 ± 5	167 ± 5	389 ± 35	375 ± 112	207 ± 25	420 ± 37	425 ± 91
		Walk	196 ± 19	51 ± 5	99 ± 22	214 ± 7	173 ± 36	109 ± 9	199 ± 12	241 ± 49
		Run	195 ± 35	57 ± 4	74 ± 3	204 ± 9	188 ± 38	103 ± 8	201 ± 14	207 ± 10
		Jump	293 ± 47	93 ± 2	124 ± 8	322 ± 22	270 ± 40	171 ± 12	364 ± 25	333 ± 91
		Average	269	81	116	282	252	148	296	**302**
	Cheetah	Walk	524 ± 42	86 ± 3	114 ± 15	395 ± 84	737 ± 54	234 ± 63	899 ± 55	916 ± 44
		Walk Backward	147 ± 81	321 ± 36	320 ± 44	346 ± 40	578 ± 137	519 ± 79	604 ± 41	878 ± 121
		Run	249 ± 62	30 ± 1	13 ± 6	69 ± 16	251 ± 18	84 ± 17	319 ± 32	306 ± 11
		Run Backward	99 ± 60	63 ± 2	61 ± 17	76 ± 9	169 ± 109	174 ± 42	144 ± 23	273 ± 8
		Average	255	125	127	222	434	253	492	**593**
	Jaco	Reach Top Left	32 ± 15	3 ± 1	18 ± 1	38 ± 10	42 ± 8	65 ± 19	24 ± 10	69 ± 13
		Reach Top Right	20 ± 3	1 ± 1	41 ± 4	39 ± 11	64 ± 27	23 ± 6	44 ± 8	51 ± 10
		Reach Bottom Left	12 ± 4	3 ± 1	18 ± 1	9 ± 1	17 ± 9	39 ± 6	25 ± 5	24 ± 3
		Reach Bottom Right	14 ± 4	1 ± 1	35 ± 7	21 ± 7	31 ± 32	33 ± 6	27 ± 4	28 ± 3
		Average	20	2	26	27	39	40	30	**43**

B. Pseudo-code

We present the pseudocode for training SRCP in the offline visual unsupervised reinforcement learning setting in Algorithm 1. Specifically, the procedure illustrates how SRCP incorporates the HILP method to learn skill-conditioned representations during the pretraining phase.

Algorithm 1 SRCP Algorithm

Inputs: pre-collected dataset $\mathcal{D}$, randomly initialized representation network f_θ, basic feature networks φ_ν, successor feature network ψ_κ, actor network π_ζ , learning rate η, mini-batch size b, number of gradient updates step M, number of gradient updates every step N, skill update period T, temperature τ.
for $m = 1$ to M **do**
  for $n = 1$ to N **do**
    Sample a mini-batch of transitions $\{(o_i, a_i, o_{i+1})\}_{i\in I} \subset \mathcal{D}$ of size $|I| = b$.
    Sample a mini-batch of target state-action pairs $\{(o'_i, a'_i)\}_{i\in I} \subset \mathcal{D}$ of size $|I| = b$.
    Sample a mini-batch of $\{z_i\}_{i\in I} \sim p(z)$ of size $|I| = b$.
    /* Generate saliency maps */
    Generate saliency maps o_α from observation o.
    /* Upadate encoder */
    Compute saliency dynamics representation learning objective $L(\theta)$ with equation (5).
    Update $\theta \leftarrow \theta - \eta\nabla L(\theta)$.
    /* Upadate basic feature and successor feature */
    Compute HILP basic feature objective $L(\nu)$ with:
    $\mathcal{L}_\nu = \sum_{i=1}^{2} \mathbb{E}_{(f(o),a,f(o'),g)} \left[(r + \gamma(1-r)\cdot V'_i(f(o'), g) - V_i(f(o), g))^2\right]$
    Compute successor measure loss with:
    $\mathcal{L}_\kappa = ||\varphi(f(o'))z + \gamma\psi(f(o'), a', z)z - \psi(f(o), a, z)z||^2$
    Update $\nu, \kappa \leftarrow \nu, \kappa - \eta\nabla L(\nu) - \eta\nabla L(\kappa)$.
    /* Upadate policy */
    Compute actor loss function $L(\zeta)$ with equation (11).
    Update $\zeta \leftarrow \zeta - \eta\nabla L(\zeta)$.
  end for
  /* Update target network parameters */
  $\kappa^- \leftarrow \tau\kappa + (1-\tau)\kappa^-$
end for

C. Theoretical analysis

Proposition 1 ([27, Prop. 16]). *Let $\tau : S \times A \to G$ be a mapping into a goal space G. Let π be a policy, and let M^π denote its successor state measure in goal space G, as defined in (10). Assume that $M^\pi(s, a, \cdot)$ is absolutely continuous with respect to a positive measure ρ on G, and let m^π be its Radon–Nikodym derivative. For any measurable function $r : G \to \mathbb{R}$, define the reward $R(s, a) := r(\tau(s, a))$. Then the Q-function of policy π under reward R satisfies*

$$\begin{aligned} Q^\pi(s,a) &= \int_G r(g)\, M^\pi(s,a,\mathrm{d}g) \\ &= \int_G r(g)\, m^\pi(s,a,g)\, \rho(\mathrm{d}g). \end{aligned} \tag{1}$$

Proof. For each time $t \geq 0$, let $P_t^\pi(s_0, a_0, \mathrm{d}g)$ denote the distribution of $g = \tau(s_t, a_t)$ under trajectories generated by policy π starting from (s_0, a_0). By definition of the successor state measure:

$$M^\pi(s,a,\mathrm{d}g) = \sum_{t\geq 0} \gamma^t\, P_t^\pi(s,a,\mathrm{d}g). \tag{2}$$

By definition of the density m^π, the Q-function of π under reward R satisfies:

$$\begin{aligned} Q^\pi(s,a) &= \sum_{t\geq 0} \gamma^t \mathbb{E}[R(s_t,a_t) \mid s_0 = s, a_0 = a, \pi] \\ &= \sum_{t\geq 0} \gamma^t \mathbb{E}[r(\tau(s_t,a_t)) \mid s_0 = s, a_0 = a, \pi] \\ &= \sum_{t\geq 0} \gamma^t \int_G r(g)\, P_t^\pi(s,a,\mathrm{d}g) \\ &= \int_G r(g)\, M^\pi(s,a,\mathrm{d}g). \end{aligned} \tag{3}$$

