VARIATIONAL SURROGATE JACOBIANS: A FRAMEWORK FOR CONSTRAINED GRADIENT PROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many learning systems couple neural networks with discrete or constrained optimisation layers, but the resulting solution maps often have zero or undefined derivatives, preventing useful gradient propagation. We introduce the *Variational Surrogate Jacobian (VSJ)* framework, which selects the surrogate Jacobian closest to a prior backward rule under a positive-definite metric and has a unique closed form solution when selected active constraints are enforced exactly. To allow constraint enforcement to respond to the local optimisation state, we also introduce adaptive weighting of inequality constraints, which uses primal slacks and dual multipliers to set the enforcement strength. We show that VSJ recovers established backward rules as particular configurations under method-specific local assumptions, and also corrects those that depend directly on the loss. Empirically, VSJ closely reproduces six of them, while validation-selected novel VSJ configurations yield lower mean test regret for every tested prior family on at least one task. As an example, composing VSJ with PEAR, a state-of-the-art method, lowers mean Knapsack regret and increases mean Portfolio cumulative return. These results establish VSJ as a unified geometric framework for constructing, analysing, and designing gradient propagation rules through constrained optimisation layers.

1 INTRODUCTION

Modern learning systems increasingly embed discrete or constrained optimisation problems as layers (Gould et al., 2016; Lahoud et al., 2025; Wang et al., 2025), where a neural network predicts problem parameters, a solver produces a decision, and a downstream loss evaluates it. Yet solver derivatives can be zero or undefined, leaving no useful gradients for end-to-end training. This challenge arises across non-differentiable task optimisation (McAllester et al., 2010; Song et al., 2016; Donti et al., 2017; Rolínek et al., 2020a; Pinto et al., 2023) and decision-focused learning (Wilder et al., 2019b; Ferber et al., 2020; Mandi et al., 2020; Rolínek et al., 2020b; Paulus et al., 2021).

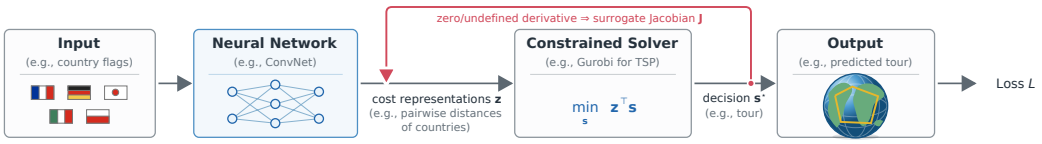
Methods for backpropagating gradients through optimisation layers include identity-style substitutes for degenerate derivatives (Hinton, 2012; Bengio et al., 2013; Sahoo et al., 2023); sensitivities derived by differentiating optimality conditions (Amos & Kolter, 2017; Agrawal et al., 2019b;a; Wilder et al., 2019a; Mandi & Guns, 2020); and loss-dependent updates constructed through surrogate losses, perturbations, or additional solver calls (Berthet et al., 2020; Pogančić et al., 2020; Mulamba et al., 2021; Niepert et al., 2021; Elmachtoub & Grigas, 2022). Lagrangian Proximal Gradient Descent (LPGD) unifies several members of the last family (Paulus et al., 2024). However, existing backward rules typically entangle the propagated update with an implicit, method-specific treatment of local constraints. Thus, arguably, the role of each constraint, its enforcement strength, and the correction geometry are not transparent or independently configurable.

These limitations motivate our central question: *How can we systematically design, geometrically unify, and improve gradient propagation through constrained optimisation layers?*

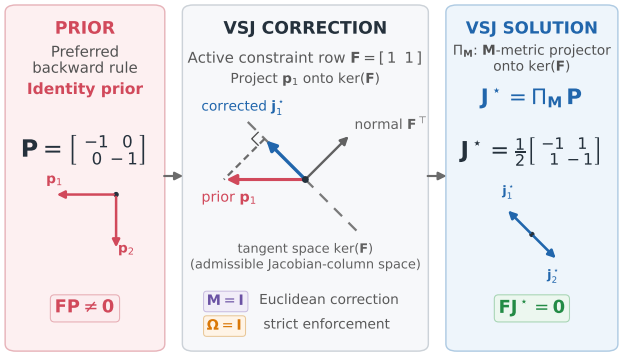
Problem. We consider the computation flow in Fig. 1(a), illustrating the setup on the Travelling Salesman Problem (TSP) (Pogančić et al., 2020): A neural network maps an input (e.g., country flags) to cost representations $\mathbf{z} \in \mathbb{R}^n$ (e.g., pairwise distances), which a constrained solver uses to compute a decision $\mathbf{s}^\star$ (e.g., a tour) by solving:

$$\mathbf{s}^\star \in \arg\min_{\mathbf{s}} \quad \mathbf{z}^\top \mathbf{s} + R(\mathbf{s}) \quad \text{subject to} \quad g(\mathbf{s}) = \mathbf{0}, \quad h(\mathbf{s}) \leq \mathbf{0}, \tag{1}$$

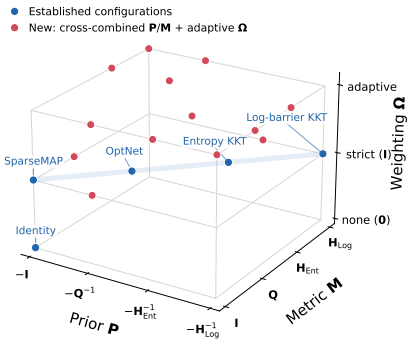
Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Typical computation flow of a neural network that incorporates a constrained solver.



(b) Geometry-guided correction of a prior Jacobian.



(c) VSJ configuration space.

Figure 1: **Overview of VSJ. (a)** Typical computation flow considered in this paper. As the solver's derivatives are zero or undefined, no useful gradient is available. A surrogate Jacobian $\mathbf{J}$ is used during backpropagation. **(b)** VSJ corrects a prior Jacobian $\mathbf{P}$ (the preferred unconstrained behavior) to respect the local constraint Jacobian $\mathbf{F}$ (gradients of the active constraints), whose null space $\ker(\mathbf{F})$ holds the constraint-preserving directions, using a metric $\mathbf{M}$ setting the correction's geometry. Under Euclidean geometry and strict enforcement ($\mathbf{M} = \mathbf{\Omega} = \mathbf{I}$), each column of $\mathbf{P}$ is projected onto $\ker(\mathbf{F})$, keeping its tangential and discarding normal components, giving $\mathbf{F}\mathbf{J}^\star = \mathbf{0}$. **(c)** Varying the prior $\mathbf{P}$, metric $\mathbf{M}$, and weighting $\mathbf{\Omega}$ recovers existing operators (blue) or yields new ones (red).

where $\mathbf{s} \in \mathbb{R}^n$ denotes the decision variables, R is an optional smooth regularizer, and $g : \mathbb{R}^n \to \mathbb{R}^{m_e}$ and $h : \mathbb{R}^n \to \mathbb{R}^{m_i}$ encode equality and inequality constraints independent of $\mathbf{z}$. A downstream loss evaluates $\mathbf{s}^\star$, and end-to-end training requires propagating its gradient through the solver to $\mathbf{z}$ and then to the network parameters. For discrete and unregularized linear problems, the solution map is typically piecewise constant in $\mathbf{z}$, with zero derivatives within regions and undefined derivatives at transitions. Hence, a surrogate Jacobian $\mathbf{J} \in \mathbb{R}^{n \times n}$ supplies the backward operation for training.

Local constraint geometry. At a solution $\mathbf{s}^\star$, let $\mathcal{A}$ index active inequalities (satisfied with equality). Stacking their gradients with the equality-constraint gradients yields the local constraint Jacobian,

$$\mathbf{F} = \begin{bmatrix} \nabla g(\mathbf{s}^\star) \\ \nabla h_{\mathcal{A}}(\mathbf{s}^\star) \end{bmatrix} \in \mathbb{R}^{m \times n}, \qquad m = m_e + |\mathcal{A}|. \tag{2}$$

Rows of $\mathbf{F}$ are local constraint-normal directions. For a small decision perturbation $\delta\mathbf{s}$, $\mathbf{F}\delta\mathbf{s}$ approximates how these constraints change (Nocedal & Wright, 2006). Staying on this constraint surface to first order in $\delta\mathbf{s}$ thus requires $\mathbf{F}\delta\mathbf{s} = \mathbf{0}$. These directions form the tangent space $\ker(\mathbf{F})$ at $\mathbf{s}^\star$.

To translate this geometry to backpropagation, the surrogate Jacobian $\mathbf{J}$ maps cost perturbations $\delta\mathbf{z}$ to decision perturbations $\delta\mathbf{s} = \mathbf{J}\delta\mathbf{z}$. For a downstream loss L, the backward pass uses $\mathbf{J}^\top \frac{\partial L}{\partial \mathbf{s}}$ as the surrogate gradient with respect to $\mathbf{z}$. Substituting $\delta\mathbf{s} = \mathbf{J}\delta\mathbf{z}$ into $\mathbf{F}\delta\mathbf{s} = \mathbf{0}$ shows that preserving this surface to first order for every $\delta\mathbf{z}$ is equivalent to $\mathbf{F}\mathbf{J} = \mathbf{0}$, as we will formalize in Theorem 1. *This characterizes the admissible surrogate Jacobians but does not select one: if* $\ker(\mathbf{F}) \neq \{\mathbf{0}\}$, *infinitely many Jacobians satisfy it.*

Variational Surrogate Jacobians (VSJ). Our proposed framework VSJ resolves this ambiguity by turning $\mathbf{F}\mathbf{J} = \mathbf{0}$ into a principled selection rule. Starting from a prior surrogate Jacobian $\mathbf{P}$ encoding the preferred unconstrained backward behavior, it selects the admissible Jacobian $\mathbf{J}^\star$ closest to $\mathbf{P}$ under a positive-definite metric $\mathbf{M}$. The construction is therefore variational: it defines $\mathbf{J}^\star$ as the optimizer of a metric criterion over the admissible Jacobians. When all rows of $\mathbf{F}$ are enforced strictly, the unique solution projects each column of $\mathbf{P}$ onto $\ker(\mathbf{F})$ in the $\mathbf{M}$ geometry, keeping its tangent component while removing its $\mathbf{M}$-orthogonal component. Fig. 1(b) depicts this correction.

Beyond this strict construction, VSJ exposes a configuration space by separating three choices that established backward rules typically couple. The prior $\mathbf{P}$ specifies the preferred unconstrained behavior, the metric $\mathbf{M}$ determines the correction geometry, and the row weighting $\mathbf{\Omega}$ controls constraint enforcement. We show that *particular VSJ configurations recover established operators, while recombining these components defines new ones.* For adaptive enforcement, we extend the row set to candidate inequalities and make $\mathbf{\Omega}$ state-dependent. At exact Karush–Kuhn–Tucker (KKT) states, inequalities with larger positive dual multipliers are enforced more strongly, while zero-dual inequalities are omitted; for approximate primal-dual states, positive slack further weakens enforcement. The correction also applies directly to loss-dependent decision-space updates without requiring an explicit prior Jacobian. Fig. 1(c) illustrates the resulting design space.

Contributions. Our main contributions can be summarised as:

- We introduce Variational Surrogate Jacobian (VSJ), which makes the prior, metric, and constraint weighting independently configurable. We derive closed-form solutions for strict and weighted enforcement, and introduce adaptive weighting based on primal slacks and dual multipliers.
- We show that VSJ is a unifying framework: six established backward rules (Identity Update, SparseMAP, OptNet, entropy-regularized and log-barrier KKT differentiations as well as PEAR) are recovered as particular VSJ configurations $(\mathbf{P}, \mathbf{M}, \mathbf{\Omega})$ under their local assumptions. VSJ also composes with loss-dependent decision-space updates, which do not admit an explicit Jacobian such as SPO+ and blackbox differentiation, and yields their geometry-aware variants.
- We explore VSJ's configuration space $(\mathbf{P}, \mathbf{M}, \mathbf{\Omega})$ and uncover new backward rules, outperforming established ones. For example, (i) a validation-selected VSJ configuration attains the lowest mean test regret on travelling salesman problem among Jacobian-based backward rules, and (ii) composing VSJ with PEAR (Lee et al., 2026) reduces mean binary knapsack regret by 13.8%.

2 Related Work

Jacobian-Based Backward Rules. A family of backward rules propagates the gradient through a surrogate Jacobian. Straight-through and identity-based variants bypass uninformative derivatives by assuming a simple relation between changes in predicted costs and decisions, such as treating the solver as an identity map during backpropagation (Hinton, 2012; Bengio et al., 2013; Oksuz et al., 2021; Sahoo et al., 2023). Differentiable optimisation layers instead differentiate optimality conditions to determine how decisions change with predicted costs, often by regularizing the problem (Amos & Kolter, 2017; Niculae et al., 2018; Agrawal et al., 2019b;a; Mandi & Guns, 2020; Pan et al., 2024; Magoon et al., 2025). In the classical projected form (Domke, 2008), the same local curvature determines both the unconstrained response and the projection metric. PEAR (Lee et al., 2026) uses this approach to derive regret gradients by rescaling cost-prediction errors by inverse curvature and projecting them onto directions that preserve active constraints. dXPP (Linghu et al., 2026) computes approximate QP Jacobians using objective curvature and smooth constraint penalties, with one multiplier-based penalty strength for equalities and another for inequalities. VSJ instead keeps equalities exact and sets each inequality's strength from its own multiplier and slack, recovers these Jacobian-based operators as specific configurations under local assumptions and allows the prior Jacobian, correction metric, and each constraint's strength to be chosen independently.

Loss-Dependent Updates. Loss-dependent backward rules construct learning updates through surrogate losses, perturbations, or additional solver calls, including SPO+ (Elmachtoub & Grigas, 2022), blackbox differentiation (Pogančić et al., 2020), Fenchel-Young losses (Berthet et al., 2020; Blondel et al., 2020), IMLE (Niepert et al., 2021), and CachingPO (Mulamba et al., 2021). These updates do not need to be represented by a loss-independent Jacobian. LPGD (Paulus et al., 2024) mainly unifies several of them through Lagrange-Moreau envelopes and perturbed solver calls. VSJ addresses a complementary question: how to correct a loss-dependent update considering local constraint geometry, thereby composing with these rules without replacing their native updates.

3 The Variational Surrogate Jacobian (VSJ) Framework

Here, we first characterize Jacobians that preserve the selected active manifold, then select the one closest to a prior under a chosen metric, and finally introduce adaptive constraint weighting. These steps define the three components of a VSJ configuration: the prior $\mathbf{P}$, metric $\mathbf{M}$, and weighting $\mathbf{\Omega}$.

3.1 Characterizing Active-Manifold-Preserving Jacobians

During backpropagation, $\mathbf{J}$ maps a cost perturbation to $\delta\mathbf{s} = \mathbf{J}\delta\mathbf{z}$. Since preserving the selected active manifold requires $\mathbf{F}\delta\mathbf{s} = \mathbf{0}$, preservation for every $\delta\mathbf{z}$ yields the following characterization.

