SURVEY: **META LEARNING** AND **SKILL COMBINATION** FOR **ROBOTICS**

IN SUPREMÆ DIGNITATIS
1343

SERGIO LATROFA - 640584

ISPR - FINAL PROJECT

OVERVIEW

The aim of this work is to analyze and draw connections between frameworks to search an **optimal initial configuration** for learning models, accelerating future training and enforcing particular reasoning-like capabilities such as *skill based planning*.

FROM GENERAL KNOWLEDGE TO SPECIALIZATION

~

Are some starting points better than others?

Although SGD would more or less converge, models are always more frequently demanded to specialize on unseen task in a small number of steps (***few-shot learning***).

~

Just a pre-training?

No. Optimized objectives are not the same of training, as well as data, coming from not necessarily similar tasks.

~

Just a theoretical trip?

Some *impactful applications* will be seen, but the horizon for future ones is still wide.

.

M.A.M.L.

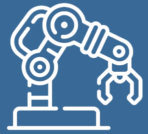
A general purpose framework to find an optimal initial configuration for gradient based models, so that future learning would be faster.

D.A.D.S.

Analyze unsupervisedl past data to find brief impactful and predictable sequences of action to further reuse for MBRL

S.p.iR.L.

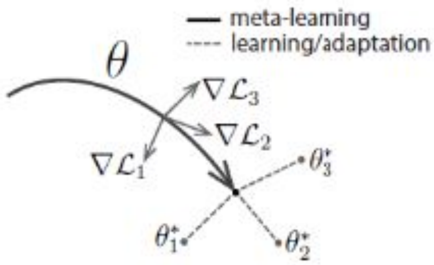
Reuse agent’s past experience to compute priors distribution towards most useful skills given states and then optimize a Hierarchical Policy.



Actionable Models

Goal reaching Q-learning based algorithm encouraging solutions composed from sequences of past experiences..

META LEARNING



Goal is to prepare a model for **few shot learning.**

Intuition is that some internal representation are more transferable than others.

How can the emerge of such general purpose features being encouraged?

MAML Explicit approach: just look at the SGD *learning rule*. Enforce parameters being sensitive to new tasks.

Such optimization is feasible for any *SGD-based* learning model, including *deep neural networks*.

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$$

Are there points where such gradient is steeper?

The answer is in its own derivative (**Hessian** of loss)

$$\min_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)}\right)$$

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}\left(f_{\theta_i'}\right)$$

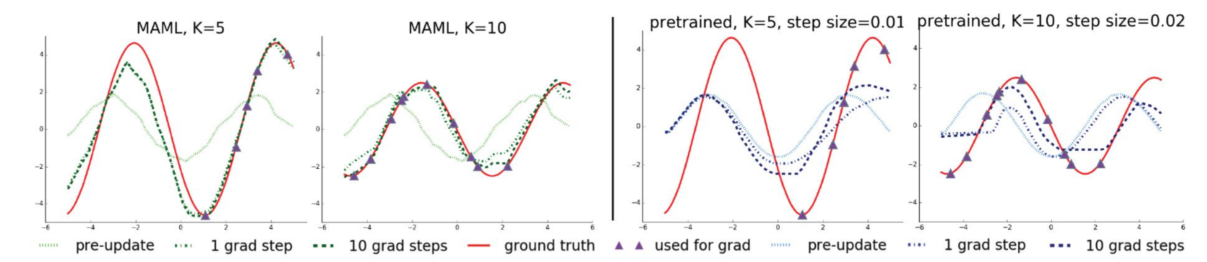
Meta learning rule:

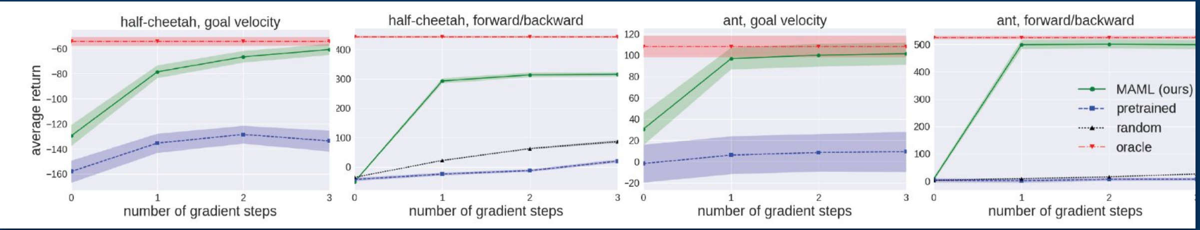
Update parameters according to the gradient of the updated model.