□

Proposition 2 ([27, Prop. 18]). *Let $f : S \times A \to \mathbb{R}$ be an arbitrary function, and define the policy π_f by $\pi_f(s) := \arg\max_a f(s,a)$. Let $r : S \times A \to \mathbb{R}$ be a bounded reward function, $Q^\star$ the optimal Q-function for r, and Q^{π_f} the Q-function of policy π_f under reward r. Then:*

$$\sup_{S\times A} |f - Q^\star| \leq \frac{2}{1-\gamma} \sup_{S\times A} |f - Q^{\pi_f}|, \tag{4}$$

and

$$\sup_{S\times A} |Q^{\pi_f} - Q^\star| \leq \frac{3}{1-\gamma} \sup_{S\times A} |f - Q^{\pi_f}|. \tag{5}$$

Proof. We restate the proof here for completeness. Define the approximation error

$$\varepsilon(s,a) := Q^{\pi_f}(s,a) - f(s,a).$$

The Q-function Q^{π_f} satisfies the Bellman equation

$$Q^{\pi_f}(s,a) = r(s,a) + \gamma \mathbb{E}_{s'|(s,a)} Q^{\pi_f}(s', \pi_f(s')).$$

Substituting $Q^{\pi_f} = f + \varepsilon$, we obtain

$$\begin{aligned} f(s,a) &= r(s,a) - \varepsilon(s,a) + \gamma \mathbb{E}_{s'|(s,a)} \big[f(s', \pi_f(s')) + \varepsilon(s', \pi_f(s')) \big] \\ &= r(s,a) - \varepsilon'(s,a) + \gamma \mathbb{E}_{s'|(s,a)} f(s', \pi_f(s')) \\ &= r(s,a) - \varepsilon'(s,a) + \gamma \mathbb{E}_{s'|(s,a)} \max_{a'} f(s', a'), \end{aligned} \tag{6}$$

where

$$\varepsilon'(s,a) := \varepsilon(s,a) - \gamma \mathbb{E}_{s'|(s,a)} \varepsilon(s', \pi_f(s')).$$

Equation (6) corresponds to the optimal Bellman equation for the modified reward function $r - \varepsilon'$. Hence, f is the optimal Q-function associated with reward $r - \varepsilon'$. Since $Q^\star$ is the optimal Q-function for the original reward r, it follows that

$$\sup_{S\times A} |f - Q^\star| \leq \frac{1}{1-\gamma} \sup_{S\times A} |\varepsilon'|.$$

By the definition of ε', we have

$$\sup_{S\times A} |\varepsilon'| \leq 2 \sup_{S\times A} |\varepsilon| = 2 \sup_{S\times A} |f - Q^{\pi_f}|,$$

which establishes the first inequality. The second inequality follows from the triangle inequality

$$|Q^{\pi_f} - Q^\star| \leq |Q^{\pi_f} - f| + |f - Q^\star|,$$

and observing that $\frac{2}{1-\gamma} + 1 \leq \frac{3}{1-\gamma}$. □

Theorem 1. *Let $\varphi : S \times A \to \mathbb{R}^d$ be a set of base features, and let $\psi^{\pi_z}(s,a) = \int \varphi(s',a')\, M^{\pi_z}(s,a,\mathrm{d}s'\mathrm{d}a')$ be the true successor features of policy π_z. Let $\hat{M}^z$ be an estimated successor measure and define the estimated successor features*

$$\hat{\psi}^z(s,a) = \int \varphi(s',a')\, \hat{M}^z(s,a,\mathrm{d}s'\mathrm{d}a').$$

For each $z \in Z$, define the policy

$$\pi_z(s) = \arg\max_a \hat{\psi}^z(s,a)^\top z.$$

Assume the reward is linearly represented by the features,

$$r(s,a) = \varphi(s,a)^\top z_r,$$

where $z_r = \mathbb{E}_\rho[\varphi\varphi^\top]^{-1}\mathbb{E}_\rho[\varphi r]$ is the least–squares coefficient under a reference distribution ρ. Let $V^\star$ be the optimal value function for r, and let $\hat{V}^{\pi_z}$ be the value function of π_z under the same reward. Then the suboptimality of the zero-shot policy π_{z_r} is controlled by the successor-feature approximation error:

$$\|\hat{V}^{\pi_{z_r}} - V^\star\|_\infty \;\leq\; \frac{3\|z_r\|_*}{1-\gamma} \sup_{s,a} \|\hat{\psi}^{z_r}(s,a) - \psi^{\pi_{z_r}}(s,a)\|. \tag{7}$$

Moreover, for any $z \in Z$, the approximation error of $\hat{\psi}$ controls the deviation from the optimal Q-function:

$$\sup_{s,a} \left|\hat{\psi}^z(s,a)^\top z - Q^\star(s,a)\right| \;\leq\; \frac{2\|z\|_*}{1-\gamma} \sup_{s,a} \|\hat{\psi}^z(s,a) - \psi^{\pi_z}(s,a)\|. \tag{8}$$

Remark. *The bounds (7)–(8) show that the suboptimality of policies derived from approximate successor features is controlled by the approximation error of ψ, providing a quantitative bound on the zero-shot generalization performance.*

Proof. Let $z \in Z$ and consider the reward $r(s,a) = \varphi(s,a)^\top z_r$. By definition of the successor feature,

$$\psi^{\pi_z}(s,a) = \int \varphi(s',a')\, M^{\pi_z}(s,a,\mathrm{d}s'\,\mathrm{d}a').$$

Similarly, the estimated successor feature satisfies

$$\hat{\psi}^z(s,a) = \int \varphi(s',a')\, \hat{M}^z(s,a,\mathrm{d}s'\,\mathrm{d}a').$$

Define the error measure between the true and estimated successor measures:

$$\varepsilon_z(s,a,\mathrm{d}s'\,\mathrm{d}a') := M^{\pi_z}(s,a,\mathrm{d}s'\,\mathrm{d}a') - \hat{M}^z(s,a,\mathrm{d}s'\,\mathrm{d}a').$$

By Proposition 1 with $G = S \times A$, the Q-function of policy π_z satisfies

$$\begin{aligned} Q^{\pi_z}(s,a) &= \int r(s',a')\, M^{\pi_z}(s,a,\mathrm{d}s'\,\mathrm{d}a') \\ &= \int r(s',a')\, \hat{M}^{z}(s,a,\mathrm{d}s'\,\mathrm{d}a') + \int r(s',a')\, \varepsilon_z(s,a,\mathrm{d}s'\,\mathrm{d}a') \\ &= \hat{\psi}^{z}(s,a)^\top z_r + \int r(s',a')\, \varepsilon_z(s,a,\mathrm{d}s'\,\mathrm{d}a'). \end{aligned} \tag{9}$$