Theorem 1 (First-Order Active-Manifold Preservation)**.** *Let* $\mathbf{F}$ *stack* $\nabla g(\mathbf{s}^\star)$ *and* $\nabla h_{\mathcal{A}}(\mathbf{s}^\star)$*. A surrogate Jacobian* $\mathbf{J}$ *preserves this active manifold to first order for every cost perturbation if and only if* $\mathbf{FJ} = \mathbf{0}$*. See Sec. B for the proof.*

Equivalently, every column of $\mathbf{J}$ is in the tangent space $\ker(\mathbf{F})$. Unless $\ker(\mathbf{F}) = \{\mathbf{0}\}$, this admits infinitely many Jacobians; VSJ resolves ambiguity by selecting the one closest to a chosen prior $\mathbf{P}$.

3.2 Strict VSJ

Existing backward rules encode preferred responses to cost perturbations. VSJ represents this behavior by a prior surrogate Jacobian $\mathbf{P} \in \mathbb{R}^{n\times n}$, the preferred unconstrained response before active-manifold preservation. For example, for shortest-path cost minimization, denoting the identity matrix by $\mathbf{I}$, the identity-style prior $\mathbf{P} = -\mathbf{I}$ assumes that increasing an edge cost decreases its corresponding decision variable without directly affecting the others (Sahoo et al., 2023).

However, because $\mathbf{P}$ does not need to satisfy $\mathbf{FP} = \mathbf{0}$, it may produce perturbations outside $\ker(\mathbf{F})$. Rather than discarding the prior, we seek an admissible surrogate Jacobian that deviates from it as little as possible. This leads naturally to our proposed VSJ framework.

Definition 2 (*Variational Surrogate Jacobian (VSJ)* framework)**.** VSJ framework formulates surrogate Jacobian design as a constrained variational optimisation problem

$$\boxed{\mathbf{J}^\star = \arg\min_{\mathbf{J}} \frac{1}{2}\operatorname{tr}\big((\mathbf{J}-\mathbf{P})^\top \mathbf{M}(\mathbf{J}-\mathbf{P})\big) \quad \text{subject to} \quad \mathbf{FJ} = \mathbf{0},} \quad \text{(VSJ defn.)} \tag{3}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ with $\mathbf{M} \succ 0$ defines the metric used to measure deviations from the prior.

Since the objective is strictly convex and the constraints are linear, Eq. (3) has a unique solution (Boyd & Vandenberghe, 2004).

Theorem 3 (Closed-Form Solution of VSJ)**.** *Suppose* $\mathbf{M} \succ 0$*. For any constraint Jacobian* $\mathbf{F}$*, including one with linearly dependent rows, Eq.* (3) *has the unique solution*

$$\boxed{\mathbf{J}^\star = \underbrace{\left(\mathbf{I} - \mathbf{M}^{-1}\mathbf{F}^\top\left(\mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top\right)^\dagger \mathbf{F}\right)}_{\Pi_{\mathbf{M}}}\mathbf{P},} \quad \textit{(VSJ soln.)} \tag{4}$$

where † *denotes the Moore–Penrose pseudoinverse. If* $\mathbf{F}$ *has full row rank, the pseudoinverse reduces to the ordinary inverse. See Sec. C for the proof.*

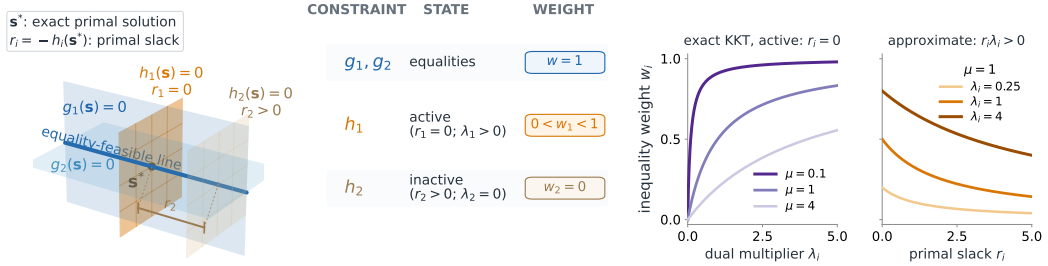
The closed-form solution separates the prior $\mathbf{P}$ from a constraint-dependent linear operator, characterized geometrically next.

Proposition 4 (Geometric Interpretation)**.** *The operator* $\Pi_{\mathbf{M}}$ *in Eq.* (4) *is the* $\mathbf{M}$*-orthogonal projector onto* $\ker(\mathbf{F})$*. See Sec. D for the proof.*

Thus, each column of $\mathbf{P}$ is projected onto $\ker(\mathbf{F})$ (see also Fig. 1b): its tangent component is retained and its $\mathbf{M}$-orthogonal normal component is removed. Consequently, VSJ applies the minimum metric correction required to preserve the selected active manifold to first order; $\mathbf{M} = \mathbf{I}$ gives Euclidean projection. Sec. 3.3 extends this strict projection through adaptive constraint weighting.

3.3 Adaptive VSJ

Although an inequality can be satisfied without holding at equality, strict VSJ treats selected active inequalities as equalities in its linearized backward response. As a toy example, at $s^\star = 0$ under $0 \le s \le 1$, strict preservation of the active lower bound requires $\delta s = 0$, giving a zero derivative even though small positive perturbations are feasible. To address this, we introduce Adaptive VSJ, which softens the enforcement of the inequality constraints using their dual multipliers as the indicators of the local influence on the optimal objective.



(a) Constraint states and their weights.

(b) Slack, temperature, and multiplier.

Figure 2: **Adaptive constraint weighting.** (a) Equalities remain strict ($w = 1$); at exact KKT states, an active inequality with $\lambda_i > 0$ has $0 < w_i < 1$, whereas a zero-dual inequality has $w_i = 0$. (b) For $w_i = \lambda_i/(\lambda_i + r_i + \mu)$, larger dual multipliers strengthen enforcement and larger μ weakens it; away from exact complementarity, positive slack further attenuates the weight.

For an inequality $h_k(\mathbf{s}) \leq 0$, let $r_k = -h_k(\mathbf{s}) \geq 0$ and $\lambda_k \geq 0$ denote its normalized primal slack and canonical KKT multiplier from the local primal-dual state; their construction and scale normalization are detailed in Sec. E. We define

$$w_k = \begin{cases} 1, & \text{if row } k \text{ is an equality,} \\ \dfrac{\lambda_k}{\lambda_k + r_k + \mu}, & \text{if row } k \text{ is an inequality,} \end{cases} \qquad \mathbf{\Omega} = \operatorname{diag}(w_1, \ldots, w_m), \tag{5}$$

where $\mu > 0$ governs inequality relaxation: larger μ suppresses constraints with weak dual support.

At an exact KKT point, complementary slackness gives $r_k \lambda_k = 0$. Hence, a positive-dual inequality has $r_k = 0$ and $w_k = \lambda_k/(\lambda_k + \mu)$, whereas any zero-dual inequality has $w_k = 0$. The general rule also accommodates approximate primal-dual states, such as solutions returned within solver tolerance, for which r_k and λ_k may both be positive, and slack then attenuates enforcement. These cases are shown in Fig. 2.

The condition $\mathbf{\Omega F J} = \mathbf{0}$ would not provide soft enforcement because every row with $w_k > 0$ would still satisfy $\mathbf{f}_k^\top \mathbf{J} = \mathbf{0}$ exactly. We therefore map each weight to a constraint stiffness

$$\gamma_k = \frac{w_k}{1 - w_k}, \qquad \mathbf{\Gamma} = \operatorname{diag}(\gamma_1, \ldots, \gamma_m). \tag{6}$$

A zero weight gives $\gamma_k = 0$, while $w_k \to 1$ gives $\gamma_k \to \infty$. Thus γ_k controls the penalty for violating first-order constraint preservation. For adaptive inequality weights in Eq. (5), $\gamma_k = \lambda_k/(r_k + \mu)$, so stronger dual support increases enforcement, while slack and temperature weaken it.

For the finite-stiffness derivation, suppose every retained row satisfies $0 < w_k < 1$; zero-weight rows are omitted and unit-weight rows are imposed separately as exact constraints. The weighted variational problem is

$$\mathbf{J}^\star = \arg\min_{\mathbf{J}} \frac{1}{2} \operatorname{tr}\left((\mathbf{J} - \mathbf{P})^\top \mathbf{M} (\mathbf{J} - \mathbf{P})\right) + \frac{1}{2} \operatorname{tr}\left(\mathbf{J}^\top \mathbf{F}^\top \mathbf{\Gamma F J}\right), \tag{7}$$

which is strictly convex because $\mathbf{M} \succ 0$ and has the unique solution

$$\mathbf{J}^\star = \left(\mathbf{M} + \mathbf{F}^\top \mathbf{\Gamma F}\right)^{-1} \mathbf{M P}. \tag{8}$$

The complete mixed strict-soft formulation omits zero-weight rows, penalizes finite-weight rows, and imposes unit-weight rows, including equalities, as exact constraints. Its Schur-complement solution is derived in Sec. E. If all selected rows are made strict, the solution converges to the strict projection $\Pi_{\mathbf{M}} \mathbf{P}$, whereas it reduces to the prior $\mathbf{P}$ if all rows are omitted.

4 The VSJ Configuration Space

In this section, we derive existing backward rules in the VSJ configuration space (Sec. 4.1) and extend VSJ to loss-dependent updates (Sec. 4.2). Tab. 1 summarizes both tracks.

Table 1: Representative backward operators derived as VSJ configurations $(\mathbf{P}, \mathbf{M}, \mathbf{\Omega})$. Loss-dependent backward rules provide native decision-space updates corrected through $(\mathbf{M}, \mathbf{\Omega})$.

Method	Prior P	Metric M	Weighting Ω
Identity Update (Sahoo et al., 2023)	$-\mathbf{I}$	$\mathbf{I}$	$\mathbf{0}$
SparseMAP (Niculae et al., 2018)	$-\tau^{-1}\mathbf{I}$	$\tau\mathbf{I}$	$\mathbf{I}$
OptNet (Amos & Kolter, 2017)	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{I}$
Entropy-KKT (Agrawal et al., 2019a)	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{I}$
Log-barrier KKT (Mandi & Guns, 2020)	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{I}$
PEAR (quadratic base) (Lee et al., 2026)	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{I}$
LPGD family (Berthet et al., 2020; Pogančić et al., 2020; Elmachtoub & Grigas, 2022; Paulus et al., 2024)	*Native update corrected by VSJ*		

4.1 Existing Methods as VSJ Configurations

A VSJ configuration $(\mathbf{P}, \mathbf{M}, \mathbf{\Omega})$ defines preferred unconstrained behavior, correction geometry, and constraint enforcement. The recovered rules range from identity propagation to curvature-induced sensitivities. Under the minimization-cost convention, the *Identity Update* uses $\mathbf{J}_{\text{Id}} = -\mathbf{I}$, passing the negated decision gradient directly to cost space. VSJ recovers it with $\mathbf{P} = -\mathbf{I}$ and $\mathbf{\Omega} = \mathbf{0}$: disabling feasibility correction gives $\mathbf{J}^\star = \mathbf{P} = -\mathbf{I}$. Because nothing is projected, $\mathbf{M}$ is immaterial; choosing $\mathbf{M} = \mathbf{I}$ gives $(-\mathbf{I}, \mathbf{I}, \mathbf{0})$ as the recovery.

Regularizer-induced backward rules differentiate the solution of Eq. (1) for a smooth regularizer R. On a fixed affine active manifold with $\mathbf{H}_R = \nabla^2 R(\mathbf{s}^\star) \succ \mathbf{0}$, the cost Jacobian is the classical equality-constrained sensitivity (Domke, 2008), derived in Sec. F:

$$\mathbf{J}_R = -\mathbf{H}_R^{-1} + \mathbf{H}_R^{-1}\mathbf{F}^\top \left(\mathbf{F}\mathbf{H}_R^{-1}\mathbf{F}^\top\right)^\dagger \mathbf{F}\mathbf{H}_R^{-1}. \tag{9}$$

VSJ recovers Eq. (9) by choosing:

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{H}_R^{-1}, \mathbf{H}_R, \mathbf{I}) \quad \Longrightarrow \quad \mathbf{J}^\star = \mathbf{J}_R, \tag{10}$$

where $\mathbf{P}$ is the unconstrained response, $\mathbf{M}$ defines the correction geometry, and $\mathbf{\Omega} = \mathbf{I}$ enforces every selected constraint strictly. Eq. (10) yields each of them through its curvature: $\mathbf{H}_R = \tau\mathbf{I}$ recovers the SparseMAP Jacobian with isotropic curvature and uniform unconstrained sensitivity, while $\mathbf{H}_R = \mathbf{Q}_\theta$ recovers OptNet's linear-cost Jacobian with anisotropic, potentially learned curvature. Entropy and log-barrier KKT Jacobians are recovered with state-dependent Hessians $\mathbf{H}_{\text{Ent}} = \epsilon\,\text{diag}(1/s_i^\star)$ and $\mathbf{H}_{\text{Log}} = \text{diag}(\mu_{\text{Log}}/(s_i^\star)^2)$. For matching curvature and active-face geometry, PEAR's base sensitivities for smoothed LPs and QPs coincide with the SparseMAP and OptNet configurations, respectively. PEAR computes its regularized-regret gradient, $\mathbf{g}_{\text{PEAR}} = -(\mathbf{J}^\star)^\top\mathbf{e}$ where $\mathbf{e} = \mathbf{z} - \mathbf{c}$ is the cost-vector prediction error and $\mathbf{c}$ is the true cost vector. These are local operator recoveries, not equivalences of full training procedures. Please see Sec. F for derivations.

4.2 Extension to Loss-Dependent Updates

Not all backward rules construct an explicit surrogate Jacobian. SPO+ and blackbox differentiation generate loss-dependent decision-space updates from differences of solver outputs. The LPGD framework (Paulus et al., 2024) includes these, Direct Loss Minimization, implicit differentiation by perturbation, and Fenchel-Young losses as special or limiting cases. VSJ can also directly correct such updates. For a native decision-space update $\mathbf{g} \in \mathbb{R}^n$ and finite stiffnesses, its VSJ correction is

$$\mathbf{g}^\star = \arg\min_{\mathbf{u}} \left\{ \frac{1}{2}(\mathbf{u}-\mathbf{g})^\top\mathbf{M}(\mathbf{u}-\mathbf{g}) + \frac{1}{2}\mathbf{u}^\top\mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\mathbf{u} \right\} = \left(\mathbf{M} + \mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\right)^{-1}\mathbf{M}\mathbf{g}. \tag{11}$$

Zero-weight rows are omitted. Unit-weight rows, including equalities under adaptive weighting, are imposed as exact constraints using the mixed formulation in Sec. E. Thus $\mathbf{\Omega} = \mathbf{0}$ returns $\mathbf{g}$, while $\mathbf{\Omega} \to \mathbf{I}$ gives $\Pi_{\mathbf{M}}\mathbf{g}$. If $\mathbf{F}\mathbf{g} = \mathbf{0}$, then $\mathbf{u} = \mathbf{g}$ is feasible and makes both objective terms zero, so $\mathbf{g}^\star = \mathbf{g}$ for any admissible $\mathbf{M}$ and $\mathbf{\Omega}$. Differences of feasible decisions preserve shared affine equalities but may violate selected active inequalities or bounds. VSJ controls their enforcement through $\mathbf{\Omega}$ and the correction geometry through $\mathbf{M}$.