MODEL AGNOSTIC META LEARNING

Meta train a *SGD-based* model on tasks $\mathcal{T} = \{\mathcal{L}(\mathbf{x}_1, \mathbf{a}_1, \ldots, \mathbf{x}_H, \mathbf{a}_H), q(\mathbf{x}_1), q(\mathbf{x}_{t+1} \mid \mathbf{x}_t, \mathbf{a}_t), H\}$ sampled from a distribution $p(T)$.

For <u>supervised</u> tasks, assume horizon $H = 1$. q_i is the transition distribution of the task.





What about RL?

Model f_θ is a policy mapping state to actions toward an horizon H.

Loss corresponds to the negative reward:

$$\mathcal{L}_{\mathcal{T}_i}(f_\phi) = -\mathbb{E}_{\mathbf{x}_t, \mathbf{a}_t \sim f_\phi, q_{\mathcal{T}_i}} \left[\sum_{t=1}^{H} R_i(\mathbf{x}_t, \mathbf{a}_t) \right]$$

<u>Optimize policy gradients!</u>

Intractable differentiation can safely be approximated **using first order models**, without any observed performance decay.

DYNAMICS AWARE DISCOVERY OF SKILLS

Goal is to learn skills with an <u>impactful outcome</u>, being as more **predictable** as possible (<u>low variance</u>).

Idea is to use **information theory:**

$$\mathcal{I}(s';z|s) = \underbrace{\mathcal{H}(z|s) - \mathcal{H}(z|s',s)} = \underbrace{\mathcal{H}(s's) - \mathcal{H}(s'|s,z)}$$

<u>How much can be known about next state given a skill</u>

<u>Diversity of transition</u> minus <u>uncertainty about next state given z.</u>

$$\mathcal{I}(s';z \mid s) = \int p(z,s,s') \log \frac{p(s'|s,z)}{p(s'|s)} ds' ds dz$$

<u>Rewritten using definition of conditional mutual information</u>

$$p(z,s,s') = p(z)p(s|z)p(s'|s,z)$$

Unknown dynamics:

Intractable

<u>Generative process: Skill prior</u> - <u>Policy induced transition</u> - <u>transition distribution under skill z.</u>

$$\mathcal{I}(s';z \mid s) = \mathbb{E}_{z,s,s'\sim p}\left[\log \frac{p(s' \mid s,z)}{p(s' \mid s)}\right]$$

<u>Apply a variational lower bound</u>

$$= \mathbb{E}_{z,s,s'\sim p}\left[\log \frac{q_\phi(s' \mid s,z)}{p(s' \mid s)}\right] + \mathbb{E}_{s,z\sim p}\left[\mathcal{D}_{KL}\left(p(s' \mid s,z) \| q_\phi(s' \mid s,z)\right)\right]$$

$$\geq \mathbb{E}_{z,s,s'\sim p}\left[\log \frac{q_\phi(s' \mid s,z)}{p(s' \mid s)}\right]$$

<u>Being KL divergence always non negative.</u>

Alternate optimization of the bounds:

<u>*Tighten variational lower bound*</u>

Minimize the *KL* w.r.t. to parameters of *q*, which corresponds to *maximize likelihood* of samples from *p* under *q*.

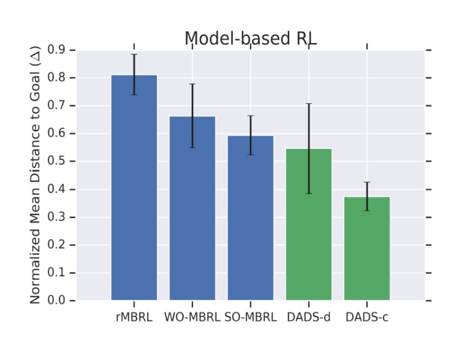
<u>Train the skill conditioned policy</u>

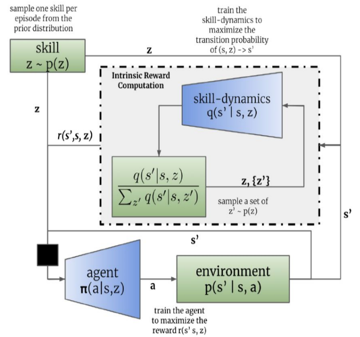
Optimize the policy maximizing the reward, again approximated sampling skills from their prior p(z).