By the dual norm property,

$$\left| \int r(s',a')\, \varepsilon_z(s,a,\mathrm{d}s'\,\mathrm{d}a') \right| \le \|z_r\|_* \, \|\hat{\psi}^{z}(s,a) - \psi^{\pi_z}(s,a)\|,$$

where $\|\cdot\|_*$ is the dual norm corresponding to the norm used in the supremum over measures. Let $f(s,a) := \hat{\psi}^{z_r}(s,a)^\top z_r$ and recall that $\pi_{z_r}(s) = \arg\max_a f(s,a)$. Then, by the Proposition 2,

$$\begin{aligned} \|\hat{V}^{\pi_{z_r}} - V^\star\|_\infty &\le \frac{3}{1-\gamma} \sup_{s,a} |f(s,a) - Q^{\pi_{z_r}}(s,a)| \\ &\le \frac{3\|z_r\|_*}{1-\gamma} \sup_{s,a} \|\hat{\psi}^{z_r}(s,a) - \psi^{\pi_{z_r}}(s,a)\|. \end{aligned} \tag{10}$$

Similarly, for any $z \in Z$,

$$\begin{aligned} \sup_{s,a} \left|\hat{\psi}^{z}(s,a)^\top z - Q^\star(s,a)\right| &\le \frac{2}{1-\gamma} \sup_{s,a} |\hat{\psi}^{z}(s,a)^\top z - Q^{\pi_z}(s,a)| \\ &\le \frac{2\|z\|_*}{1-\gamma} \sup_{s,a} \|\hat{\psi}^{z}(s,a) - \psi^{\pi_z}(s,a)\|. \end{aligned} \tag{11}$$

This completes the proof, establishing that the suboptimality of policies derived from approximate successor features is controlled by the approximation error of $\hat{\psi}$. □

D. Extended Background

Reinforcement Learning (RL) [26, 29, 32] has achieved remarkable success in learning task-specific behaviors when guided by handcraft extrinsic rewards. However, such supervision often requires extensive manual effort and limits generalization to novel tasks. Inspired by the effectiveness of unsupervised pretraining in computer vision [**?**] and natural language processing [**?**], Unsupervised Reinforcement Learning (URL) has emerged as a promising paradigm that enables agents to acquire reusable representations [17, 31] and behaviors from reward-free interactions [4, 20]. In URL, agents are trained using task-agnostic objectives, such as intrinsic rewards or self-supervised losses, to explore environments and acquire general-purpose skills or latent representations. These pretrained components can later be adapted to downstream tasks with improved sample efficiency. Among various URL approaches, unsupervised skill discovery (USD) methods [1, 10, 14, 19] aim to developing diverse skills via reward-free learning to enable task generalization through fine-tuning. Despite these advancements, most existing USD methods focus primarily on discover diverse skills, while lacking the ability to directly generalize to new tasks without additional fine-tuning, limiting the zero-shot generalization in URL.

A prominent line of work in zero-shot URL builds upon the framework of successor representations (SR) [7], which decouples environment dynamics from reward specification to support generalization across tasks. Extensions such as successor features (SF) [3] further enhance this formulation by allowing skill-conditioned value estimation. State-of-the-art zero-shot methods instantiate this idea via universal successor features (USFs) [24] or forward-backward (FB) representations [27, 28]. USF-based methods typically rely on learned basic features, and a variety of techniques have been proposed to facilitate this, including Laplacian eigenfunctions [30], low-rank transition approximations [23], contrastive learning [2], and hilbert space representation learning [18]. In contrast, FB methods avoid explicit feature learning by directly modeling successor measures through jointly learned forward and backward representations. While existing SR methods exhibit strong zero-shot generalization in structured state-based environments, they struggle to scale to visual tasks due to the challenges of learning meaningful representations from high-dimensional inputs. To address this limitation, we propose Saliency-guided Representation with Consistency Policy learning (SRCP), a novel framework designed to improve generalization in visual unsupervised reinforcement learning. SRCP enhances SR-based learning through two key innovations: (1) a saliency-guided dynamics objective that learns dynamics-relevant representations by decoupling them from SR training, and (2) a consistency-based diffusion actor that models diverse skill-conditioned behaviors with improved action controllability.

Diffusion models [11] have shown impressive success in modeling complex, high-dimensional distributions and generating diverse, high-quality samples across a variety of domains, including image generation, planning [25], and control [**?**]. These properties make them especially appealing for RL, where the ability to capture expressive and multi-modal action distributions is crucial for balancing exploration and exploitation [5, 12, 21]. Recent works have demonstrated that diffusion-based policies can outperform traditional MLP-based actors, particularly in tasks requiring diverse and multi-modal behaviors. However, the adoption of diffusion models in RL comes with a significant computational cost. Their iterative sampling process leads to high inference latency, making them inefficient for real-time decision-making and online learning scenarios. To mitigate this, recent advances have introduced consistency models [6, 9, 15] as efficient one-step approximations of diffusion processes. These models retain the generative expressiveness of diffusion while enabling faster inference, making them promising candidates for use in RL actor networks. Empirical results have shown that consistency models can match or even surpass diffusion policies in continuous control tasks, with substantially lower computational overhead. Despite their promising properties, consistency models remain underexplored in visual URL [13], which demands learning multi-modal skill-conditioned behaviors from high-dimensional visual inputs. To bridge this gap, we propose a consistency policy equipped with URL-specific classifier-free guidance, which promotes multi-modal action distribution modeling and skill controllability in visual tasks and improves generalization in visual URL.

E. Experimental Setting

Environments We adopt the URL Benchmark [13] and ExORL datasets [8] to evaluate the task generalization performance in visual URL. The benchmark comprises 16 high-dimensional continuous control tasks across 4 distinct domains: *Walker*, *Quadruped*, *Cheetah*, and *Jaco*. All environments provide image-based observations with a resolution of $64 \times 64 \times 3$. Agents should extract effective representations and learn skill-conditioned behaviors from high-dimensional visual observations, making this benchmark particularly challenging for evaluating zero-shot task generalization in visual URL. We detail the four domains as follows:

- **Walker Domain** involves humanoid locomotion with action space $\mathcal{A} \in \mathbb{R}^6$. It includes the tasks of *Walker Stand*, *Walker Walk*, *Walker Flip*, and *Walker Run*.
- **Quadruped Domain** features four-legged locomotion with higher action space $\mathcal{A} \in \mathbb{R}^{16}$. It includes the tasks of *Quadruped Stand*, *Quadruped Walk*, *Quadruped Jump*, and *Quadruped Run*.
- **Cheetah Domain** involves fast planar locomotion with action space $\mathcal{A} \in \mathbb{R}^6$. It includes the tasks of *Cheetah Walk*, *Cheetah Walk Backward*, *Cheetah Run*, and *Cheetah Run Backward*.
- **Jaco Domain** includes manipulation tasks using a 9-DoF Jaco robotic arm. It includes the tasks of *Jaco Reach Top Left*, *Jaco Reach Top Right*, *Jaco Reach Bottom Left*, and *Jaco Reach Bottom Right*.

Datasets To promote generalization and support reward-free pretraining, the datasets in each domain are collected using 4 distinct unsupervised exploration strategies: *Random Network Distillation (RND)*, *Active Pretraining with Successor Features (APS)*, *Active Pretraining (APT)*, and *Prototype-based Exploration (PROTO)*. Each method is designed to encourage diverse trajectories without relying on task-specific reward signals.

