A Supplementary material

Contents

A.1 Theoretical results in continuous time

Summary. In this section, we present all the results derived in *continuous* time. More precisely:

- We formally define our model (Def. A.1) and constrained optimization problem (Def. A.2) in *continuous time*.
- We state and prove our algorithm baseline, the *continuous adjoint state method* (ASM) for this constrained optimization problem (Theorem A.1). To ease the comparison of the continuous ASM with *Recurrent Hamiltonian Echo Learning* (RHEL) in continuous time, we state re-parametrized version of the continuous ASM where time is indexed *backwards* (Corollary A.1). Finally, we also state a variant of the continuous ASM when the loss function is only defined at the *final* timestep (Corollary A.2) to ease the comparison between the continuous ASM and *Hamiltonian Echo Backprop* (HEB, [10]).
- We introduce two technical results (Lemma A.1–A.2) which enable us to prove the *time-reversal invariance* property of our model (Lemma A.3). We then show that the direct consequence of this property is the *time-reversibily* of our model upon momentum flipping (Corollary A.3), the key mechanics which fundamentally underpins our algorithm. All these intermediate results allow us to finally introduce RHEL in continuous time and prove its equivalence with the continuous ASM (Theorem A.2).
- Lastly, we connect HEB [10] with the continuous ASM when the loss is defined only at the final time step (Theorem A.4). We highlight some key differences between HEB and RHEL.

A.1.1 Definitions & assumptions

Definition A.1 (Continuous Hamiltonian model)**.** *Given $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$, $T \in \mathbb{R}_+^\star$ and an input sequence $t \to \boldsymbol{u}(t) \in \left(\mathbb{R}^{d_{\boldsymbol{u}}}\right)^{[-T,0]}$, the continuous Hamiltonian model prediction $t \to \boldsymbol{\Phi}(t) \in \left(\mathbb{R}^{d_{\boldsymbol{\Phi}}}\right)^{[-T,0]}$ is, by definition, implicitly given as the solution of the following ODE:*

$$\boldsymbol{\Phi}(-T) = \boldsymbol{x}, \quad \forall t \in [-T, 0] : \partial_t \boldsymbol{\Phi}(t) = \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)], \quad \boldsymbol{J} := \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{I} & \boldsymbol{0} \end{bmatrix}.$$

We assume that:

1. *H is time-reversal invariant:*

$$\forall \boldsymbol{\Phi} \in \mathbb{R}^{d_{\boldsymbol{\Phi}}}, \quad \forall \boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}, \forall \boldsymbol{u} \in \mathbb{R}^{d_{\boldsymbol{u}}} : \quad H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] = H[\boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}], \quad \boldsymbol{\Sigma}_z := \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{I} \end{bmatrix}$$

2. *$\boldsymbol{\Phi} \to H[\boldsymbol{\Phi}, \cdot, \cdot]$ is twice continuously differentiable,*
3. *$\boldsymbol{\theta} \to H[\cdot, \boldsymbol{\theta}, \cdot]$ is differentiable,*
4. *$\nabla^2_{1,2} H$ exists and is continuous with respect to t,*
5. *$\boldsymbol{u} \to H[\cdot, \cdot, \boldsymbol{u}]$ is continuous,*
6. *$\boldsymbol{\Phi} \to \nabla_1 H[\boldsymbol{\Phi}, \cdot, \cdot]$ and $\boldsymbol{\Phi} \to \nabla_{2,1} H[\boldsymbol{\Phi}, \cdot, \cdot]$ are Lipschitz continuous.*

Remark 1. *While some of these assumptions will be used explicitly in our derivations, they are all needed to guarantee the existence of partial derivatives of s as an* implicit function *of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ through Eq. (1) and we refer to [28] for such claims given these assumptions.*

Definition A.2 (Continuous constrained optimization optimization problem)**.** *Given a continuous Hamiltonian model (Def. A.1), we consider the following constrained optimization problem:*

$$\min_{\boldsymbol{\theta}} L := \int_{-T}^{0} dt \ell[t, \boldsymbol{\Phi}(t), \boldsymbol{\theta}] \quad \textit{s.t.} \quad \forall t \in [-T, 0] : \ \partial_t \boldsymbol{\Phi}(t) = \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)],$$

where we assume that:

1. *ℓ is time-reversal invariant:*

$$\forall \boldsymbol{\Phi} \in \mathbb{R}^{d_{\boldsymbol{\Phi}}}, \quad \forall \boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}, \ \forall t \in [-T, 0] : \quad \ell[t, \boldsymbol{\Phi}, \boldsymbol{\theta}] = \ell[t, \boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}, \boldsymbol{\theta}]$$

2. *$t \to \ell[t, \cdot, \cdot]$ is continuous,*
3. *$\boldsymbol{\theta} \to \ell[\cdot, \cdot, \boldsymbol{\theta}]$ is differentiable,*
4. *$t \to \nabla_{\boldsymbol{\theta}} \ell[t, \cdot, \cdot]$ is continuous.*
5. *$\boldsymbol{\Phi} \to \ell[\cdot, \boldsymbol{\Phi}, \cdot]$ is twice differentiable,*

Remark 2. *Note that while we did not assume in the main part of this manuscript that ℓ depended on $\boldsymbol{\theta}$, we assume it in the appendix for the generality of our derivations.*

A.1.2 Proof of the continuous adjoint state method (ASM)

Theorem A.1 (Continuous adjoint state method ([28]))**.** *Given assumptions A.1–A.2, the gradients of L with respect to $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are given by:*

$$d_{\boldsymbol{\theta}} L = \int_{-T}^{0} g_{\boldsymbol{\theta}}(t) dt, \quad d_x L = \boldsymbol{\lambda}(0)$$

with $t \to g_{\boldsymbol{\theta}} \in \left(\mathbb{R}^{d_{\boldsymbol{\theta}}}\right)^{[-T,0]}$ defined as:

$$g_{\boldsymbol{\theta}}(t) := \nabla_{\theta} \ell[t, \boldsymbol{\Phi}(t), \boldsymbol{\theta}] + \nabla^2_{\boldsymbol{\Phi}, \boldsymbol{\theta}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) \quad \forall t \in [-T, 0]$$

*and $\boldsymbol{\lambda}$ solving for the **adjoint** ODE:*

$$\begin{cases} \boldsymbol{\lambda}(0) & = \mathbf{0} \\ \partial_t \boldsymbol{\lambda}(t) & = -\nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) - \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(t), \boldsymbol{\theta}] \end{cases}$$

Proof of Theorem A.1. With slight adaptations, our proof mostly follows that of [32] and also *assume* the existence of the partial derivatives of $\boldsymbol{\Phi} = \boldsymbol{\Phi}(t, \boldsymbol{x}, \boldsymbol{\theta})$ as an *implicit* function of $t, \boldsymbol{x}$ and $\boldsymbol{\theta}$ – we defer to [28] for the proof of this claim.

We start off defining the Lagrangian associated with the constraint optimization problem:

$$\mathcal{L}(\boldsymbol{\Phi}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{u}) := \int_{-T}^{0} dt \left(\ell[t, \boldsymbol{\Phi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}] + \boldsymbol{\lambda}^\top(t) \cdot (\boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}, \boldsymbol{u}(t)] - \partial_t \boldsymbol{\Phi}(t, \boldsymbol{\theta})) \right),$$

where $t \to \boldsymbol{\lambda}(t) \in \left(\mathbb{R}^{d_{\boldsymbol{\Phi}}}\right)^{[-T,0]}$ denotes the Lagrangian multiplier associated to the constraint.

Derivation of $d_{\boldsymbol{\theta}} L$. For readability, we emphasize the dependence of $\boldsymbol{\Phi}$ on t and $\boldsymbol{\theta}$, as we will leverage the existence of its partial derivatives with respect to these variables. Then, given Assumptions A.1, the total derivative of the Lagrangian with respect to $\boldsymbol{\theta}$ exists and reads:

$$\begin{aligned} d_{\boldsymbol{\theta}} \mathcal{L} = & \int_{-T}^{0} dt \left(\partial_{\boldsymbol{\theta}} \boldsymbol{\Phi}(t, \boldsymbol{\theta})^\top \cdot \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}] + \nabla_{\boldsymbol{\theta}} \ell[t, \boldsymbol{\Phi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}] \right) \\ & + \int_{-T}^{0} dt \left[\partial_{\boldsymbol{\theta}} \boldsymbol{\Phi}(t, \boldsymbol{\theta})^\top \cdot \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{J}^\top + \nabla^2_{\boldsymbol{\theta}, \boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t, \boldsymbol{\theta}), \boldsymbol{\theta}, \boldsymbol{u}(t)]^\top \cdot \boldsymbol{J}^\top - \partial^2_{\boldsymbol{\theta}, t} \boldsymbol{\Phi}(t, \boldsymbol{\theta})^\top \right] \cdot \boldsymbol{\lambda}(t). \end{aligned}$$

We can transform the last term of the integrand with $\partial_{\boldsymbol{\theta},t}\boldsymbol{\Phi}(t,\boldsymbol{\theta})$ by applying Schwartz's theorem and and integration by parts as:

$$\begin{aligned}
-\int_{-T}^{0} dt \partial_{\boldsymbol{\theta},t}\boldsymbol{\Phi}(t,\boldsymbol{\theta})^\top \cdot \boldsymbol{\lambda}(t) &= -\int_{-T}^{0} dt \partial_{t,\boldsymbol{\theta}}\boldsymbol{\Phi}(t,\boldsymbol{\theta})^\top \cdot \boldsymbol{\lambda}(t) \\
&= \left[-\partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(t,\boldsymbol{\theta})^\top \cdot \boldsymbol{\lambda}(t)\right]_{-T}^{0} + \int_{-T}^{0} dt \partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(t,\boldsymbol{\theta})^\top \cdot \partial_t\boldsymbol{\lambda}(t) \\
&= -\partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(0,\boldsymbol{\theta})^\top \cdot \boldsymbol{\lambda}(0) + \int_{-T}^{0} dt \partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(t,\boldsymbol{\theta})^\top \cdot \partial_t\boldsymbol{\lambda}(t)
\end{aligned}$$

where the contribution of the first term at $t = -T$ vanishes because:

$$\boldsymbol{\Phi}(-T,\boldsymbol{\theta}) = \boldsymbol{x} \Rightarrow \partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(0,\boldsymbol{\theta}) = 0$$

Plugging this back into $d_{\boldsymbol{\theta}}\mathcal{L}$ yields:

$$\begin{aligned}
d_{\boldsymbol{\theta}}\mathcal{L} = &-\partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(0,\boldsymbol{\theta})^\top \cdot \boldsymbol{\lambda}(0) \\
&+ \int_{-T}^{0} dt \partial_{\boldsymbol{\theta}}\boldsymbol{\Phi}(t,\boldsymbol{x},\boldsymbol{\theta})^\top \cdot \left[\nabla_{\boldsymbol{\Phi}}\ell[t,\boldsymbol{\Phi}(t,\boldsymbol{\theta}),\boldsymbol{\theta}] + \nabla^2_{\boldsymbol{\Phi}}H[\boldsymbol{\Phi}(t,\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{\theta}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) + \partial_t\boldsymbol{\lambda}(t)\right] \\
&+ \int_{-T}^{0} dt \left(\nabla_{\boldsymbol{\theta}}\ell[t,\boldsymbol{\Phi}(t,\boldsymbol{\theta}),\boldsymbol{\theta}] + \nabla^2_{\boldsymbol{\Phi},\boldsymbol{\theta}}H[\boldsymbol{\Phi}(t,\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{\theta}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t)\right)
\end{aligned}$$

Denoting $\boldsymbol{\Phi}_*$ the solution of the (primal) ODE and by defining $\boldsymbol{\lambda}_*$ as the solution of the adjoint ODE:

$$\partial_t\boldsymbol{\lambda}_*(t) = -\nabla^2_{\boldsymbol{\Phi}}H[\boldsymbol{\Phi}_*(t,\boldsymbol{\theta}),\boldsymbol{\theta},\boldsymbol{u}(t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_*(t) - \nabla_{\boldsymbol{\Phi}}\ell[t,\boldsymbol{\Phi}_*(t,\boldsymbol{\theta}),\boldsymbol{\theta}], \quad \boldsymbol{\lambda}_*(0) = 0$$

we have:

$$\begin{aligned}
d_{\boldsymbol{\theta}}L &:= d_{\boldsymbol{\theta}} \int_{-T}^{0} dt \ell[t,\boldsymbol{\Phi}(t,\boldsymbol{\theta}),\boldsymbol{\theta}] \\
&= d_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\Phi}_*,\boldsymbol{\lambda}_*,\boldsymbol{\theta}) \\
&= \int_{-T}^{0} dt \left(\nabla_{\boldsymbol{\theta}}\ell[t,\boldsymbol{\Phi}_*(t,\boldsymbol{\theta}),\boldsymbol{\theta}] + \nabla^2_{\boldsymbol{\Phi},\boldsymbol{\theta}}H[\boldsymbol{\Phi}_*(t,\boldsymbol{x},\boldsymbol{\theta}),\boldsymbol{\theta}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_*(t)\right)
\end{aligned}$$

Derivation of $d_{\boldsymbol{x}}L$. Similarly, we can prove that:

$$\begin{aligned}
d_{\boldsymbol{x}}\mathcal{L} = &\,\partial_{\boldsymbol{x}}\boldsymbol{\Phi}(0,\boldsymbol{x})^\top \cdot [\nabla\ell[\boldsymbol{\Phi}(0,\boldsymbol{x})] - \boldsymbol{\lambda}(0)] \\
&+ \int_{-T}^{0} dt \partial_{\boldsymbol{x}}\boldsymbol{\Phi}(t,\boldsymbol{x})^\top \cdot \left[\nabla_{\boldsymbol{\Phi}}\ell[t,\Phi(t,\boldsymbol{\theta}),\boldsymbol{\theta}] + \nabla^2_{\boldsymbol{\Phi}}H[\boldsymbol{\Phi}(t,\boldsymbol{x}),\boldsymbol{\theta},\boldsymbol{u}(t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) + \partial_t\boldsymbol{\lambda}(t)\right] \\
&+ \boldsymbol{\lambda}(0)
\end{aligned}$$

Using the same $\boldsymbol{\Phi}_*$ and $\boldsymbol{\lambda}_*$, we get $d_{\boldsymbol{x}}L = \boldsymbol{\lambda}(0)$. □

Reparametrization of the continuous adjoint method. To ease the comparison of the continuous adjoint state method with our algorithm, we slightly reparametrize the variables introduced in Theorem A.1.

Corollary A.1. *Under the same assumptions as Theorem A.1, the gradients of L with respect to $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are given by:*

$$d_{\boldsymbol{\theta}}L = \int_0^T g_{\boldsymbol{\theta}}(t)dt, \quad d_xL = \boldsymbol{\lambda}(T)$$

with $t \to g_{\boldsymbol{\theta}} \in \left(\mathbb{R}^{d_{\boldsymbol{\theta}}}\right)^{[0,T]}$ defined as:

$$g_{\boldsymbol{\theta}}(t) := \nabla_\theta\ell[-t,\boldsymbol{\Phi}(-t),\boldsymbol{\theta}] + \nabla^2_{\boldsymbol{\Phi},\boldsymbol{\theta}}H[\boldsymbol{\Phi}(-t),\boldsymbol{\theta},\boldsymbol{u}(-t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) \quad \forall t \in [0,T]$$

*and $\boldsymbol{\lambda}$ solving for the **adjoint** ODE:*

$$\begin{cases} \boldsymbol{\lambda}(0) & = \mathbf{0} \\ \partial_t\boldsymbol{\lambda}(t) & = \nabla^2_{\boldsymbol{\Phi}}H[\boldsymbol{\Phi}(-t),\boldsymbol{\theta},\boldsymbol{u}(-t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) + \nabla_{\boldsymbol{\Phi}}\ell[-t,\boldsymbol{\Phi}(-t),\boldsymbol{\theta}] \end{cases}$$

Proof of Corollary A.1. Immediately stems from Theorem A.1. □

Edge case: loss at the final timestep only. The backward ODE can be differently parametrized when a loss is defined only at the last time step. We need this formulation for later convenience.

Corollary A.2. *Under the same assumptions as Theorem A.1, and assuming:*

$$\ell[t,\cdot,\cdot] = 0 \quad \forall t \in [0,T), \quad \ell[T,\cdot,\cdot] := \ell_T,$$

the gradients of ℓ_T with respect to $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are given by:

$$d_{\boldsymbol{\theta}}\ell_T[\boldsymbol{\Phi}(0),\boldsymbol{\theta}] = \nabla_{\boldsymbol{\theta}}\ell_T[\boldsymbol{\Phi}(0),\boldsymbol{\theta}] + \int_0^T g_{\boldsymbol{\theta}}(t)dt, \quad d_x L = \boldsymbol{\lambda}(T)$$

with $t \to g_{\boldsymbol{\theta}} \in \left(\mathbb{R}^{d_{\boldsymbol{\theta}}}\right)^{[0,T]}$ defined as:

$$g_{\boldsymbol{\theta}}(t) := \nabla^2_{\boldsymbol{\Phi},\boldsymbol{\theta}} H[\boldsymbol{\Phi}(-t),\boldsymbol{\theta},\boldsymbol{u}(-t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) \quad \forall t \in [0,T]$$

*and $\boldsymbol{\lambda}$ solving for the **adjoint** ODE:*

$$\begin{cases} \boldsymbol{\lambda}(0) & = \nabla_{\boldsymbol{\Phi}}\ell_T[\boldsymbol{\Phi}(0),\boldsymbol{\theta}] \\ \partial_t\boldsymbol{\lambda}(t) & = \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t),\boldsymbol{\theta},\boldsymbol{u}(-t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) \end{cases}$$

Proof of Corollary A.2. Starting from the Lagrangian:

$$\mathcal{L}(\boldsymbol{\Phi},\boldsymbol{\lambda},\boldsymbol{\theta},\boldsymbol{u}) := \ell_T[\boldsymbol{\Phi}(0,\boldsymbol{\theta}),\boldsymbol{\theta}] + \int_{-T}^0 dt \left(\boldsymbol{\lambda}^\top(t) \cdot (\boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t,\boldsymbol{\theta}),\boldsymbol{\theta},\boldsymbol{u}(t)] - \partial_t\boldsymbol{\Phi}(t,\boldsymbol{\theta}))\right),$$

the proof reads in the exact same fashion as that of Theorem A.1. □

A.1.3 Proof of Theorem 3.1

Technical Lemmas. We first introduce two technical Lemmas which will be needed for the derivation of our main result.