DADS - WRAPPING UP

Reward function approximation is $r_z(s, a, s') = \log \frac{q_\phi(s'|s,z)}{\sum_{i=1}^{L} q_\phi(s'|s,z_i)} + \log L, \quad z_i \sim p(z)$ whit the summation approximating *intractable probabilities integral*. Such formulation encourages transitions predictability and also skills exploration (**diversity**) due to samling. Alternate optimization ends up returning a state-skill conditioned policy $\pi(a \mid s, z)$ and a skill-transition dynamics model $q_\phi(s' \mid s, z)$.

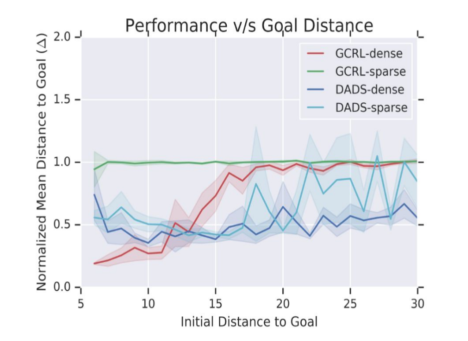
The planning problem can be solved in the **latent skills space** implicit extend the horizon by an skill length factor, allowing *temporal abstraction*. Authors propose an adaptation of the **MPC** paradigm (model-predictive-control), modeling plans as sequences of Gaussians, with their parameters refined updated for R steps and K samples using the **MPPI** (Model Predictive Path Integral) controller.





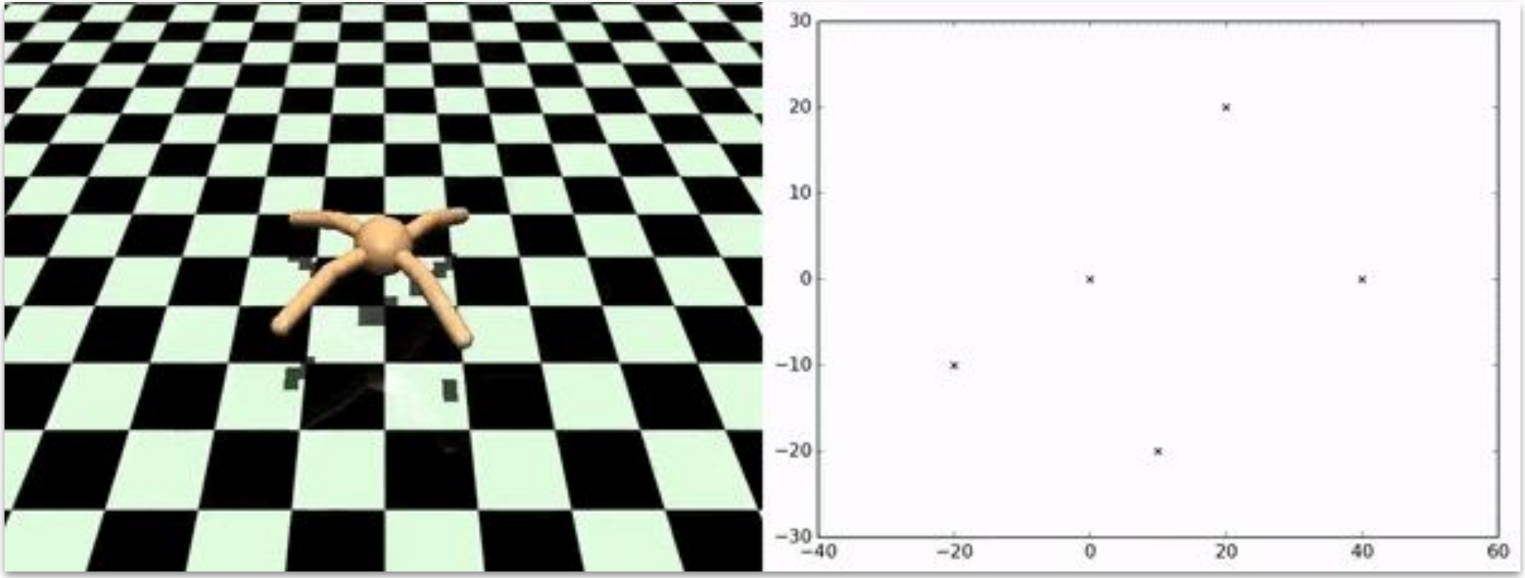
Algorithm 1: Dynamics-Aware Discovery of Skills (DADS)