- **Random Network Distillation (RND)** [4] leverages the prediction error of a randomly initialized neural network as an intrinsic reward. This encourages agents to explore novel states with high prediction error, thereby promoting broad state coverage.
- **Active Pretraining with Successor Features (APS)** [16] guides exploration by maximizing the norm of successor features computed from learned representations. This drives the agent toward informative and discriminative states under the successor feature formulation.
- **Active Pretraining (APT)** [17] maximizes a non-parametric entropy in an abstract representation space to drive exploration, avoiding explicit density modeling.
- **Prototype-based Exploration (PROTO)** [31] learns a set of prototypical representations that summarize the agent's exploratory experience. These prototypes serve as a compact basis for state encoding and are learned in a self-supervised manner without downstream task labels, facilitating efficient exploration.

These strategies generate diverse and task-agnostic trajectories, which serve as a foundation for evaluating zero-shot generalization in visual URL.

F. Baseline Methods

To comprehensively evaluate the generalization of SRCP framework, we compare it against a wide range of successor representation (SR) approaches. These methods can be broadly categorized into two groups:

(1) Successor Feature (SF) Methods with Different Basic Feature Learning Techniques [7]. These approaches follow the classical successor feature formulation but differ in how they learn or define the underlying basic representations:

- **Auto encoder (AE)** [22]: This method learns state embeddings by reconstructing raw observations through a bottleneck neural network. It encodes each input into a compact latent representation and then decodes it back to the original observation space. By minimizing reconstruction error, the learned embeddings capture salient visual features of the environment, compressing redundant information while preserving the dominant factors of variation.
- **Successor Features with Laplacian Eigenfunctions (Lap)** [30]: This method learns state embeddings by computing the eigenfunctions of the graph Laplacian constructed from state transitions. It minimizes differences between adjacent state representations under a behavior policy while ensuring orthonormality, promoting clustering among nearby states and separation of distant ones.
- **Contrastive Learning (CL)** [2]: CL applies a SimCLR-style contrastive objective to learn representations by distinguishing positive state pairs (e.g., temporally close) from negatives (e.g., randomly sampled). It effectively learns features aligned with the spectral structure of the successor measure but requires full trajectories and sufficient data diversity.
- **Low-Rank Approximation of Successor Representations (LRA-SR)** [30]: LRA-SR extends CL by using temporal-difference learning to estimate the successor measure, improving stability compared to Monte Carlo estimation. It normalizes feature vectors to enforce orthogonality, resulting in a low-rank approximation of the successor representation matrix.
- **Forward Dynamics Model (FDM)** [28]: FDM leverages a learned forward model to predict future state embeddings from current ones, implicitly encouraging features to capture dynamics. It is a simple and scalable way to encode temporal structure for successor learning.
- **Hilbert Representations (HILP)** [18]: HILP learns basic representations by optimizing a goal-conditioned value function through inner products in a Hilbert space. Unlike contrastive or reconstruction-based methods, HILP directly aligns feature learning with the goal-reaching structure of the MDP, enabling more effective downstream adaptation.

(2) Forward-Backward Representation (FB) [28]: The FB method avoids explicit feature learning by directly estimating successor measures using forward and backward representations. It circumvents the challenges of learning stable embeddings and has achieved state-of-the-art results in zero-shot URL.

These baselines cover a wide spectrum of successor representation learning strategies for URL. SRCP is evaluated in comparison to each of these to highlight its generalization capabilities in visual URL. In particular, SRCP differs by incorporating saliency-guided dynamics representation learning and consistency policy modeling, allowing it to achieve superior zero-shot performance on high-dimensional visual tasks.

G. Hyper-parameter settings

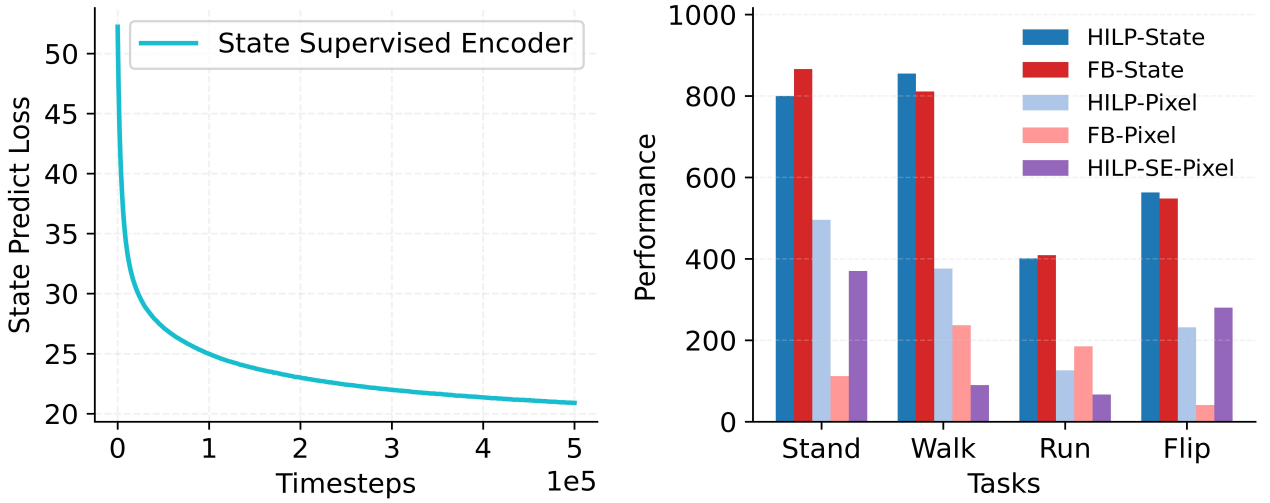
In this section, we provide the detailed hyper-parameter settings of our proposed SRCP, as shown in Table 2.

Table 2. Hyper-parameter settings.