Lemma A.1 (Block-wise Pauli matrices and associated properties)**.** *Defining* $\boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_y, \boldsymbol{\Sigma}_z \in \mathbb{R}^{d_{\boldsymbol{\Phi}} \times d_{\boldsymbol{\Phi}}}$ *as:*

$$\boldsymbol{\Sigma}_x := \begin{bmatrix} \mathbf{0} & \boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2} \\ \boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\Sigma}_y := \begin{bmatrix} \mathbf{0} & -i\boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2} \\ i\boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2} & \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\Sigma}_z := \begin{bmatrix} \boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2} & \mathbf{0} \\ \mathbf{0} & -\boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2} \end{bmatrix}$$

where i *denotes the imaginary unit, the following equalities hold:*

1. $\boldsymbol{J} = i\boldsymbol{\Sigma}_y$
2. $\boldsymbol{\Sigma}_x^2 = \boldsymbol{\Sigma}_y^2 = \boldsymbol{\Sigma}_z^2 = \boldsymbol{I}_{d_{\boldsymbol{\Phi}}/2}$,
3. $\boldsymbol{\Sigma}_x \cdot \boldsymbol{\Sigma}_y = i\boldsymbol{\Sigma}_z$, $\boldsymbol{\Sigma}_y \cdot \boldsymbol{\Sigma}_z = i\boldsymbol{\Sigma}_x$, $\boldsymbol{\Sigma}_z \cdot \boldsymbol{\Sigma}_x = i\boldsymbol{\Sigma}_y$,
4. $\boldsymbol{\Sigma}_i \cdot \boldsymbol{\Sigma}_j = -\boldsymbol{\Sigma}_j \cdot \boldsymbol{\Sigma}_i$ *for any* $i \neq j \in \{x, y, z\}$.

Proof of Lemma A.1. Because of the block-wise structure of $\boldsymbol{\Sigma}_x, \boldsymbol{\Sigma}_y, \boldsymbol{\Sigma}_z$, these equalities can be easily checked. □

Lemma A.2. *Under the assumptions of Def. A.1, the following equalities hold for all* $\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}$*:*

$$\begin{aligned} \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] &= \boldsymbol{\Sigma}_z \cdot \nabla_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}], \\ \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] &= \boldsymbol{\Sigma}_z \cdot \nabla^2_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}] \cdot \boldsymbol{\Sigma}_z, \\ \nabla_{\boldsymbol{\Phi}, \boldsymbol{\theta}} H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] &= \nabla_{\boldsymbol{\Phi}^\star, \boldsymbol{\theta}} H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}] \cdot \boldsymbol{\Sigma}_z \end{aligned}$$

Proof of Lemma A.2. The above equalities can be simply obtained by differentiating through the time-reversal invariance hypothesis – which is possible because of the differentiability of H with respect to $\boldsymbol{\Phi}$ and $\boldsymbol{\theta}$ – and using the chain rule. Namely, given some $\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}$:

$$\begin{aligned} \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] &= \nabla_{\boldsymbol{\Phi}} \left(H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}] \right) \\ &= \left(\partial_{\boldsymbol{\Phi}} \boldsymbol{\Phi}^\star \right)^\top \cdot \nabla_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}] \\ &= \boldsymbol{\Sigma}_z^\top \cdot \nabla_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}] = \boldsymbol{\Sigma}_z \cdot \nabla_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star, \boldsymbol{\theta}, \boldsymbol{u}], \end{aligned}$$

since $\boldsymbol{\Phi}^\star := \boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}$. The other equalities are derived in the same way. □

Time-reversal invariance. We highlight here how the assumption $H[\boldsymbol{\Phi}, \cdot, \cdot] = H[\boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}, \cdot, \cdot]$ given inside Def. A.1 entails time-reversal invariance of the dynamics – up to time-reversal the input sequence.

Lemma A.3 (Time-reversal invariance of the dynamics)**.** *Under the assumptions given in Def. A.1, if* $\boldsymbol{\Phi}$ *the solution of the ODE:*

$$\partial_t \boldsymbol{\Phi}(t) = \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)],$$

the function $\widetilde{\boldsymbol{\Phi}}^\star : t \to \boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}(-t) = [\boldsymbol{\phi}^\top(-t), -\boldsymbol{\pi}^\top(-t)]^\top$ *is solution of the same ODE with the time-reversed input sequence* $t \to \widetilde{\boldsymbol{u}}(t) := \boldsymbol{u}(-t)$.

Proof. Let $t \in [-T, 0]$. We have:

$$\begin{aligned} \partial_t \widetilde{\boldsymbol{\Phi}}^\star(t) &= -\boldsymbol{\Sigma}_z \cdot \partial_t \boldsymbol{\Phi}(-t) && \text{(Def. of } \widetilde{\boldsymbol{\Phi}}^\star \text{ and chain rule)} \\ &= -\boldsymbol{\Sigma}_z \cdot \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] && \text{(by assumption)} \\ &= +\boldsymbol{J} \cdot \boldsymbol{\Sigma}_z \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] && \text{(Lemma A.1)} \\ &= \boldsymbol{J} \cdot \boldsymbol{\Sigma}_z^2 \cdot \nabla_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star(-t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] && \text{(Lemma A.2)} \\ &= \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}^\star} H[\widetilde{\boldsymbol{\Phi}}^\star(t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] && \text{(Lemma A.1)} \end{aligned}$$

□

Time reversibility. A direct consequence of the time-reversal of the dynamics under consideration (Lemma A.3) is *time reversibility*: upon flipping the momentum of $\boldsymbol{\Phi}$ at time $t = 0$ ($\boldsymbol{\pi}(0) \leftarrow -\boldsymbol{\pi}(0)$) and presenting the input sequence in reversed order, the system evolves backward to its initial state.

Corollary A.3 (Reversibility of the dynamics)**.** *Under the same assumptions as Lemma A.3, we define $\boldsymbol{\Phi}^e$ as the solution of the ODE:*

$$\boldsymbol{\Phi}^e(0) = \boldsymbol{\Phi}^\star(0), \quad \partial_t \boldsymbol{\Phi}^e(t) = \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}^e} H[\boldsymbol{\Phi}^e(t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] \quad \forall t \in [0, T].$$

Then:

$$\forall t \in [0, T]: \quad \boldsymbol{\Phi}^e(t) = \widetilde{\boldsymbol{\Phi}}^\star(t)$$

Proof. $\boldsymbol{\Phi}^e$ and $\widetilde{\boldsymbol{\Phi}}^\star$ satisfy: i) the same initial conditions ($\widetilde{\boldsymbol{\Phi}}^\star(0) = \boldsymbol{\Phi}^\star(0) = \boldsymbol{\Phi}^e(0)$), ii) the same ODE (Lemma A.3), therefore by unicity of the solution of the ODE, they are equal at all time over the domain of definition of $\boldsymbol{\Phi}$. □

Main result. We are now ready to state and demonstrate our main result in continuous time.

Theorem A.2 (Equivalence between RHEL and the continuous ASM)**.** *Under the assumptions of Def. A.1 and Def. A.2, let $\boldsymbol{\Phi}$ be the solution of the ODE for $t \in [-T, 0]$:*

$$\boldsymbol{\Phi}(-T) = \boldsymbol{x}, \quad \partial_t \boldsymbol{\Phi}(t) = \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)].$$

Given $\boldsymbol{\Phi}$, let $\boldsymbol{\Phi}^e$ be defined as the solution of the other ODE for $t \in [0, T]$:

$$\boldsymbol{\Phi}^e(0) = \boldsymbol{\Phi}^\star(0), \quad \partial_t \boldsymbol{\Phi}^e(t, \epsilon) = \boldsymbol{J} \nabla_{\boldsymbol{\Phi}^e} H[\boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)] - \epsilon \boldsymbol{J} \nabla_{\boldsymbol{\Phi}^e} \ell[-t, \boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}].$$

Defining:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t, \epsilon) := \nabla_{\boldsymbol{\theta}} \ell[-t, \boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}] + \frac{1}{2\epsilon} \left(\nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)] - \nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t, -\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)] \right),$$

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t, \epsilon) := \frac{1}{2\epsilon} \boldsymbol{\Sigma}_x \cdot \left(\boldsymbol{\Phi}^e(t, \epsilon) - \boldsymbol{\Phi}^e(t, -\epsilon) \right),$$

we have:

$$\forall t \in [0, T], \quad \boldsymbol{\lambda}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t, \epsilon), \quad g_{\boldsymbol{\theta}}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t, \epsilon)$$

where $\boldsymbol{\lambda}$ and $g_{\boldsymbol{\theta}}$ are defined in Corollary A.1.

Proof. Defining $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t) := \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t, \epsilon)$ and $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t) := \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t, \epsilon)$, note that:

$$\begin{aligned}
\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t) &= \boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t, \epsilon)|_{\epsilon=0} \\
\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t) &= \nabla_{\boldsymbol{\theta}} \ell[-t, \boldsymbol{\Phi}^e(t, 0), \boldsymbol{\theta}] + \partial_\epsilon \left(\nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)] \right)|_{\epsilon=0}
\end{aligned}$$

Derivation of $\boldsymbol{\lambda}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t, \epsilon)$. Given $t \in [0, T]$, differentiating the ODE satisfied by $\boldsymbol{\Phi}^e$ with respect to ϵ at $\epsilon = 0$ yields:

$$\begin{aligned}
\partial_t (\partial_\epsilon \boldsymbol{\Phi}^e(t, \epsilon)|_{\epsilon=0}) &= \partial_\epsilon (\partial_t \boldsymbol{\Phi}^e(t, \epsilon))|_{\epsilon=0} && \text{(Schwartz Theorem)} \\
&= \partial_\epsilon (\boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}^e} H[\boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}, \boldsymbol{u}(t)] - \epsilon \boldsymbol{J} \nabla_{\boldsymbol{\Phi}^e} \ell[t, \boldsymbol{\Phi}^e(t, \epsilon), \boldsymbol{\theta}])|_{\epsilon=0} \\
&= \boldsymbol{J} \cdot \nabla^2_{\boldsymbol{\Phi}^e} H[\boldsymbol{\Phi}^e(t, 0)] \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t, \epsilon)|_{\epsilon=0} - \boldsymbol{J} \nabla_{\boldsymbol{\Phi}^e} \ell[t, \boldsymbol{\Phi}^e(t, 0), \boldsymbol{\theta}].
\end{aligned}$$

By Lemma A.3:

$$\boldsymbol{\Phi}^e(t, 0) = \widetilde{\boldsymbol{\Phi}}^\star(t) \quad \forall t \in [0, T],$$

therefore:

$$\begin{aligned}
\partial_t (\partial_\epsilon \boldsymbol{\Phi}^e(t, \epsilon)|_{\epsilon=0}) &= \boldsymbol{J} \cdot \nabla^2_{\boldsymbol{\Phi}^\star} H[\boldsymbol{\Phi}^\star(-t)] \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t, \epsilon)|_{\epsilon=0} - \boldsymbol{J} \nabla_{\boldsymbol{\Phi}^\star} \ell[t, \boldsymbol{\Phi}^\star(-t), \boldsymbol{\theta}] \\
&= \boldsymbol{J} \cdot \boldsymbol{\Sigma}_z \cdot \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t, 0)] \cdot \boldsymbol{\Sigma}_z \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t, \epsilon)|_{\epsilon=0} - \boldsymbol{J} \cdot \boldsymbol{\Sigma}_z \cdot \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(-t), \boldsymbol{\theta}] && \text{(Lemma A.2)}
\end{aligned}$$

Additionally, note that we have by Lemma A.1:

$$\boldsymbol{J} \cdot \boldsymbol{\Sigma}_z = -\boldsymbol{\Sigma}_x, \quad \boldsymbol{\Sigma}_z = \boldsymbol{J} \cdot \boldsymbol{\Sigma}_x,$$

so that:

$$\partial_t(\partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0}) = -\boldsymbol{\Sigma}_x \cdot \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t,0)] \cdot (\boldsymbol{J} \cdot \boldsymbol{\Sigma}_x) \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0} + \boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(-t), \boldsymbol{\theta}] \quad (19)$$

Left multiplying Eq. (19) on both sides by $\boldsymbol{\Sigma}_x$ yields:

$$\begin{aligned}
\partial_t \left(\boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0}\right) &= -\boldsymbol{\Sigma}_x^2 \cdot \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t,0)] \cdot \boldsymbol{J} \cdot (\boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0}) + \boldsymbol{\Sigma}_x^2 \cdot \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(-t), \boldsymbol{\theta}] \\
&= -\nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t,0)] \cdot \boldsymbol{J} \cdot (\boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0}) + \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(-t), \boldsymbol{\theta}] && \text{(Lemma A.1)} \\
&= \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t,0)] \cdot \boldsymbol{J}^\top \cdot (\boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0}) + \nabla_{\boldsymbol{\Phi}} \ell[t, \boldsymbol{\Phi}(-t), \boldsymbol{\theta}] && (\boldsymbol{J}^\top = -\boldsymbol{J})
\end{aligned}$$

Finally, note that because $\boldsymbol{\Phi}^e(0) = \boldsymbol{\Phi}^\star$ does not depend on ϵ, we have that:

$$\partial_t \left(\boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}^e(0,\epsilon)|_{\epsilon=0}\right) = 0,$$

so that all in all, $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}$ satisfies:

$$\begin{cases} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(0) & = \mathbf{0} \\ \partial_t \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t) & = \nabla^2_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(-t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] \cdot \boldsymbol{J}^\top \cdot \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t) + \nabla_{\boldsymbol{\Phi}} \ell[-t, \boldsymbol{\Phi}(-t), \boldsymbol{\theta}] \end{cases}$$

Therefore $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}$ and $\boldsymbol{\lambda}$ (as defined in Corollary A.1) satisfy the same initial conditions and the same ODE, therefore they are equal at all times.

Derivation of $g_{\boldsymbol{\theta}}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t,\epsilon)$. Note that by Lemma A.3 and time-reversal invariance of ℓ:

$$\begin{aligned}
\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t) &= \nabla_{\boldsymbol{\theta}} \ell[-t, \boldsymbol{\Phi}^\star(-t,0), \boldsymbol{\theta}] + \partial_\epsilon \left(\nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)]\right)|_{\epsilon=0} \\
&= \nabla_{\boldsymbol{\theta}} \ell[-t, \boldsymbol{\Phi}(-t,0), \boldsymbol{\theta}] + \partial_\epsilon \left(\nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)]\right)|_{\epsilon=0}
\end{aligned}$$

As the first term of $\Delta^{\mathrm{RHEL}}(t)$ and $g_{\boldsymbol{\theta}}(t)$ coincide, the remainder of the derivation focuses on the second term of $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t)$. Given $t \in [0,T]$, we have:

$$\begin{aligned}
\nabla^2_{\boldsymbol{\Phi},\boldsymbol{\theta}} H[\boldsymbol{\Phi}(-t), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) &= \nabla^2_{\boldsymbol{\Phi}^e,\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,0), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{\Sigma}_z \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}(t) && \text{(Lemma A.2)} \\
&= -\nabla^2_{\boldsymbol{\Phi}^e,\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,0), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{\Sigma}_z \cdot i\boldsymbol{\Sigma}_y \cdot \boldsymbol{\lambda}(t) && (\boldsymbol{J} = i\boldsymbol{\Sigma}_y) \\
&= +\nabla^2_{\boldsymbol{\Phi}^e,\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,0), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot i\boldsymbol{\Sigma}_y \cdot \boldsymbol{\Sigma}_z \cdot \boldsymbol{\lambda}(t) && \text{(Lemma A.1)} \\
&= -\nabla^2_{\boldsymbol{\Phi}^e,\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,0), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{\Sigma}_x \cdot \boldsymbol{\lambda}(t) && \text{(Lemma A.1)} \\
&= -\nabla^2_{\boldsymbol{\Phi}^e,\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,0), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \boldsymbol{\Sigma}_x^2 \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0} && (\boldsymbol{\lambda} = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}) \\
&= -\nabla^2_{\boldsymbol{\Phi}^e,\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,0), \boldsymbol{\theta}, \boldsymbol{u}(t)] \cdot \partial_\epsilon \boldsymbol{\Phi}^e(t,\epsilon)|_{\epsilon=0} && \text{(Lemma A.1)} \\
&= -\partial_\epsilon \left(\nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(t)]\right)|_{\epsilon=0},
\end{aligned}$$

which finishes to prove $g_{\boldsymbol{\theta}}(t) = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t)$ for $t \in [0,T]$. □

A.1.4 Connection to Hamiltonian Echo Backpropagation (HEB)

Remark 3. *The above setup and implementation of RHEL is not exactly that of Hamiltonian Echo Backprop (HEB, [10]). In particular:*

- *the loss function in HEB is only defined at the* final *time step,*
- *the interaction with ℓ does not happen* simultaneously *with H,*
- *finally, we would like to recover the HEB formula giving the gradient estimate of the loss with respect to the initial state of the neurons.*

In the following corollary, we make slight algorithmic adjustments to match the seminal HEB implementation as much as possible ***while preserving the generality of the sequence modelling setting****,* i.e. *dependence of H with $\boldsymbol{\theta}$ and $\boldsymbol{u}$. Note that in the seminal HEB work, H does not depend on a static set of parameters $\boldsymbol{\theta}$ nor on an input sequence. $\boldsymbol{u}$.*

Corollary A.4. *Under the assumptions of Def. A.1 and Def. A.2, and assuming additionally:*

$$\ell[t,\cdot,\cdot] = 0 \quad \forall t \in [0,T), \quad \ell[T,\cdot,\cdot] := \ell_T,$$

let $\boldsymbol{\Phi}$ be the solution of the ODE, for $t \in [-T, 0]$:

$$\boldsymbol{\Phi}(-T) = \boldsymbol{x}, \quad \partial_t \boldsymbol{\Phi}(t) = \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} H[\boldsymbol{\Phi}(t), \boldsymbol{\theta}, \boldsymbol{u}(t)],$$

and the solution of another ODE, for $t \in [0, \epsilon]$:

$$\partial_t \boldsymbol{\Phi}(t) = \boldsymbol{J} \nabla_{\boldsymbol{\Phi}} \ell_T[\boldsymbol{\Phi}(t), \boldsymbol{\theta}].$$

Let $\boldsymbol{\Phi}^e$ be the solution of the following ODE, for $t \in [\epsilon, T]$:

$$\boldsymbol{\Phi}^e(0,\epsilon) = \boldsymbol{\Phi}(\epsilon)^\star, \quad \partial_t \boldsymbol{\Phi}^e(t,\epsilon) = \boldsymbol{J} \nabla_{\boldsymbol{\Phi}^e} H[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)],$$

Defining:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t,\epsilon) := \frac{1}{2\epsilon} \left(\nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)] - \nabla_{\boldsymbol{\theta}} H[\boldsymbol{\Phi}^e(t,-\epsilon), \boldsymbol{\theta}, \boldsymbol{u}(-t)] \right),$$
$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t,\epsilon) := \frac{1}{2\epsilon} \boldsymbol{\Sigma}_x \cdot \left(\boldsymbol{\Phi}^e(t,\epsilon) - \boldsymbol{\Phi}^e(t,-\epsilon) \right),$$

we have:

$$\forall t \in [0,T], \quad \boldsymbol{\lambda}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t,\epsilon), \quad g_{\boldsymbol{\theta}}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t,\epsilon)$$

where $\boldsymbol{\lambda}$ and $g_{\boldsymbol{\theta}}$ are defined in Corollary A.2. In particular:

$$-i\epsilon d^w_{\boldsymbol{x}^\star} \ell_T[\boldsymbol{\Phi}(0), \boldsymbol{\theta}] = \boldsymbol{\Phi}^e(T,\epsilon)^\star - \boldsymbol{\Phi}(-T) + \mathcal{O}(\epsilon^2)$$

where $d^w_{\boldsymbol{x}^\star} \equiv d_{\boldsymbol{\Phi}}$ denotes the total Wirtinger derivative with respect to $\boldsymbol{x}^\star$ and i the imaginary unit.