5 Experimental Evaluation

We evaluate (1) how well VSJ recovers established backward rules; (2) whether new VSJ configurations can be effective, and if so whether we can highlight a practical recipe for exploring the configuration space; and (3) whether the recipe generalises to the state-of-the-art approach PEAR.

Table 2: **Benchmark selection.** Different benchmarks probe complementary constraint geometries and support comparisons with established backward rules. Arrows indicate the preferred metric direction. Formulation and evaluation protocols are included in Secs. G.1 and G.2, respectively.

Benchmark	Input, predictor, and task	Benchmark Selection Rationale	Metrics
Travelling Salesman Problem (TSP) *Synthetic data*	**Input:** *Vision*; 40 flag images. **Predictor:** ConvNet for 3D coordinates. **Task:** min.-cost tour from pairwise distances.	NP-hard integer optimisation with generated subtour inequalities tests adaptive constraint weighting in a discrete decision pipeline.	Relative regret ↓
Knapsack *Fractional / binary* *Synthetic data*	**Input:** *Tabular*; 48/100 items. **Predictor:** linear $8 \to 1$ / $5 \to 100$. **Task:** fractional / binary packing.	Capacity and bound inequalities test adaptive weighting for continuous and binary decisions. Binary variant allows comparing with PEAR.	Relative regret ↓
Portfolio Optimisation *Synthetic data*	**Input:** *Tabular*; five asset features. **Predictor:** linear $5 \to 50$ map. **Task:** budget- and risk-constrained allocation.	A nonlinear quadratic-risk constraint tests whether VSJ remains useful beyond affine constraint geometry.	Absolute regret ↓
Time-Series Portfolio *Real-world data*	**Input:** *Time series*; 63-day histories, 89 assets. **Predictor:** DLinear, 21-day return forecasts. **Task:** Allocate investments across S&P assets.	mean-variance task (Hwang et al., 2025) used by PEAR (Lee et al., 2026), enabling comparison with it under time-varying covariance geometry.	Norm. regret ↓; returns ↑; Sharpe ↑; signed MDD ↑

Table 3: **Consistency of matched VSJ configurations on fractional Knapsack, as an example task.** The matched VSJ configuration of each native backward rule (Tab. 1) closely recovers its test regret. The full comparison in Tab. 9 supports the same observation across benchmarks.

Implementation	Identity Update	SparseMAP	OptNet	Entropy-KKT	Log-barrier KKT
Native	0.1634 ± 0.0050	0.0534 ± 0.0099	0.0745 ± 0.0160	0.0503 ± 0.0037	0.1537 ± 0.0849
Matched VSJ	0.1634 ± 0.0050	0.0534 ± 0.0099	0.0721 ± 0.0153	0.0503 ± 0.0037	0.1494 ± 0.0868

5.1 Experimental Setup

Benchmarks. We employ five commonly-used benchmarks summarized in Tab. 2. The benchmarks include continuous and discrete decisions, and affine and nonlinear constraint geometries. TSP (Pogančić et al., 2020), fractional Knapsack (Mandi et al., 2024), and synthetic Portfolio (Mandi et al., 2024) probe generated inequalities, capacity and bound activity, and quadratic risk constraints, respectively. Binary Knapsack (Lee et al., 2026) tests continuous backward surrogates for discrete decisions and time-series Portfolio (Hwang et al., 2025) extends the evaluation to temporal prediction in a real-world dataset. Formulations of all benchmarks are provided in Sec. G.1.

Evaluation. For each benchmark and method, we use the same data splits, predictors, optimizers, training schedules and seeds. We select VSJ configurations based on validation performance and report held-out test mean $\pm$ standard deviation over five seeds unless otherwise explicitly stated. Tab. 2 includes the metrics for each benchmark, and the details of the metrics are included in Sec. G.2.

5.2 Recovery Fidelity of VSJ

To verify the theoretical recoveries numerically, we compare five explicit-Jacobian backward rules (Identity, SparseMAP, OptNet, Entropy-KKT, and Log-barrier KKT) with their matched VSJ configurations. We test both end-to-end training behavior and the stronger requirement that each native operator match its VSJ counterpart at a fixed state (problem data, cost, solution, and local geometry). Replacing each native backward operator with its VSJ counterpart yields overlapping test-regret mean $\pm$ standard-deviation intervals for every reported pair in Tab. 3 (and in Tab. 9), supporting consistent training behavior. Controlled fixtures and a deterministic sweep across 39 states further yield maximum spectral and relative spectral Jacobian errors of 6.51×10^{-8} and 7.32×10^{-8}, respectively (details in Sec. G.5). *Together, these results support the theoretical recoveries through high-precision operator agreement and end-to-end consistency.*

5.3 Exploring the VSJ Configuration Space

Explicit Jacobians. Established backward rules implicitly couple a prior, metric, and constraint weighting as a fixed update, while VSJ exposes them as independent design axes as its configurations. We now explore their recombinations together with different constraint enforcement strategies: no correction ($\mathbf{\Omega} = \mathbf{0}$), strict correction ($\mathbf{\Omega} = \mathbf{I}$), or adaptive weighting ($\mathbf{\Omega} = \mathbf{\Omega}_\mu$). Explored VSJ configurations on validation set are detailed in Sec. G.11. For each prior, Tab. 4 compares its native update, bare prior as no correction, and validation-selected VSJ on the test set. On the NP-hard TSP, the selected VSJ configurations lower mean test regret relative to all four native methods. For example, the OptNet-prior reduces regret from 0.2619 to 0.0992, the lowest TSP mean. On Knapsack, validation selects adaptive weighting for all four prior families, consistent with the ben-

Table 4: **Exploring new VSJ configurations across prior Jacobians corresponding to existing backward rules.** For each prior $\mathbf{P}$, a VSJ configuration is selected by mean validation regret, among different metric-weighting combinations detailed in Sec. G.11. Bare form uses the uncorrected prior ($\mathbf{\Omega} = \mathbf{0}$, $\mathbf{J}^\star = \mathbf{P}$), thus SparseMAP's bare form is Identity Update (marked by *). Test mean regret $\pm$ SD over five seeds. Δ_{native} = selected VSJ $-$ native. Green: gain, red: worse, bold: best mean.

Benchmark	Native method (prior $\mathbf{P}$)	Native	Bare prior	Validation-selected VSJ	Selected $(\mathbf{M}, \mathbf{\Omega})$	Δ_{native}
TSP	SparseMAP $(-\mathbf{I})$	0.1676 ± 0.0041	$0.2675 \pm 0.1313^*$	0.1157 ± 0.0191	$(\mathbf{I}, \mathbf{\Omega}_{10^{-2}})$	-0.0519
	OptNet $(-\mathbf{Q}_\theta^{-1})$	0.2619 ± 0.0738	0.1853 ± 0.0118	$\mathbf{0.0992} \pm 0.0067$	$(\mathbf{Q}_\theta, \mathbf{\Omega}_{10^{-2}})$	-0.1627
	Entropy-KKT $(-\mathbf{H}_{\text{Ent}}^{-1})$	0.1871 ± 0.0134	0.1727 ± 0.0502	0.1727 ± 0.0502	$\mathbf{\Omega} = \mathbf{0}$	-0.0144
	Log-barrier KKT $(-\mathbf{H}_{\text{Log}}^{-1})$	0.1970 ± 0.0117	0.3076 ± 0.1614	0.1146 ± 0.0324	$(\mathbf{Q}_\theta, \mathbf{I})$	-0.0824
Fractional Knapsack	SparseMAP $(-\mathbf{I})$	0.0534 ± 0.0099	$0.1634 \pm 0.0050^*$	0.0535 ± 0.0049	$(\mathbf{Q}_\theta, \mathbf{\Omega}_{10^{-3}})$	$+0.0001$
	OptNet $(-\mathbf{Q}_\theta^{-1})$	0.0745 ± 0.0160	0.4591 ± 0.0181	0.0540 ± 0.0021	$(\mathbf{Q}_\theta, \mathbf{\Omega}_{10^{-1}})$	-0.0205
	Entropy-KKT $(-\mathbf{H}_{\text{Ent}}^{-1})$	0.0503 ± 0.0037	0.1737 ± 0.0047	$\mathbf{0.0501} \pm 0.0034$	$(\mathbf{H}_{\text{Ent}}, \mathbf{\Omega}_{10^{-2}})$	-0.0002
	Log-barrier KKT $(-\mathbf{H}_{\text{Log}}^{-1})$	0.1537 ± 0.0849	0.1654 ± 0.0111	0.0527 ± 0.0058	$(\mathbf{H}_{\text{Log}}, \mathbf{\Omega}_{10^{-1}})$	-0.1010
Portfolio Optimisation	SparseMAP $(-\mathbf{I})$	$\mathbf{0.0007} \pm 0.0000$	$0.0204 \pm 0.0001^*$	$\mathbf{0.0007} \pm 0.0000$	$(\tau\mathbf{I}, \mathbf{I})$	0.0000
	OptNet $(-\mathbf{Q}_\theta^{-1})$	$\mathbf{0.0007} \pm 0.0000$	0.0495 ± 0.0068	0.0011 ± 0.0008	$(\mathbf{Q}_\theta, \mathbf{I})$	$+0.0004$
	Entropy-KKT $(-\mathbf{H}_{\text{Ent}}^{-1})$	0.0575 ± 0.0010	$\mathbf{0.0007} \pm 0.0000$	$\mathbf{0.0007} \pm 0.0000$	$\mathbf{\Omega} = \mathbf{0}$	-0.0568
	Log-barrier KKT $(-\mathbf{H}_{\text{Log}}^{-1})$	0.0936 ± 0.0041	0.0779 ± 0.0029	0.0779 ± 0.0029	$\mathbf{\Omega} = \mathbf{0}$	-0.0157

Table 5: **Exploring VSJ corrections to loss-dependent updates** (mean regret). The conventions of Tab. 4 apply. Similar to prior Jacobians, VSJ can improve loss dependent updates.

Benchmark	Native Update	Native baseline	Validation-selected VSJ	Selected $(\mathbf{M}, \mathbf{\Omega})$	Δ_{native}
TSP	SPO+	$\mathbf{0.0000} \pm 0.0000$	$\mathbf{0.0000} \pm 0.0000$	$(\mathbf{I}, \mathbf{\Omega}_{10^{-2}})$	0.0000
	Blackbox	$\mathbf{0.0000} \pm 0.0000$	$\mathbf{0.0000} \pm 0.0000$	$(\mathbf{I}, \mathbf{\Omega}_1)$	0.0000
Fractional Knapsack	SPO+	0.0732 ± 0.0453	$\mathbf{0.0491} \pm 0.0035$	$(\mathbf{I}, \mathbf{\Omega}_{10^{-1}})$	-0.0241
	Blackbox	0.0496 ± 0.0031	0.0497 ± 0.0030	$(\mathbf{I}, \mathbf{\Omega}_1)$	$+0.0001$
Portfolio Optimisation	SPO+	0.0013 ± 0.0000	$\mathbf{0.0008} \pm 0.0000$	$(\mathbf{I}, \mathbf{\Omega}_{10^{-2}})$	-0.0005
	Blackbox	0.1805 ± 0.0000	0.1045 ± 0.0094	$(\mathbf{I}, \mathbf{I})$	-0.0760

efit of retaining update directions that change which capacity or bound constraints are tight. VSJ lowers OptNet from 0.0745 to 0.0540 and Log-barrier KKT from 0.1537 to 0.0527. On Portfolio, VSJ lowers Entropy-KKT from 0.0575 to 0.0007, showing empirical utility beyond affine geometry. *Thus, every explicit prior family improves on its native coupling in mean test regret on at least one benchmark, and a VSJ configuration attains the lowest displayed mean on each benchmark.*

Loss-dependent updates. We use SPO+ and Blackbox as representative backward rules with distinct perturbation mechanisms: cost reflection for SPO+ and loss-gradient probing for Blackbox. As discussed in Sec. 4.2, differences of feasible decisions preserve shared affine equalities, but finite perturbations can change which inequalities are tight. Hence, the resulting update need not be tangent to the current active manifold, allowing VSJ to moderate its constraint-normal components. To investigate the benefit of this mechanism, we explore $\mathbf{M}$ and $\mathbf{\Omega}$ (details in Sec. G.11). In Tab. 5, adaptive VSJ lowers SPO+ mean regret from 0.0732 to 0.0491 on Knapsack. On TSP, native SPO+ and Blackbox attain zero regret on their uncollapsed seeds, which their selected VSJ settings retain (Sec. G.8). On Portfolio, strict VSJ lowers Blackbox from 0.1805 to 0.1045, showing empirical gain beyond affine geometry. Thus, VSJ correction can also improve loss-dependent updates.

Practical starting recipe. While applying VSJ to a new benchmark, the gains of VSJ over native priors in Tab. 4 suggest a simple starting point: retain the native $\mathbf{P}$ and $\mathbf{M}$ of the underlying backward rule, and tune how the constraints are enforced. This recipe corresponds to 10 out of 12 validation-selected VSJ configurations in Tab. 4, restricting the initial configuration search to only $\mathbf{\Omega}$. We next test this strategy using PEAR as a state-of-the-art baseline.

5.4 Comparison with State-of-the-art and Further Analyses

We compare with PEAR (Lee et al., 2026) using its benchmark protocols: (i) 100-item binary Knapsack with degree-8 synthetic data, and (ii) time-series Portfolio with 89 assets, 63-day histories, and 21-day forecasts (Hwang et al., 2025). We also include the MSE baseline minimizing squared prediction error on profits/returns, before using the predictions to optimize decisions.

Applying the recipe to PEAR. Following Tab. 1, VSJ recovers PEAR's underlying sensitivity with $\mathbf{P} = \mathbf{H}^{-1}$, $\mathbf{M} = \mathbf{H}$, and $\mathbf{\Omega} = \mathbf{I}$. The positive prior reflects profit/return maximization, reversing the table's minimization-cost convention. Following PEAR, we use $\mathbf{H} = 0.1\mathbf{I}$ for Knapsack and $\mathbf{H} = 2(\mathbf{L}\mathbf{L}^\top + 10^{-6}\mathbf{I})$ for Portfolio, where $\mathbf{L}$ is the Cholesky factor of the covariance estimated

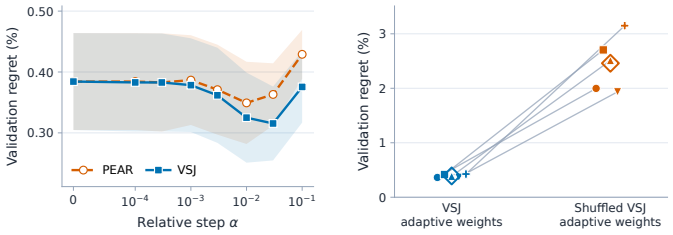
Table 6: **Test performance on the PEAR benchmark protocols.** Mean ± sample std. dev. over five seeds. Parentheses give relative changes from PEAR, $100(\bar{x} - \bar{x}_{\text{PEAR}})/|\bar{x}_{\text{PEAR}}|$. Green is better, red worse, following metric arrows. Bold: best, underline: second-best. PEAR is our reproduction.