Hyper-parameter	Setting
Pre-training frames	$5e^5$
Zero-shot selection frames	$1e^4$
RL replay buffer size	$1e6$
Frame stack	3
Action repeat	1
z vector dimensions	50
z vector space	continuous
RL backbone algorithm	DDPG
Return discount	0.99
Batch size	512
Optimizer	Adam
Learning rate	$1e-4$
Actor activation	$\mathrm{layernorm(Tanh)} \rightarrow \mathrm{ReLU}$
Agent update frequency	2
Target critic network EMA	0.01
Exploration stddev clip	0.3
Exploration stddev value	0.2
coefficient ω	3
coefficient β	0.5
coefficient λ_1	0.2
coefficient λ_1	0.2

H. Understanding the Limits of Visual Representation Learning for Successor Representations

Can We Directly Learn Physical State Representations from Pixels? To assess whether physical state supervision can effectively guide representation learning in visual URL, we explore the feasibility of regressing low-dimensional physical state vectors directly from image observations. Specifically, we decouple the training into two stages: (1) an encoder is trained via supervised regression to predict physical states from visual inputs, and (2) the resulting features are used to train a successor measure module for downstream tasks. As shown in Fig.3(a), the representation learned directly from ground-truth physical states achieves convergence in state prediction loss, but the converged value remains high. Furthermore, Fig.3(b) shows that, compared to learning representations directly from visual inputs using the SR objective, using ground-truth physical states does not lead to significant performance improvement. This suggests that, even with access to accurate physical states, the encoder still struggles to recover precise state dynamics. This limitation carries over to the successor feature learning phase, where no notable improvement in generalization is observed. These findings suggest that accurately reconstructing physical states from high-dimensional images is inherently challenging, and that such reconstruction-based objectives may be insufficient for effective successor training and task generalization.



(a) State Prediction Loss

(b) Performance Across Tasks

Figure 3. Experimental results of training an encoder using physical states as supervision in the Walker domain. (a) Prediction loss of the encoder during training; (b) Comparison of HILP and FB across different tasks under state input, visual input, and HILP with state-supervised encoder.

Beyond Physical States: The Role of Dynamics-Relevant Attention. Recognizing the limitations of physical state regression, we pose a more fundamental question: what kind of representation is truly useful for successor representation (SR) learning in visual URL? To explore this, we conduct attention analysis comparing the focus of encoders trained with conventional SR objectives versus those guided by our proposed saliency-based approach. As illustrated in the main text, conventional SR methods tend to capture static or background regions in observations, which offer poor dynamics-relevant representations. In contrast, our Saliency-Guided Dynamics Encoder (SDE) consistently attends to dynamics-relevant regions that are crucial for accurate successor estimation. This reveals a key insight: rather than attempting to reconstruct physical states, it is more effective to learn dynamics-relevant representations that suitable for successor measure. By introducing the saliency-guided dynamic representation learning, our method explicitly encourages the encoder to focus on dyanmics-relevant regions, resulting in significantly improved generalization in visual URL.

I. Effectiveness of Saliency-Guided Dynamic Representation Learning

To evaluate the impact of Saliency-Guided Dynamic Representation Learning on learned representations, we provide detailed comparisons of saliency maps and activation heatmaps produced by our encoder and those from existing SR methods. As shown in Fig.4, baseline methods such as FB and HILP tend to focus on task-irrelevant regions of the observation, whereas our SRCP framework consistently attends to more accurate dynamics-relevant regions. This demonstrates the effectiveness of our method in guiding the encoder to learn more meaningful dynamics-relevant representations.

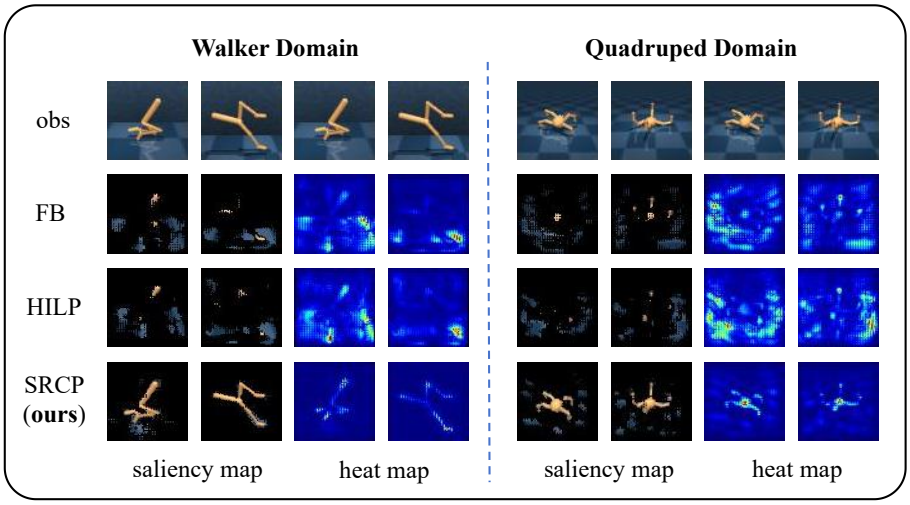


Figure 4. Detailed attention heat map of SR methods.

To further evaluate the quality of representations learned by SRCP's saliency-guided dynamics encoder, we assess its ability to capture essential physical properties of the environment. Specifically, we compare SRCP with two SR method baselines, HILP and FB, by examining how effectively their pretrained encoders support the prediction of physical state information. Following a linear probing protocol, we freeze the encoder parameters and append a randomly initialized linear layer. This linear layer is trained to regress the underlying physical state vectors from the latent representations in the *Walker* domain. We track both the convergence speed and final regression accuracy to measure the informativeness of the learned features. As shown in Fig.5, the SRCP encoder achieves significantly faster convergence and lower prediction error than those from HILP and FB. This indicates that the saliency-guided dynamic representation learning in SRCP leads to more physically effective representations, which are better aligned with the true underlying dynamics of the environment. Such representations are essential not only for accurate successor measure estimation but also for zero-shot task generalization in visual URL.

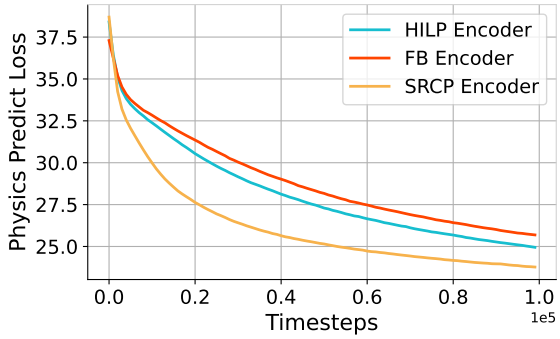


Figure 5. Prediction loss of physical states in walker domain.

J. Full Ablation Study in RND Dataset

To investigate the contribution of each component in the SRCP framework, we conduct an ablation study on all 16 tasks from the RND dataset using 4 random seeds. The compared variants include:

- **HILP**: A baseline that jointly trains the encoder and successor features using the Hilbert basic feture learning and SR objective.
- **SRCP w/o SE**: A variant of SRCP that removes the saliency-guided dynamics encoder, using the consistency policy with an encoder trained via Hilbert basic feature learning and the SR objective.
- **SRCP w/o CM**: A variant of SRCP that removes the consistency actor, employing the saliency-guided dynamics encoder while using standard policy learning.
- **SRCP**: The complete SRCP method, combining saliency-guided dynamics representation learning with consistency-based policy learning.