Initialize π, q_ϕ;
while *not converged* **do**
  Sample a skill $z \sim p(z)$ every episode;
  Collect new M on-policy samples;
  Update q_ϕ using K_1 steps of gradient descent on M transitions;
  Compute $r_z(s, a, s')$ for M transitions;
  Update π using any RL algorithm;
end



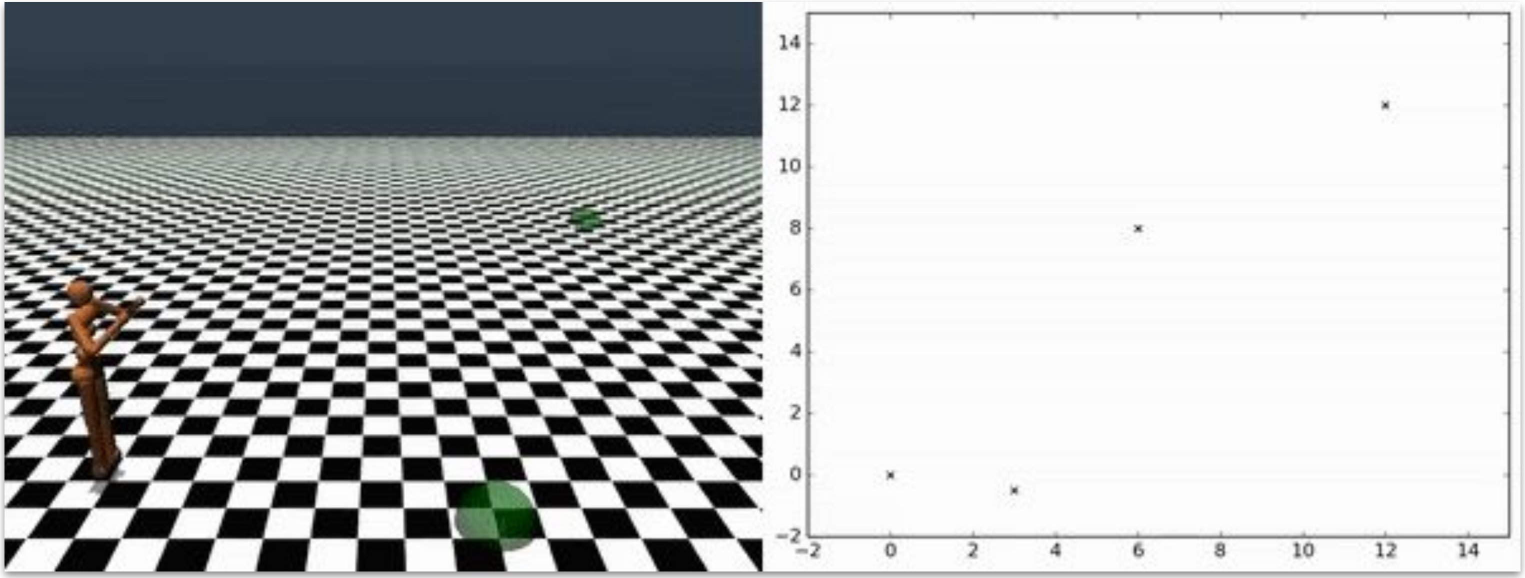
DADS - APPLICATIONS - 1

Goals (green ellipsoids on the left and 'x' on the right) are updated online, and the agent only sees the current goal. There is no training on the task, that is, it solves it in zero-shot.