Proof of Corollary A.4. The derivation is almost exactly similar to that of Theorem 3.1 with two key differences:

- the version of the continuous ASM against which this version of RHEL is compared is different (Corollary A.2),
- the interaction of $\boldsymbol{\Phi}$ with ℓ and H do not happen simultaneously but on disjoint intervals.

We will simply show that the interaction with ℓ and conjugation $\boldsymbol{\Phi} \to \boldsymbol{\Phi}^\star$ yields the correct initial conditions and defer to the proof of Theorem 3.1 for the remainder. We will also use the same notations and denote $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t) := \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t,\epsilon)$, $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t) := \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(t,\epsilon)$.

Derivation of $\boldsymbol{\lambda}(t) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(t,\epsilon)$. Integrating the ODE satisfied by $\boldsymbol{\Phi}$ between 0 and T yields:

$$\begin{aligned}\boldsymbol{\Phi}(\epsilon) &= \boldsymbol{\Phi}(0) + \int_0^\epsilon dt \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}(t), \boldsymbol{\theta}] \\ &= \boldsymbol{\Phi}(0) + \epsilon \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}(0), \boldsymbol{\theta}] + \mathcal{O}(\epsilon^2)\end{aligned}$$

Therefore, the initial state of $\boldsymbol{\Phi}^e$ can be written as:

$$\begin{aligned}\boldsymbol{\Phi}^e(\epsilon) &= \boldsymbol{\Phi}^\star(0) + \epsilon \boldsymbol{\Sigma}_z \cdot \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}(0), \boldsymbol{\theta}] + \mathcal{O}(\epsilon^2) \\ &= \boldsymbol{\Phi}^\star(0) + \epsilon \boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}(0), \boldsymbol{\theta}] + \mathcal{O}(\epsilon^2) \qquad \text{(Lemma A.1).}\end{aligned}$$

By differentiating the last equality with respect to ϵ at $\epsilon = 0$, we obtain:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(0) = \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}(0), \boldsymbol{\theta}].$$

Proceeding exactly as in the proof of Theorem A.2, we obtain that:

$$\begin{cases} \Delta_{\boldsymbol{\Phi}}^{\mathrm{RHEL}}(0) & = \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}(0), \boldsymbol{\theta}], \\ \partial_t \Delta_{\boldsymbol{\Phi}}^{\mathrm{RHEL}}(t) & = \nabla_{\boldsymbol{\Phi}}^2 H[\boldsymbol{\Phi}(-t), \boldsymbol{\theta}, \boldsymbol{u}(-t)] \cdot \boldsymbol{J}^\top \cdot \Delta_{\boldsymbol{\Phi}}^{\mathrm{RHEL}}(t) \end{cases}$$

Therefore $\Delta_{\boldsymbol{\Phi}}^{\mathrm{RHEL}}$ and $\boldsymbol{\lambda}$ (as defined in Corollary A.2) satisfy the same initial conditions and the same ODE, therefore they are equal at all times.

Derivation of $g_{\boldsymbol{\theta}}(t) = \lim_{\epsilon \to 0} \Delta_{\boldsymbol{\theta}}^{\mathrm{RHEL}}(t, \epsilon)$. See proof of Theorem A.2.

Connection to HEB formula. In particular, we have:

$$\begin{aligned} d_{\boldsymbol{x}} \ell_T[\boldsymbol{\Phi}(0), \boldsymbol{\theta}] = \boldsymbol{\lambda}(T) &= \boldsymbol{\Sigma}_x \cdot \partial_\epsilon \left(\boldsymbol{\Phi}(T, \epsilon) \right|_{\epsilon=0} \\ &= \frac{1}{\epsilon} \boldsymbol{\Sigma}_x \left(\boldsymbol{\Phi}^e(T, \epsilon) - \boldsymbol{\Phi}^e(T, 0) \right) + \mathcal{O}(\epsilon) \\ &= \frac{1}{\epsilon} \boldsymbol{\Sigma}_x \left(\boldsymbol{\Phi}^e(T, \epsilon) - \boldsymbol{\Phi}^\star(-T) \right) + \mathcal{O}(\epsilon) && \text{(Lemma A.3)} \\ &= -\frac{i}{\epsilon} \boldsymbol{\Sigma}_y \cdot \boldsymbol{\Sigma}_z \cdot \left(\boldsymbol{\Phi}^e(T, \epsilon) - \boldsymbol{\Phi}^\star(-T) \right) + \mathcal{O}(\epsilon) && \text{(Lemma A.1)} \\ &= -\frac{i}{\epsilon} \boldsymbol{\Sigma}_y \cdot \left(\boldsymbol{\Phi}^e(T, \epsilon)^\star - \boldsymbol{\Phi}(-T) \right) + \mathcal{O}(\epsilon) \end{aligned}$$

Left multiplying on both sides by $i\epsilon \boldsymbol{\Sigma}_y$ and noticing that $i d_{\boldsymbol{\Phi}^\star}^w \equiv -i \boldsymbol{\Sigma}_y \cdot d_{\boldsymbol{\Phi}}$, we finally obtain:

$$-i\epsilon d_{\boldsymbol{x}^\star}^w \ell_T[\boldsymbol{\Phi}(0), \boldsymbol{\theta}] = \boldsymbol{\Phi}^e(T, \epsilon)^\star - \boldsymbol{\Phi}(-T) + \mathcal{O}(\epsilon^2)$$

□

A.2 Theoretical results in discrete time

Summary. In this section, we introduce all the results derived in *discrete* time. More precisely:

- We first formally define *Hamiltonian Recurrent Units* (HRUs, Definition A.3). HRUs can be regarded as the discrete-time counterpart of the continuous model introduced in the previous section (Definition A.1), namely as an *explicit* and *symplectic* integrator of the continuous Hamiltonian model which preserves the time-reversal invariance and time-reversibility properties in discrete time. We also introduce the constrained optimization problem naturally associated with HRUs (Definition A.4), which is the discrete time counterpart of the constrained continuous optimization problem introduced in the previous section (Definition A.2).
- We then formally define *Hamiltonian State Space Models* (HSSMs) as stacks of HRUs (Definition A.5) and the *multilevel* constrained optimization problem which is naturally associated to these models (Definition A.6).
- We state and prove our algorithmic baseline, *Backpropagation Through Time* (BPTT), through the lens of the *Lagrangian* formalism to establish a clear connection with the continuous ASM. We first introduce and derive BPTT in its general form (Theorem A.3) and then apply it more specifically to a HRU as defined in Definition A.3 (Corollary A.5).
- As we did in continuous time, we introduce a series of technical Lemmas needed to extend RHEL in discrete time. We first demonstrate the time-reversibility of HRUs on a single time step (Lemma A.4), which then enables us to extend the time reversibility property derived in continuous time (Corollary A.3) to *discrete* time (Corollary A.6). After introducing one last technical result (Lemma A.5), we then state and prove RHEL in discrete time when applied to HRUs (Corollary A.7). As the algorithm prescribed by Corollary A.7 includes solving an *implicit equation*, we finally introduce a slight practical (*i.e.* fully explicit) variant of RHEL in discrete time (Corollary A.8).
- Lastly, we show how to estimate gradients end-to-end in HSSMs by using *RHEL–chaining* (Theorem A.4). We also highlight that in practice, when using feedforward transformations across HRUs, the algorithm prescribed by Theorem A.4 implicitly requires to chain RHEL through HRUs and *automatic differentiation* through these feedforward transformations (Remark 8). This remark fundamentally underpins the actual algorithmic implementation of RHEL which was used throughout our experiments.

A.2.1 Definitions & assumptions

Definition A.3 (Hamiltonian Recurrent Unit)**.** *Given $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$, $K \in \mathbb{N}^\star$ and an input sequence $(\boldsymbol{u}_k)_{k\in[-K,0]} \in \left(\mathbb{R}^{d_{\boldsymbol{u}}}\right)^K$, the Hamiltonian Recurrent Unit (HRU) prediction is given by:*

$$\boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k] \quad \forall k = -K \cdots -1,$$

with $H := T + V$ and:

$$\mathcal{M}_{H,\delta} := \mathcal{M}_{T,\delta/2} \circ \mathcal{M}_{V,\delta} \circ \mathcal{M}_{T,\delta/2}, \quad \mathcal{M}_{T,\delta} := \boldsymbol{\Phi} + \delta \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} T, \quad \mathcal{M}_{V,\delta} := \boldsymbol{\Phi} + \delta \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}} V$$

We assume that:

1. *H is separable,* i.e. *V and T only depend on $\boldsymbol{\phi}$ and $\boldsymbol{\pi}$ respectively:*

$$V[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] = V[\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{u}], \quad T[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] = T[\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{u}]$$

2. *T and V are time-reversal invariant:*

$$\forall \boldsymbol{\Phi} \in \mathbb{R}^{d_{\boldsymbol{\Phi}}},\ \forall \boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}},\ \forall \boldsymbol{u} \in \mathbb{R}^{d_{\boldsymbol{u}}} : \quad \begin{cases} T[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] = T[\boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}], \\ V[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] = V[\boldsymbol{\Sigma}_z \cdot \boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] \end{cases}$$

3. *T and V are twice differentiable with respect to $\boldsymbol{\Phi}$, $\boldsymbol{\theta}$ and $\boldsymbol{u}$.*

Remark 4. *Note that $\mathcal{M}_{H,\delta}$ is simply a Leapfrog integrator associated with H. We justify each of our design choices below:*

- ***3 steps-parametrization.*** *We write the Leapfrog integrator in a three-steps fashion to yield a* reversible *integrator.*
- ***Separability of the Hamiltonian.*** *In the case where ϕ and ϕ can be separated out in the Hamiltonian function, the Leapfrog integrator becomes explicit [33].*
- T ***and*** V ***as functions of*** $\mathbf{\Phi}$***.*** *Although T and V only depend on $\boldsymbol{\pi}$ and ϕ respectively, we choose to define them as functions of $\mathbf{\Phi}$ so that the proof of RHEL in the continuous case seamlessly translates to the discrete case.*

Definition A.4 (Constrained optimization optimization problem in discrete time)**.** *Given a continuous Hamiltonian model (Def. A.1), we consider the following constrained optimization problem:*

$$\min_{\boldsymbol{\theta}} L := \sum_{k=-K+1}^{0} \ell[\mathbf{\Phi}_k, k] \quad \textit{s.t.} \quad \mathbf{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\mathbf{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k] \quad \forall k = -K \cdots -1$$

where we assume that:

1. *ℓ is time-reversal invariant:*

$$\forall \mathbf{\Phi} \in \mathbb{R}^{d_{\mathbf{\Phi}}},\ \forall \boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}},\ \forall k = -K, \cdots, 0: \quad \ell_k[\mathbf{\Phi}, \boldsymbol{\theta}] = \ell_k[\mathbf{\Sigma}_z \cdot \mathbf{\Phi}, \boldsymbol{\theta}]$$

2. *ℓ is twice differentiable with respect to $\mathbf{\Phi}$ and $\boldsymbol{\theta}$.*

Definition A.5 (Hamiltonian State Space Models)**.** *Given $(\boldsymbol{\theta}^{(1)}, \cdots, \boldsymbol{\theta}^{(N)}) \in \left(\mathbb{R}^{d_{\boldsymbol{\theta}}}\right)^N$, $K \in \mathbb{N}^\star$ and an input sequence $(\boldsymbol{u}_k)_{k\in[-K,0]} \in \left(\mathbb{R}^{d_{\boldsymbol{u}}}\right)^K$, a Hamiltonian State Space Model (HSSM) is defined as the composition of HRUs defined in Def. A.3 as:*

$$\mathbf{\Phi}^{(0)} := \overline{\boldsymbol{u}}, \quad \forall \ell \in [\![0, N-1]\!], \quad \forall k \in [\![-K, 0]\!]: \quad \mathbf{\Phi}_{k+1}^{(\ell+1)} = \mathcal{M}_{H^{(\ell)},\delta}^{(\ell)}[\mathbf{\Phi}_k^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \mathbf{\Phi}_k^{(\ell)}],$$

or in a vectorized fashion as:

$$\mathbf{\Phi}^{(0)} := \overline{\boldsymbol{u}}, \quad \forall \ell \in [\![0, N-1]\!]: \ \overline{\mathbf{\Phi}}^{(\ell+1)} = \boldsymbol{\mathcal{M}}_{H^{(\ell)},\delta}^{(\ell)}[\overline{\mathbf{\Phi}}^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \overline{\mathbf{\Phi}}^{(\ell)}],$$

Definition A.6 (Multilevel optimization problem in discrete time)**.** *Given a HSSM (Def. A.5), we consider the following constrained optimization problem:*

$$\min_{\boldsymbol{\theta}} L := \sum_{k=-K+1}^{0} \ell[\mathbf{\Phi}_k, k] \quad \textit{s.t.} \quad \mathbf{\Phi}^{(0)} := \overline{\boldsymbol{u}},$$
$$\forall \ell \in [\![0, N-1]\!]: \ \overline{\mathbf{\Phi}}^{(\ell+1)} = \boldsymbol{\mathcal{M}}_{H^{(\ell)},\delta}^{(\ell)}\left[\overline{\mathbf{\Phi}}^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \overline{\mathbf{\Phi}}^{(\ell)}\right]$$

where we assume that ℓ satisfies the same assumptions as in Def. A.4.

A.2.2 Backpropagation Through Time (BPTT)

General form. We first state and prove Backpropagation Through Time (BPTT) for any integrator $\mathcal{M}_{H,\delta}$.

Theorem A.3 (Backpropagation Through Time (BPTT))**.** *Given assumptions in Def. A.3–A.4, the gradients of the loss with respect to the parameters $\boldsymbol{\theta}$ and the inputs $\boldsymbol{u}_{-k}$ are given by:*

$$d_{\boldsymbol{\theta}} L = \sum_{k=0}^{K-1} g_{\boldsymbol{\theta}}(k), \quad d_{\boldsymbol{u}_{-(k+1)}} L = g_{\boldsymbol{u}}(k) \quad \forall k \in [\![0, K-1]\!],$$

with:

$$\begin{cases} g_{\boldsymbol{\theta}}(k) & = \nabla_2 \ell[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, -k] + \partial_2 \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]^\top \cdot \boldsymbol{\lambda}_k \\ g_{\boldsymbol{u}}(k) & = \partial_3 \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]^\top \cdot \boldsymbol{\lambda}_k, \end{cases}$$

and where $(\boldsymbol{\lambda}_k)$ satisfy the following recursion relationship:

$$\begin{cases} \boldsymbol{\lambda}_0 = \nabla_1 \ell[\boldsymbol{\Phi}_0, 0], \\ \boldsymbol{\lambda}_{k+1} = \partial_1 \mathcal{M}_{H,\delta}(\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)})^\top \cdot \boldsymbol{\lambda}_k + \nabla_1 \ell[\boldsymbol{\Phi}_{-(k+1)}, -(k+1)] \quad \forall k = 0, \cdots, K-1 \end{cases}$$

Proof of Theorem A.3. BPTT can classically be derived through the application of the "chain rule" backward through the inference computational graph defined in Def. A.3. Another useful viewpoint though, which directly connects BPTT as the discrete counterpart of the continuous ASM and will be useful later in the appendix, is to derive it through *the method of Lagrangian multipliers*. Namely, the Lagrangian associated to the constrained optimization problem in Def. A.4 reads as:

$$\mathcal{L}(\boldsymbol{\Phi}, \boldsymbol{\lambda}, \boldsymbol{\theta}, \boldsymbol{u}) = \sum_{k=0}^{K-1} \ell[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, -k] + \boldsymbol{\lambda}_k^\top \cdot \left(\mathcal{M}_{H,\delta}\left[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}\right] - \boldsymbol{\Phi}_{-k} \right)$$

Extremizing $\mathcal{L}$ with respect to $\boldsymbol{\Phi}$ and $\boldsymbol{\lambda}$ yield $\boldsymbol{\Phi}_{k,*}$ and $\boldsymbol{\lambda}_{k,*}$:

$$\begin{aligned} \forall k = 0, \cdots, K-1: \quad & \partial_{\boldsymbol{\lambda}_k} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u}) = \mathcal{M}_{H,\delta}\left[\boldsymbol{\Phi}_{-(k+1),*}, \boldsymbol{\theta}, \boldsymbol{u}_{-k}\right] - \boldsymbol{\Phi}_{-k,*} = 0, \\ & \partial_{\boldsymbol{\Phi}_0} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u}) = \nabla_1 \ell[\boldsymbol{\Phi}_{0,*}, \boldsymbol{\theta}, 0] - \boldsymbol{\lambda}_{0,*} = 0 \\ \forall k = 1, \cdots, K-1: \quad & \partial_{\boldsymbol{\Phi}_{-k}} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u}) = \nabla_1 \ell[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, -k] + \partial_1 \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, \boldsymbol{u}_{-k}]^\top \cdot \boldsymbol{\lambda}_{k-1} - \boldsymbol{\lambda}_k = 0, \end{aligned}$$