As shown in Table 3, both **SRCP w/o SE** and **SRCP w/o CM** outperform the HILP baseline across most domains, confirming the benefits of both the saliency-guided dynamics encoder and the consistency policy. Specifically, **SRCP w/o CM** performs better than HILP in most tasks, demonstrating that saliency-guided representations improve generalization. Meanwhile, **SRCP w/o SE** achieves higher performance in several tasks, indicating the effectiveness of the consistency policy in modeling multi-modal skill-conditioned behaviors. The full **SRCP** model consistently achieves the best overall results across all domains, validating that both components work synergistically to enhance generalization in visual URL.

Table 3. Full Ablation Study

Domain	Task	HILP	SRCP w/o SE	SRCP w/o CM	SRCP
Walker	Stand	496 ± 73	519 ± 12	658 ± 38	671 ± 51
	Walk	376 ± 52	423 ± 12	449 ± 75	520 ± 32
	Run	126 ± 8	145 ± 4	173 ± 21	194 ± 38
	Flip	232 ± 41	293 ± 5	303 ± 13	369 ± 25
	Average	308	345	396	439
Quadruped	Stand	327 ± 126	476 ± 9	595 ± 30	703 ± 22
	Walk	163 ± 45	253 ± 15	305 ± 14	339 ± 27
	Run	148 ± 19	289 ± 35	310 ± 32	361 ± 24
	Jump	244 ± 122	391 ± 34	413 ± 18	535 ± 14
	Average	221	352	406	485
Cheetah	Walk	895 ± 33	893 ± 44	807 ± 26	859 ± 34
	Walk Backward	927 ± 35	951 ± 23	954 ± 22	934 ± 45
	Run	276 ± 46	277 ± 14	292 ± 25	322 ± 44
	Run Backward	297 ± 46	278 ± 3	338 ± 6	293 ± 42
	Average	599	600	598	602
Jaco	Top Left	29 ± 4	30 ± 8	49 ± 3	53 ± 9
	Top Right	35 ± 13	54 ± 2	63 ± 9	60 ± 7
	Bottom Left	25 ± 3	43 ± 3	23 ± 4	43 ± 8
	Bottom Right	31 ± 11	49 ± 7	39 ± 4	42 ± 8
	Average	30	44	43	50

K. Effectiveness of Representation Learning Methods for Visual URL

To evaluate the impact of different representation learning strategies under the HILP framework, we compare five variants on the RND dataset: the original HILP, HILP-TACO, HILP-TACOP, HILP-SE (SRCP w/o CM), and the full SRCP method. HILP-TACO and HILP-TACOP incorporate the state-of-the-art representation learning methods TACO [33] and TACO-Premier [34] into HILP, aiming to decouple the encoder from the successor representation objective. In contrast, HILP-SE and SRCP adopt our proposed saliency-guided dynamics encoder to learn dynamics-relevant visual representations.

As shown in Table 4, HILP-TACO and HILP-TACOP achieve limited improvements over HILP, with performance gains in some tasks (e.g., walker_stand or jaco_reach_top_left) but substantial drops in others (e.g., walker_walk or cheetah_run_backward), leading to inconsistent generalization. This suggests that applying generic representation learning alone is insufficient for achieving robust generalization performance across diverse tasks. In contrast, HILP-SE significantly outperforms all three baselines in every domain, particularly in *Walker* and *Quadruped*, where it improves average zero-shot returns by over 90% compared to HILP. Furthermore, SRCP consistently achieves the best performance across most tasks, demonstrating that combining the saliency-guided dynamics encoder with consistency policy learning further enhances multi-modal action distribution modeling ability and generalization. These results clearly highlight the superiority of the proposed saliency-guided dynamics representation in visual URL. It enables the agent to focus on dynamics-relevant features and supports more effective zero-shot generalization.

Table 4. Zero-shot Performance of HILP Variants on the RND Dataset

Domain	Task	HILP	HILP-TACO	HILP-TACOP	HILP-SE	SRCP
Walker	Stand	496 ± 73	388 ± 8	622 ± 24	658 ± 38	671 ± 51
	Walk	376 ± 52	111 ± 6	339 ± 14	449 ± 75	520 ± 32
	Run	126 ± 8	73 ± 3	135 ± 10	173 ± 21	194 ± 38
	Flip	232 ± 41	99 ± 8	199 ± 24	303 ± 13	369 ± 25
	Average	308	168	324	396	439
Quadruped	Stand	327 ± 126	234 ± 18	195 ± 16	595 ± 30	703 ± 22
	Walk	163 ± 45	91 ± 19	122 ± 13	305 ± 14	339 ± 27
	Run	148 ± 19	73 ± 10	101 ± 6	310 ± 32	361 ± 24
	Jump	244 ± 122	86 ± 13	149 ± 16	413 ± 18	535 ± 14
	Average	221	121	142	406	485
Cheetah	Walk	895 ± 33	693 ± 33	599 ± 33	807 ± 26	859 ± 34
	Walk Backward	927 ± 35	422 ± 15	534 ± 89	954 ± 22	934 ± 45
	Run	276 ± 46	136 ± 6	185 ± 7	292 ± 25	322 ± 44
	Run Backward	297 ± 46	80 ± 2	124 ± 24	338 ± 6	293 ± 42
	Average	599	333	361	598	602
Jaco	Top Left	29 ± 4	46 ± 4	33 ± 2	49 ± 3	53 ± 9
	Top Right	35 ± 13	43 ± 3	24 ± 3	63 ± 9	60 ± 7
	Bottom Left	25 ± 3	25 ± 6	41 ± 3	23 ± 4	43 ± 8
	Bottom Right	31 ± 11	23 ± 4	24 ± 6	39 ± 4	35 ± 10
	Average	30	34	31	43	48

L. Experimental Result on Forward Backward Representation

To further validate the generality of SRCP, we integrate the Forward-Backward (FB) [27] representation into the SRCP framework, denoted as SRCP(FB). Table 5 presents the detailed zero-shot performance of FB and SRCP(FB) across four domains (Walker, Quadruped, Cheetah, and Jaco) in the APS dataset. SRCP(FB) consistently achieves higher returns in all tasks, with particularly notable improvements in the Walker and Cheetah domains. These results align with the findings in the main paper, confirming that SRCP provides a general and effective framework for enhancing the representation quality, skill expressiveness, and zero-shot generalization of various SR-based methods.