Method	Binary Knapsack (Degree 8)	Time-series Portfolio (Real-world Dataset)				
	Regret (%) ↓	Regret (%) ↓	Cumulative return (%) ↑	Annualized return (%) ↑	Sharpe ↑	Signed max. drawdown (%) ↑
MSE	2.2844 ± 0.2041	98.61 ± 0.94	40.27 ± 7.06	21.86 ± 2.01	0.58 ± 0.06	−52.55 ± 1.82
PEAR (Lee et al., 2026)	0.4358 ± 0.0439	82.41 ± 4.68	265.70 ± 105.56	76.08 ± 18.72	1.80 ± 0.42	−28.67 ± 6.70
VSJ recovery of PEAR	0.4261 ± 0.0408	82.41 ± 4.68	265.70 ± 105.56	76.08 ± 18.72	1.80 ± 0.42	−28.67 ± 6.70
Strict VSJ (Eq. (3))	2.4781 ± 1.3871 (+468.6%)	80.81 ± 2.86 (−1.9%)	292.99 ± 63.90 (+10.3%)	**81.87** ± 12.49 (+7.6%)	1.92 ± 0.22 (+7.1%)	−27.97 ± 4.62 (+2.4%)
Adaptive VSJ (Eq. (7))	0.3838 ± 0.0234 (−11.9%)	81.00 ± 1.81 (−1.7%)	286.52 ± 41.03 (+7.8%)	80.57 ± 8.03 (+5.9%)	1.92 ± 0.13 (+7.1%)	−32.91 ± 4.61 (−14.8%)
Adaptive VSJ-PEAR (Eq. (11))	**0.3756** ± 0.0312 (−13.8%)	**80.48** ± 3.77 (−2.3%)	**298.10** ± 75.67 (+12.2%)	81.46 ± 14.48 (+7.1%)	**1.98** ± 0.32 (+10.3%)	**−22.28** ± 2.65 (+22.3%)

from historical and forecast returns. Adaptive VSJ retains these prior and metric choices and explores adaptive inequality enforcement, while Strict VSJ enforces the selected linearized constraints exactly. Adaptive VSJ-PEAR applies the correction in Eq. (11) directly to PEAR's native update.

Performance comparison. Tab. 6 shows that VSJ closely reproduces PEAR's performance (grey row), and both adaptive variants achieve lower mean regret than MSE on both tasks. Following the starting recipe, Adaptive VSJ retains PEAR's prior and metric and lowers mean regret relative to PEAR by 11.9% on Knapsack and 1.7% on Portfolio. Matched strict-adaptive controls show why native enforcement remains worth testing: adaptive enforcement lowers Knapsack mean regret from 2.4781% to 0.3838%, whereas strict enforcement gives lower Portfolio mean regret. Adaptive VSJ-PEAR composition achieves the lowest displayed mean regret, reducing it by 13.8% on Knapsack and 2.3% on Portfolio and raising mean cumulative return and Sharpe by 12.2% and 10.3%. At matched budget and tuning over ten seeds, Adaptive VSJ and VSJ-PEAR improve on PEAR on every seed, by 12.5% and 7.0% on average (Sec. G.2).

Update direction versus magnitude. To provide insights on VSJ's gains against PEAR, we investigate if PEAR or Adaptive VSJ yields lower regret from identical predictions after an equally-sized step. For this, we obtain 480 Knapsack predicted-profit vectors $\mathbf{z} \in \mathbb{R}^{100}$ using a mix of PEAR and VSJ checkpoints on validation set. At every vector, we evaluate both updates, regardless of which checkpoint produced it. We rescale each nonzero update to satisfy $\|\Delta\mathbf{z}\|_2 = \alpha\|\mathbf{z}\|_2$, so both move predictions equally and only their direction differs. In Fig. 3a, VSJ's direction yields lower mean regret at every displayed nonzero step size, showing that VSJ provides more useful update directions.



(a) PEAR vs. VSJ: equal steps. (b) Shuffling adaptive weights.

Figure 3: (a) Equal-norm prediction steps; bands: sample SD of five seeds. (b) Validation regret when we shuffle the adaptive weights. Lines pair seeds, diamonds mark means. Protocol details: Sec. G.9.1.

Does weight assignment matter? In a separate ablation on Knapsack, we shuffle adaptive weights among constraints of the same type, preserving their values at each backward evaluation but changing their assignment. In Fig. 3b, mean validation regret rises from 0.3913% to 2.4596%, with deterioration in all five seeds, further supporting the importance of constraint-specific weight assignment.

6 Conclusion

We introduced VSJ, which separates backward propagation through constrained optimisation into a prior $\mathbf{P}$, metric $\mathbf{M}$, and constraint weighting $\mathbf{\Omega}$. It supports strict active-manifold preservation and adaptive inequality enforcement, recovers established operators under local assumptions, and corrects loss-dependent updates. Experiments verify these recoveries and show decision-quality gains from component recombination and update correction, including notable gains over PEAR. VSJ thus turns coupled backward rules into independently configurable design choices.

AI USE STATEMENT

In this work, we used generative AI tools to help draft and revise the manuscript and to get feedback on it, to draft some mathematical derivations and proofs (for proofs of Theorem 1, Proposition 4, Proposition 5, derivation of PEAR as a VSJ configuration). We did not use generative AI tools to develop the ideas, add references, generate data or experimental results, and other disclosure categories are not applicable to this work. All AI-assisted material was critically reviewed and revised by the authors. In particular, the authors independently checked proofs and derivations against their assumptions, verified computational claims using the accompanying code and tests, and checked factual statements and references against the cited sources. Generative AI output was treated as a suggestion rather than evidence or authoritative verification. We take full responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The assumptions and algorithmic definitions underlying VSJ are stated in the main text, with proofs and detailed derivations in Secs. B to F. The evaluation protocol is described in Sec. 5.1; the appendix provides the benchmark formulations, architectures, training hyperparameters, operator-validation fixtures, ablation details, and runtime protocol in Secs. G.1, G.5 and G.8 to G.10. The supplementary code contains the VSJ implementation and its dependency specification, and the configurations and scripts that reproduce the fractional Knapsack results in Tabs. 4 and 5 and and those for the PEAR comparison in Tab. 6.

REFERENCES

Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a.

Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M. Moursi. Differentiating through a cone program. *Journal of Applied and Numerical Optimization*, 1(2):107–115, 2019b.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145. PMLR, 06–11 Aug 2017.

Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. In *Advances in Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Mathieu Blondel, André F. T. Martins, and Vlad Niculae. Learning with Fenchel-Young losses. *Journal of Machine Learning Research*, 21(35):1–69, 2020.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, pp. 2292–2300, 2013.

George Dantzig, Ray Fulkerson, and Selmer Johnson. Solution of a large-scale traveling-salesman problem. *Operations Research*, 2(4):393–410, 1954.

Justin Domke. Learning convex inference of marginals. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

Priya L. Donti, Brandon Amos, and J. Zico Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, NIPS'17, pp. 5490–5500, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68 (1):9–26, 2022. doi: 10.1287/mnsc.2020.3922.

Aaron Ferber, Bryan Wilder, Bistra Dilkina, and Milind Tambe. Mipaal: Mixed integer program as a layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:1504–1511, 04 2020. doi: 10.1609/aaai.v34i02.5509.

Stephen Gould, Basura Fernando, Anoop Cherian, Peter Anderson, Rodrigo Santa Cruz, and Edison Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization, 2016.

Geoffrey Hinton. Neural networks for machine learning, coursera. Coursera, video lectures, 2012.

Yoontae Hwang, Yaxuan Kong, Stefan Zohren, and Yongjae Lee. Decision-informed neural networks with large language model integration for portfolio optimization, 2025.

Alan Lahoud, Ahmad Khan, Erik Schaffernicht, Marco Trincavelli, and Johannes Stork. Predict-and-optimize techniques for data-driven optimization problems: A review. *Neural Processing Letters*, 57, 04 2025. doi: 10.1007/s11063-025-11746-w.

Junhyeong Lee, Sangjin Jin, and Yongjae Lee. Decision-focused learning via tangent-space projection of prediction error. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

Yuxuan Linghu, Zhiyuan Liu, and Qi Deng. A penalty approach for differentiation through black-box quadratic programming solvers. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*. PMLR, 2026.

Connor W. Magoon, Fengyu Yang, Noam Aigerman, and Shahar Z. Kovalsky. Differentiation through black-box quadratic programming solvers. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jayanta Mandi and Tias Guns. Interior point solving for lp-based prediction+optimisation. In *Advances in Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Jayanta Mandi, Emir Demirović, Peter J. Stuckey, and Tias Guns. Smart predict-and-optimize for hard combinatorial optimization problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:1603–1610, 04 2020. doi: 10.1609/aaai.v34i02.5521.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80, September 2024. ISSN 1076-9757. doi: 10.1613/jair.1.15320.

David McAllester, Tamir Hazan, and Joseph Keshet. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems*, NIPS'10, pp. 1594–1602, Red Hook, NY, USA, 2010. Curran Associates Inc.

Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. Contrastive losses and solution caching for predict-and-optimize. In Zhi-Hua Zhou (ed.), *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pp. 2833–2840. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/390. Main Track.

Vlad Niculae, Andre Martins, Mathieu Blondel, and Claire Cardie. SparseMAP: Differentiable sparse structured inference. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3799–3808. PMLR, 10–15 Jul 2018.

Mathias Niepert, Pasquale Minervini, and Luca Franceschi. Implicit mle: backpropagating through discrete exponential family distributions. In *Advances in Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer series in operations research and financial engineering. Springer, 2. ed. edition, 2006.

Kemal Oksuz, Baris Can Cam, Emre Akbas, and Sinan Kalkan. Rank & Sort Loss for Object Detection and Instance Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2989–2998, Los Alamitos, CA, USA, October 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00300.

Jianming Pan, Zeqi Ye, Xiao Yang, Xu Yang, Weiqing Liu, Lewen Wang, and Jiang Bian. BPQP: A differentiable convex optimization framework for efficient end-to-end learning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Anselm Paulus, Michal Rolinek, Vit Musil, Brandon Amos, and Georg Martius. Comboptnet: Fit the right np-hard problem by learning integer programming constraints. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8443–8453. PMLR, 18–24 Jul 2021.

Anselm Paulus, Georg Martius, and Vít Musil. Lpgd: a general framework for backpropagation through embedded optimization layers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019. doi: 10.1561/2200000073.

André Susano Pinto, Alexander Kolesnikov, Yuge Shi, Lucas Beyer, and Xiaohua Zhai. Tuning computer vision models with task rewards. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 33229–33239. PMLR, 23–29 Jul 2023.

Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. Differentiation of blackbox combinatorial solvers. In *Proceedings of the International Conference on Learning Representations*, 2020.

Roman Polyak. Modified barrier functions (theory and methods). *Mathematical Programming*, 54 (1–3):177–222, 1992.

Michal Rolínek, Vít Musil, Anselm Paulus, Marin Vlastelica, Claudio Michaelis, and Georg Martius. Optimizing rank-based metrics with blackbox differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7617–7627, 2020a.

Michal Rolínek, Paul Swoboda, Dominik Zietlow, Anselm Paulus, Vít Musil, and Georg Martius. Deep graph matching via blackbox differentiation of combinatorial solvers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *16th European Conference on Computer Vision*, pp. 407–424, Cham, 2020b. Springer International Publishing. ISBN 978-3-030-58604-1.

Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. Backpropagation through combinatorial algorithms: Identity with projection works. In *Proceedings of the International Conference on Learning Representations*, 2023.

Philipp J. Schneider and Daniel Kuhn. Soft-radial projection for constrained end-to-end learning. 2026.

Yang Song, Alexander G. Schwing, Richard S. Zemel, and Raquel Urtasun. Training deep neural networks via direct loss minimization. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, volume 48 of *ICML'16*, pp. 2169–2177. JMLR.org, 2016.

Andrey N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4:1035–1038, 1963.

Po-Wei Wang, Priya L. Donti, Bryan Wilder, and J. Zico Kolter. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 6545–6554, 2019.

Runzhong Wang, Yunhao Zhang, Ziao Guo, Tianyi Chen, Xiaokang Yang, and Junchi Yan. LinSATNet: The positive linear satisfiability neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36605–36625. PMLR, 23–29 Jul 2023.

Yanzhi Wang, Jianxiao Wang, Haoran Zhang, and Jie Song. Bridging prediction and decision: Advances and challenges in data-driven optimization. *Nexus*, 2(1):100057, 2025. ISSN 2950-1601. doi: 10.1016/j.ynexs.2025.100057.

Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: decision-focused learning for combinatorial optimization. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence and 31st Innovative Applications of Artificial Intelligence Conference and 9th AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019a. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33011658.

Bryan Wilder, Eric Ewing, Bistra Dilkina, and Milind Tambe. End to end learning and optimization on graphs. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.

Max A Woodbury. Inverting modified matrices. Technical Report 42, Statistical Research Group, Princeton University, 1950.

TABLE OF CONTENTS: APPENDIX

A Relation to Differentiable Forward Layers **14**

B Active-Manifold Preservation, Local Feasibility and Proof of Theorem 1 **15**

C Proof of the Closed-Form VSJ Solution Theorem 3 **16**

D Proof of the Geometric Interpretation **16**

E Further Details on the Adaptively Weighted VSJ Correction **17**

E.1 Constraint Rows and primal-dual State 17

E.2 Adaptive Weights and Stiffnesses 19

E.3 Adaptive VSJ as a Shifted-Barrier Sensitivity 20

E.4 Finite-Weight Solution 20

E.5 Mixed Strict-Soft Correction 22

E.6 Numerical Implementation 23

E.7 Comparison with dXPP 24

F Recovery of Existing Backward Operators **25**

G Further Details on Experiments and Results **29**

G.1 Formulation of the Benchmarks 29

G.2 Evaluation Metrics and Protocols 30

G.3 Attribution of the Portfolio PEAR Composition 32

G.4 End-to-End Consistency of Matched VSJ Configurations 33

G.5 Direct Operator Validation 34

G.6 Linear Dependence and the Pseudoinverse: MNIST Matching 35

G.7 Pascal VOC-U Graph Matching 36

G.8 Experimental Validation and Training Details 37

G.9 Mechanism and Axis-Ablation Details 37

G.10 Computational Efficiency 41

G.11 Validation Results for Configuration Selection 42

A RELATION TO DIFFERENTIABLE FORWARD LAYERS

Differentiable forward constructions. Several approaches construct a forward computation that supports differentiation. For example, SATNet (Wang et al., 2019) solves a semidefinite relaxation of a satisfiability problem and differentiates its solution. Sinkhorn iterations can be used to compute entropy-regularized optimal transport through matrix scaling (Cuturi, 2013). LinSATNet (Wang et al., 2023) extends this normalization approach to enforce linear equalities and inequalities with nonnegative coefficients and right-hand sides. These approaches overlap with differentiable optimisation layers and have a different focus than our paper.

Forward feasibility versus backward constraint preservation. Our VSJ framework addresses a different question: which backward operator to use for a chosen forward solver. It leaves the

computed decision unchanged and selects a surrogate Jacobian using a prior $\mathbf{P}$, correction metric $\mathbf{M}$, and constraint weighting $\mathbf{\Omega}$. Under strict enforcement, $\mathbf{FJ} = \mathbf{0}$ requires the implied decision perturbations to preserve the selected active manifold to first order. This is not a projection of the forward output onto the feasible set, nor does it guarantee feasibility after finite perturbations. Adaptive weighting relaxes inequality preservation in the backward rule, while equalities remain strict. Forward feasibility remains the responsibility of the chosen solver.

