We thank the reviewers (R1, R2, R3) for their detailed feedback. Key concerns are briefly addressed below.

Novelty of learning rule and key contributions (R2): While the final expression for our learning rule is superficially similar to other tri-factor rules (as suggested by the name), ours is derived in a principled manner from a global objective. Importantly, our solution is by construction designed for recurrent networks, unlike the references mentioned by R2, which do not learn recurrent weights (note that we had originally cited two of them). The closest solution to ours at the technical level is the REINFORCE algorithm (Williams, 1992), but the Hebbian term in that formulation is different and produces subtly different experimental predictions:

- Our learning rule: $\Delta w_{ij} \propto \alpha(\mathbf{Dr}, \mathbf{s}) \left(r_i r_j - \langle r_i r_j \rangle_{p(\mathbf{r}|\mathbf{s})} \right) - 2\lambda_W w_{ij}$
- REINFORCE: $\Delta w_{ij} \propto \alpha(\mathbf{Dr}, \mathbf{s}) \sum_{t=0}^{T} f'(h_i(t))(r_i(t) - \bar{r}_i(t)) r_j(t) - 2\lambda_W w_{ij}$,

where $f(\cdot)$ is the activation function, h_i is the pre-activation of neuron i, and $\bar{r}_i$ is the expected average activation for that neuron. One notable difference is that learning is gated by deviations from the mean co-activation of the pre- and post-synaptic neurons, and not by post-synaptic activity alone – which is unique to our solution and potentially testable.

More generally, to our knowledge, no previous work has used tri-factor learning rules to study the effects of task constraints and intrinsic noise on the learned representations. Our derivation of the three-factor rule facilitates this goal, but our main contribution is these novel analyses (as noted by R1).

Disentangling sensory information from task (R3): While one tends to think of sensory representations as being determined exclusively by input statistics, with task constraints only affecting decoding/decision-making circuitry, there is a substantial body of experimental evidence showing that early sensory cortices can change in a task-specific manner in the presence of neuromodulation (Polley & Merzenich, 2006; Froemke & Schreiner 2007). Our results speak to these experimental observations. More generally, the task specificity of the learned code will depend on several factors – the set of tasks the system needs to perform, and various resource constraints (architecture, total number of neurons, etc). We believe that generalizations of our circuit model should allow one to dissect the contribution of each of these elements to the final representation.

Symmetric weights (R1): Having a proper energy significantly simplifies the analysis of how the noise is being reshaped during learning, but is not strictly necessary for our framework. We are working on a generalization to arbitrary weights, which results in qualitatively similar learning rules, with additional temporal integration via eligibility traces. Also note that our current learning rule will naturally converge to symmetric weights, even for arbitrary synapse initialization (cf. Kolen & Pollack, 1994), so symmetry is an emergent property of the network.

Results clarifications (R1): Prior manipulations were made throughout learning, not only at test time. The volume fraction is defined as $VF(\mathbf{s}) = \frac{\det \mathbf{C}_D}{\det \mathbf{C}_{PCA}}$, where $\mathbf{C}_D$ is the covariance matrix of $\mathbf{r}$ projected onto the two output dimensions, and $\mathbf{C}_{PCA}$ is the covariance matrix of $\mathbf{r}$ projected onto the first two principal components of the neural activity for a fixed stimulus $\mathbf{s}$. In Fig. 2a, ellipses give 95% confidence range for network outputs, for a range of test stimuli marked with corresponding black crosses.

"It is not clear in what way *intrinsic noise in the recurrent dynamics ... allows us to derive closed-form probabilistic expressions for the objective function gradients.*": when we say that noise is necessary for our learning rule, we mean that the probabilistic description of neural activity is essential for the derivation of the learning rule (Eq. 3), not that the noise levels must be large. Intrinsic noise is what allows for this probabilistic description. In particular, in the absence of noise, our learning rule produces weight updates of 0 (i.e., no learning).

Intrinsic noise is not always bad (R3): Increasing the magnitude of the noise is generally detrimental for performance, at least in our setup (Fig. 3c). We thought that the network might learn to use its stochasticity to encode uncertainty, but saw no evidence for this for our simple tasks. We'll rephrase the statement about noise being detrimental for encoding, mentioning potential benefits of sampling for Bayesian computation.

The network just plays as a filter of the noise in the framework of MSE (point estimate). You are right that the output of the network is a point estimate and not a probability. But there is a subtle point to be made here: although we have a probabilistic formulation for the encoding model, the computational goals of the circuit are not explicitly Bayesian. The task objectives are defined by marginalizing the prior input statistics and the intrinsic noise.

Clarity and missing details (R1-3): We will correct the typos, add the requested additional information and the suggested references in the updated version. A Github code repository will also be provided to facilitate reproducibility.