Finally, the total derivative of L with respect to $\boldsymbol{\theta}$ reads as:

$$\begin{aligned} d_{\boldsymbol{\theta}} L &= d_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u}) \\ &= \partial_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u}) + \partial_{\boldsymbol{\theta}} \boldsymbol{\Phi}_*^\top \cdot \underbrace{\partial_{\boldsymbol{\Phi}} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u})}_{=0} + \partial_{\boldsymbol{\theta}} \boldsymbol{\lambda}_*^\top \cdot \underbrace{\partial_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\Phi}_*, \boldsymbol{\lambda}_*, \boldsymbol{\theta}, \boldsymbol{u})}_{=0} \\ &= \sum_{k=0}^{K-1} \nabla_2 \ell[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, -k] + \partial_2 \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]^\top \cdot \boldsymbol{\lambda}_k \end{aligned}$$

The total derivative of L with respect to $\boldsymbol{u}_{-k}$ is derived in the exact same fashion. □

Remark 5. *Note that using the vectorized notations introduced in subsection 3.3, the Lagrangian of the constrained optimization problem defined in Def. A.4 re-writes:*

$$\mathcal{L} = \mathbf{1}^\top \cdot \ell[\widetilde{\boldsymbol{\Phi}}, \boldsymbol{\theta}] + \mathrm{Tr}\left[\left(\boldsymbol{\mathcal{M}}_{\boldsymbol{H},\boldsymbol{\delta}}[\widetilde{\boldsymbol{\Phi}}, \boldsymbol{\theta}, \widetilde{\boldsymbol{u}}] - \widetilde{\boldsymbol{\Phi}} \right) \cdot \overline{\boldsymbol{\lambda}}^\top \right]$$

with Tr *denoting the trace matrix operator and:*

$$\mathbf{1} := \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{K \times 1}, \quad \ell[\widetilde{\boldsymbol{\Phi}}, \boldsymbol{\theta}] := \begin{bmatrix} \ell[\boldsymbol{\Phi}_0, \boldsymbol{\theta}, 0] \\ \vdots \\ \ell[\boldsymbol{\Phi}_{-(K-1)}, \boldsymbol{\theta}, -(K-1)] \end{bmatrix} \in \mathbb{R}^{K \times 1},$$

$$\boldsymbol{\mathcal{M}}_{H,\delta}[\widetilde{\boldsymbol{\Phi}}, \boldsymbol{\theta}, \widetilde{\boldsymbol{u}}] := \begin{bmatrix} \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_0, \boldsymbol{\theta}, \boldsymbol{u}] \\ \vdots \\ \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_{-(K-1)}, \boldsymbol{\theta}, \boldsymbol{u}] \end{bmatrix} \in \mathbb{R}^{K \times d_{\boldsymbol{\Phi}}}$$

Detailed BPTT. For the needs of our derivation of RHEL in discrete time, we now introduce a finer-grained version of BPTT given model assumptions given in Def. A.3.

Corollary A.5 (Detailed BPTT)**.** *Given assumptions in Def. A.3–A.4, the gradients of the loss with respect to the parameters $\boldsymbol{\theta}$ and the inputs $\boldsymbol{u}_{-k}$ are given by:*

$$d_{\boldsymbol{\theta}} L = \sum_{k=0}^{K-1} g_{\boldsymbol{\theta}}(k), \quad d_{\boldsymbol{u}_{-k}} L = g_{\boldsymbol{u}}(k),$$

with:

$$g_{\boldsymbol{\theta}}(k) := \nabla_2 \ell[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, -k] + \frac{\delta}{2} \nabla^2_{1,2} T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_k$$
$$+ \delta \nabla^2_{1,2} V[\boldsymbol{\Phi}_{-(k+2/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+1/3} + \frac{\delta}{2} \nabla^2_{1,2} T[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+2/3},$$

$$g_{\boldsymbol{u}}(k) := \frac{\delta}{2} \nabla^2_{1,3} T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_k$$
$$+ \delta \nabla^2_{1,3} V[\boldsymbol{\Phi}_{-(k+2/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+1/3} + \frac{\delta}{2} \nabla^2_{1,3} T[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+2/3},$$

and where $(\boldsymbol{\lambda}_k)$ satisfy the following recursion relationship, with $\boldsymbol{\lambda}_0 = \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}_0]$ and $\forall k \in [0, K-1]$:

$$\begin{cases} \boldsymbol{\lambda}_{k+1/3} & = \boldsymbol{\lambda}_k + \frac{\delta}{2} \nabla^2_1 T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_k \\ \boldsymbol{\lambda}_{k+2/3} & = \boldsymbol{\lambda}_{k+1/3} + \delta \nabla^2_1 V[\boldsymbol{\Phi}_{-(k+2/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+1/3} \\ \boldsymbol{\lambda}_{k+1} & = \boldsymbol{\lambda}_{k+2/3} + \frac{\delta}{2} \nabla^2_1 T[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+2/3} + \nabla_1 \ell[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}] \end{cases}$$

Proof of Corollary A.5. Direct application of Theorem A.3 with the inference computational graph details defined inside Def. A.3. □

A.2.3 Proof of Theorem 3.2

Time reversibility in discrete time. We first derive the discrete counterpart of Lemma A.3 as a technical pre-requisite for the extension of RHEL to the discrete-time setting.

Lemma A.4 (Reversibility of $\mathcal{M}_{H,\delta}$)**.** *Under the assumptions of Def. A.3–A.4:*

$$\forall \boldsymbol{\Phi}_k \in \mathbb{R}^{d_{\boldsymbol{\Phi}}},\ \forall \boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}},\ \forall \boldsymbol{u} \in \mathbb{R}^{d_{\boldsymbol{u}}}: \quad \boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}] \Rightarrow \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}^\star_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}] = \boldsymbol{\Phi}^\star_k$$

Proof of Lemma A.4. Let $\boldsymbol{\Phi}_k, \boldsymbol{\Phi}_{k+1} \in \mathbb{R}^{d_{\boldsymbol{\Phi}}}$ be such that:

$$\boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}]$$

which rewrites, given Def. A.3:

$$\mathcal{M}_{H,\delta}: \begin{cases} \boldsymbol{\phi}_{k+1/3} &= \boldsymbol{\phi}_k + \frac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_k, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}_{k+1/3} &= \boldsymbol{\pi}_k \\ \boldsymbol{\phi}_{k+2/3} &= \boldsymbol{\phi}_{k+1/3} \\ \boldsymbol{\pi}_{k+2/3} &= \boldsymbol{\pi}_{k+1/3} - \delta\nabla_{\boldsymbol{\phi}} V[\boldsymbol{\phi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\phi}_{k+1} &= \boldsymbol{\phi}_{k+2/3} + \frac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}_{k+1} &= \boldsymbol{\pi}_{k+2/3} \end{cases} \tag{20}$$

It becomes apparent from Eq. (20) that $\mathcal{M}_{H,\delta}$ is invertible with respect to its first argument and that inverting $\mathcal{M}_{H,\delta}$ amounts to change δ to $-\delta$:

$$\mathcal{M}^{-1}_{H,\delta}: \begin{cases} \boldsymbol{\phi}_{k+2/3} &= \boldsymbol{\phi}_{k+1} - \frac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}_{k+2/3} &= \boldsymbol{\pi}_{k+1} \\ \boldsymbol{\phi}_{k+1/3} &= \boldsymbol{\phi}_{k+2/3} \\ \boldsymbol{\pi}_{k+1/3} &= \boldsymbol{\pi}_{k+2/3} + \delta\nabla_{\boldsymbol{\phi}} V[\boldsymbol{\phi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\phi}_{k} &= \boldsymbol{\phi}_{k+1/3} - \frac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}_{k} &= \boldsymbol{\pi}_{k+1/3} \end{cases} \tag{21}$$

and therefore:

$$\boldsymbol{\Phi}_k = \mathcal{M}^{-1}_{H,\delta}[\boldsymbol{\Phi}_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}] = \mathcal{M}_{H,-\delta}[\boldsymbol{\Phi}_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}],$$

Denoting $\boldsymbol{\pi}^\star := -\boldsymbol{\pi}$, note that by time-reversal invariance hypothesis in Def. A.3 and Lemma A.2, we have that $\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{u}] = -\nabla_{\boldsymbol{\pi}^\star} T[\boldsymbol{\pi}^\star, \boldsymbol{\theta}, \boldsymbol{u}]$. Therefore, Eq. (21) rewrites as:

$$\begin{cases} \boldsymbol{\phi}_{k+2/3} &= \boldsymbol{\phi}_{k+1} + \frac{\delta}{2}\nabla_{\boldsymbol{\pi}^\star} T[\boldsymbol{\pi}^\star_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}^\star_{k+2/3} &= \boldsymbol{\pi}^\star_{k+1} \\ \boldsymbol{\phi}_{k+1/3} &= \boldsymbol{\phi}_{k+2/3} \\ \boldsymbol{\pi}^\star_{k+1/3} &= \boldsymbol{\pi}^\star_{k+2/3} - \delta\nabla_{\boldsymbol{\phi}} V[\boldsymbol{\phi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\phi}_{k} &= \boldsymbol{\phi}_{k+1/3} + \frac{\delta}{2}\nabla_{\boldsymbol{\pi}^\star} T[\boldsymbol{\pi}^\star_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}^\star_{k} &= \boldsymbol{\pi}^\star_{k+1/3} \end{cases}, \tag{22}$$

where equations bearing on $\boldsymbol{\pi}$ have been multiplied on both sides by -1. Finally note that Eq. (22) simply rewrites as:

$$\mathcal{M}_{H,\delta}[\boldsymbol{\Phi}^\star_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}] = \boldsymbol{\Phi}^\star_k$$

□

Corollary A.6. *Under the assumptions of Def. A.3, if $\boldsymbol{\Phi}$ satisfies the following recursive equations:*

$$\boldsymbol{\Phi}_{-K} = \boldsymbol{x}, \quad \forall k = -K, \cdots, -1: \quad \boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k]$$

and $\boldsymbol{\Phi}^e$ is subsequently defined as:

$$\boldsymbol{\Phi}^e_0 = \boldsymbol{\Phi}^\star_0, \quad \forall t = 0, \cdots, K-1: \quad \boldsymbol{\Phi}^e_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}^e_k, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]$$

Then:

$$\forall t = 0, \cdots, K-1: \quad \boldsymbol{\Phi}^e_k = \boldsymbol{\Phi}^\star_{-k}$$

Proof of Corollary A.6. This result is immediately obtained by iterating Lemma A.4 over the whole trajectory. □

A technical pre-requisite. Finally, we need one last technical Lemma to handle subtleties pertaining to Jacobian evaluation which only occur in discrete time.

Lemma A.5. *Under the assumptions of Def. A.3, if we have, for some $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}$ and $\boldsymbol{u} \in \mathbb{R}^{d_{\boldsymbol{u}}}$:*

$$\boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}],$$

then:

$$\begin{cases} T[\boldsymbol{\Phi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}] = T[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}], \\ V[\boldsymbol{\Phi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}] = V[\boldsymbol{\Phi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ T[\boldsymbol{\Phi}_{k+1}, \boldsymbol{\theta}, \boldsymbol{u}] = T[\boldsymbol{\Phi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}] \end{cases}$$

Proof. This can be seen by simply writing $\mathcal{M}_{H,\delta}$ explicitly for $\boldsymbol{\phi}$ and $\boldsymbol{\pi}$:

$$\mathcal{M}_{H,\delta} : \begin{cases} \boldsymbol{\phi}_{k+1/3} &= \boldsymbol{\phi}_k + \frac{\delta}{2} \nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_k, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}_{k+1/3} &= \boldsymbol{\pi}_k \\ \boldsymbol{\phi}_{k+2/3} &= \boldsymbol{\phi}_{k+1/3} \\ \boldsymbol{\pi}_{k+2/3} &= \boldsymbol{\pi}_{k+1/3} - \delta \nabla_{\boldsymbol{\phi}} V[\boldsymbol{\phi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\phi}_{k+1} &= \boldsymbol{\phi}_{k+2/3} + \frac{\delta}{2} \nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}] \\ \boldsymbol{\pi}_{k+1} &= \boldsymbol{\pi}_{k+2/3} \end{cases}$$

□

Discrete-time RHEL. We are now ready to state the main result of this section.

Corollary A.7. *Under the assumptions of Def. A.3–A.4, let $(\boldsymbol{\Phi}_k)_k$ satisfy the recursive equation:*

$$\boldsymbol{\Phi}_{-K} = \boldsymbol{x}, \quad \boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k] \quad \forall k = -K, \cdots, -1,$$

and let $\boldsymbol{\Phi}^e$ satisfy:

$$\begin{cases} \boldsymbol{\Phi}^e_0(\epsilon) = \boldsymbol{\Phi}^\star_0 + \epsilon \boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}_0, \boldsymbol{\theta}, 0], \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\Phi}^e_{k+1/3}(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}^e_{k+2/3}(\epsilon) = \mathcal{M}_{V,\delta}[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}^e_{k+1}(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] - \epsilon \boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}^e} \ell[\boldsymbol{\Phi}^e_{k+1}, \boldsymbol{\theta}, -(k+1)], \end{cases}$$

Then defining:

$$\begin{aligned} H^{1/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] &:= \frac{1}{2} \left(H[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + H[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \right), \\ \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k, \epsilon) &:= \nabla_2 \ell[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, -k] \\ &\quad - \frac{\delta}{2\epsilon} \left(\nabla_2 H^{1/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] - \nabla_2 H^{1/2}[\boldsymbol{\Phi}^e_k(-\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \right), \\ \Delta^{\mathrm{RHEL}}_{\boldsymbol{u}}(k, \epsilon) &:= -\frac{\delta}{2\epsilon} \left(\nabla_3 H^{1/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] - \nabla_3 H^{1/2}[\boldsymbol{\Phi}^e_k(-\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \right), \\ \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k, \epsilon) &:= \frac{1}{2\epsilon} \boldsymbol{\Sigma}_x \cdot (\boldsymbol{\Phi}^e_k(\epsilon) - \boldsymbol{\Phi}^e_k(-\epsilon)), \end{aligned}$$

we have:

$$\forall k = 0, \cdots, K-1 : \quad \boldsymbol{\lambda}_k = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k, \epsilon), \quad g_{\boldsymbol{\theta}}(k) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k, \epsilon), \quad g_{\boldsymbol{u}}(k) = \lim_{\epsilon \to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{u}}(k, \epsilon),$$

where $(\boldsymbol{\lambda}_k)_{k \in [\![0,K]\!]}$, $(g_{\boldsymbol{\theta}}(k))_{k \in [\![0,K-1]\!]}$ and $(g_{\boldsymbol{u}}(k))_{k \in [\![0,K-1]\!]}$ are defined inside Corollary (A.5).

Proof of Corollary A.7. Let $k \in [0, K-1]$. We define:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k) := \lim_{\epsilon\to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k,\epsilon) = \boldsymbol{\Sigma}_x \cdot \partial_\epsilon \boldsymbol{\Phi}(k,\epsilon)|_{\epsilon=0}$$
$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k) := \lim_{\epsilon\to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k,\epsilon) = \nabla_2 \ell[\boldsymbol{\Phi}^e_k(0), \boldsymbol{\theta}, -k] - \delta\partial_\epsilon \left(\nabla_2 H^{1/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right)|_{\epsilon=0}$$
$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{u}}(k) := \lim_{\epsilon\to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{u}}(k,\epsilon) = -\delta\partial_\epsilon \left(\nabla_3 H^{1/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right)|_{\epsilon=0}$$

Derivation of $\boldsymbol{\lambda}_k = \lim_{\epsilon\to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k,\epsilon)$. We proceed exactly as in Theorem A.2 with some subtle adaptations which we highlight. Differentiating the dynamics of $\boldsymbol{\Phi}^e$ between k and $k+1/3$ and proceeding as in the proof of Theorem A.2 using Lemma A.1, Lemma A.2 and Corollary A.6 (as the discrete counterpart of Lemma A.3 which was used for Theorem A.2), we obtain:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+1/3) = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k) + \frac{\delta}{2}\nabla^2_{\boldsymbol{\Phi}} T[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k)$$

However note that this does not correctly match the dynamics satisfied by $\boldsymbol{\lambda}$ inside Corollary A.5 between k and $k+1/3$: the Hessian $\nabla^2_{\boldsymbol{\Phi}} T$ should instead be evaluated at $\boldsymbol{\Phi}_{-(k+1/3)}$. Fortunately, using Lemma A.5:

$$\nabla^2_{\boldsymbol{\Phi}} T[\boldsymbol{\Phi}_{-k}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] = \nabla^2_{\boldsymbol{\Phi}} T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]$$

Therefore we get:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+1/3) = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k) + \frac{\delta}{2}\nabla^2_{\boldsymbol{\Phi}} T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k)$$

Proceeding the same way on $\boldsymbol{\Phi}^e_{k+2/3}$ and $\boldsymbol{\Phi}^e_{k+1}$, we get altogether:

$$\begin{cases} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+1/3) = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k) + \frac{\delta}{2}\nabla^2_{\boldsymbol{\Phi}} T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k) \\ \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+2/3) = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+1/3) + \delta\nabla^2_{\boldsymbol{\Phi}} V[\boldsymbol{\Phi}_{-(k+2/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+1/3) \\ \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+1) = \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+2/3) + \frac{\delta}{2}\nabla^2_{\boldsymbol{\Phi}} T[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k+2/3) \\ \qquad +\nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, -(k+1)] \end{cases}$$

$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}$ satisfying the same equations as $(\boldsymbol{\lambda}_k)_k$ given by BPTT, together with same initial conditions:

$$\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(0) = \nabla_{\boldsymbol{\Phi}} \ell[\boldsymbol{\Phi}_0, \boldsymbol{\theta}, 0]$$

yields the desired equality.