Compatibility and scope. Our VSJ framework does not require a differentiable relaxation or added forward regularization, but it does not exclude these choices. When a regularized forward problem is retained, suitable VSJ configurations recover its cost sensitivity under the corresponding local assumptions. For example, the curvature-coupled configuration applies to entropy-regularized transport at a positive solution with fixed marginals. Such correspondence concerns the local operator at a common forward solution, not an equivalence of solver implementations or complete training procedures. Other choices of prior, metric, and enforcement deliberately produce surrogate backward rules. The distinction is therefore between constructing a differentiable forward map and independently designing the backward operator, rather than between differentiating a map and differentiating a solver. A concurrent forward-side treatment reports the same obstruction from the other direction: Schneider & Kuhn (2026) observe that orthogonal projection layers induce rank-deficient Jacobians that nullify gradients orthogonal to active constraints, and avoid it by reparameterizing the forward map into the interior of the feasible set. VSJ leaves the forward solver untouched and relaxes the same obstruction in the backward operator through finite inequality weights.

B Active-Manifold Preservation, Local Feasibility and Proof of Theorem 1

The equality condition in Theorem 1 may appear stronger than local feasibility: at an active inequality, a one-sided feasible direction needs to only satisfy $\nabla h_i(\mathbf{s}^\star)^\top \delta\mathbf{s} \leq 0$. A Jacobian, however, describes responses to both $\delta\mathbf{z}$ and $-\delta\mathbf{z}$; requiring feasibility in both directions forces equality and therefore preserves the manifold defined by a fixed set of selected active rows rather than the full feasible tangent cone. The full active set in Eq. (2) is the canonical choice, while implementation-specific selection is detailed in Sec. E. This tangent-space interpretation assumes continuously differentiable constraints and locally constant rank; identifying the selected rows with a stable full active set additionally requires standard sensitivity conditions such as LICQ, strict complementarity, and second-order sufficiency, whereas the algebraic VSJ solution requires only fixed $\mathbf{F}$ and $\mathbf{M} \succ 0$.

Proof of Theorem 1. Let $c_{\mathcal{A}}$ collect the equality and selected active inequality constraints, so that $c_{\mathcal{A}}(\mathbf{s}^\star) = \mathbf{0}$ and $\mathbf{F} = \nabla c_{\mathcal{A}}(\mathbf{s}^\star)$. For any parameter perturbation $\delta\mathbf{z}$, differentiability gives

$$c_{\mathcal{A}}\big(\mathbf{s}^\star + t\mathbf{J}\delta\mathbf{z}\big) = t\mathbf{F}\mathbf{J}\delta\mathbf{z} + o(t), \tag{12}$$

where t is a small scalar and $o(t)$ denotes higher-order terms that vanish faster than t as $t \to 0$. If $\mathbf{FJ} = \mathbf{0}$, the right-hand side is $o(t)$ for every $\delta\mathbf{z}$, so the selected active manifold is preserved to first order.

Conversely, first-order preservation implies

$$t\mathbf{F}\mathbf{J}\delta\mathbf{z} + o(t) = o(t), \tag{13}$$

for every $\delta\mathbf{z}$. Dividing by t and taking $t \to 0$ gives $\mathbf{FJ}\delta\mathbf{z} = \mathbf{0}$ for every $\delta\mathbf{z}$, and hence $\mathbf{FJ} = \mathbf{0}$. □

For completeness, local feasibility along $\delta\mathbf{z}$ at an active inequality implies

$$\nabla h_i(\mathbf{s}^\star)^\top \mathbf{J}\delta\mathbf{z} \leq 0. \tag{14}$$

Applying the same condition along $-\delta\mathbf{z}$ gives the reverse inequality, and therefore

$$\nabla h_i(\mathbf{s}^\star)^\top \mathbf{J}\delta\mathbf{z} = 0. \tag{15}$$

Thus, a two-sided differentiable solution map that remains feasible locally is tangent to every active inequality.

C Proof of the Closed-Form VSJ Solution Theorem 3

Proof of Theorem 3. The objective in Equation 3 is strictly convex because $\mathbf{M} \succ 0$, and the constraints are linear and feasible. Therefore, the problem has a unique minimizer, and its KKT conditions are necessary and sufficient (Boyd & Vandenberghe, 2004, §5.5.3).

Introducing $\mathbf{\Lambda} \in \mathbb{R}^{m \times n}$, the Lagrangian is

$$\mathcal{L}(\mathbf{J}, \mathbf{\Lambda}) = \frac{1}{2} \operatorname{tr}\big((\mathbf{J} - \mathbf{P})^\top \mathbf{M} (\mathbf{J} - \mathbf{P})\big) + \operatorname{tr}\big(\mathbf{\Lambda}^\top \mathbf{F} \mathbf{J}\big). \tag{16}$$

Stationarity with respect to $\mathbf{J}$ gives

$$\mathbf{M}(\mathbf{J} - \mathbf{P}) + \mathbf{F}^\top \mathbf{\Lambda} = \mathbf{0}, \tag{17}$$

and hence

$$\mathbf{J} = \mathbf{P} - \mathbf{M}^{-1} \mathbf{F}^\top \mathbf{\Lambda}. \tag{18}$$

Define

$$\mathbf{A} = \mathbf{F} \mathbf{M}^{-1} \mathbf{F}^\top. \tag{19}$$

Because $\mathbf{M}^{-1} \succ 0$,

$$\mathbf{y}^\top \mathbf{A} \mathbf{y} = \left\| \mathbf{M}^{-1/2} \mathbf{F}^\top \mathbf{y} \right\|_2^2, \tag{20}$$

so

$$\ker(\mathbf{A}) = \ker(\mathbf{F}^\top). \tag{21}$$

Since $\mathbf{A}$ is symmetric,

$$\operatorname{range}(\mathbf{A}) = \ker(\mathbf{A})^\perp = \ker(\mathbf{F}^\top)^\perp = \operatorname{range}(\mathbf{F}). \tag{22}$$

Every column of $\mathbf{FP}$ therefore lies in $\operatorname{range}(\mathbf{A})$. The Moore–Penrose identity $\mathbf{A}\mathbf{A}^\dagger \mathbf{v} = \mathbf{v}$ for $\mathbf{v} \in \operatorname{range}(\mathbf{A})$ consequently gives

$$\mathbf{A}\mathbf{A}^\dagger \mathbf{F}\mathbf{P} = \mathbf{F}\mathbf{P}. \tag{23}$$

Choose the minimum-norm multiplier

$$\mathbf{\Lambda} = \mathbf{A}^\dagger \mathbf{F}\mathbf{P}. \tag{24}$$

Substitution into the stationarity expression yields

$$\mathbf{J}^\star = \mathbf{P} - \mathbf{M}^{-1} \mathbf{F}^\top \mathbf{A}^\dagger \mathbf{F}\mathbf{P}. \tag{25}$$

This candidate is feasible because

$$\begin{aligned} \mathbf{F}\mathbf{J}^\star &= \mathbf{F}\mathbf{P} - \mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top \mathbf{A}^\dagger \mathbf{F}\mathbf{P} \\ &= \mathbf{F}\mathbf{P} - \mathbf{A}\mathbf{A}^\dagger \mathbf{F}\mathbf{P} = \mathbf{0}. \end{aligned} \tag{26}$$

It also satisfies stationarity by construction. Hence, it satisfies the KKT conditions and is the unique minimizer of Equation 3. Although the primal minimizer $\mathbf{J}^\star$ is unique, the multiplier need not be unique when $\mathbf{F}$ has dependent rows. □

D Proof of the Geometric Interpretation

Proof of Proposition 4. Let

$$\mathbf{A} = \mathbf{F} \mathbf{M}^{-1} \mathbf{F}^\top. \tag{27}$$

As established in the proof of Theorem 3,

$$\operatorname{range}(\mathbf{A}) = \operatorname{range}(\mathbf{F}). \tag{28}$$

Consequently, every column of $\mathbf{F}$ lies in $\operatorname{range}(\mathbf{A})$, and the Moore–Penrose identity gives

$$\mathbf{A}\mathbf{A}^\dagger \mathbf{F} = \mathbf{F}. \tag{29}$$

By the definition of $\Pi_{\mathbf{M}}$ in Eq. (4), left-multiplication by $\mathbf{F}$ gives

$$\begin{aligned} \mathbf{F}\Pi_{\mathbf{M}} &= \mathbf{F} - \mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top \mathbf{A}^\dagger \mathbf{F} \\ &= \mathbf{F} - \mathbf{A}\mathbf{A}^\dagger \mathbf{F} = \mathbf{0}. \end{aligned} \tag{30}$$

Hence,

$$\mathrm{range}(\Pi_{\mathbf{M}}) \subseteq \ker(\mathbf{F}). \tag{31}$$

Conversely, if $\mathbf{x} \in \ker(\mathbf{F})$, then

$$\Pi_{\mathbf{M}}\mathbf{x} = \mathbf{x} - \mathbf{M}^{-1}\mathbf{F}^{\top}\mathbf{A}^{\dagger}\mathbf{F}\mathbf{x} = \mathbf{x}. \tag{32}$$

Thus, $\Pi_{\mathbf{M}}$ acts as the identity on $\ker(\mathbf{F})$. Together with $\mathrm{range}(\Pi_{\mathbf{M}}) \subseteq \ker(\mathbf{F})$, this shows that

$$\mathrm{range}(\Pi_{\mathbf{M}}) = \ker(\mathbf{F}), \tag{33}$$

and

$$\Pi_{\mathbf{M}}^2 = \Pi_{\mathbf{M}}. \tag{34}$$

Therefore, $\Pi_{\mathbf{M}}$ is a projector onto $\ker(\mathbf{F})$.

It remains to establish orthogonality under the inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{M}} = \mathbf{x}^{\top}\mathbf{M}\mathbf{y}. \tag{35}$$

For any $\mathbf{y} \in \ker(\mathbf{F})$ and any $\mathbf{x} \in \mathbb{R}^n$,

$$\begin{aligned} \langle \mathbf{y}, (\mathbf{I} - \Pi_{\mathbf{M}})\mathbf{x} \rangle_{\mathbf{M}} &= \mathbf{y}^{\top}\mathbf{M}\mathbf{M}^{-1}\mathbf{F}^{\top}\mathbf{A}^{\dagger}\mathbf{F}\mathbf{x} \\ &= (\mathbf{F}\mathbf{y})^{\top}\mathbf{A}^{\dagger}\mathbf{F}\mathbf{x} = 0. \end{aligned} \tag{36}$$

Hence, the projection residual $(\mathbf{I} - \Pi_{\mathbf{M}})\mathbf{x}$ is $\mathbf{M}$-orthogonal to $\ker(\mathbf{F})$. Therefore, $\Pi_{\mathbf{M}}$ is the $\mathbf{M}$-orthogonal projector onto $\ker(\mathbf{F})$. □

E Further Details on the Adaptively Weighted VSJ Correction

This appendix derives the closed-form adaptive VSJ correction and clarifies the treatment of zero, finite, and unit constraint weights. Throughout, let $\mathbf{J}, \mathbf{P} \in \mathbb{R}^{n \times p}$ and let $\mathbf{M} \in \mathbb{R}^{n \times n}$ be symmetric positive definite.

E.1 Constraint Rows and primal-dual State

Constraint rows used by strict and adaptive VSJ. In the main text, $\mathcal{A}$ denotes the full set of binding inequalities. The VSJ formulas apply equally when $\mathbf{F}$ contains a selected subset of these rows; in that case, the preserved manifold is the one defined by the selected constraints rather than the complete active manifold.

Our implementation selects inequality rows from the backward primal-dual certificate, not from an activity test at a potentially different forward decision. It first considers all model inequalities and finite variable bounds, including, for TSP, the inequalities present in the final DFJ model. A candidate row is retained when its supplied multiplier is nonzero and its relaxation is finite. Consequently, at an exact primal-dual optimum, positive-dual binding rows are retained, whereas zero-dual rows, including degenerate binding rows, are omitted. A strict weight enforces each retained row exactly but does not cause an omitted zero-dual row to enter the system. Approximate or noncomplementary primal-dual states may additionally retain rows with positive slack.

Throughout this appendix, $\mathbf{F}$ denotes the row set used by the construction at hand. Strict VSJ uses the equality rows and selected active inequalities with zero relaxation. Adaptive VSJ starts from the equality rows and candidate inequalities, omits zero-weight rows, and assigns the finite relaxation $d_k = (r_k + \mu)/\lambda_k$ to each retained finite-weight inequality. Conditional on a fixed retained row set, the strict and adaptive formulas differ only in enforcement strength; adaptive weighting may also change which candidate rows contribute.

If the retained rows satisfy $\mathrm{rank}(\mathbf{F}) = n$, strict VSJ gives $\mathbf{J}^{\star} = \mathbf{0}$, irrespective of $\mathbf{P}$ and $\mathbf{M}$. This can occur at a nondegenerate LP vertex; using an LP relaxation or omitting zero-dual rows does not guarantee a nontrivial null space. Finite inequality weights relax these tangency requirements while preserving the strictly enforced equality rows.

Relation to forward smoothing on linear programs. Zero LP curvature does not imply a singular reduced KKT system: at a stable nondegenerate optimal vertex, this system is nonsingular and the cost Jacobian is zero. The strict projector vanishes exactly when $\mathrm{rank}(\mathbf{F}) = n$. BPQP (Pan et al., 2024, Eq. 17) and PEAR (Lee et al., 2026, Sec. 4.3) add quadratic curvature to the forward objective. In our notation, $R(\mathbf{s}) = \frac{\lambda}{2}\|\mathbf{s}\|_2^2$ gives $(-\lambda^{-1}\mathbf{I}, \lambda\mathbf{I}, \mathbf{I})$, the isotropic configuration in Tab. 1. For fixed $\mathbf{F}$, $\Pi_{\lambda\mathbf{I}}$ is independent of λ and the Jacobian scales with λ^{-1}; changing λ can also change the forward solution and active face. In contrast, adaptive VSJ modifies only the backward operator through finite inequality stiffnesses, imposing equalities separately as exact constraints (Sec. E).

Multiplier sign convention. Throughout, λ_k denotes the nonnegative KKT multiplier associated with the canonical inequality $h_k(\mathbf{s}) \le 0$, rather than a raw signed marginal returned by a solver. A maximization problem is first written as minimization of the negated objective, and constraints with the opposite orientation are rewritten accordingly. In particular, lower and upper bounds are represented as

$$\ell_i - s_i \le 0, \qquad s_i - u_i \le 0. \tag{37}$$

Under this convention, every inequality multiplier satisfies $\lambda_k \ge 0$, independently of the objective direction or constraint type. Equality multipliers remain unrestricted in sign, but do not enter the adaptive rule because equality rows are assigned unit weight.

Solver-reported multipliers are converted to this nonnegative canonical convention, according to the solver's documented objective and constraint orientations, before computing the adaptive weights. For dual-feasible one-sided inequalities, this conversion is equivalent to taking the corresponding multiplier magnitude.