Table 5. Zero-shot Performance of FB and SRCP(FB) on the APS Dataset

Domain	Task	FB	SRCP(FB)
Walker	Stand	304 ± 122	680 ± 14
	Walk	81 ± 14	478 ± 18
	Run	51 ± 13	152 ± 12
	Flip	66 ± 8	274 ± 13
	Average	126	**396**
Quadruped	Stand	421 ± 16	583 ± 21
	Walk	239 ± 12	294 ± 10
	Run	208 ± 14	293 ± 8
	Jump	294 ± 9	421 ± 32
	Average	318	**398**
Cheetah	Walk	12 ± 2	291 ± 84
	Walk Backward	48 ± 7	495 ± 19
	Run	13 ± 5	57 ± 7
	Run Backward	9 ± 4	103 ± 13
	Average	21	**237**
Jaco	Top Left	22 ± 6	46 ± 6
	Top Right	23 ± 4	21 ± 3
	Bottom Left	36 ± 13	48 ± 15
	Bottom Right	19 ± 6	21 ± 5
	Average	25	**34**

M. Full Hyperparameter Ablation Study

We provide a comprehensive ablation study on the sensitivity of SRCP to two key hyperparameters, ω and β. Table 6 shows effect of ω in Walker domain, ω controls the strength of skill-conditioned guidance. Setting $\omega = 0$ leads to a marked performance drop, while moderate values enhance generalization. In contrast, large ω over-constrains the policy and slightly reduces performance. Table 7 analyzes the effect of β in Walker domain, which controls the weight of saliency guidance in representation learning. Increasing β improves generalization up to a point, after which excessive emphasis on salient regions weakens the model's ability to capture dynamics cues. Overall, SRCP maintains stable performance across a wide range of hyperparameters, with balanced ω and β achieving the best trade-off between skill and saliency guidance.

Walker	$\omega = 0$	$\omega = 2$	$\omega = 3$	$\omega = 4$	$\omega = 6$
Stand	176 ± 7	609 ± 40	$\mathbf{671 \pm 51}$	498 ± 30	363 ± 16
Walk	42 ± 4	415 ± 40	$\mathbf{520 \pm 32}$	464 ± 16	79 ± 13
Run	32 ± 6	136 ± 18	$\mathbf{194 \pm 38}$	164 ± 11	70 ± 14
Flip	59 ± 7	$\mathbf{374 \pm 18}$	369 ± 25	358 ± 18	206 ± 10
Average	77	384	**439**	371	180

Table 6. Ablation study of parameter ω in SRCP on Walker domain in RND dataset, with 4 random seeds per task.

Walker	$\beta = 0$	$\beta = 0.2$	$\beta = 0.5$	$\beta = 1$
Stand	544 ± 20	550 ± 26	$\mathbf{671 \pm 51}$	583 ± 18
Walk	354 ± 42	458 ± 19	$\mathbf{520 \pm 32}$	484 ± 48
Run	156 ± 12	170 ± 35	$\mathbf{194 \pm 38}$	169 ± 23
Flip	310 ± 15	315 ± 11	$\mathbf{369 \pm 25}$	346 ± 5
Average	341	373	**439**	396

Table 7. Ablation study of parameter β in SRCP on Walker domain in RND dataset, with 4 random seeds per task.

N. Computational Cost

As shown in Table 8, the proposed SRCP framework achieves training time and memory consumption that are comparable to those of the baseline methods across all experimental settings. By explicitly decoupling representation learning and value learning into separate optimization objectives, SRCP avoids the computationally expensive joint optimization commonly adopted in conventional end-to-end frameworks. This structural design effectively offsets the additional computational overhead introduced by saliency map generation, leading to similar wall-clock training time compared to the baselines. Overall, SRCP only incurs minor and acceptable extra overhead. On the one hand, the decoupled training paradigm eliminates chained backpropagation across multiple modules, which significantly reduces gradient computation complexity. On the other hand, the saliency inference module is designed to be lightweight and efficient, introducing negligible latency during training. Meanwhile, the consistency-based policy regularization imposes computational costs comparable to those of standard policy optimization objectives, without introducing heavy additional computation. Consequently, the total training cost of SRCP remains competitive with state-of-the-art approaches despite the integration of multiple learning objectives.

Table 8. Computational cost on an A800 GPU for 5M steps.

Metric	FB	HILP	SRCP
Training Time (h)↓	**23.1**	24.2	24.4
Memory Usage (GB)↓	9.6	**8.8**	9.1

O. Visually Complex Environments

To further validate the generalization ability of SRCP in complex and noisy visual environments, we evaluate its performance on the DMC-GB benchmark, which contains various types of visual distractions. Specifically, we train SRCP on environments with clean backgrounds and then test it under two challenging visual settings for the Walker task: Color Easy (mild color perturbations) and Color Hard (severe color variations). As reported in Table 9, SRCP consistently outperforms the baseline methods FB and HILP across both settings, which demonstrates its strong generalization capability against diverse task dynamics and visual appearance variations. In addition to visual robustness, we assess the scalability and cross-domain applicability of SRCP beyond locomotion tasks. To this end, we conduct experiments on the Point Mass Maze navigation task, a representative goal-directed control problem distinct from continuous locomotion. The quantitative results are presented in Table 10. SRCP achieves significantly improved performance compared with all baselines, indicating that the proposed framework can effectively transfer its learning paradigm to different task domains and maintain strong decision-making performance. For a more comprehensive evaluation under complex visual perturbations, we further test SRCP on the full DMC-GB benchmark with multiple types of visual disturbances. The agent is trained on clean backgrounds and then evaluated under three increasingly challenging test scenarios: dynamic easy color perturbations, hard color perturbations, and natural video distractions. Consistent with previous observations, the results in Table 9 confirm that SRCP still outperforms FB and HILP by a clear margin, further verifying its superior generalization ability in the presence of complex and realistic visual variations. In summary, experiments on both the DMC-GB visual generalization benchmark and the Point Mass Maze navigation task demonstrate that SRCP not only achieves strong robustness to visual noise and domain shifts but also scales effectively across different task categories, showing wide applicability in visual reinforcement learning.

Table 9. Performance on DMC-GB Walker.

		Color Easy			Color Hard			Video Easy		
Domain	Task	FB	HILP	SRCP	FB	HILP	SRCP	FB	HILP	SRCP
Walker	Stand	243 ± 80	364 ± 89	639 ± 69	165 ± 66	282 ± 74	506 ± 58	127 ± 42	176 ± 32	277 ± 48
	Walk	62 ± 29	110 ± 54	423 ± 47	54 ± 27	73 ± 34	279 ± 45	47 ± 14	62 ± 13	162 ± 47
	Run	46 ± 25	55 ± 18	165 ± 31	39 ± 16	49 ± 21	113 ± 36	38 ± 8	40 ± 5	78 ± 10
	Flip	59 ± 22	98 ± 34	371 ± 42	51 ± 25	69 ± 36	156 ± 38	47 ± 11	56 ± 11	123 ± 44
	Average	103	157	**400**	77	118	**264**	65	84	**160**

Table 10. Performance on Point Mass Maze navigation tasks.