Derivation of $g_{\boldsymbol{\theta}}(k) = \lim_{\epsilon\to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k,\epsilon)$. Proceeding in the same way as in the derivation of Theorem A.2, starting from the expression of $g_{\boldsymbol{\theta}}(k)$ derived in Corollary A.5, using $\Delta^{\mathrm{RHEL}}_{\boldsymbol{\Phi}}(k) = \boldsymbol{\lambda}_k$ and paying attention to evaluating Jacobian at the right places using Lemma A.5, we obtain $\forall k = 0, \cdots, K-1$:

$$\begin{aligned} g_{\boldsymbol{\theta}}(k) &= \nabla_2 \ell[\boldsymbol{\Phi}^e_k(0), \boldsymbol{\theta}, -k] \\ &\quad - \frac{\delta}{2}\,\partial_\epsilon \left\{\nabla_{\boldsymbol{\theta}} T[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + 2\nabla_{\boldsymbol{\theta}} V[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + \nabla_{\boldsymbol{\theta}} T[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right\}\Big|_{\epsilon=0}. \end{aligned}$$

There again, using Lemma A.5:

$$\nabla_{\boldsymbol{\theta}} T[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] = \nabla_{\boldsymbol{\theta}} T[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}], \quad \nabla_{\boldsymbol{\theta}} V[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] = \nabla_{\boldsymbol{\theta}} V[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}],$$

therefore:

$$\begin{aligned} g_{\boldsymbol{\theta}}(k) &= \nabla_2 \ell[\boldsymbol{\Phi}^e_k(0), \boldsymbol{\theta}, -k] - \frac{\delta}{2}\,\partial_\epsilon \left\{\nabla_{\boldsymbol{\theta}} T[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + \nabla_{\boldsymbol{\theta}} V[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right. \\ &\quad \left. +\nabla_{\boldsymbol{\theta}} V[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + \nabla_{\boldsymbol{\theta}} T[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right\}\Big|_{\epsilon=0}. \\ &= \nabla_2 \ell[\boldsymbol{\Phi}^e_k(0), \boldsymbol{\theta}, -k] - \frac{\delta}{2}\,\partial_\epsilon \left(H[\boldsymbol{\Phi}^e_{k+1/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + H[\boldsymbol{\Phi}^e_{k+2/3}(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right)\Big|_{\epsilon=0} \\ &= \nabla_2 \ell[\boldsymbol{\Phi}^e_k(0), \boldsymbol{\theta}, -k] - \delta\partial_\epsilon \left(H^{1/2}[\boldsymbol{\Phi}^e_k(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}]\right)\Big|_{\epsilon=0} = \lim_{\epsilon\to 0} \Delta^{\mathrm{RHEL}}_{\boldsymbol{\theta}}(k,\epsilon) \end{aligned}$$

Derivation of $g_{\boldsymbol{u}}(k) = \lim_{\epsilon\to 0} \Delta_{\boldsymbol{u}}^{\mathrm{RHEL}}(k,\epsilon)$. Strictly identical to the above paragraph. □

Remark 6. *Note that the echo dynamics prescribed by Corollary A.7 are **implicit**:*

$$\left\{\begin{array}{l} \boldsymbol{\Phi}_0^e(\epsilon) = \boldsymbol{\Phi}_0^\star + \epsilon\boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}}\ell[\boldsymbol{\Phi}_0, \boldsymbol{\theta}, 0], \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\Phi}_{k+1/3}^e(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}_k^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}_{k+2/3}^e(\epsilon) = \mathcal{M}_{V,\delta}[\boldsymbol{\Phi}_{k+1/3}^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}_{k+1}^e(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}_{k+2/3}^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] - \epsilon\boldsymbol{J} \cdot \nabla_{\boldsymbol{\Phi}^e}\ell[\boldsymbol{\Phi}_{k+1}^e, \boldsymbol{\theta}, -(k+1)], \end{array}\right.$$

where $\boldsymbol{\Phi}_{k+1}^e$ appears, as highlighted in red, on both sides of the last step. A straightforward way to make the echo dynamics explicit while preserving the theoretical guarantees of Corollary A.7 is to linearize *the nudging signal, namely using instead the following set of equations:*

$$\left\{\begin{array}{l} \boldsymbol{\Phi}_0^e(\epsilon) = \boldsymbol{\Phi}_0^\star + \epsilon\boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}}\ell[\boldsymbol{\Phi}_0, \boldsymbol{\theta}, 0], \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\Phi}_{k+1/3}^e(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}_k^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}_{k+2/3}^e(\epsilon) = \mathcal{M}_{V,\delta}[\boldsymbol{\Phi}_{k+1/3}^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}_{k+1}^e(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}_{k+2/3}^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + \epsilon\boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}^e}\ell[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, -(k+1)], \end{array}\right.$$

Note that the $\epsilon\boldsymbol{J}$ of the original implicit equation becomes $-\epsilon\boldsymbol{\Sigma}_x$ in its linearized counterpart.

This remark leads us to a slight variant of Corollary A.7.

Corollary A.8. *Under the assumptions of Def. A.3–A.4, let $(\boldsymbol{\Phi}_k)_k$ satisfy the recursive equation:*

$$\boldsymbol{\Phi}_{-K} = \boldsymbol{x}, \quad \boldsymbol{\Phi}_{k+1} = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k] \quad \forall k = -K, \cdots, -1,$$

and let $\boldsymbol{\Phi}^e$ satisfy:

$$\left\{\begin{array}{l} \boldsymbol{\Phi}_0^e(\epsilon) = \boldsymbol{\Phi}_0^\star + \epsilon\boldsymbol{\Sigma}_x \cdot \boldsymbol{y}_0, \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\Phi}_{k+1/3}^e(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}_k^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}_{k+2/3}^e(\epsilon) = \mathcal{M}_{V,\delta}[\boldsymbol{\Phi}_{k+1/3}^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \\ \quad \boldsymbol{\Phi}_{k+1}^e(\epsilon) = \mathcal{M}_{T,\delta/2}[\boldsymbol{\Phi}_{k+2/3}^e(\epsilon), \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] + \epsilon\boldsymbol{\Sigma}_x \cdot \boldsymbol{y}_{-(k+1)}, \end{array}\right.$$

where $\boldsymbol{y} \in \mathbb{R}^{K\times d_{\boldsymbol{\Phi}}}$ does not depend on $\boldsymbol{\Phi}$. Then the same conclusions as Corollary A.7 hold, with $(\boldsymbol{\lambda}_k)_{k\in[\![0,K-1]\!]}$ satisfying $\boldsymbol{\lambda}_0 = \boldsymbol{y}_0$ and $\forall k \in [0, K-1]$:

$$\left\{\begin{array}{lll} \boldsymbol{\lambda}_{k+1/3} & = \boldsymbol{\lambda}_k + \frac{\delta}{2}\nabla_1^2 T[\boldsymbol{\Phi}_{-(k+1/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_k \\ \boldsymbol{\lambda}_{k+2/3} & = \boldsymbol{\lambda}_{k+1/3} + \delta\nabla_1^2 V[\boldsymbol{\Phi}_{-(k+2/3)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+1/3} \\ \boldsymbol{\lambda}_{k+1} & = \boldsymbol{\lambda}_{k+2/3} + \frac{\delta}{2}\nabla_1^2 T[\boldsymbol{\Phi}_{-(k+1)}, \boldsymbol{\theta}, \boldsymbol{u}_{-(k+1)}] \cdot \boldsymbol{J}^\top \cdot \boldsymbol{\lambda}_{k+2/3} + \boldsymbol{y}_{-(k+1)} \end{array}\right.$$

Proof Corollary A.8. Identical to the proof of Corollary A.7. □

A.2.4 Proof of Theorem 3.3

Theorem A.4. *Assuming a HSSM (Def. A.5) and the optimization problem depicted in Def. A.6, we have:*

$$\forall \ell = 0, \cdots, N-1: \quad d_{\boldsymbol{\theta}^{(\ell)}} L = \lim_{\epsilon \to 0} \Delta \boldsymbol{\theta}^{(\ell)}(\epsilon),$$

*where $\Delta\boldsymbol{\theta}^{(\ell)}(\epsilon)$ can be recursively computed backwards, starting from the **top-most block** as:*

$$\begin{aligned}
\overline{\boldsymbol{\Phi}}^{e,(N)}(\epsilon) &= \boldsymbol{\mathcal{M}}_{H^{(N-1)},\delta}[\overline{\boldsymbol{\Phi}}^{e,(N)}(\epsilon), \boldsymbol{\theta}^{(N-1)}, \widetilde{\boldsymbol{\Phi}}^{(N-1)}] + \epsilon \boldsymbol{\Sigma}_x \cdot \nabla_{\boldsymbol{\Phi}^{(N)}} L \\
\Delta\boldsymbol{\theta}^{(N-1)}(\epsilon) &= \sum_{k=0}^{K-1} \nabla_2 \ell[\boldsymbol{\Phi}_k^e, \boldsymbol{\theta}^{(L-1)}, -k] \\
&\quad - \frac{\delta}{2\epsilon} \sum_{k=0}^{K-1} \left(\boldsymbol{\nabla_2} H^{1/2}[\boldsymbol{\Phi}_k^{e,(L)}(\epsilon), \boldsymbol{\theta}^{(N-1)}, \boldsymbol{\Phi}_{-k}^{(N-1)}] - \boldsymbol{\nabla_2} H^{1/2}[\boldsymbol{\Phi}_k^{e,(N)}(-\epsilon), \boldsymbol{\theta}^{(N-1)}, \boldsymbol{\Phi}_{-k}^{(N-1)}] \right), \\
\overline{\Delta}_{\boldsymbol{\Phi}^{(N-1)}}(\epsilon) &= -\frac{\delta}{2\epsilon} \left(\boldsymbol{\nabla_3} H^{1/2}[\overline{\boldsymbol{\Phi}}^{e,(N)}(\epsilon), \boldsymbol{\theta}^{(N-1)}, \widetilde{\boldsymbol{\Phi}}^{(N-1)}] - \boldsymbol{\nabla_3} H^{1/2}[\overline{\boldsymbol{\Phi}}^{e,(L)}(-\epsilon), \boldsymbol{\theta}^{(N-1)}, \widetilde{\boldsymbol{\Phi}}^{(N-1)}] \right)
\end{aligned}$$

*and subsequently for **upstream blocks**,* i.e. $\forall \ell = N-2, \cdots, 0$*:*

$$\begin{aligned}
\overline{\boldsymbol{\Phi}}^{e,(\ell+1)}(\epsilon) &= \boldsymbol{\mathcal{M}}_{H^{(\ell)},\delta}[\overline{\boldsymbol{\Phi}}^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}^{(\ell)}, \widetilde{\boldsymbol{\Phi}}^{(\ell)}] + \epsilon \boldsymbol{\Sigma}_x \cdot \widetilde{\Delta}_{\boldsymbol{\Phi}^{(\ell+1)}}(\epsilon) \\
\Delta\boldsymbol{\theta}^{(\ell)}(\epsilon) &= -\frac{\delta}{2\epsilon} \sum_{k=0}^{K-1} \left(\boldsymbol{\nabla_2} H^{1/2}[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-k}^{(\ell)}] - \boldsymbol{\nabla_2} H^{1/2}[\boldsymbol{\Phi}_k^{e,(\ell+1)}(-\epsilon), \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-k}^{(\ell)}] \right) \\
\Delta_{\boldsymbol{\Phi}^{(\ell)}}(\epsilon) &= -\frac{\delta}{2\epsilon} \left(\boldsymbol{\nabla_3} H^{1/2}[\overline{\boldsymbol{\Phi}}^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}^{(\ell)}, \widetilde{\boldsymbol{\Phi}}^{(\ell)}] - \boldsymbol{\nabla_3} H^{1/2}[\overline{\boldsymbol{\Phi}}^{e,(\ell+1)}(-\epsilon), \boldsymbol{\theta}^{(\ell)}, \widetilde{\boldsymbol{\Phi}}^{(\ell)}] \right)
\end{aligned}$$

Proof. Re-using the vectorized notations introduced in subsection 3.3 and used in Remark 5, the Lagrangian associated to the optimization problem depicted in Def. A.6 reads:

$$\begin{aligned}
\mathcal{L} &= \mathbf{1}^\top \cdot \ell[\widetilde{\boldsymbol{\Phi}}^{(L)}, \boldsymbol{\theta}^{(L-1)}] + \sum_{\ell=0}^{L-1} \mathrm{Tr}\left[\left(\boldsymbol{\mathcal{M}}_{H^{(\ell)},\delta}[\widetilde{\boldsymbol{\Phi}}^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \widetilde{\boldsymbol{u}}^{(\ell)}] - \widetilde{\boldsymbol{\Phi}}^{(\ell+1)} \right) \cdot \left(\overline{\boldsymbol{\lambda}}^{(\ell+1)} \right)^\top \right] \\
&= \sum_{k=0}^{K-1} \ell[\boldsymbol{\Phi}_{-k}^{(L)}, \boldsymbol{\theta}^{(L)}, -k] + \sum_{\ell=0}^{L-1} \left(\boldsymbol{\lambda}_k^{(\ell+1)} \right)^\top \cdot \left(\mathcal{M}_{H^{(\ell)},\delta}[\boldsymbol{\Phi}_{-(k+1)}^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell)}] - \boldsymbol{\Phi}_{-k}^{(\ell+1)} \right)
\end{aligned}$$

We proceed by induction on the block index starting from $\ell = L$.

Initialization ($\ell = L$). Let $\boldsymbol{\Phi}_k^{(N)}$ and $\boldsymbol{\lambda}_k^{(N)}$ for $k \in [\![0, K-1]\!]$ be the critical points of $\mathcal{L}$. By Theorem A.3:

$$d_{\boldsymbol{\theta}^{(N-1)}} L = \sum_{k=0}^{K-1} \nabla_2 \ell \left[\boldsymbol{\Phi}_{-k}^{(N)}, \boldsymbol{\theta}^{(N-1)}, -k \right] + \partial_2 \mathcal{M}_{H^{(N-1)},\delta} \left[\boldsymbol{\Phi}_{-(k+1)}^{(N)}, \boldsymbol{\theta}^{(L-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(N-1)} \right]^\top \cdot \boldsymbol{\lambda}_k^{(N)},$$

with $(\boldsymbol{\lambda}_k^{(N)})_{k \in [\![0,K-1]\!]}$ satisfying the following recursion relationship:

$$\begin{cases}
\boldsymbol{\lambda}_0^{(N)} = \nabla_1 \ell[\boldsymbol{\Phi}_0^{(N)}, \boldsymbol{\theta}^{(N-1)}, 0], \\
\forall k = 0, \cdots, K-1: \\
\quad \boldsymbol{\lambda}_{k+1}^{(N)} = \partial_1 \mathcal{M}_{H^{(N-1)},\delta} \left[\boldsymbol{\Phi}_{-(k+1)}^{(N)}, \boldsymbol{\theta}^{(N-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(N-1)} \right]^\top \cdot \boldsymbol{\lambda}_k^{(N)} + \nabla_1 \ell \left[\boldsymbol{\Phi}_{-(k+1)}^{(N)}, \boldsymbol{\theta}, -(k+1) \right]
\end{cases}$$

Given the definition of the dynamics of $\boldsymbol{\Phi}^{e,(N)}$ *by hypothesis*, we can directly apply Corollary A.7 to obtain:

$$\begin{aligned}
d_{\boldsymbol{\theta}^{(N-1)}} L &= \sum_{k=0}^{K-1} \nabla_2 \ell \left[\boldsymbol{\Phi}_{-k}^{(N)}, \boldsymbol{\theta}^{(N-1)}, -k \right] - \delta \, \partial_\epsilon \left(\nabla_2 H^{(N-1),1/2} \left[\boldsymbol{\Phi}_k^{e,(N)}(\epsilon), \boldsymbol{\theta}^{(L-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(N-1)} \right] \right) \Big|_{\epsilon=0}, \\
d_{\boldsymbol{\Phi}_{k-1}^{(N-1)}} L &= \partial_3 \mathcal{M}_{H^{(N-1)},\delta}[\boldsymbol{\Phi}_{-(k+1)}^{(N)}, \boldsymbol{\theta}^{(N-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(N-1)}]^\top \cdot \boldsymbol{\lambda}_k^{(N)} \\
&= -\delta \, \partial_\epsilon \left(\nabla_3 H^{(N-1),1/2} \left[\boldsymbol{\Phi}_k^{e,(N)}(\epsilon), \boldsymbol{\theta}^{(L-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(N-1)} \right] \right) \Big|_{\epsilon=0}
\end{aligned}$$

Induction ($\ell+1 \to \ell$). Let us assume that the desired property is satisfied at layer $\ell+1$. We denote again $\boldsymbol{\Phi}_k^{(\ell)}$ and $\boldsymbol{\lambda}_k^{(\ell)}$ for $k \in [\![0, K-1]\!]$ the critical point of $\mathcal{L}$. We have, for $k \in [\![0, K-1]\!]$:

$$\begin{cases} \boldsymbol{\lambda}_0^{(\ell)} = \partial_3 \mathcal{M}_{H^{(\ell)},\delta}\left[\boldsymbol{\Phi}_0^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_0^{(\ell)}\right]^\top \cdot \boldsymbol{\lambda}_0^{(\ell+1)} \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\lambda}_{k+1}^{(\ell)} = \partial_1 \mathcal{M}_{H^{(\ell-1)},\delta}\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell)}, \boldsymbol{\theta}^{(\ell-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell-1)}\right]^\top \cdot \boldsymbol{\lambda}_k^{(\ell)} + \partial_3 \mathcal{M}_{H^{(\ell)},\delta}\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell)}\right]^\top \cdot \boldsymbol{\lambda}_k^{(\ell+1)} \end{cases}$$

Using the induction hypothesis at layer $(\ell+1)$:

$$\begin{aligned} \partial_3 \mathcal{M}_{H^{(\ell)},\delta}\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell+1)}, \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell)}\right]^\top \cdot \boldsymbol{\lambda}_k^{(\ell+1)} &= -\delta\, \partial_\epsilon \left(\nabla_3 H^{(\ell),1/2}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell)}\right]\right)\Big|_{\epsilon=0} \\ &= \lim_{\epsilon \to 0} \Delta_{\boldsymbol{\Phi}_{(k+1)}^{(\ell)}}(\epsilon) \end{aligned}$$

Therefore on the one hand, denoting $\Delta_{\boldsymbol{\Phi}^{(\ell)}} := \lim_{\epsilon\to 0} \Delta_{\boldsymbol{\Phi}^{(\ell)}}(\epsilon) \in \mathbb{R}^{K \times d_{\boldsymbol{\Phi}}}$, the dynamics on $\boldsymbol{\lambda}$ rewrite:

$$\begin{cases} \boldsymbol{\lambda}_0^{(\ell)} = \Delta_{\boldsymbol{\Phi}_0^{(\ell)}} \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\lambda}_{k+1}^{(\ell)} = \partial_1 \mathcal{M}_{H^{(\ell-1)},\delta}\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell)}, \boldsymbol{\theta}^{(\ell-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell-1)}\right]^\top \cdot \boldsymbol{\lambda}_k^{(\ell)} + \Delta_{\boldsymbol{\Phi}_{(k+1)}^{(\ell)}} \end{cases}$$