Constraint normalization and rescaling invariance. Before forming the adaptive weights, we normalize each adaptively weighted inequality row together with its slack and multiplier.

Proposition 5 (Constraint-Rescaling Invariance)**.** *Let $\mathbf{f}_k^\top = \nabla h_k(\mathbf{s}^\star)^\top$ be an adaptively weighted inequality row, with slack $r_k = -h_k(\mathbf{s}^\star) \ge 0$, multiplier $\lambda_k \ge 0$, and $\kappa_k = \|\mathbf{f}_k\|_2 > 0$. Define*

$$\bar{\mathbf{f}}_k = \frac{\mathbf{f}_k}{\kappa_k}, \qquad \bar{r}_k = \frac{r_k}{\kappa_k}, \qquad \bar{\lambda}_k = \kappa_k \lambda_k. \tag{38}$$

Then the normalized row, adaptive weight, and resulting VSJ correction are invariant under $h_k \mapsto a_k h_k$ for any $a_k > 0$.

Proof. Under $h_k \mapsto a_k h_k$, the constraint row and slack transform as

$$\mathbf{f}_k \mapsto a_k \mathbf{f}_k, \qquad r_k \mapsto a_k r_k. \tag{39}$$

To preserve the same Lagrangian term, the multiplier transforms as $\lambda_k \mapsto \lambda_k / a_k$, while $\kappa_k \mapsto a_k \kappa_k$. Hence,

$$\frac{a_k \mathbf{f}_k}{a_k \kappa_k} = \bar{\mathbf{f}}_k, \qquad \frac{a_k r_k}{a_k \kappa_k} = \bar{r}_k, \qquad (a_k \kappa_k)\frac{\lambda_k}{a_k} = \bar{\lambda}_k. \tag{40}$$

Thus, the normalized row, slack, and multiplier are unchanged. Therefore, so are

$$w_k = \frac{\bar{\lambda}_k}{\bar{\lambda}_k + \bar{r}_k + \mu}, \qquad \gamma_k = \frac{w_k}{1 - w_k} = \frac{\bar{\lambda}_k}{\bar{r}_k + \mu}. \tag{41}$$

The contribution of row k to $\mathbf{F}^\top \mathbf{\Gamma} \mathbf{F}$ is

$$\gamma_k \bar{\mathbf{f}}_k \bar{\mathbf{f}}_k^\top, \tag{42}$$

which is also unchanged. Consequently, the weighted objective is unchanged and, since $\mathbf{M} \succ 0$, so is its unique minimizer $\mathbf{J}^\star$. □

The proposition concerns adaptively weighted inequality rows. A row assigned zero weight is omitted, while a unit-weight inequality may retain its original scaling because nonzero row scaling does not change exact enforcement. Equality rows are likewise strict and may retain their original scaling without changing their null space or the resulting hard projection. Adaptively weighted rows with $\kappa_k = 0$ are locally degenerate and are omitted from the correction. In the remainder, we suppress the bars for adaptively weighted inequalities and write $\mathbf{f}_k$, r_k, and λ_k for their normalized row, slack, and multiplier, as in Sec. 3.3; $\mathbf{F}$ is assembled from these rows and the strict rows as described above.

Dependence on objective scale and certificate choice. Proposition 5 concerns rescaling a constraint representation, not the objective. Multiplying the complete objective by $a > 0$ leaves its optimizer and primal slacks unchanged but scales KKT inequality multipliers as $\lambda_k \mapsto a\lambda_k$; for an unregularized linear objective, this is equivalent to $\mathbf{z} \mapsto a\mathbf{z}$. Therefore, with μ fixed, the adaptive weight generally changes to

$$\frac{a\lambda_k}{a\lambda_k + r_k + \mu}. \tag{43}$$

We consequently treat the objective normalization and μ as part of the VSJ configuration and keep the cost preprocessing fixed within each empirical comparison. Moreover, at a dual-degenerate optimum, multiple valid multiplier vectors may induce different weights. The implementation uses the canonicalized multiplier returned by the stated solver, so adaptive VSJ is uniquely defined conditional on that primal-dual certificate, but is not claimed to be invariant to objective rescaling or independent of the solver's dual selection.

primal-dual state used for weighting. The primal slacks and dual multipliers are obtained from the optimisation state associated with the predicted costs. When the forward program provides its own primal-dual state, we obtain the certificate from that state. Otherwise, we solve the corresponding continuous LP relaxation with HiGHS and use its optimal primal solution and constraint marginals. This auxiliary solution is used only to construct the backward correction and does not replace the forward decision. For a discrete forward solver, candidate-row selection, slacks, and multipliers are therefore evaluated at the relaxation solution, not at the discrete decision. The resulting $\mathbf{F}$ defines a relaxation-derived surrogate geometry: the VSJ projection results apply conditional on this $\mathbf{F}$, but do not identify it with the active manifold at the discrete forward decision. For an exact primal-dual optimum of an LP or QP, complementary slackness implies that any inequality row with $\lambda_k > 0$ has $r_k = 0$. Its adaptive weight and relaxation therefore reduce to

$$w_k = \frac{\lambda_k}{\lambda_k + \mu}, \qquad d_k = \frac{\mu}{\lambda_k}. \tag{44}$$

Thus, $\mu > 0$ provides finite softening even at an exact KKT point. A row with $\lambda_k = 0$ receives zero weight and is omitted, consistently with the limit $d_k \to \infty$ as $\lambda_k \to 0^+$. The general expression involving both r_k and λ_k also accommodates approximate or noncomplementary primal-dual states, for which both quantities may be positive. Our implementation neither requires them to be simultaneously positive nor assumes a particular interior-point tolerance.

E.2 Adaptive Weights and Stiffnesses

A constraint weight $w_k \in [0, 1]$ is converted into a nonnegative stiffness

$$\gamma_k = \frac{w_k}{1 - w_k}. \tag{45}$$

This transformation has the desired endpoint behavior:

$$w_k = 0 \implies \gamma_k = 0, \qquad w_k \to 1 \implies \gamma_k \to \infty. \tag{46}$$

Thus, a zero-weight row is ignored, a finite weight produces a soft penalty, and a unit weight is interpreted as exact enforcement in the limiting sense.

For the adaptive choice used in the main text,

$$w_k = \begin{cases} 1, & \text{if row } k \text{ is an equality,} \\ \dfrac{\lambda_k}{\lambda_k + r_k + \mu}, & \text{if row } k \text{ is an inequality,} \end{cases} \tag{47}$$

where $r_k \geq 0$ is the primal slack, $\lambda_k \geq 0$ is the dual multiplier, and $\mu > 0$ is the temperature. For an inequality with $\lambda_k > 0$, substituting Equation 47 into Equation 45 gives

$$\gamma_k = \frac{\lambda_k}{r_k + \mu}. \tag{48}$$

Its reciprocal,

$$d_k = \gamma_k^{-1} = \frac{r_k + \mu}{\lambda_k}, \tag{49}$$

will be called the *relaxation*. Large stiffness γ_k corresponds to small relaxation d_k and therefore stronger enforcement.

Properties of the Adaptive Weighting

Lemma 6 (Adaptive-Weight Properties)**.** *For an inequality, let*

$$w_k = \frac{\lambda_k}{\lambda_k + r_k + \mu}, \qquad \lambda_k, r_k \geq 0, \quad \mu > 0. \tag{50}$$

Then $w_k \in [0, 1)$, is strictly increasing in λ_k, and is nonincreasing in r_k and μ. Moreover,

$$\lim_{\lambda_k \to 0} w_k = 0, \qquad \lim_{\lambda_k \to \infty} w_k = 1. \tag{51}$$

Proof. The bounds follow from $0 \leq \lambda_k < \lambda_k + r_k + \mu$. Furthermore,

$$\frac{\partial w_k}{\partial \lambda_k} = \frac{r_k + \mu}{(\lambda_k + r_k + \mu)^2} > 0, \qquad \frac{\partial w_k}{\partial r_k} = \frac{\partial w_k}{\partial \mu} = -\frac{\lambda_k}{(\lambda_k + r_k + \mu)^2} \leq 0. \tag{52}$$

The limiting statements follow directly from the definition. □

E.3 Adaptive VSJ as a Shifted-Barrier Sensitivity

Proposition 7 (Adaptive VSJ as a shifted-barrier sensitivity)**.** *Let R be twice differentiable with $\mathbf{H} = \nabla^2 R(\mathbf{s}^\star) \succ 0$, let the retained inequalities be affine, and fix the primal-dual state. Replace each retained inequality with $0 < w_k < 1$ by the shifted log-barrier $\psi_k(\mathbf{s}) = -\lambda_k(r_k+\mu)\log\big(r_k(\mathbf{s})+\mu\big)$, whose coefficient is fixed at the current state (a modified barrier in the sense of Polyak (1992) when $r_k = 0$), keep unit-weight rows as constraints, and drop zero-weight rows. Then $\mathbf{s}^\star$ remains stationary, and the derivative of the resulting solution with respect to $\mathbf{z}$ at $\mathbf{s}^\star$ equals the adaptive VSJ Jacobian with $(\mathbf{P}, \mathbf{M}) = (-\mathbf{H}^{-1}, \mathbf{H})$.*

Proof. Write each retained inequality as $h_k(\mathbf{s}) = \mathbf{f}_k^\top \mathbf{s} - b_k \leq 0$ with slack $r_k(\mathbf{s}) = -h_k(\mathbf{s})$, so $r_k(\mathbf{s}^\star) = r_k$, and let $\mathbf{F}_\mathrm{s}$ collect the unit-weight rows. With $c_k = \lambda_k(r_k + \mu)$,

$$\nabla\psi_k(\mathbf{s}) = \frac{c_k}{r_k(\mathbf{s}) + \mu}\mathbf{f}_k, \qquad \nabla^2\psi_k(\mathbf{s}) = \frac{c_k}{(r_k(\mathbf{s}) + \mu)^2}\mathbf{f}_k\mathbf{f}_k^\top. \tag{53}$$

At $\mathbf{s}^\star$ these equal $\lambda_k\mathbf{f}_k$ and $\gamma_k\mathbf{f}_k\mathbf{f}_k^\top$ with $\gamma_k = \lambda_k/(r_k + \mu) = w_k/(1 - w_k)$. Zero-weight rows have $\lambda_k = 0$, so the KKT condition of the original problem is the stationarity condition of the modified problem, $\mathbf{z} + \nabla R(\mathbf{s}) + \sum_k \nabla\psi_k(\mathbf{s}) + \mathbf{F}_\mathrm{s}^\top\boldsymbol{\nu} = \mathbf{0}$ with $\mathbf{F}_\mathrm{s}\mathbf{s}$ fixed, at $\mathbf{s}^\star$ with the original multipliers. Differentiating with respect to $\mathbf{z}$ gives

$$\mathbf{I} + \big(\mathbf{H} + \mathbf{F}^\top\boldsymbol{\Gamma}\mathbf{F}\big)\,\mathbf{J} + \mathbf{F}_\mathrm{s}^\top\frac{\partial\boldsymbol{\nu}}{\partial\mathbf{z}} = \mathbf{0}, \qquad \mathbf{F}_\mathrm{s}\mathbf{J} = \mathbf{0}. \tag{54}$$

With $\mathbf{P} = -\mathbf{H}^{-1}$ and $\mathbf{M} = \mathbf{H}$, the first-order conditions of the mixed strict-soft problem of Sec. E.5 read $\mathbf{H}(\mathbf{J} - \mathbf{P}) + \mathbf{F}^\top\boldsymbol{\Gamma}\mathbf{F}\mathbf{J} + \mathbf{F}_\mathrm{s}^\top\boldsymbol{\Lambda} = \mathbf{0}$ and $\mathbf{F}_\mathrm{s}\mathbf{J} = \mathbf{0}$, and $\mathbf{H}\mathbf{P} = -\mathbf{I}$ makes them the same system. Since $\mathbf{H} + \mathbf{F}^\top\boldsymbol{\Gamma}\mathbf{F} \succ 0$, both have the same unique solution; without strict rows it is $\mathbf{J} = -(\mathbf{H} + \mathbf{F}^\top\boldsymbol{\Gamma}\mathbf{F})^{-1}$, Eq. (8). □

For a nonlinear inequality, $\nabla^2\psi_k(\mathbf{s}^\star)$ gains $\lambda_k\nabla^2 h_k(\mathbf{s}^\star)$, its Lagrangian curvature, and the statement holds with that term included in $\mathbf{H}$. At $\mu = 0$, $\gamma_k = \lambda_k/r_k$ is the barrier curvature of an interior-point method at its central-path multiplier; for an active row with $\lambda_k > 0$, $\gamma_k \to \infty$ as $\mu \to 0$ and the hard constraint is recovered. Adaptive weighting therefore differentiates a shifted log-barrier model of the retained inequalities, with the slack shifted by μ, and its stiffness stays finite at active rows, where the hard-constrained derivative can vanish. At $\mu = 0$ and an interior primal-dual iterate, this is the interior-point KKT derivative used by OptNet.

E.4 Finite-Weight Solution

We first assume that all retained weights satisfy $0 < w_k < 1$, so that every stiffness is finite and positive. Define

$$\boldsymbol{\Gamma} = \mathrm{diag}(\gamma_1, \ldots, \gamma_m) \succ 0. \tag{55}$$

The weighted VSJ objective is

$$\mathcal{L}(\mathbf{J}) = \frac{1}{2}\,\mathrm{tr}\big((\mathbf{J} - \mathbf{P})^\top\mathbf{M}(\mathbf{J} - \mathbf{P})\big) + \frac{1}{2}\,\mathrm{tr}\big(\mathbf{J}^\top\mathbf{F}^\top\boldsymbol{\Gamma}\mathbf{F}\mathbf{J}\big)\,. \tag{56}$$

The first term measures the deviation from the prior under the metric $\mathbf{M}$. The second penalizes the linearized constraint residual $\mathbf{FJ}$, with one stiffness for each constraint row.

Since $\mathbf{M}$ and $\mathbf{\Gamma}$ are symmetric, differentiating Equation 56 with respect to $\mathbf{J}$ gives

$$\nabla_{\mathbf{J}}\mathcal{L}(\mathbf{J}) = \mathbf{M}(\mathbf{J}-\mathbf{P}) + \mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\mathbf{J}. \tag{57}$$

The first-order optimality condition is therefore

$$\left(\mathbf{M}+\mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\right)\mathbf{J} = \mathbf{M}\mathbf{P}. \tag{58}$$

For every nonzero $\mathbf{x}\in\mathbb{R}^n$,

$$\mathbf{x}^\top\left(\mathbf{M}+\mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\right)\mathbf{x} = \mathbf{x}^\top\mathbf{M}\mathbf{x} + (\mathbf{F}\mathbf{x})^\top\mathbf{\Gamma}(\mathbf{F}\mathbf{x}) \tag{59}$$

$$> 0, \tag{60}$$

because $\mathbf{M}\succ 0$ and $\mathbf{\Gamma}\succeq 0$. Hence, $\mathbf{M}+\mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}$ is positive definite, regardless of the rank of $\mathbf{F}$. The objective is strictly convex and has the unique solution

$$\boxed{\mathbf{J}^\star = \left(\mathbf{M}+\mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\right)^{-1}\mathbf{M}\mathbf{P}.} \tag{61}$$

Equation 61 is already a complete closed-form solution. In numerical implementations, the inverse should not be formed explicitly; one instead solves the positive-definite linear system in Equation 58.