DADS - APPLICATIONS - 2

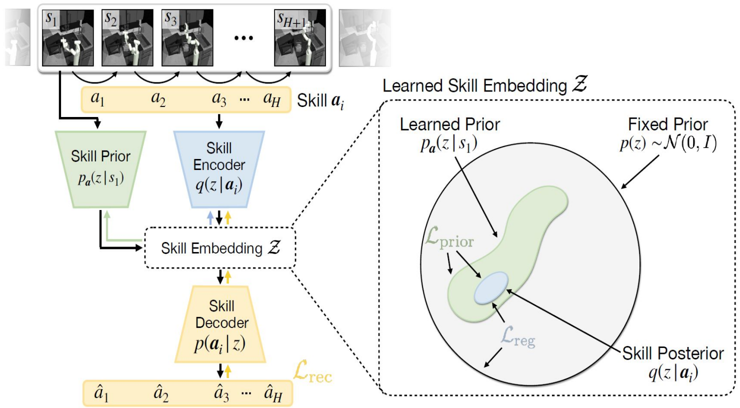
A similar demonstration for the Humanoid agent, which composes its learnt skills to follow the sequence of goals. The feasible sequence of goals is restricted compared to Ant, however, skill composition using planning can still be leveraged. The video has been sped up 2x.



ISPR - FINAL PROJECT

SKILL PRIOR RL

The proposed framework works for **entropy regularized RL** algorithms, and is based on learning a prior distribution $p_{\boldsymbol{a}}(z \mid s_t)$ over skills conditioned under states. Once skills are computed a higher level policy $\pi_\theta(z|s_t)$ can be learned, *planning in the skill space* with a further horizon.



Variational Inference again

Skills are handled through amortized variational inference, via two deep neural networks: encoder $q(z \mid \boldsymbol{a}_i)$ and $p(\boldsymbol{a}_i \mid z)$ which output the parameters of the posterior and output distributions (Gaussian).

Each skill corresponds to a sequence of action with fixed horizon $\mathcal{H}$ mapped in latent space $\mathcal{Z}$.

$$\log p(\boldsymbol{a}_i) \geq \mathbb{E}_q[\underbrace{\log p(\boldsymbol{a}_i \mid z)}_{\text{reconstruction}} - \beta(\underbrace{\log q(z \mid \boldsymbol{a}_i) - \log p(z)}_{\text{regularization}})]$$

Skill latent representation is based on sampling training instances to maximize the ELBO.
Priors are learned jointly with the encoder and decoder optimization (stability).

$$\mathbb{E}_{(s,\boldsymbol{a}_i)\sim\mathcal{D}} D_{\mathrm{KL}}\left(q(z \mid \boldsymbol{a}_i), p_{\boldsymbol{a}}(z \mid s_t)\right)$$

Minimizing the reverse KL ensures mode covering: represent all skills observed in the state

ISPR - FINAL PROJECT

SPiRL - RESULTS

Authors tested the effectiveness of the learned skill prior in a soft actor critic (**SAC**) RL model, **regularized by entropy** . Impact of the $|\mathcal{Z}|$ latente space dimensionality and the *horizon* $|H|$, with the first one to be tuned according to problem complexity and the second regulating the long term planning capacity vs. exploration tradeoff.

$$J(\theta)=\mathbb{E}_\pi\left[\sum_{t=1}^{T}\gamma^t r\left(s_t,a_t\right)+\alpha\mathcal{H}\left(\pi\left(a_t\mid s_t\right)\right)\right]$$

SAC Loss is adapted re defining the of the entropy term

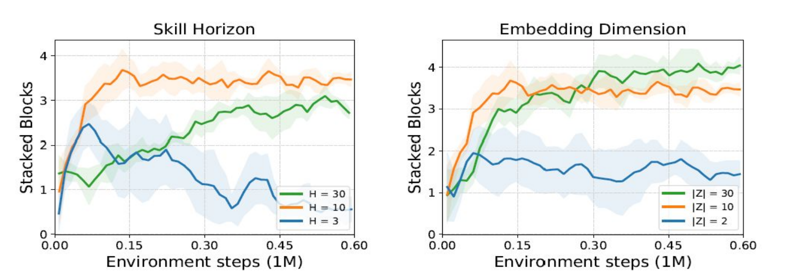
$$\mathcal{H}\left(\pi\left(a_t\mid s_t\right)\right)=-\mathbb{E}_\pi\left[\log\pi\left(a_t\mid s_t\right)\right]\propto -D_{\mathrm{KL}}\left(\pi\left(a_t\mid s_t\right),U\left(a_t\right)\right)$$