On the other hand, the dynamics of $\boldsymbol{\Phi}^{e,(\ell)}$ read *by hypothesis*:

$$\begin{cases} \boldsymbol{\Phi}_0^{(\ell)} = \left(\boldsymbol{\Phi}_0^{(\ell)}\right)^\star + \epsilon \boldsymbol{\Sigma}_x \cdot \Delta_{\boldsymbol{\Phi}_0^{(\ell)}}(\epsilon) \\ \forall k = 0, \cdots, K-1 : \\ \quad \boldsymbol{\Phi}_{k+1}^{e,(\ell)} = \mathcal{M}_{H^{(\ell-1)},\delta}\left[\boldsymbol{\Phi}_{k+1}^{e,(\ell)}, \boldsymbol{\theta}^{(\ell-1)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell-1)}\right] + \epsilon \boldsymbol{\Sigma}_x \cdot \Delta_{\boldsymbol{\Phi}_{(k+1)}^{(\ell)}} \end{cases}$$

using Corollary A.8 with $y = \Delta_{\boldsymbol{\Phi}^{(\ell)}}$, we conclude that:

$$d_{\boldsymbol{\theta}^{(\ell)}} L = \lim_{\epsilon \to 0} \Delta \boldsymbol{\theta}^{(\ell)}(\epsilon)$$

□

Remark 7. *Theorem A.4, and more generally our definition of HSSMs (Def. A.5) assume that the connectivity pattern of the HRU units is a* linear chain. *Note that while we chose this hypothesis for the sake of clarity of our results and their derivations, Theorem A.4 could be seamlessly extended to any* Directed Acyclic Graph *(DAG) of HRUs. This allows, for instance as a simple and realistic case, to use skip connections across HRUs within HSSMs.*

Remark 8. *Note that RHEL chaining as prescribed by Theorem A.4* ***implicitly chains RHEL and automatic differentiation****. Indeed, if H explicitly parametrizes feedforward mappings across HRUs as:*

$$H^{(\ell)}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell)}\right] = H^{(\ell)}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}_\alpha^{(\ell)}, F\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell)}, \boldsymbol{\theta}_\beta^{(\ell)}\right]\right], \tag{23}$$

then, denoting $\boldsymbol{u}^{(\ell)} := F\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell)}, \boldsymbol{\theta}_\beta^{(\ell)}\right]$, we have:

$$\begin{aligned} &\partial_\epsilon \left(\nabla_3 H^{(\ell),1/2}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}^{(\ell)}, \boldsymbol{\Phi}_{-(k+1)}^{(\ell)}\right]\right)\Big|_{\epsilon=0} \\ &= \partial_1 F\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell)}, \boldsymbol{\theta}_\beta^{(\ell)}\right]^\top \cdot \partial_\epsilon \left(\nabla_3 H^{(\ell),1/2}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}_\alpha^{(\ell)}, \boldsymbol{u}^\ell\right]\right)\Big|_{\epsilon=0} \\ &\approx \partial_1 F\left[\boldsymbol{\Phi}_{-(k+1)}^{(\ell)}, \boldsymbol{\theta}_\beta^{(\ell)}\right]^\top \cdot \frac{1}{2\epsilon}\left(\nabla_3 H^{(\ell),1/2}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(\epsilon), \boldsymbol{\theta}_\alpha^{(\ell)}, \boldsymbol{u}^\ell\right] - \nabla_3 H^{(\ell),1/2}\left[\boldsymbol{\Phi}_k^{e,(\ell+1)}(-\epsilon), \boldsymbol{\theta}_\alpha^{(\ell)}, \boldsymbol{u}^\ell\right]\right) \end{aligned}$$

The red part is done by automatic differentiation and the blue part by RHEL. This underpins the implementation of RHEL chaining we used in our own code.

A.3 Models and algorithms details

Summary. In this section, we provide details about our models and algorithms. More precisely:

- In section A.3.1, we first describe our *toy model* used inside Fig. 2 in terms of its Hamiltonian, resulting continuous-time dynamics and RHEL gradient estimators as prescribed by Theorem 3.1.
- In section A.3.2, we provide details about the HSSMs which we used in our experiments. We describe in greater details HSSMs made up of *linear* HRU blocks (section A.3.3). We describe their Hamiltonian, their resulting dynamics, the associated gradient estimators prescribed by RHEL and explain how *parallel scan* can be used on these models, especially when applying RHEL. Similarly, we describe HSSMs made up of *nonlinear* HRU blocks (section A.3.4).
- In section A.3.5, we show how the time discretization δ can itself be trained by absorbing it into the definition of the Hamiltonian function.
- Finally, in the light of Remark 8, we highlight in section A.3.6 how our implementation hybridizes *Automatic Differentiation* (AD) and RHEL using Algorithms 3–4.

A.3.1 Toy model

In this section, we provide the gradient estimators for the parameters of the toy model. The toy model is a simple network of six mechanically coupled oscillators. Each oscillator is described by a state $\boldsymbol{\Phi^i} = (\phi^i, \pi^i)$ where $\phi^i \in \mathbb{R}$ is the position of the oscillator and $\pi^i \in \mathbb{R}$ is its momentum. It has a mass parameter m_i and spring parameter k_i. Any pair of oscillators (i, j) is coupled via the spring parameter k_{ij}. The input to the models is a time-varying external force $\boldsymbol{u}(t) \in \mathbb{R}$ coupled to oscillator 1. During the echo passes, the nudging force is modelled by a spring coupling with parameter $\epsilon \in \mathbb{R}$ to an external force $y(t) \in \mathbb{R}$. The Hamiltonian of the system is given by:

$$H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \mathbf{u}] = \sum_i \frac{(\pi^i)^2}{2m_i} + \frac{1}{2}\sum_i k_i(\phi^i)^2 + \frac{1}{2}\sum_i \sum_{j>i} k_{ij}(\phi^j - \phi^i)^2 + \mathbf{u}\,\phi^1 \tag{24}$$

Which gives the following equations of motion:

$$\begin{cases} \dot{\phi}^i = \frac{\pi^i}{m_i} \quad \text{for all } i \in \{1, 6\} \\ \dot{\pi}^i = -k_i\phi^i + \sum_{j,j\neq i} k_{ij}(\phi^j - \phi^i), \quad i \in \{2, 3, 5, 6\} \\ \dot{\pi}^1 = -k_1\phi^1 + \sum_{j,j\neq 1} k_{1j}(\phi^j - \phi^1) + \boldsymbol{u} \\ \dot{\pi}^4 = -k_4\phi^4 + \sum_{j,j\neq 4} k_{4j}(\phi^j - \phi^4) - \delta_e\epsilon(\boldsymbol{\phi}^4 - y) \end{cases}$$

where δ_e is the indicator function of the echo pass, it's equal to 1 during the echo pass and 0 otherwise.

RHEL gradient estimators of the model parameters. For the mass m_i, we have:

$$\begin{aligned} \Delta_{m_i}^{RHEL} &= -\frac{1}{2\epsilon}\left(\nabla_{m_i} H\left[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}\right] - \nabla_{m_i} H\left[\boldsymbol{\Phi}^e(t,-\epsilon), \boldsymbol{\theta}, \boldsymbol{u}\right]\right) \\ &= \frac{1}{2\epsilon}\left(\frac{\left(\pi^i(t,\epsilon)\right)^2}{2m_i^2} - \frac{\left(\pi^i(t,-\epsilon)\right)^2}{2m_i^2}\right) \end{aligned} \tag{25}$$

For the spring parameters k_i, we haves:

$$\begin{aligned} \Delta_{k_i}^{RHEL} &= -\frac{1}{2\epsilon}\left(\nabla_{k_i} H[\boldsymbol{\Phi}^e(t,\epsilon), \boldsymbol{\theta}, \boldsymbol{u}] - \nabla_{k_i} H[\boldsymbol{\Phi}^e(t,-\epsilon), \boldsymbol{\theta}, \boldsymbol{u}]\right) \\ &= -\frac{1}{2\epsilon}\left(\left(\phi^i(t,\epsilon)\right)^2 - \left(\phi^i(t,-\epsilon)\right)^2\right) \end{aligned} \tag{26}$$

$$\tag{27}$$

For the coupling parameters k_{ij}, we have:

$$\begin{aligned}\Delta_{k_{ij}}^{RHEL} &= -\frac{1}{2\epsilon}\left(\nabla_{k_{ij}} H[\mathbf{\Phi}^e(t,\epsilon),\boldsymbol{\theta},\boldsymbol{u}] - \nabla_{k_{ij}} H[\mathbf{\Phi}^e(t,-\epsilon),\boldsymbol{\theta},\boldsymbol{u}]\right) \\ &= -\frac{1}{2\epsilon}\left(\left(\phi^j(t,\epsilon)-\phi^i(t,\epsilon)\right)^2 - \left(\phi^j(t,-\epsilon)-\phi^i(t,-\epsilon)\right)^2\right) \end{aligned} \tag{28}$$

RHEL gradient estimators of the state sensitivities For the state position sensitivities, we have:

$$\begin{aligned}\Delta_{\phi^i}^{RHEL} &= -\frac{1}{2\epsilon}\mathbf{\Sigma}_x\left(\mathbf{\Phi}^e(t,\epsilon) - \mathbf{\Phi}^e(t,-\epsilon)\right) \\ &= -\frac{1}{2\epsilon}\left(\pi^i(t,\epsilon) - \pi^i(t,-\epsilon)\right) \end{aligned} \tag{29}$$

For the state momentum sensitivities, we have:

$$\begin{aligned}\Delta_{\pi^i}^{RHEL} &= -\frac{1}{2\epsilon}\mathbf{\Sigma}_x\left(\mathbf{\Phi}^e(t,\epsilon) - \mathbf{\Phi}^e(t,-\epsilon)\right) \\ &= -\frac{1}{2\epsilon}\left(\phi^i(t,\epsilon) - \phi^i(t,-\epsilon)\right) \end{aligned} \tag{30}$$

A.3.2 HSSM architecture

In this section, we outline the detailed architecture of a full multi-layer HSSM architecture. For the inference, we re-use the same stacking architecture of recurrent blocks and feedforward elements as the LinOSS model [29]. The model starts by encoding an input sequence $\overline{\boldsymbol{u}} \in \mathbb{R}^{d_{\boldsymbol{u}} \times K}$ via an affine transformation. The transformed sequence then progresses through multiple HSSM blocks, linear (see Appendix A.3.3), or nonlinear (see Appendix A.3.4), directly followed by nonlinear transformations. These transformations include the Gaussian error linear unit (GELU) [55] and the Gated Linear Unit (GLU) [56], defined as $\text{GLU}(\mathbf{x}) = \text{sigmoid}(\mathbf{W}_1\mathbf{x}) \circ \mathbf{W}_2\mathbf{x}$ where $\mathbf{W}_{1,2}$ represent trainable weight matrices, accompanied by a residual connection. The sequence output from the final block undergoes a second affine transformation to produce the model output.

The full linear and nonlinear HSSM is further presented in Algorithm 2. For clarity, when applying operations to sequence elements denoted with an overline (e.g., $\overline{\boldsymbol{u}}$), these operations are implicitly broadcast across the time dimension. Specifically, for any function f applied to $\overline{\boldsymbol{u}}$, we have $f(\overline{\boldsymbol{u}})_t = f(\boldsymbol{u}_t)$ for all time steps $t \in \{1, 2, ..., K\}$.

For the inference of the linear HSSMs, we keep the same recurrent block as LinOSS with a slight change in the integrator (see A.3.3). For the nonlinear HSSM, we replace the recurrent block by a UniCORRN recurrent block [30] for which we use the same integrator as for the linear HSSM (see A.3.4).

Algorithm 2 HSSM model

Input: Input sequence $\overline{\boldsymbol{u}}$, model type **type** $\in \{\text{linear}, \text{nonlinear}\}$
Output: HSSM output sequence $\overline{\boldsymbol{u}}$
$\overline{\boldsymbol{u}}^{(0)} \leftarrow \boldsymbol{W}_{enc}\overline{\boldsymbol{u}} + \boldsymbol{b}_{enc}$
for $\ell = 1, \ldots, N$ **do**
  if type $=$ linear **then**
    $\overline{\mathbf{\Phi}}^{(\ell)}, \_, \_ \leftarrow \text{LINEARHRU}(\overline{\boldsymbol{u}}^{(\ell-1)}, 0)$ ▷ Via parallel scan
  else
    $\overline{\mathbf{\Phi}}^{(\ell)}, \_, \_ \leftarrow \text{NONLINEARHRU}(\overline{\boldsymbol{u}}^{(\ell-1)}, 0)$ ▷ Sequentially
  end if
  $\overline{\boldsymbol{x}}^{(\ell)} \leftarrow \boldsymbol{C}\overline{\boldsymbol{\phi}}^{(\ell)} + \boldsymbol{D}\overline{\boldsymbol{u}}^{\ell-1}$
  $\overline{\boldsymbol{x_g}}^{(\ell)} \leftarrow \text{GELU}(\overline{\boldsymbol{x}}^{(\ell)})$
  $\overline{\boldsymbol{u}}^{(\ell)} \leftarrow \text{GLU}(\overline{\boldsymbol{x_g}}^{(\ell)} + \overline{\boldsymbol{u}}^{(\ell-1)})$
end for
$\overline{\boldsymbol{o}} \leftarrow \boldsymbol{W}_{dec}\overline{\boldsymbol{u}}^{(N)} + \boldsymbol{b}_{dec}$

A.3.3 Linear HRU Block

Hamiltonian of the recurrence. The linear HRU block is the composition of a nonlinear spatial transformation and a linear recurrent transformation (see Eq. 17). Here we provide more details about the linear recurrence that is computed with the RHEL gradient estimator. The linear recurrence is defined by the following Hamiltonian:

$$\begin{aligned} H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] &= T[\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{u}] + V[\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{u}] \\ &= \frac{1}{2}\|\boldsymbol{\pi}\|^2 + \left(\frac{1}{2}\boldsymbol{\phi}^\top \boldsymbol{A}\boldsymbol{\phi} - \boldsymbol{\phi}^\top \boldsymbol{B}\boldsymbol{u}\right) \end{aligned} \tag{31}$$

Dynamics. The dynamics of the linear HRU block are defined by the following equations:

$$\begin{cases} \dot{\boldsymbol{\phi}} = \boldsymbol{\pi} \\ \dot{\boldsymbol{\pi}} = -\boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{B}\boldsymbol{u} \end{cases} \tag{32}$$

Which, after time-discretization with the integrator defined in A.2.1, gives the following equations:

$$\begin{cases} \boldsymbol{\phi}_{k+1/3} = \boldsymbol{\phi}_k + \dfrac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k] \\ \qquad\qquad = \boldsymbol{\phi}_k + \dfrac{\delta}{2}\boldsymbol{\pi}_k \\ \boldsymbol{\pi}_{k+1/3} = \boldsymbol{\pi}_k \\ \boldsymbol{\phi}_{k+2/3} = \boldsymbol{\phi}_{k+1/3} \\ \boldsymbol{\pi}_{k+2/3} = \boldsymbol{\pi}_{k+1/3} + \delta\nabla_{\boldsymbol{\phi}} V[\boldsymbol{\phi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}_k] \\ \qquad\qquad = \boldsymbol{\pi}_{k+1/3} - \delta\boldsymbol{A}\boldsymbol{\phi}_{k+1/3} + \delta\boldsymbol{B}\boldsymbol{u}_k \\ \boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_{k+2/3} + \dfrac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}_k] \\ \qquad\qquad = \boldsymbol{\phi}_{k+2/3} + \dfrac{\delta}{2}\boldsymbol{\pi}_{k+2/3} \\ \boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_{k+2/3} \end{cases} \tag{33}$$

with the initial condition $\boldsymbol{\Phi}_{-K} = (\boldsymbol{\phi}_{-K}^\top, \boldsymbol{\pi}_{-K}^\top)^\top = \boldsymbol{x}$.