Woodbury form. When the number of retained constraint rows m is smaller than the decision dimension n, it can be preferable to solve an $m\times m$ system instead of the $n\times n$ system in Equation 58. This follows from the Woodbury matrix identity (Woodbury, 1950):

$$(\mathbf{A}+\mathbf{U}\mathbf{C}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{V}\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}\mathbf{A}^{-1}. \tag{62}$$

The identity rewrites the inverse of a matrix plus a structured update in terms of a system whose dimension is determined by the inner matrix $\mathbf{C}$.

Applying Equation 62 with

$$\mathbf{A}=\mathbf{M}, \qquad \mathbf{U}=\mathbf{F}^\top, \qquad \mathbf{C}=\mathbf{\Gamma}, \qquad \mathbf{V}=\mathbf{F}, \tag{63}$$

gives

$$\left(\mathbf{M}+\mathbf{F}^\top\mathbf{\Gamma}\mathbf{F}\right)^{-1} = \mathbf{M}^{-1} - \mathbf{M}^{-1}\mathbf{F}^\top\left(\mathbf{\Gamma}^{-1}+\mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top\right)^{-1}\mathbf{F}\mathbf{M}^{-1}. \tag{64}$$

Multiplying Equation 64 by $\mathbf{MP}$ yields

$$\mathbf{J}^\star = \left[\mathbf{I} - \mathbf{M}^{-1}\mathbf{F}^\top\left(\mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top+\mathbf{D}\right)^{-1}\mathbf{F}\right]\mathbf{P}, \qquad \mathbf{D}=\mathbf{\Gamma}^{-1}. \tag{65}$$

For the adaptive inequality weights,

$$\mathbf{D} = \operatorname{diag}\left(\frac{r_k+\mu}{\lambda_k}\right). \tag{66}$$

The matrix

$$\mathbf{S} = \mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top \tag{67}$$

is, up to sign, the Schur complement (Boyd & Vandenberghe, 2004, §A.5.5) obtained by eliminating the primal variable from the block system

$$\begin{bmatrix}\mathbf{M} & \mathbf{F}^\top\\ \mathbf{F} & \mathbf{0}\end{bmatrix}. \tag{68}$$

Intuitively, $\mathbf{S}$ measures interactions among the constraint normals under the geometry induced by $\mathbf{M}$. Equation 65 therefore computes the correction by solving in constraint space rather than directly in the original n-dimensional decision space.

E.5 MIXED STRICT-SOFT CORRECTION

The derivation above assumes $0 < w_k < 1$. The endpoint cases are handled as follows.

A zero-weight row has $\gamma_k = 0$ and contributes nothing to $\mathbf{F}^\top \mathbf{\Gamma} \mathbf{F}$. Such a row can therefore be omitted from the correction system. Equivalently, it corresponds formally to $d_k \to \infty$ in the Woodbury representation.

A unit-weight row has $\gamma_k \to \infty$ and is more naturally written as the hard constraint

$$\mathbf{F}_k \mathbf{J} = \mathbf{0}. \tag{69}$$

To combine strict and soft rows, define

$$\mathcal{S} = \{k : w_k = 1\}, \qquad \mathcal{Q} = \{k : 0 < w_k < 1\}, \tag{70}$$

and let $\mathbf{F}_\mathcal{S}$ and $\mathbf{F}_\mathcal{Q}$ contain the corresponding rows. The exact mixed formulation is

$$\begin{aligned} \mathbf{J}^\star = \arg\min_{\mathbf{J}} \quad & \frac{1}{2} \operatorname{tr}\big((\mathbf{J} - \mathbf{P})^\top \mathbf{M} (\mathbf{J} - \mathbf{P})\big) \\ & + \frac{1}{2} \operatorname{tr}\big(\mathbf{J}^\top \mathbf{F}_\mathcal{Q}^\top \mathbf{\Gamma}_\mathcal{Q} \mathbf{F}_\mathcal{Q} \mathbf{J}\big) \\ \text{subject to} \quad & \mathbf{F}_\mathcal{S} \mathbf{J} = \mathbf{0}. \end{aligned} \tag{71}$$

This formulation makes the semantics of the weights explicit:

- $w_k = 0$: row k is ignored;
- $0 < w_k < 1$: row k is penalized softly;
- $w_k = 1$: row k is enforced exactly.

Let

$$\mathbf{F}_\mathcal{R} = \begin{bmatrix} \mathbf{F}_\mathcal{S} \\ \mathbf{F}_\mathcal{Q} \end{bmatrix}, \qquad \mathbf{D}_\mathcal{R} = \operatorname{blkdiag}\left(\mathbf{0}, \mathbf{\Gamma}_\mathcal{Q}^{-1}\right). \tag{72}$$

The KKT conditions of Equation 71 yield

$$\boxed{\mathbf{J}^\star = \left[\mathbf{I} - \mathbf{M}^{-1} \mathbf{F}_\mathcal{R}^\top \left(\mathbf{F}_\mathcal{R} \mathbf{M}^{-1} \mathbf{F}_\mathcal{R}^\top + \mathbf{D}_\mathcal{R}\right)^\dagger \mathbf{F}_\mathcal{R}\right] \mathbf{P}.} \tag{73}$$

Thus, the compact constraint-space expression remains valid when strict and soft rows are combined, provided zero-weight rows are omitted and strict rows are assigned zero relaxation. The pseudoinverse accommodates dependencies among the strict rows; when the mixed Schur matrix is nonsingular, it reduces to the ordinary inverse.

Derivation of the mixed form. For completeness, introduce a multiplier matrix $\mathbf{\Lambda}_\mathcal{S}$ for the strict rows and define

$$\mathbf{U}_\mathcal{Q} = \mathbf{\Gamma}_\mathcal{Q} \mathbf{F}_\mathcal{Q} \mathbf{J}. \tag{74}$$

The stationarity condition for Equation 71 is

$$\mathbf{M}(\mathbf{J} - \mathbf{P}) + \mathbf{F}_\mathcal{S}^\top \mathbf{\Lambda}_\mathcal{S} + \mathbf{F}_\mathcal{Q}^\top \mathbf{U}_\mathcal{Q} = \mathbf{0}. \tag{75}$$

Stacking

$$\mathbf{U} = \begin{bmatrix} \mathbf{\Lambda}_\mathcal{S} \\ \mathbf{U}_\mathcal{Q} \end{bmatrix}, \tag{76}$$

gives

$$\mathbf{J} = \mathbf{P} - \mathbf{M}^{-1} \mathbf{F}_\mathcal{R}^\top \mathbf{U}. \tag{77}$$

The strict constraints and the definition of $\mathbf{U}_\mathcal{Q}$ together imply

$$\left(\mathbf{F}_\mathcal{R} \mathbf{M}^{-1} \mathbf{F}_\mathcal{R}^\top + \mathbf{D}_\mathcal{R}\right) \mathbf{U} = \mathbf{F}_\mathcal{R} \mathbf{P}. \tag{78}$$

The right-hand side lies in the range of the mixed Schur matrix, so the system is consistent even when redundant strict rows make its multiplier nonunique. Choosing the minimum-norm solution,

$$\mathbf{U} = \left(\mathbf{F}_\mathcal{R} \mathbf{M}^{-1} \mathbf{F}_\mathcal{R}^\top + \mathbf{D}_\mathcal{R}\right)^\dagger \mathbf{F}_\mathcal{R} \mathbf{P}, \tag{79}$$

and substituting it into Equation 77 proves Equation 73.

Projection versus soft correction. For finite weights, the operator

$$\mathcal{C}_{\mathbf{M},\mathbf{\Omega}} = \left(\mathbf{M} + \mathbf{F}^\top \mathbf{\Gamma} \mathbf{F}\right)^{-1} \mathbf{M}, \tag{80}$$

is generally not an exact projection. In particular, it need not satisfy

$$\mathcal{C}^2_{\mathbf{M},\mathbf{\Omega}} = \mathcal{C}_{\mathbf{M},\mathbf{\Omega}}, \tag{81}$$

or

$$\mathbf{F}\mathcal{C}_{\mathbf{M},\mathbf{\Omega}} = \mathbf{0}. \tag{82}$$

It instead shrinks components that violate heavily weighted constraints while retaining more of the prior along weakly weighted directions.

For $\mathbf{M} \succ 0$,

$$\lim_{\mathbf{\Omega}\to\mathbf{I}} \mathcal{C}_{\mathbf{M},\mathbf{\Omega}} = \Pi_{\mathbf{M}} = \mathbf{I} - \mathbf{M}^{-1}\mathbf{F}^\top \left(\mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top\right)^\dagger \mathbf{F}. \tag{83}$$

At the opposite endpoint,

$$\mathbf{\Omega} = \mathbf{0} \quad \Longrightarrow \quad \mathbf{\Gamma} = \mathbf{0}, \qquad \mathcal{C}_{\mathbf{M},\mathbf{\Omega}} = \mathbf{I}, \qquad \mathbf{J}^\star = \mathbf{P}. \tag{84}$$

Consequently, strict weighting recovers the metric projection, whereas zero weighting recovers the unmodified prior. This includes STE whenever $\mathbf{P}$ is chosen as the STE prior, irrespective of the constraints used in the forward problem.

Schur regularization and rank deficiency. The relaxation matrix $\mathbf{D}$ adds a positive diagonal term to the Schur matrix $\mathbf{S} = \mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top$. This is a form of *Tikhonov regularization* (Tikhonov, 1963): a possibly singular positive-semidefinite matrix is stabilized by adding a positive diagonal matrix before solving the linear system. In the all-soft case, $\mathbf{D} \succ 0$, and therefore

$$\mathbf{S} + \mathbf{D} \succ 0, \tag{85}$$

regardless of whether the rows of $\mathbf{F}$ are linearly independent.

In the mixed hard-soft case, $\mathbf{D}_\mathcal{R}$ contains zeros for strictly enforced rows. The soft relaxations regularize dependencies involving soft rows, but they do not remove dependencies among the strict rows themselves. Indeed, for any partitioned vector $\mathbf{y} = [\mathbf{y}_\mathcal{S}; \mathbf{y}_\mathcal{Q}]$,

$$\mathbf{y}^\top \left(\mathbf{F}_\mathcal{R}\mathbf{M}^{-1}\mathbf{F}_\mathcal{R}^\top + \mathbf{D}_\mathcal{R}\right)\mathbf{y} = \left\|\mathbf{M}^{-1/2}\mathbf{F}_\mathcal{R}^\top\mathbf{y}\right\|_2^2 + \mathbf{y}_\mathcal{Q}^\top \mathbf{\Gamma}_\mathcal{Q}^{-1}\mathbf{y}_\mathcal{Q}. \tag{86}$$

This quantity can vanish only if $\mathbf{y}_\mathcal{Q} = \mathbf{0}$ and $\mathbf{F}_\mathcal{S}^\top \mathbf{y}_\mathcal{S} = \mathbf{0}$. Hence, the mixed Schur matrix is positive definite when the strict rows are linearly independent. If strict rows are redundant, the mixed Schur matrix is singular, but the primal minimizer remains unique because $\mathbf{M} \succ 0$. The pseudoinverse in Eq. (73) selects one of the nonunique multiplier solutions without changing that primal minimizer.

E.6 Numerical Implementation

We implement the mixed strict-soft correction in constraint space. After omitting rows with nonfinite relaxation, we form

$$\mathbf{S} = \mathbf{F}\mathbf{M}^{-1}\mathbf{F}^\top + \mathbf{D}. \tag{87}$$

Let $\mathbf{G}$ denote the condition-aware generalized inverse described below. We compute

$$\mathbf{U} = \mathbf{G}\mathbf{F}\mathbf{P}, \qquad \mathbf{J} = \mathbf{P} - \mathbf{M}^{-1}\mathbf{F}^\top\mathbf{U}. \tag{88}$$

The backward pass applies the same correction directly to the incoming vector, without materializing $\mathbf{J}$; the dense matrix is formed only when the Jacobian is explicitly requested.

Before computing $\mathbf{G}$, we symmetrically scale $\mathbf{S}$ to unit diagonal wherever its diagonal is positive. Let

$$q_i = \begin{cases} \sqrt{S_{ii}}, & S_{ii} > 0, \\ 1, & S_{ii} \le 0, \end{cases} \qquad \mathbf{Q} = \operatorname{diag}(q_1, \ldots, q_m), \tag{89}$$

and define

$$\widehat{\mathbf{S}} = \mathbf{Q}^{-1}\mathbf{S}\mathbf{Q}^{-1}. \tag{90}$$

The fallback value $q_i = 1$ is defensive; the Schur matrix is theoretically positive semidefinite with a nonnegative diagonal. This scaling prevents large entries of $\mathbf{D}$ from dominating the numerical rank test.

Because $\widehat{\mathbf{S}}$ is symmetric, we assess its conditioning using the absolute values of its eigenvalues. Let

$$\rho_{\min} = \min_i \left|\lambda_i(\widehat{\mathbf{S}})\right|, \qquad \rho_{\max} = \max_i \left|\lambda_i(\widehat{\mathbf{S}})\right|, \tag{91}$$

and let ε denote machine precision in the working dtype. We classify the scaled system as numerically ill conditioned when

$$\rho_{\max}\sqrt{\varepsilon} > \rho_{\min}. \tag{92}$$

We then compute

$$\widehat{\mathbf{G}} = \begin{cases} \widehat{\mathbf{S}}^{-1}, & \text{if the scaled system is well conditioned,} \\ \widehat{\mathbf{S}}^{\dagger}, & \text{if it is rank deficient or numerically ill conditioned,} \end{cases} \qquad \mathbf{G} = \mathbf{Q}^{-1}\widehat{\mathbf{G}}\mathbf{Q}^{-1}. \tag{93}$$

Here † denotes the Hermitian Moore–Penrose pseudoinverse. The branch is selected independently for each matrix in a batch.

When $\mathbf{S}$ is singular, the scaled construction yields a reflexive generalized inverse of $\mathbf{S}$. Although this matrix need not equal the Moore–Penrose pseudoinverse of the unscaled $\mathbf{S}$, it produces the same unique primal correction; only the multipliers associated with redundant strict rows are nonunique. Thus, dependencies among strictly enforced rows are handled by the constraint-space linear algebra rather than by the adaptive relaxation.

For structurally represented transport constraints (Peyré & Cuturi, 2019), the marginal equalities have one known linear dependency. When the marginal block is undamped, we omit one redundant row before the constraint-space computation. This preserves its row space and therefore the resulting projector. When any marginal row is damped, all rows are retained because removing one would change the weighted operator.

Finally, a candidate inequality or bound row with zero dual multiplier has $d_k = +\infty$ and contributes nothing to the correction. More generally, any row whose computed relaxation is nonfinite is omitted before constructing $\mathbf{F}$ and $\mathbf{D}$, rather than represented by an arbitrarily large finite diagonal entry.