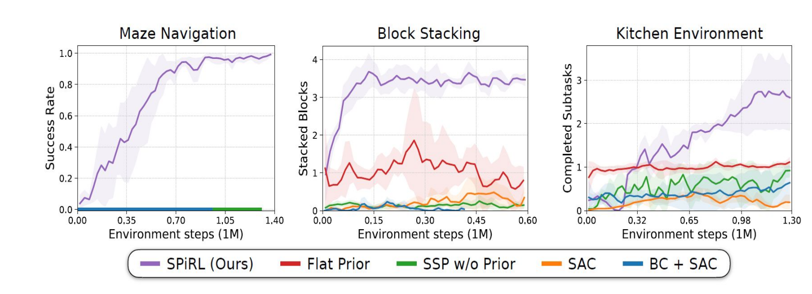
Entropy corresponds to the negated KL divergence between policy and uniform AC prior over actions.

$$J(\theta)=\mathbb{E}_\pi\left[\sum_{t=1}^{T}\tilde{r}\left(s_t,z_t\right)-\alpha D_{\mathrm{KL}}\left(\pi\left(z_t\mid s_t\right),p_{\boldsymbol{a}}\left(z_t\mid s_t\right)\right)\right]$$

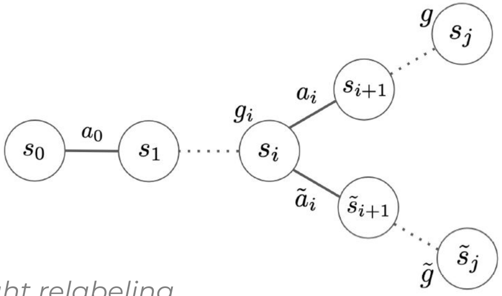
Algorithm 1 SPiRL: Skill-Prior RL

Inputs: H-step reward function $\tilde{r}(s_t,z_t)$, discount γ, target divergence δ, learning rates $\lambda_\pi,\lambda_Q,\lambda_\alpha$, target update rate τ.
Initialize replay buffer $\mathcal{D}$, high-level policy $\pi_\theta(z_t|s_t)$, critic $Q_\phi(s_t,z_t)$, target network $Q_{\bar{\phi}}(s_t,z_t)$
for each iteration **do**
  for every H environment steps **do**
    $z_t\sim\pi(z_t|s_t)$ ▷ sample skill from policy
    $s_{t'}\sim p(s_{t+H}|s_t,z_t)$ ▷ execute skill in environment
    $\mathcal{D}\leftarrow\mathcal{D}\cup\{s_t,z_t,\tilde{r}(s_t,z_t),s_{t'}\}$ ▷ store transition in replay buffer
  for each gradient step **do**
    $\bar{Q}=\tilde{r}(s_t,z_t)+\gamma\left[Q_{\bar{\phi}}(s_{t'},\pi_\theta(z_{t'}|s_{t'}))-\alpha D_{\mathrm{KL}}\left(\pi_\theta(z_{t'}|s_{t'}),p_{\boldsymbol{a}}(z_{t'}|s_{t'})\right)\right]$ ▷ compute Q-target
    $\theta\leftarrow\theta-\lambda_\pi\nabla_\theta\left[Q_\phi(s_t,\pi_\theta(z_t|s_t))-\alpha D_{\mathrm{KL}}(\pi_\theta(z_t|s_t),p_{\boldsymbol{a}}(z_t|s_t))\right]$ ▷ update policy weights
    $\phi\leftarrow\phi-\lambda_Q\nabla_\phi\left[\frac{1}{2}\left(Q_\phi(s_t,z_t)-\bar{Q}\right)^2\right]$ ▷ update critic weights
    $\alpha\leftarrow\alpha-\lambda_\alpha\nabla_\alpha\left[\alpha\cdot(D_{\mathrm{KL}}(\pi_\theta(z_t|s_t),p_{\boldsymbol{a}}(z_t|s_t))-\delta)\right]$ ▷ update alpha
    $\bar{\phi}\leftarrow\tau\phi+(1-\tau)\bar{\phi}$ ▷ update target network weights
return trained policy $\pi_\theta(z_t|s_t)$





ACTIONABLE MODELS - SETUP



Reuse **past trajectories** (*or transferred skills*), to learn a conservative Q-Learning model with *hindsight relabeling*.

$$Q^{\pi}(s_t, a_t, g) = \mathbb{E}_{\pi}\left[\sum_t \gamma^t R(s_t, a_t, g)\right] = P^{\pi}(s_T = g \mid s_t, a_t)$$