For the echo passes, the initial condition are $\boldsymbol{\Phi}_0^e = \boldsymbol{\Phi}_0^\star \pm \epsilon\boldsymbol{\Sigma}_x \cdot \Delta_{\boldsymbol{\Phi}}\ell[\boldsymbol{\Phi}_0, 0]$. The dynamics equations follow below, with modifications from equation 33 highlighted in blue::

$$\begin{cases} \boldsymbol{\phi}^e_{k+1/3} = \boldsymbol{\phi}^e_k + \dfrac{\delta}{2}\boldsymbol{\pi}^e_k \\ \boldsymbol{\pi}^e_{k+1/3} = \boldsymbol{\pi}^e_k \\ \boldsymbol{\phi}^e_{k+2/3} = \boldsymbol{\phi}^e_{k+1/3} \\ \boldsymbol{\pi}^e_{k+2/3} = \boldsymbol{\pi}^e_{k+1/3} - \delta\boldsymbol{A}\boldsymbol{\phi}^e_{k+1/3} + \delta\boldsymbol{B}\boldsymbol{u}_{-(k+1)} \\ \boldsymbol{\phi}^e_{k+1} = \boldsymbol{\phi}^e_{k+2/3} + \dfrac{\delta}{2}\boldsymbol{\pi}^e_{k+2/3} + \epsilon\nabla_{\boldsymbol{\pi}}\ell[\boldsymbol{\Phi}_{-(k+1)}] \\ \boldsymbol{\pi}^e_{k+1} = \boldsymbol{\pi}^e_{k+2/3} + \epsilon\nabla_{\boldsymbol{\phi}}\ell[\boldsymbol{\Phi}_{-(k+1)}] \end{cases} \tag{34}$$

RHEL gradient estimators. The gradient estimators of the parameters of the linear HRU are:

$$\begin{aligned}
\Delta_{\boldsymbol{A}}^{RHEL}(k,\epsilon) &= -\frac{\delta}{2\epsilon}\left(\nabla_{\boldsymbol{A}} H^{1/2}[\boldsymbol{\Phi}_k^e(\epsilon),\boldsymbol{\theta},\boldsymbol{u}_{-(k+1)}] - \nabla_{\boldsymbol{A}} H^{1/2}[\boldsymbol{\Phi}_k^e(-\epsilon),\boldsymbol{\theta},\boldsymbol{u}_{-(k+1)}]\right) \\
&= -\frac{\delta}{4\epsilon}\Big[\left(\boldsymbol{\phi}_{k+1/3}^e(\epsilon)^\top \boldsymbol{\phi}_{k+1/3}^e(\epsilon) + \boldsymbol{\phi}_{k+2/3}^e(\epsilon)^\top \boldsymbol{\phi}_{k+2/3}^e(\epsilon)\right) \\
&\qquad - \left(\boldsymbol{\phi}_{k+1/3}^e(-\epsilon)^\top \boldsymbol{\phi}_{k+1/3}^e(-\epsilon) + \boldsymbol{\phi}_{k+2/3}^e(-\epsilon)^\top \boldsymbol{\phi}_{k+2/3}^e(-\epsilon)\right)\Big] \\
&= -\frac{\delta}{2\epsilon}\left[\left(\boldsymbol{\phi}_{k+1/3}^e(\epsilon)\right)^\top \left(\boldsymbol{\phi}_{k+1/3}^e(\epsilon)\right) - \left(\boldsymbol{\phi}_{k+1/3}^e(-\epsilon)\right)^\top \left(\boldsymbol{\phi}_{k+1/3}^e(-\epsilon)\right)\right] \qquad (35) \\
\Delta_{\boldsymbol{B}}^{RHEL}(k,\epsilon) &= -\frac{\delta}{2\epsilon}\left(\nabla_{\boldsymbol{B}} H^{1/2}[\boldsymbol{\Phi}_k^e(\epsilon),\boldsymbol{\theta},\boldsymbol{u}_{-(k+1)}] - \nabla_{\boldsymbol{B}} H^{1/2}[\boldsymbol{\Phi}_k^e(-\epsilon),\boldsymbol{\theta},\boldsymbol{u}_{-(k+1)}]\right) \\
&= \frac{\delta}{4\epsilon}\Big[\left(\boldsymbol{\phi}_{k+1/3}^e(\epsilon) + \boldsymbol{\phi}_{k+2/3}^e(\epsilon)\right)\boldsymbol{u}_{-(k+1)}^\top \\
&\qquad - \left(\boldsymbol{\phi}_{k+1/3}^e(-\epsilon) + \boldsymbol{\phi}_{k+2/3}^e(-\epsilon)\right)\boldsymbol{u}_{-(k+1)}^\top\Big] \\
&= \frac{\delta}{2\epsilon}\left[\boldsymbol{\phi}_{k+1/3}^e(\epsilon)\boldsymbol{u}_{-(k+1)}^\top - \boldsymbol{\phi}_{k+1/3}^e(-\epsilon)\boldsymbol{u}_{-(k+1)}^\top\right] \qquad (36)
\end{aligned}$$

The gradient estimator with respect to the input of the recurrent transformation is:

$$\begin{aligned}
\Delta_{\boldsymbol{u}}^{RHEL}(k,\epsilon) &= -\frac{\delta}{2\epsilon}\left(\nabla_{\boldsymbol{u}} H^{1/2}[\boldsymbol{\Phi}_k^e(\epsilon),\boldsymbol{\theta},\boldsymbol{u}_{-(k+1)}] - \nabla_{\boldsymbol{u}} H^{1/2}[\boldsymbol{\Phi}_k^e(-\epsilon),\boldsymbol{\theta},\boldsymbol{u}_{-(k+1)}]\right) \\
&= \frac{\delta}{4\epsilon}\Big[\boldsymbol{B}^\top\left(\boldsymbol{\phi}_{k+1/3}^e(\epsilon) + \boldsymbol{\phi}_{k+2/3}^e(\epsilon)\right) \\
&\qquad - \boldsymbol{B}^\top\left(\boldsymbol{\phi}_{k+1/3}^e(-\epsilon) + \boldsymbol{\phi}_{k+2/3}^e(-\epsilon)\right)\Big] \\
&= \frac{\delta}{2\epsilon}\left[\boldsymbol{B}^\top\boldsymbol{\phi}_{k+1/3}^e(\epsilon) - \boldsymbol{B}^\top\boldsymbol{\phi}_{k+1/3}^e(-\epsilon)\right] \qquad (37)
\end{aligned}$$

Parallel Scan. Similarly to the LinOSS model [29], we can compute the recurrence of Linear HRU with a parallel scan [5] to reduce the computational time. To implement the parallel scan, we need to put the discretized dynamics in a form that can be computed in parallel. For this we vectorize the Equation 33:

$$\begin{aligned}
\boldsymbol{\Phi}_{k+1} &= \begin{bmatrix} \boldsymbol{I} & \frac{\delta}{2}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\cdot\boldsymbol{\Phi}_{k+2/3} \\
&= \begin{bmatrix} \boldsymbol{I} & \frac{\delta}{2}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\cdot\left(\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\delta\boldsymbol{A} & \boldsymbol{I}\end{bmatrix}\cdot\boldsymbol{\Phi}_{k+1/3} + \begin{bmatrix}\boldsymbol{0} \\ \delta\boldsymbol{B}\cdot\boldsymbol{u}_k\end{bmatrix}\right) \\
&= \begin{bmatrix} \boldsymbol{I} & \frac{\delta}{2}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\cdot\begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ -\delta\boldsymbol{A} & \boldsymbol{I}\end{bmatrix}\cdot\begin{bmatrix} \boldsymbol{I} & \frac{\delta}{2}\boldsymbol{I} \\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\cdot\boldsymbol{\Phi}_k + \begin{bmatrix}\frac{\delta^2}{2}\boldsymbol{B}\cdot\boldsymbol{u}_k \\ \delta\boldsymbol{B}\cdot\boldsymbol{u}_k\end{bmatrix}
\end{aligned}$$

which gives us the discrete dynamics in matrix form:

$$\boldsymbol{\Phi}_{k+1} = M\boldsymbol{\Phi}_k + F_k \qquad (38)$$

with:

$$M = \begin{bmatrix} I - \frac{\delta^2}{2}\boldsymbol{A} & \delta[I - \frac{\delta^2}{4}\boldsymbol{A}] \\ -\delta\boldsymbol{A} & I - \frac{\delta^2}{2}\boldsymbol{A}\end{bmatrix}, \quad F_k = \begin{bmatrix}\frac{\delta^2}{2}\boldsymbol{B}\cdot\boldsymbol{u}_k \\ \delta\boldsymbol{B}\cdot\boldsymbol{u}_k\end{bmatrix} \qquad (39)$$

To compute the echo passes $\boldsymbol{\Phi}_k^e, k \in \{1,\cdots,K\}$, we only need to adapt F_k to get for the positive nudging $+\epsilon$:

$$F_k^e = \begin{bmatrix}\frac{\delta^2}{2}\boldsymbol{B}\cdot\boldsymbol{u}_{-(k+1)} + \epsilon\nabla_{\boldsymbol{\pi}}\ell[\boldsymbol{\Phi}_{-(k+1)}] \\ \delta\boldsymbol{B}\cdot\boldsymbol{u}_{-(k+1)} + \epsilon\nabla_{\boldsymbol{\phi}}\ell[\boldsymbol{\Phi}_{-(k+1)}]\end{bmatrix} \qquad (40)$$

A.3.4 Nonlinear HRU

Hamiltonian of the recurrence. The nonlinear HRU block is the composition of a nonlinear spatial transformation and a nonlinear recurrent transformation (see Eq. 18). The Hamiltonian of the nonlinear HRU block is defined by the following equations:

$$\begin{aligned} H[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}] &= T[\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{u}] + V[\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{u}] \\ &= \frac{1}{2}\|\boldsymbol{\pi}\|^2 + \frac{\alpha}{2}\|\boldsymbol{\phi}\|^2 + \left(\boldsymbol{A}^{-\top} \cdot \log\left(\cosh\left(\boldsymbol{A} \cdot \boldsymbol{\phi} + \boldsymbol{B} \cdot \boldsymbol{u} + \boldsymbol{b}\right)\right)\right) \end{aligned} \tag{41}$$

Dynamics. The dynamics of the nonlinear HRU block are defined by the following equations:

$$\begin{cases} \dot{\boldsymbol{\phi}} = \boldsymbol{\pi} \\ \dot{\boldsymbol{\pi}} = -\left(\tanh\left(\boldsymbol{A}\boldsymbol{\phi} + \boldsymbol{B}\boldsymbol{u} + b\right) + \alpha\boldsymbol{\phi}\right) \end{cases} \tag{42}$$

Which, after time-discretization with the integrator defined in A.2.1, gives the following equations:

$$\begin{cases} \boldsymbol{\phi}_{k+1/3} = \boldsymbol{\phi}_k + \dfrac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k] \\ \qquad\qquad = \boldsymbol{\phi}_k + \dfrac{\delta}{2}\boldsymbol{\pi}_k \\ \boldsymbol{\pi}_{k+1/3} = \boldsymbol{\pi}_k \\ \boldsymbol{\phi}_{k+2/3} = \boldsymbol{\phi}_{k+1/3} \\ \boldsymbol{\pi}_{k+2/3} = \boldsymbol{\pi}_{k+1/3} + \delta\nabla_{\boldsymbol{\phi}} V[\boldsymbol{\phi}_{k+1/3}, \boldsymbol{\theta}, \boldsymbol{u}_k] \\ \qquad\qquad = \boldsymbol{\pi}_{k+1/3} - \delta(\tanh\left(\boldsymbol{A}\boldsymbol{\phi}_{k+1/3} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{b}\right) + \alpha\boldsymbol{\phi}_{k+1/3}) \\ \boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_{k+2/3} + \dfrac{\delta}{2}\nabla_{\boldsymbol{\pi}} T[\boldsymbol{\pi}_{k+2/3}, \boldsymbol{\theta}, \boldsymbol{u}_k] \\ \qquad\qquad = \boldsymbol{\phi}_{k+2/3} + \dfrac{\delta}{2}\boldsymbol{\pi}_{k+2/3} \\ \boldsymbol{\pi}_{k+1} = \boldsymbol{\pi}_{k+2/3} \end{cases} \tag{43}$$

A.3.5 Differentiating the time-discretization

In training HRUs, one can also train the multidimensional time-discretization $\delta \in \mathbb{R}^{d_{\boldsymbol{\Phi}} \times d_{\boldsymbol{\Phi}}}$ that is assumed to be diagonal. For this, it suffices to reparametrize the integrator and the Hamiltonian, so that the time discretization becomes a parameter of the Hamiltonian. Hence the HRU equation 9 becomes:

$$\boldsymbol{\Phi}_{k+1} = \mathcal{M}_{\hat{H},1}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k, \delta] \quad \forall k = -K \cdots - 1, \tag{44}$$

with $\hat{H}$ is a reparametrization of the Hamiltonian H such that $\mathcal{M}_{\hat{H},1}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k, \delta] = \mathcal{M}_{H,\delta}[\boldsymbol{\Phi}_k, \boldsymbol{\theta}, \boldsymbol{u}_k]$.

For instance, for the linear HRU, we can reparametrize the Hamiltonian as:

$$\hat{H}[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}, \delta] = \frac{1}{2}\boldsymbol{\pi}^\top \delta \boldsymbol{\pi} + \left(\frac{1}{2}\boldsymbol{\phi}^\top(\delta\boldsymbol{A})\boldsymbol{\phi} - \boldsymbol{\phi}^\top(\delta\boldsymbol{B})\boldsymbol{u}\right) \tag{45}$$

And for the nonlinear HRU, we can reparametrize the Hamiltonian as:

$$\hat{H}[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}, \delta] = \frac{1}{2}\boldsymbol{\pi}^\top \delta \boldsymbol{\pi} + \frac{\alpha}{2}\boldsymbol{\phi}^\top \delta \boldsymbol{\phi} + \frac{1}{2}\left(\delta\boldsymbol{A}^{-\top} \cdot \log\left(\cosh\left(\boldsymbol{A} \cdot \boldsymbol{\phi} + \boldsymbol{B} \cdot \boldsymbol{u} + \boldsymbol{b}\right)\right)\right) \tag{46}$$

$$\tag{47}$$

Note that to match with previous implementations of the nonlinear HRU [30], we used the following parametrization for the discretization step:

$$\hat{H}[\boldsymbol{\Phi}, \boldsymbol{\theta}, \boldsymbol{u}, \delta] = \frac{1}{2}\boldsymbol{\pi}^\top \sigma(\delta) \boldsymbol{\pi} + \frac{\alpha}{2}\boldsymbol{\phi}^\top \sigma(\delta) \boldsymbol{\phi} + \frac{1}{2}\left(\sigma(\delta)\boldsymbol{A}^{-\top} \cdot \log\left(\cosh\left(\boldsymbol{A} \cdot \boldsymbol{\phi} + \boldsymbol{B} \cdot \boldsymbol{u} + \boldsymbol{b}\right)\right)\right) \tag{48}$$

where $\sigma(\delta)$ is a diagonal matrix with $\sigma(\delta)_{ii} = 0.5 + 0.5\tanh(\delta_{ii}/2)$.

Algorithm 3 Custom Reverse Mode Automatic Differentiation for an arbitrary HRU

@custom_autodiff
function HRU($\boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}}$)
    $\overline{\boldsymbol{\Phi}} \leftarrow$ HRU-HELPER$(\mathbf{0}, \boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}}, \overline{\mathbf{0}})$
    return $\overline{\boldsymbol{\Phi}}$
end function

function FORWARDHRU($\boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}}$)
    $\overline{\boldsymbol{\Phi}} \leftarrow$ HRU-HELPER$(\mathbf{0}, \boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}}, \overline{\mathbf{0}})$
    return $\overline{\boldsymbol{\Phi}}, \boldsymbol{\Phi}_K$
end function

function BACKWARDHRU($\boldsymbol{r}$,$\overline{\boldsymbol{g}}$, input_forward)
    $\boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}} \leftarrow$ input_forward
    $\tilde{\boldsymbol{u}} \leftarrow$ REVERSE$(\overline{\boldsymbol{u}})$ ▷ Reverse the input sequence
    $\boldsymbol{\Phi}_K \leftarrow \boldsymbol{r}$
    $\boldsymbol{\Phi}^e_{0,\pm\epsilon} \leftarrow \boldsymbol{\Sigma}_x \boldsymbol{\Phi}_K + \epsilon \boldsymbol{\Sigma}_x \boldsymbol{g}_K$
    $\overline{\boldsymbol{\Phi}}^e_{\pm\epsilon} \leftarrow$ HRU-HELPER$(\boldsymbol{\Phi}^e_{0,\pm\epsilon}, \boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{g}})$
    $\Delta_{\boldsymbol{\theta}_{HRU}}, \Delta_{\overline{\boldsymbol{u}}} \leftarrow$ LEARNINGHRU$(\overline{\boldsymbol{\Phi}}^e_{\pm\epsilon})$
    return $\Delta_{\boldsymbol{\theta}_{HRU}}, \Delta_{\overline{\boldsymbol{u}}}$ ▷ Loss gradient with respect to the input of FORWARDHRU$(\cdot, \cdot)$
end function

HRU.define_custom_autodiff(ForwardHRU, BackwardHRU)

Algorithm 4 Helper functions for custom Automatic Differentiation

function HRU-HELPER($\boldsymbol{\Phi}_0, \boldsymbol{\theta}_{HRU}, \overline{\boldsymbol{u}}, \overline{\boldsymbol{n}}$) ▷ See Dynamics. in App. A.3.3 and A.3.4 for concrete examples
    Input: Initial State $\boldsymbol{\Phi}_0$, Parameters of the recurrence $\boldsymbol{\theta}_{HRU}$, Inputs $\overline{\boldsymbol{u}}$, Nudging $\overline{\boldsymbol{n}}$
    Output: State sequence $\overline{\boldsymbol{\Phi}}$
end function

function LEARNINGHRU($\overline{\boldsymbol{\Phi}}^e_{\pm\epsilon}$, $\overline{\boldsymbol{u}}$) ▷ See RHEL gradient estimator in App. A.3.3 for a concrete example
    Input: Echo passes $\overline{\boldsymbol{\Phi}}^e_{\pm\epsilon}$, input sequence time-reversed $\tilde{\boldsymbol{u}}$
    Output: Loss gradient w.r.t to the input of FORWARDHRU$(\cdot, \cdot)$: $\Delta_{\boldsymbol{\theta}_{HRU}}, \Delta_{\overline{\boldsymbol{u}}}$
end function

A.3.6 Echo passes with automatic differentiation

As mentioned in Remark 8, learning in HSSMs with RHEL involves chaining RHEL gradient estimators with Automatic Differentiation (AD). This design makes RHEL implementation readily compatible with modern automatic differentiation frameworks such as PyTorch or JAX. These libraries provide the capability to create custom functions that, when invoked within a computational graph, are automatically differentiated. Algorithm 3 demonstrates how to implement RHEL/AD chaining for the recurrent component of an HRU block.

To implement a custom autodiff function `HRU`, we must define two additional functions required by AD: `ForwardHRU` and `BackwardHRU`. The `ForwardHRU` function is invoked when the `HRU` function is called within a computational graph to be differentiated. The `ForwardHRU` function returns two elements: its output for computing the remainder of the computational graph $\bar{\boldsymbol{\Phi}}$, and a *residual* to be stored for the backward pass. For the HRU, the residual consists only of the final state $\boldsymbol{\Phi}_K$ from the forward pass. The `BackwardHRU` function is called during graph backpropagation. It receives as input the loss gradient from the upstream portion of the computational graph $\bar{g}$, the input from the forward pass $\bar{u}$, and the residual $\boldsymbol{\Phi}_K$. The `BackwardHRU` function computes the gradient of the loss with respect to the HRU block parameters, which is subsequently used to update the HRU parameters.

It also computes the gradient of the loss with respect to the forward pass input $\bar{u}$, which is then used to propagate the loss gradient backward through the computational graph.

These three functions are then registered with the autodiff library using the `HRU.define_custom_autodiff` function, enabling the library to automatically differentiate the `HRU` function when it is called within a computational graph.

The three functions utilize two helper functions: `HRU-HELPER` and `LearningHRU`. The `HRU-HELPER` function implements the HRU dynamics and is employed in both the forward and backward passes. The `LearningHRU` function implements the RHEL gradient estimator and is used in the backward pass to compute the gradient of the loss with respect to the HRU block parameters.s

A.4 Experimental details

Summary. This last section provides additional experimental details. More precisely:

- We provide details about the datasets at use, the devices we used, as well as detailed hyperparameters.
- Importantly, we detail **how RHEL can be subject to numerical *underflow* and how we mitigated this issue** – see Fig. 5–6 for details.