Method	Reach Top Left	Reach Top Right	Reach Bottom Left	Reach Bottom Right	Average
FB	14 ± 5	3 ± 1	1 ± 1	0 ± 0	5
HILP	471 ± 51	112 ± 19	36 ± 10	2 ± 1	155
SRCP	635 ± 95	166 ± 55	67 ± 18	10 ± 6	**220**

P. Difference between Diffusion Models and Consistency Models

We further compare SRCP with the representative diffusion policy on the challenging Proto Quadruped locomotion tasks to verify its efficiency and practical deployment potential. The quantitative comparison is summarized in Table 11. On one hand, the full diffusion model with 40 denoising steps achieves slightly higher task performance than SRCP, but this improvement comes at the cost of drastically increased computation: it consumes nearly 2.5× the total training time of SRCP. On the other hand, when diffusion is constrained to only a single denoising step for inference efficiency, its performance drops significantly and becomes considerably worse than SRCP. In contrast, SRCP maintains strong task performance with inherently efficient single-step execution, without relying on iterative denoising. These results clearly demonstrate that the proposed consistency policy effectively balances performance and computational overhead, yielding a superior performance–computation trade-off compared with diffusion-based policies for real-world robotic control.

Table 11. Comparison between diffusion models and SRCP.

Method	Stand↑	Walk↑	Run↑	Jump↑	Average↑	Time (h) ↓
Diffusion (1 step)	626 ± 50	308 ± 37	324 ± 18	397 ± 40	414	**24.7**
Diffusion (40 steps)	724 ± 30	357 ± 43	354 ± 17	531 ± 26	**492**	64.8
SRCP	703 ± 22	339 ± 27	361 ± 24	535 ± 14	485	25.4

Q. Inference Budget

We further evaluate the sensitivity of SRCP to the number of task-specific transitions used during adaptation, by conducting experiments on the Walker domain with a wide range of inference budgets (as shown in Table 12). When only 100 task-specific transitions are available, the overall performance shows a slight decrease due to the limited task-specific information. However, across the range from 500 up to 20,000 transitions, the performance of SRCP remains stable and consistent without significant degradation. This observation demonstrates that SRCP can achieve reliable adaptation with only a modest number of task-specific transitions, and is robust to the choice of inference budget, making it suitable for real-world scenarios where collecting large amounts of task-specific data can be costly or impractical.

Table 12. Effect of inference transition on SRCP performance.

Inference Transitions:	100	500	1k	5k	10k	20k
Stand	828 ± 26	817 ± 19	823 ± 35	834 ± 34	830 ± 39	834 ± 32
Walk	466 ± 87	478 ± 35	508 ± 47	504 ± 46	504 ± 66	506 ± 31
Run	207 ± 42	216 ± 33	224 ± 22	220 ± 26	227 ± 16	238 ± 12
Flip	376 ± 86	418 ± 26	413 ± 24	434 ± 35	428 ± 75	420 ± 28
blue!10 Average	469	482	492	498	497	**500**

References

[1] Chenjia Bai, Rushuai Yang, Qiaosheng Zhang, Kang Xu, Yi Chen, Ting Xiao, and Xuelong Li. Constrained ensemble exploration for unsupervised skill discovery. In *International Conference on Machine Learning*, pages 2418–2442, 2024. 8

[2] Randall Balestriero and Yann LeCun. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. In *Advances in Neural Information Processing Systems*, pages 26671–26685, 2022. 8, 10

[3] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, page 4058–4068, 2017. 8

[4] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. 8, 9

[5] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. In *International Conference on Learning Representations*, 2023. 8

[6] Yuhui Chen, Haoran Li, and Dongbin Zhao. Boosting continuous control with consistency policy. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, page 335–344, 2024. 8

[7] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. In *Neural Computation*, pages 613–624, 1993. 8, 10

[8] Yarats Denis, Brandfonbrener David, Liu Hao, Laskin Michael, Abbeel Pieter, Lazaric Alessandro, and Pinto Lerrel. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022. 9

[9] Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *International Conference on Learning Representations*, 2024. 8

[10] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. 8

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 8

[12] Bingyi Kang, Xiao Ma, Chao Du, Tianyu Pang, and Shuicheng Yan. Efficient diffusion policies for offline reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. 8

[13] Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, and Pieter Abbeel. Urlb: Unsupervised reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 2021. 8, 9

[14] Michael Laskin, Hao Liu, Xue Bin Peng, Denis Yarats, Aravind Rajeswaran, and Pieter Abbeel. Unsupervised reinforcement learning with contrastive intrinsic control. pages 34478–34491, 2022. 8

[15] Haoran Li, Zhennan Jiang, yuhui Chen, and Dongbin Zhao. Generalizing consistency policy to visual rl with prioritized proximal experience regularization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 8

[16] Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pages 6736–6747, 2021. 9

[17] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, pages 18459–18473, 2021. 8, 9

[18] Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *Proceedings of the 41st International Conference on Machine Learning*, pages 39737–39761. PMLR, 2024. 8, 10

[19] Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. In *International Conference on Learning Representations*, 2024. 8

[20] Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, pages 5062–5071, 2019. 8

[21] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *International Conference on Learning Representations*, 2023. 8

[22] Lucas Pinheiro Cinelli, Matheus Araújo Marins, Eduardo Antúnio Barros da Silva, and Sérgio Lima Netto. Variational autoencoder. In *Variational methods for machine learning with applications to deep networks*, pages 111–149. Springer, 2021. 10

[23] Tongzheng Ren, Tianjun Zhang, Lisa Lee, Joseph E Gonzalez, Dale Schuurmans, and Bo Dai. Spectral decomposition representation for reinforcement learning. In *International Conference on Learning Representations*, 2023. 8

[24] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International Conference on Machine Learning*, pages 1312–1320, 2015. 8

[25] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning*, pages 32211–32252. PMLR, 2023. 8

[26] R.S. Sutton and A.G. Barto. Reinforcement learning: An introduction. *IEEE Transactions on Neural Networks*, 9(5):1054–1054, 1998. 8

[27] Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems*, pages 13–23, 2021. 5, 8, 16

[28] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *International Conference on Learning Representations*, 2023. 8, 10

[29] J Wang, Q Zhang, and D Zhao. Dynamic-horizon model-based value estimation with latent imagination. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):8812–8825, 2024. 8

[30] Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. In *International Conference on Learning Representations*, 2018. 8, 10

[31] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. In *International Conference on Machine Learning*, pages 11920–11931. PMLR, 2021. 8, 9

[32] Qichao Zhang, Yinfeng Gao, Yikang Zhang, Youtian Guo, Dawei Ding, Yunpeng Wang, Peng Sun, and Dongbin Zhao. Trajgen: Generating realistic and diverse trajectories with reactive and feasible agent behaviors for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):24474–24487, 2022. 8

[33] Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé, and Furong Huang. Taco: temporal latent action-driven contrastive loss for visual reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, 2023. 15

[34] Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé III, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-taco is a few-shot policy learner: pretraining multitask representation via temporal action-driven contrastive loss. In *Proceedings of the 41st International Conference on Machine Learning*. JMLR.org, 2024. 15