Episode terminates when a goal state is reached: TD-learning to maximize the expected return

$$\pi(a \mid s, g) = \arg\max_a Q(s, a, g)$$

Associated policy.

$$\mathcal{L}_g(\theta) = \min_{\theta} \mathbb{E}_{(s_t, a_t, s_{t+1}, g) \sim \mathcal{D}}\left[\left(Q_{\theta}(s_t, a_t, g) - y(s_{t+1}, g)\right)^2 + \underbrace{\mathbb{E}_{\tilde{a} \sim \exp(Q_{\theta})}\left[\left(Q_{\theta}(s, \tilde{a}, g) - 0\right)^2\right]}\right]$$

$$y(s_{t+1}, g) = \begin{cases} 1 & \text{if } s_{t+1} = g \\ \gamma \mathbb{E}_{a \sim \pi}\left[Q_{\theta}(s_{t+1}, a, g)\right] & \text{otherwise.} \end{cases}$$

Follow as much as possible **positive trajectories** present in the dataset (those ones ending up **reaching a goal state**), *penalizing* Q-values for *unseen* actions: $\mathbb{E}_{\tilde{a}_t \sim p_{\tilde{\mathcal{A}}}}[Q^{\pi}(s_t, \tilde{a}_t, g)] = 0$

Conservative Regularization term:

sampler of *unseen actions* $\tilde{a} \in \tilde{A}(s, g)$.

Useful when there is no pathway through goal.

Not differentiable, hence not optimized

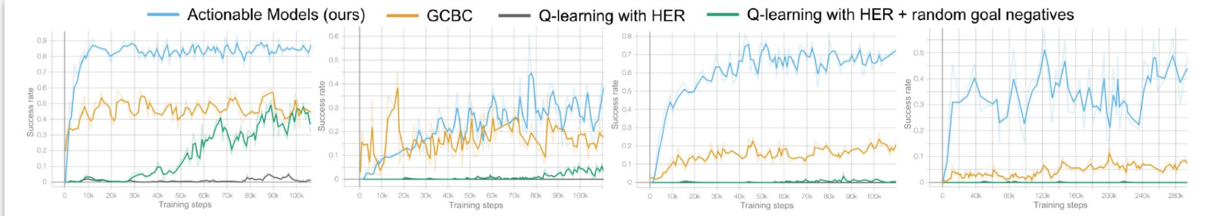
TD -target based on Q value, which would **propagate** from eventual subsequent goal states, allowing discovery of goal chaining points.

ACTIONABLE MODELS - RESULTS

Task	Success rate
Instance grasping	92%
Rearrangement	74%
Container placing	66%

(a) Real world goal reaching

Approach was tested adapting the **QT-Opt** framework [7], dealing with visual goals (*fixed camera images*), in both simulated and real environments, also proving **goal chaining.**



(a) Pick-and-place (b) Stacking (c) Fixture placing (d) Food object grasping

(b) Goal chaining ablation (c) Bottle grasping (d) Banana grasping

Task	No pre-training	With pre-training
Grasp box	0%	27%
Grasp banana	4%	20%
Grasp milk	1%	20%

Table 1. Success rates of learning real world instance grasping tasks from a small amount of data with task-specific rewards