A.4.1 Datasets

This series of tasks was recently introduced as a subset of the University of East Anglia (UEA) datasets with the longest sequences for increased difficulty and recently used to benchmark the linear HSSM previously introduced [29]

The classification datasets are drawn from a recently introduced benchmark [34] that selects a subset of the University of East Anglia (UEA) datasets [35], specifically choosing those with the longest sequences to increase difficulty, and which has been recently employed to evaluate the linear HSSM model [29]. These datasets include EigenWorms (17,984 sequence length, 5 classes), SelfRegulationSCP1 (896 length, 2 classes), SelfRegulationSCP2 (1,152 length, 2 classes), EthanolConcentration (1,751 length, 4 classes), Heartbeat (405 length, 2 classes), and MotorImagery (3,000 length, 2 classes).

Additionally, we evaluate our HSSMs on the PPG-DaLiA dataset [36], a multivariate time series regression dataset designed for heart rate prediction using data collected from a wrist-worn device. It includes recordings from fifteen individuals, each with approximately 150 minutes of data sampled at a maximum rate of 128 Hz. The dataset consists of six channels: blood volume pulse, electrodermal activity, body temperature, and three-axis acceleration. After splitting the data, a sliding window of length 49920 and step size 4992 is applied, representing a challenging very long-range interaction task.

A.4.2 Simulation details and ressource consumption

Simulation details .The code to run the experiments is implemented using the JAX auto-differentiation framework [57]. All experiments were run on Nvidia V100 GPUs, except for the PPG experiments, which were run on Nvidia Tesla A100 GPUs due to larger memory demands.

Ressource consumption. For the linear HSSM, training time is approximately 30 minutes for both classification (SCP1 dataset) and regression (PPG dataset) tasks across all training algorithms (RHEL and BPPT). The nonlinear HSSM requires approximately 1.5 hours for classification tasks with both algorithms. For the regression tasks, both BPTT and RHEL require approximately between 10 and 20 hours.

RAM consumption remained consistent across all conditions (learning algorithms, datasets, and models), reaching 30GB for regression tasks and 24.15GB for classification (SCP1 dataset). This represents 75% pre-allocation of GPU memory by JAX. While the regression task exceeded the V100's 32GB memory capacity, it fit within the A100's 40GB RAM.

A.4.3 Hyperparameters

We adopted the hyperparameters of [29] without modification, as their experiments utilized the IMEX integrator, which, like our approach, derives from the LeapFrog integrator. The hyperparameters are: learning rate (lr), number of layers (#blocks), number of hidden neurons (hidden dim), state-space dimension (state dim), and whether the time dimension is sent as input (include time). These hyperparameters were found by grid search and are presented in Table 3.

We note that the implementation of the linear HSSM models is based on the code of [29] and uses complex numbers for implementation, which corresponds to doubling the state-space dimension mentioned in Table 3. We found that this dimensional doubling was necessary to recover the performance reported in the original paper, so we maintained this approach. We did not need to apply the same doubling for the nonlinear HSSM.covering the performance reported in the paper, so we kept

Table 3: Hyperparameters for the linear and nonlinear HSSM model

Dataset	lr	hidden dim	state dim	#blocks	include time
Worms	0.0001	64	16	2	False
SCP1	0.0001	64	256	6	False
SCP2	0.00001	64	256	6	True
Ethanol	0.00001	16	256	4	False
Heartbeat	0.00001	64	16	2	True
Motor	0.0001	16	256	6	True
PPG	0.0001	64	16	2	True

For the $RHEL$ algorithm we have two additional hyperparameters: the nudging strength ϵ and the scaling factor γ (see Appx. A.4.4). The nudging strength ϵ was set to 10^{-1} without prior tunning. For the scaling factor γ we did a grid search over the values $\{10^0, 10^4, 10^8, 10^12\}$ for the regression task (PPG-DaLiA) and found that the best performing parameter was 10^4. For the classification tasks, we did a grid search over the values $\{10^0, 10^1, 10^2, 10^4\}$ and found that the best performing scaling was 10^4 for the averaged score.

We employ different initialization schemes for linear and nonlinear HSSM variants. For the linear HSSM model, we follow the same initialization scheme as [29]:

- $\boldsymbol{A} \sim \text{Uniform}(0, 1)$
- $\boldsymbol{B} \sim \text{Uniform}\left(-\frac{1}{\text{hidden\_dim}}, \frac{1}{\text{hidden\_dim}}\right)$
- $\boldsymbol{C} \sim \text{Uniform}\left(-\frac{1}{\text{state\_dim}}, \frac{1}{\text{state\_dim}}\right)$
- $\boldsymbol{D} \sim \mathcal{N}(0, 1)$
- $\boldsymbol{\delta} \sim \text{Uniform}(0, 1)$

For the nonlinear HSSM model, we adopt the initialization scheme from [30]:

- $\boldsymbol{A} \sim \text{Uniform}(0.5, 1)$
- $\boldsymbol{B} \sim \text{Uniform}\left(-\frac{1}{\text{hidden\_dim}}, \frac{1}{\text{hidden\_dim}}\right)$
- $\boldsymbol{C} = 0$
- $\boldsymbol{D} \sim \mathcal{N}(0, 1)$
- $\boldsymbol{b} \sim \mathcal{N}(0, 1)$
- $\alpha \sim \text{Uniform}(0.1, 1.0)$
- $\boldsymbol{\delta} \sim \text{Uniform}(-1, 1)$

A.4.4 Gradient re-scaling for dealing with numerical instabilities

Analysis From the theoretical results on RHEL, the nudging strength ϵ should be chosen as small as possible to accurately estimate the gradient of the loss function. However, naive numerical implementation can encounter underflow issues. In RHEL, gradient information is encoded in small perturbations to the state $\boldsymbol{\Phi}$ that generate the echo trajectory $\boldsymbol{\Phi}^e$. This presents numerical challenges when the perturbations and state values differ by several orders of magnitude.

In practice, when using finite precision representations (such as floating-point numbers), the perturbations may be lost due to discretization error, where small values cannot be accurately represented or distinguished from zero in finite precision arithmetic.

To address this numerical challenge, we employ a simple gradient rescaling method. We multiplicatively scale the output loss $\ell[\cdot, \cdot]$ by a constant $\gamma > 1$ during the echo dynamics computation. This amplification ensures that the perturbation-induced changes in the loss remain within the representable range of floating-point arithmetic. Subsequently, we divide the RHEL parameter gradient estimate $\left(\Delta_{\boldsymbol{\theta}}^{\text{RHEL}}(k, \epsilon)\right)$ by the same scaling factor γ to recover the unbiased gradient.

We now demonstrate the effectiveness of this gradient rescaling method on both Linear and Nonlinear HSSM. To evaluate the effect of gradient scaling, we compare the gradients of RHEL and BPTT. Given a HSSM model, we randomly sample a tuple $(\boldsymbol{u}_i, \boldsymbol{y}_i) \sim \mathcal{D}$ from the SCP1 dataset (see App. A.4.1) and compute the gradients of the loss function with respect to the parameters of the recurrence of the HRUs in Fig. 5A for the linear HSSM and Fig. 6A for the nonlinear HSSM.

To gain a more fine-grained understanding, we also compute the gradients with respect to the inputs of the third layer of the HSSM (see Eq. 17 and 18) in Fig. 5B and Fig. 6B. Compared to the parameter gradients, these input gradients are not time-averaged and hence reveal more detail about the underflow issues.

Gradient rescaling recovers the underflow issues. As a general pattern across both architectures, we observe that the unscaled ($\gamma = 10^0$) parameter and input gradients tend to be biased: they have a norm ratio above 1.0 and cosine similarity below 1.0. We also note that for both the linear and nonlinear cases, there exists a scaling factor (10^6 for linear and 10^4 for nonlinear) that recovers a nearly perfect match between RHEL and BPTT.

Additionally, for the input gradients that are not time-averaged (first column of Fig. 5B and Fig. 6B), there is a large amount of noise in the unscaled gradients, which is eliminated for the best-performing scaling factor (10^6 for linear and 10^4 for nonlinear). We conjecture that this noise is due to the underflow effects described above.

Linear HSSM: scaling improves gradient matching For the linear analysis, we observe a general pattern: the more we scale the RHEL gradient, the better it matches BPTT, both for input gradients and parameter gradients (Fig. 5A and B).

Nonlinear HSSM: optimal scaling balances underflow and linearization errors For the nonlinear analysis, we also observe that scaling the RHEL gradient improves the match between RHEL and BPTT gradients. However, for higher values of the scaling factor ($\gamma = 10^6$), we observe that both input and parameter gradients show a drop in matching quality. We conjecture that this is due to linearization errors. If the loss gradient is too large, the nudging it drives will be large, and the finite difference method used for the learning rule will no longer be valid.

Solution Implemented. For the experiments, we conducted a grid search over the scaling parameter γ. In our initial implementation, we applied gradient scaling without the corresponding downscaling step, effectively amplifying the gradient magnitude throughout training. This provided valuable insights into the sensitivity of RHEL dynamics to gradient scaling and improved performance. The complete rescaling procedure (including both upscaling and downscaling) will be implemented in the camera-ready version to provide a more comprehensive analysis of the numerical stabilization approach.

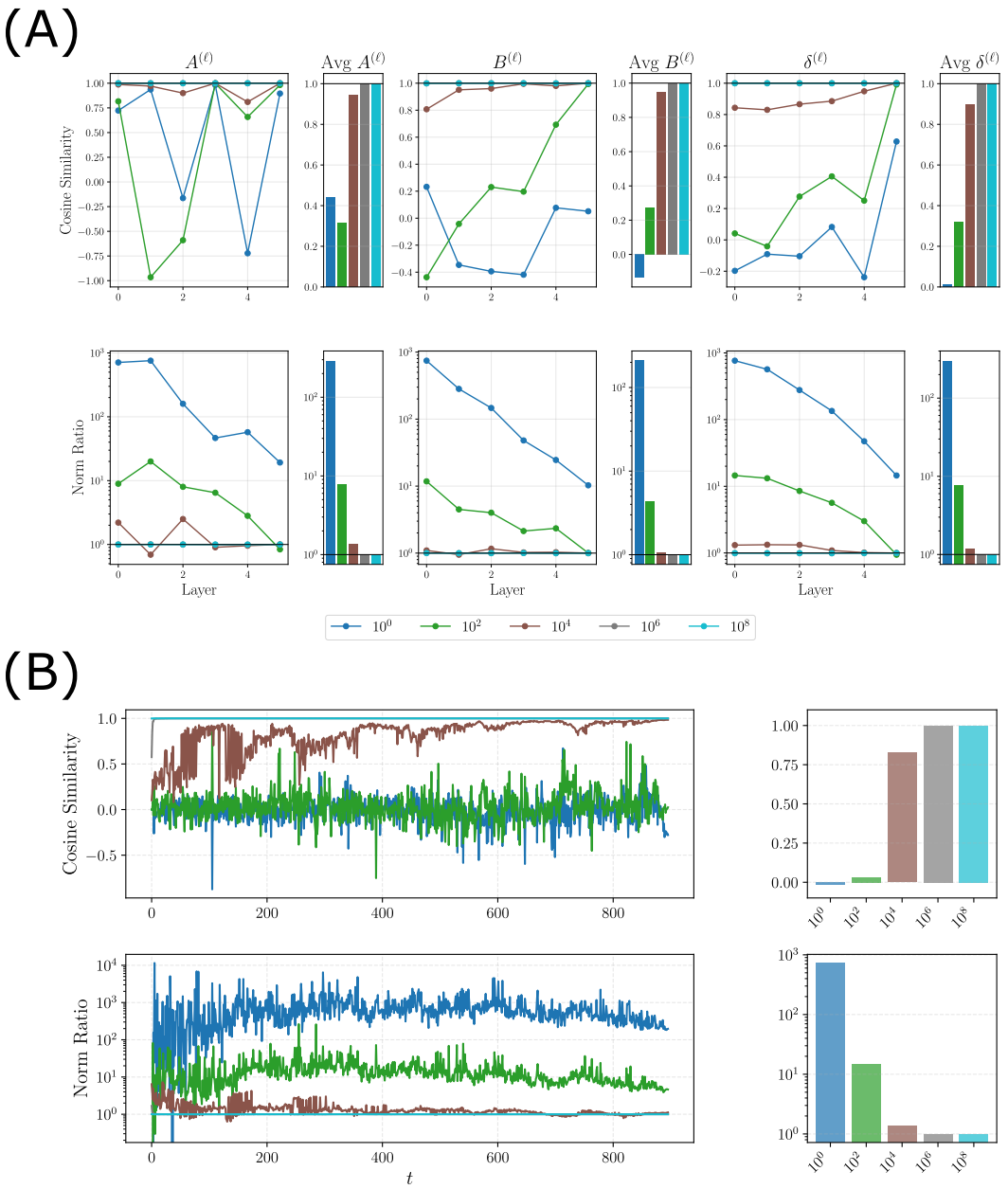

Figure 5: **(A): Parameters gradients comparison between RHEL and BPTT for HSSM.** Given some $(\boldsymbol{u}_i, \boldsymbol{y}_i) \sim \mathcal{D}$, we perform BPTT and RHEL on six blocks-deep linear HSSMs for different scaling factor γ of RHEL (different colors). We measure, per layer (line plot) and when averaged across layers (bar-plot), the cosine similarity (top panels) and norm ratio (bottom panels) between RHEL and BPTT parameters gradients of a linear HSSM (Eq. (17)). **(B): Inputs gradient comparison between RHEL and BPTT for HSSM.** Same setting as (A) but we focus on the gradients with respect to the inputs of the third layer ($\boldsymbol{u}^{(3)}$, see Eq. 17). We measure, per time steps (line plot) and when averaged across time (bar-plot), the cosine similarity (top panels) and norm ratio (bottom panels) between RHEL and BPTT inputs gradients.

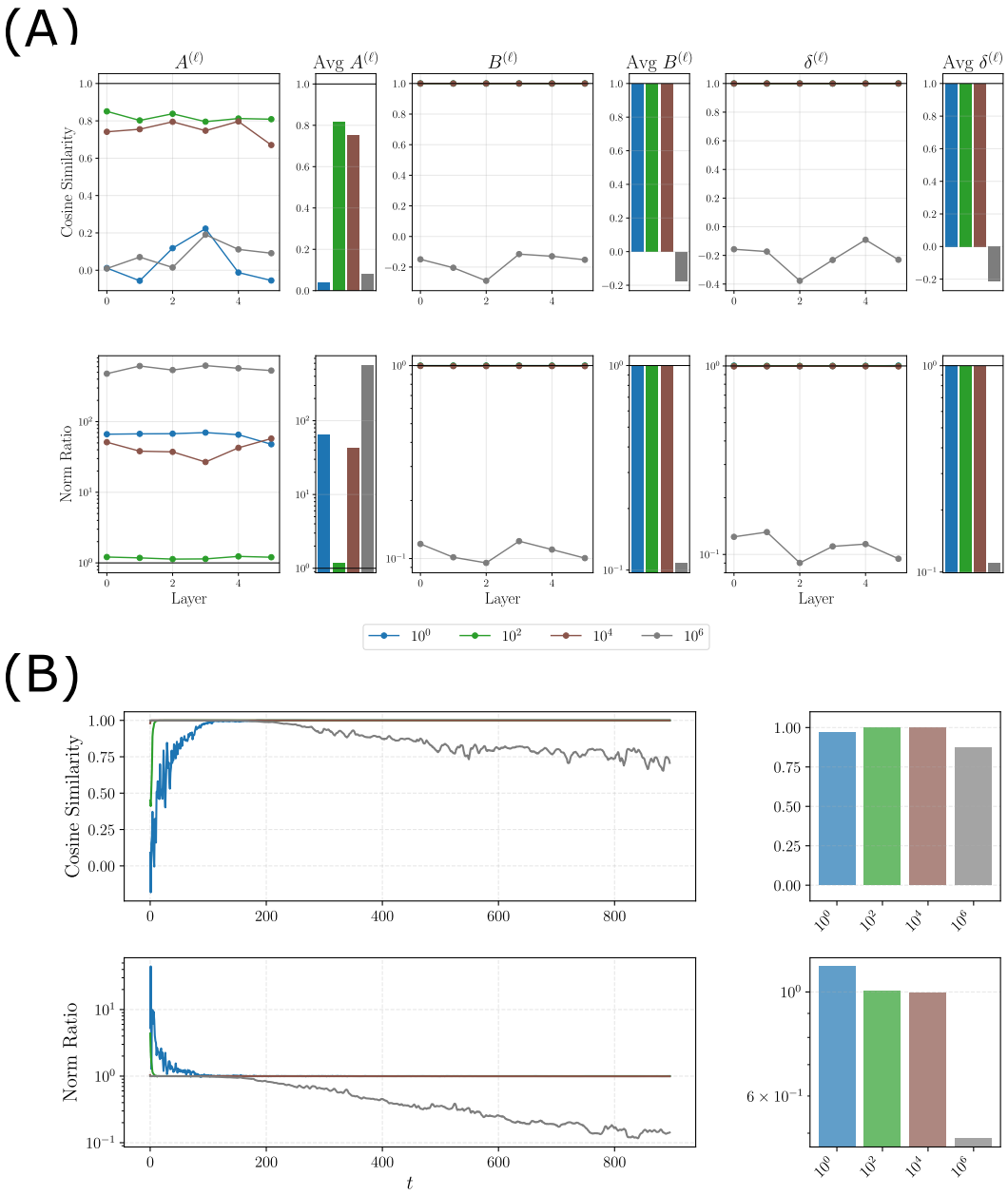

Figure 6: **(A): Parameters gradients comparison between RHEL and BPTT for HSSM.** Given some $(\boldsymbol{u}_i, \boldsymbol{y}_i) \sim \mathcal{D}$, we perform BPTT and RHEL on six blocks-deep nonlinear HSSMs for different scaling factor γ of RHEL (different colors). We measure, per layer (line plot) and when averaged across layers (bar-plot), the cosine similarity (top panels) and norm ratio (bottom panels) between RHEL and BPTT parameters gradients of a nonlinear HSSM (Eq. (18)). **(B): Inputs gradient comparison between RHEL and BPTT for HSSM.** Same setting as (A) but we focus on the gradients with respect to the inputs of the third layer ($\boldsymbol{u}^{(3)}$, see Eq. 18). We measure, per time steps (line plot) and when averaged across time (bar-plot), the cosine similarity (top panels) and norm ratio (bottom panels) between RHEL and BPTT inputs gradients. For (A) and (B), $\gamma = 10^8$ was also tested but produced numerical instabilities leading to NaN values and is therefore omitted from the results.