We thank the reviewers for their time and constructive comment. We appreciate that reviewers notice the relevance of the presented problem, given its complexity, we will put extra effort in the presentation to clarify the mathematical analysis and explanations as suggested. We first provide a general description of our work to clarify some of the raised concerns, and then address the particulars. The revised paper will address all the comments.

Our goal is to provide a formulation to describe the generalization of agents trained on a finite number of levels (instances) like the case of CoinRun. Even though the underlying model is a POMDP, for any given instance the state-observation-reward sequences are deterministic for a given sequence of actions. We define this as the instance-specific deterministic trajectory function, and define how it is derived from the underlying POMDP (Eq 8). This level-specific trajectory function mirrors the role of samples in traditional supervised learning, and a model can potentially memorize this sample without generalizing to new samples from the same distribution; yielding policies that are perfectly tuned to the level, we draw a parallel between this phenomenon and the practice of speedrunning, where human players attempt to complete a particular video-game level as fast as possible, the techniques used for this are often level-specific and exploit particularities of the level that are not common throughout the rest of the environment.

Lemmas 1 and 2 show that this instance sampling procedure is internally consistent with the standard POMDP formulation. Lemma 3 notes that an agent trained over a finite level set may exploit the characteristics of these level-specific trajectory functions even further than one trained over an unbounded level set, yielding a policy that may exploit essentially spurious instance-specific correlations and may not generalize to unseen instances. This is aggravated for POMDP's because the states are unobserved and we cannot force the agent to act only based on the posterior state distribution given past observations, rewards and actions, since these also carry information on the particular instance the agent is acting on. This is emphasized by Lemma 4, by using standard supervised learning generalization bounds to the learned value function. We also propose (and evaluate) how to mitigate the lack of generalization using ensembles. All lemmas will be further explained. All suggested references and a glossary with all variables will be added.

R1 *What is the Markov process considered here?* The entire game is considered a POMDP, we discuss a dual representation of the environment as the standard POMDP, or the set of all possible instances sampled according to Eq 8. Generalization is modeled as the performance difference between observed and unobserved levels.

R1 *How is the reward function formalized?* State and reward at time t in the POMDP are modelled as $s_t, r_t \sim T(r_t \mid s_t, a_{t-1})T(s_t \mid s_{t-1}, a_{t-1})$, we will explicitly state this in the revised version.

R1 *Remark 1 is already proven in literature.* Citation is provided in line 109, the remark is added for context. *How many episodes can an instance have?* An instance function can generate as many episodes as there are distinct action sequences, so for a maximum episode length of N and number of actions A there could be as many as A^N distinct episodes in the trajectory function.

R1,R2 *On trajectory generation.* At every node in the trajectory tree, and for every possible action a, the transition samples its future transition from $O(o_t \mid s_t, k)T(r_t \mid s_t, a_t)T(s_t \mid a_{t-1}, s_{t-1})$. The states are latent to the environment and unobserved by the agent. Eq 8 describes how trajectory functions are created, not how the agent perceives them (e.g., if the agent tries the same action sequence twice on the same instance, it will get the same result, like in Eq 9).

R1 *About Eq 9 Expectation across all levels.* Follows from Eq 8 by construction on the unbounded instance set, does not necessarily hold for finite levels. This is better shown in Lemma 2. *Histories.* from the agent's perspective (with access to states), the transition matrix of the next state, observation and reward is the average transition over all instances i such that the transition function along the current action sequence matches the current history, δ is a Kronecker delta, non-stochastic distribution, $\tau^i(a_{0:t} \mid a_t)$ should read $\tau^i(a_{0:t})$, the trajectory of instance i along action sequence $a_{0:t}$.

R2 *Within a set of instances, how are instances selected?* Uniformly at random at the start of the episode. *I suggest minimizing the use of beliefs in the presentation* We will clarify that we are not ascribing any properties to the encoding of the agent, these are merely latent states of our policy. The use of belief was meant to address the capabilities of "optimal" agents.

R2 *On Lemma 3.* Your reading of the lemma is correct, and agree on ensembles only addressing specialization at the policy level, specialization at the representation level is addressed only via ℓ_2 regularization and could be improved.

R2, R3, R4 *Evaluation to other envs* We evaluated the method over three of the ProcGen environments (chaser, plunder and dodgeball), IAPE outperforms the baseline with ℓ_2 and batchnorm by 8% to 12% on total test-time episode reward. Extended comparisons will be added.

R3 *Lemma 2* Yes, the policy is independent of the instance set for the reasons you described.

R1,R2 *Notations* $\mathcal{B}$: set of possible beliefs (e.g., $\mathbb{R}^b$). $\mathbf{H}$: entropy, K: set of observation styles (e.g., background or agent sprites), no influence on state, action, reward dynamics. H_t in Eq 9 is a single episode from the trajectory function, termed "full" history because it includes states, as well as observations. T^I vs. T, T^I: transition matrix over instance set I, T: transition matrix of full POMDP. Expectation notation: Will be cleaned up; $E_A[B \mid C]$ is "expectation of B w.r.t. distribution A given C". Compatible means non-zero probability.