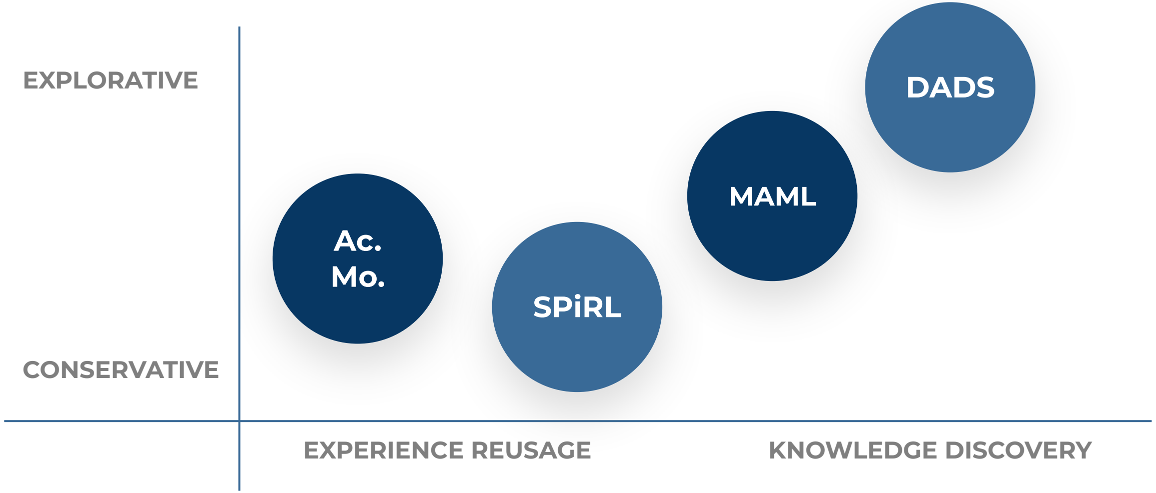
APPROACHES **COMPARISON**

Meta learning idea conceptually is the *fil rouge* among the analyzed algorithms.

SPiRL algorithm is the one closer to **MAML**, as it found an *optimal initial prior parametrization.*

On the other hand **DADS** look for features to whose environment *is more sensitive* itself.

Actionable models propose a way to explicitly *transfer and reuse* such already learned skills.



M.A.M.L.

- Intuitive and effective
- Simple approach
- No abstraction on skill.

D.A.D.S.

- Totally Unsupervised
- Provide a model too
- Entropy/Inf. Th. based
- Long term planning

S.p.iR.L.

- Knowledge transfer
- Entropy/probability based
- Long term planning

Actionable Models

- Knowledge transfer
- Q-Learning
- Goal chaining
- Good as auxiliary loss

CONCLUSIONS

About Meta Learning and Skills based Reinforcement Learning:

- **Strong theoretical foundation** (classic ML & RL, probability, information, game theory, variational calculus).
- Very effective in term of **learning capability improvement.**
- Encourage **data reusage.**
- Allow **saving** computation and money on **long training.**
- Skills enforce **long term planning**
- **Temporal abstraction** allows to solve **complex tasks.**
- **Few shot learning** is not a pre training issue
- Challenging due to **multidisciplinarity** (math, robotics, ...)

Possible future works

- ❏ **Reservoir computing** may benefit a lot from meta learned initial weights configurations
- ❏ **Continual learning** models would benefit from always taking into account configuration to achieve faster specialization.
- ❏ It would be interesting to test the impact of Meta Learning with **unbalanced data** in the downstream task.
- ❏ Draw more connections with **neuroscience** and **biological plausibility** related to neural configuration to fasten learning.

SOURCE PAPERS

1. Chebotar, Yevgen & Hausman, Karol & Lu, Yao & Xiao, Ted & Kalashnikov, Dmitry & Varley, Jake & Irpan, Alex & Eysenbach, Benjamin & Julian, Ryan & Finn, Chelsea & Levine, Sergey. (2021). **Actionable Models: Unsupervised Offline Reinforcement Learning of Robotic Skills.**

2. Sharma, Archit & Gu, Shixiang & Levine, Sergey & Kumar, Vikash & Hausman, Karol. (2019). **Dynamics-Aware Unsupervised Discovery of Skills.**

3. Finn, Chelsea & Abbeel, Pieter & Levine, Sergey. (2017). **Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.**

4. Pertsch, Karl & Lee, Youngwoon & Lim, Joseph. (2020). **Accelerating Reinforcement Learning with Learned Skill Priors.**

ADDITIONAL BIBLIOGRAPHY

5 Hausman, Karol & Springenberg, Jost Tobias & Wang, Ziyu & Heess, Nicolas, & Riedmiller, Martin. (2018). **Learning an embedding space for transferable robot skills.** *6th International Conference on Learning Representations, ICLR 2018*

6 Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., ... Zaremba, W. (2017). **Hindsight experience replay**. *Advances in Neural Information Processing Systems, 2017-December*(Nips)

7 Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., ... Levine, S. (2018). ***QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation***. (CoRL)

8 Gupta, A., Kumar, V., Lynch, C., Levine, S., & Hausman, K. (2019). ***Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning***. (CoRL)

9 Eysenbach, B., Ibarz, J., Gupta, A., & Levine, S. (2019). **Diversity is all you need: Learning skills without a reward function**. *7th International Conference on Learning Representations, ICLR 2019*.