E.7 Comparison with dXPP

dXPP (Linghu et al., 2026) and our adaptive VSJ both use primal-dual information to construct a backward operator without changing the solver's forward solution. We compare them for the QP specialization of Eq. (1), with objective $\frac{1}{2}\mathbf{s}^\top\mathbf{H}\mathbf{s} + \mathbf{z}^\top\mathbf{s}$, $\mathbf{H} \succ 0$, and affine constraints. Let $\boldsymbol{\nu}$ and $\boldsymbol{\lambda} \geq \mathbf{0}$ denote the equality and inequality multipliers at $\mathbf{s}^\star$, and let $r_k = -h_k(\mathbf{s}^\star)$. The coefficient comparisons below assume matching constraint rows and scaling; VSJ normalizes each inequality's row, slack, and multiplier together as in Proposition 5.

Shared penalty magnitudes and slack-dependent stiffnesses. dXPP sets one penalty magnitude for equalities and another for inequalities using the largest multiplier magnitude in each group:

$$\rho = \zeta\|\boldsymbol{\nu}\|_\infty, \qquad \alpha = \zeta\|\boldsymbol{\lambda}\|_\infty, \qquad \zeta \geq 1. \tag{94}$$

Using the softplus function $p_\delta(t) = \delta\log(1+\exp(t/\delta))$, with smoothing parameter $\delta > 0$, it forms the penalty objective

$$\begin{aligned} \Phi_\delta(\mathbf{s};\mathbf{z}) = \tfrac{1}{2}\mathbf{s}^\top\mathbf{H}\mathbf{s} + \mathbf{z}^\top\mathbf{s} + \alpha\sum_{k=1}^{m_i} p_\delta(h_k(\mathbf{s})) \\ + \rho\sum_{j=1}^{m_e}\left[p_\delta(g_j(\mathbf{s})) + p_\delta(-g_j(\mathbf{s}))\right]. \end{aligned} \tag{95}$$

The coefficients of the constraint rows in its Hessian are the penalty second derivatives, not simply ρ and α. At the feasible QP solution, they are

$$\begin{aligned} \gamma^{\mathrm{eq}}_{j,\delta} &= 2\rho p''_\delta(0) = \frac{\rho}{2\delta}, \\ \gamma^{\mathrm{ineq}}_{k,\delta} &= \alpha p''_\delta(-r_k) = \frac{\alpha}{\delta}\frac{\exp(-r_k/\delta)}{\left(1+\exp(-r_k/\delta)\right)^2}. \end{aligned} \tag{96}$$

Thus dXPP uses each inequality's slack through the smooth penalty curvature, while its multiplier dependence enters through the shared maximum in Eq. (94). These coefficients are stiffnesses, corresponding to $\gamma_k = w_k/(1 - w_k)$ in VSJ, rather than bounded weights w_k. For $\alpha > 0$, an inactive inequality has positive stiffness at finite smoothing, decreasing with its slack. dXPP also supports active-set pruning, which omits these inactive contributions.

Comparison with adaptive enforcement. Our adaptive VSJ instead uses $\gamma_k = \lambda_k/(r_k + \mu)$ from Eq. (48) and imposes equality rows exactly through Eq. (71). At an exact KKT point, $r_k\lambda_k = 0$, giving the cases in Tab. 7. In particular, dXPP assigns the same coefficient to all active inequalities, whereas adaptive VSJ distinguishes them by their individual multipliers. An active zero-dual row is omitted by adaptive VSJ but can still be penalized by dXPP when another inequality has a positive multiplier.

Table 7: Constraint treatment in the backward operator at an exact KKT point, for matching constraint scaling. Entries are stiffnesses, not bounded VSJ weights. The dXPP column describes the unpruned construction; pruning removes inactive rows.

Constraint state	dXPP	Adaptive VSJ
Equality	$\rho/(2\delta)$	Exact preservation
Active inequality, $\lambda_k > 0$	$\alpha/(4\delta)$	λ_k/μ
Active inequality, $\lambda_k = 0$	$\alpha/(4\delta)$	0
Inactive inequality, $r_k > 0$	$\alpha p'_\delta(-r_k)$	0

For approximate primal-dual states, both r_k and λ_k may be positive, so slack also weakens a positive adaptive VSJ stiffness. Both therefore use slack and dual information, but dXPP combines slack-dependent curvature with a shared dual scale, while adaptive VSJ uses each inequality's own multiplier and slack. The equality distinction concerns the backward response: both leave the forward constraints unchanged.

Operator recovery and independent design choices. dXPP evaluates the penalty derivatives at the original QP solution rather than solving the smoothed penalty problem. Let $\mathbf{\Gamma}_\delta$ collect the coefficients in Eq. (96), and let $\mathbf{F}$ contain the corresponding equality and inequality rows. Holding the other QP data and the penalty parameters fixed during differentiation, its plug-in linear-cost Jacobian satisfies

$$\left(\mathbf{H} + \mathbf{F}^\top \mathbf{\Gamma}_\delta \mathbf{F}\right) \mathbf{J}_{\mathrm{dXPP}} = -\mathbf{I}. \tag{97}$$

For matching rows and coefficients, the general finite-weight VSJ construction recovers this operator with

$$\mathbf{M} = \mathbf{H}, \qquad \mathbf{P} = -\mathbf{H}^{-1}, \qquad \mathbf{\Gamma} = \mathbf{\Gamma}_\delta, \qquad w_k = \frac{\gamma_k}{1 + \gamma_k}. \tag{98}$$

Indeed, $\mathbf{MP} = -\mathbf{I}$, so Eq. (58) becomes Eq. (97). Zero-coefficient or pruned rows are omitted. This recovery uses finite penalties on equality rows as well as inequalities; it is therefore a configuration of the general weighted framework, not the default adaptive rule with strict equalities. dXPP couples the prior and metric through objective curvature, whereas VSJ allows them to be chosen independently.

This comparison identifies the local cost operator. Differentiation with respect to parameter-dependent objective curvature or constraints introduces additional right-hand-side terms in dXPP and is outside this recovery. dXPP targets an approximation to the exact QP sensitivity and analyzes convergence as $\delta \to 0$ under its stated regularity assumptions. VSJ instead permits independently chosen surrogate configurations, which need not be exact derivatives of the solver's solution map. The operator match does not imply equivalence of complete training procedures.

F Recovery of Existing Backward Operators

This section details the recoveries summarized in Tab. 1. We use the minimization-cost convention and compare local backward operators at the same forward solution and fixed selected active manifold. These recoveries do not imply equivalence of the corresponding forward problems or training procedures.

Identity and STE. As a boundary configuration, $\mathbf{\Omega} = \mathbf{0}$ omits every constraint row, so the VSJ solution equals its prior. Choosing

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{I}, \mathbf{I}, \mathbf{0}), \qquad \mathbf{J}^\star = -\mathbf{I}, \tag{99}$$

recovers the Identity update (Sahoo et al., 2023) and the classical straight-through rule under the minimization-cost convention (Hinton, 2012; Bengio et al., 2013). Setting $\mathbf{\Omega} = \mathbf{I}$ instead recovers the projection variant of the same work, which removes the component of $-\mathbf{I}$ normal to the solution hyperplane. Up to the scale $1/\tau$, this is the SparseMAP configuration $(-\tau^{-1}\mathbf{I}, \tau\mathbf{I}, \mathbf{I})$, so its results are the SparseMAP native and matched rows of Tabs. 4 and 9. VSJ adds the metric and the per-row enforcement strength as independent choices on top of it. Because no constraint correction is applied, the metric is immaterial.

Regularizer-induced KKT sensitivities. Consider Eq. (1) on a fixed active manifold represented locally by affine constraints $\mathbf{F}\mathbf{s} = \mathbf{b}$. Let $\boldsymbol{\nu}^\star$ denote their multipliers and suppose

$$\mathbf{H}_R = \nabla^2 R(\mathbf{s}^\star) \succ \mathbf{0}. \tag{100}$$

The local KKT conditions are

$$\mathbf{z} + \nabla R(\mathbf{s}^\star) + \mathbf{F}^\top \boldsymbol{\nu}^\star = \mathbf{0}, \qquad \mathbf{F}\mathbf{s}^\star = \mathbf{b}. \tag{101}$$

Differentiating with respect to $\mathbf{z}$ gives

$$\mathbf{H}_R\mathbf{J} + \mathbf{I} + \mathbf{F}^\top\mathbf{B} = \mathbf{0}, \qquad \mathbf{F}\mathbf{J} = \mathbf{0}, \qquad \mathbf{B} = \frac{\partial \boldsymbol{\nu}^\star}{\partial \mathbf{z}}. \tag{102}$$

If $\mathbf{F}$ has full row rank, eliminating $\mathbf{B}$ yields

$$\frac{\partial \mathbf{s}^\star}{\partial \mathbf{z}} = -\mathbf{H}_R^{-1} + \mathbf{H}_R^{-1}\mathbf{F}^\top \left(\mathbf{F}\mathbf{H}_R^{-1}\mathbf{F}^\top\right)^{-1} \mathbf{F}\mathbf{H}_R^{-1}. \tag{103}$$

This is exactly the strict VSJ solution obtained from

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{H}_R^{-1}, \mathbf{H}_R, \mathbf{I}). \tag{104}$$

For dependent rows, replacing the inverse in Eq. (103) by the Moore–Penrose pseudoinverse gives the same unique primal Jacobian, although the multiplier derivative need not be unique.

This is the classical equality-constrained sensitivity of Domke (2008), stated there for a general convex objective; a linear cost gives $\partial^2 F / \partial \mathbf{s}\, \partial \mathbf{z} = \mathbf{I}$ and reduces it to the form above. That statement assumes $\mathbf{F}$ has full row rank; Theorem 3 replaces the inverse by a pseudoinverse and admits dependent rows.

SparseMAP. In the direct-feature specialization considered here, SparseMAP applies quadratic regularization to a structured marginal $\mathbf{s} \in \mathcal{M}$, where $\mathcal{M} = \mathrm{conv}(\mathcal{Y})$:

$$\mathbf{s}^\star(\mathbf{z}) = \arg\min_{\mathbf{s} \in \mathcal{M}} \left\{ \mathbf{z}^\top \mathbf{s} + \frac{\tau}{2}\|\mathbf{s}\|_2^2 \right\}, \qquad \tau > 0. \tag{105}$$

Here $\mathbf{z}$ is the negative of the conventional SparseMAP score vector. Since $\mathbf{H}_R = \tau\mathbf{I}$, Eq. (104) becomes

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = \left(-\tau^{-1}\mathbf{I}, \tau\mathbf{I}, \mathbf{I}\right). \tag{106}$$

For a fixed active face, this recovers the SparseMAP Jacobian when the selected rows of $\mathbf{F}$ span the normal space of that face, equivalently when $\ker(\mathbf{F})$ equals its tangent space. This qualification is needed because SparseMAP identifies the face through active structures, whereas VSJ represents it through constraint rows.

OptNet. OptNet differentiates quadratic programs with objective (Amos & Kolter, 2017)

$$\frac{1}{2}\mathbf{s}^\top \mathbf{Q}_\theta \mathbf{s} + \mathbf{z}^\top \mathbf{s}, \qquad \mathbf{Q}_\theta \succ \mathbf{0}. \tag{107}$$

Taking $\mathbf{H}_R = \mathbf{Q}_\theta$ gives

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{Q}_\theta^{-1}, \mathbf{Q}_\theta, \mathbf{I}). \tag{108}$$

Thus, VSJ recovers the derivative with respect to the linear cost $\mathbf{z}$ when the curvature and constraint data are held fixed. OptNet parameterizes learnable positive-definite curvature as

$$\mathbf{Q}_\theta = \mathbf{L}\mathbf{L}^\top + \delta\mathbf{I}, \qquad \delta > 0. \tag{109}$$

Derivatives with respect to $\mathbf{L}$ or other QP parameters are additional layer-parameter derivatives and are not represented by the cost Jacobian in Eq. (108).

Entropy regularization. Entropy-regularized prediction uses (Agrawal et al., 2019a)

$$R_{\mathrm{Ent}}(\mathbf{s}) = \epsilon \sum_i s_i(\log s_i - 1), \qquad \mathbf{H}_{\mathrm{Ent}} = \epsilon \operatorname{diag}\left(\frac{1}{s_i^\star}\right), \qquad s_i^\star > 0. \tag{110}$$

Its strict VSJ configuration is

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{H}_{\mathrm{Ent}}^{-1}, \mathbf{H}_{\mathrm{Ent}}, \mathbf{I}). \tag{111}$$

On the simplex, $\mathbf{s}^\star = \mathrm{softmax}(-\mathbf{z}/\epsilon)$, and Eq. (103) reduces to

$$\frac{\partial \mathbf{s}^\star}{\partial \mathbf{z}} = -\frac{1}{\epsilon}\left(\operatorname{diag}(\mathbf{s}^\star) - \mathbf{s}^\star(\mathbf{s}^\star)^\top\right). \tag{112}$$

Thus, the softmax cost Jacobian is a strict curvature-induced VSJ configuration.

The same configuration applies verbatim when the feasible set is a transport polytope rather than a simplex. There the entropic problem is entropic optimal transport, whose solution is computed by Sinkhorn iterations (Cuturi, 2013; Peyré & Cuturi, 2019), and $\mathbf{H}_{\mathrm{Ent}}$ retains the same diagonal form. We state this only as a consequence of the derivation: it is outside our benchmark set and we do not verify it numerically, so we do not list it among the recoveries in Tab. 1.

Log-barrier KKT sensitivity. The log-barrier formulation considered by Mandi & Guns (2020) uses

$$R_{\mathrm{Log}}(\mathbf{s}) = -\mu_{\mathrm{Log}} \sum_i \log s_i, \qquad \mathbf{H}_{\mathrm{Log}} = \operatorname{diag}\left(\frac{\mu_{\mathrm{Log}}}{(s_i^\star)^2}\right), \qquad s_i^\star > 0. \tag{113}$$

Applying Eq. (104) gives

$$(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{H}_{\mathrm{Log}}^{-1}, \mathbf{H}_{\mathrm{Log}}, \mathbf{I}). \tag{114}$$

Here positivity is encoded by the barrier, while $\mathbf{F}$ contains the remaining affine constraints. This configuration recovers the local log-barrier KKT sensitivity, but not the homogeneous self-dual backward system used by the full IntOpt method.

PEAR: base sensitivity and regret gradient. PEAR (Lee et al., 2026, Secs. 3.3–3.4) obtains its base regret gradient by projecting curvature-scaled prediction error. We derive its relation to the cost Jacobian and the configurations in Tab. 1.

Setting and assumptions. Consider Eq. (1) with fixed affine constraints and a twice continuously differentiable convex regularizer R independent of $\mathbf{z}$. Let $\mathbf{z}$ denote predicted costs, $\mathbf{c}$ the true costs, and $\mathbf{e} = \mathbf{z} - \mathbf{c}$. At $\mathbf{s}^\star = \mathbf{s}^\star(\mathbf{z})$, assume $\mathbf{H}_R = \nabla^2 R(\mathbf{s}^\star) \succ \mathbf{0}$, independent equality and active-inequality rows in $\mathbf{F}$, and strict complementarity for the active inequalities. These conditions give a differentiable solution map with a locally fixed active set. The derivatives below hold R and all constraint data fixed; $\mathbf{H}_R$ is evaluated at the current solution and need not be constant across solutions.

Cost sensitivity. Writing $\mathbf{J}_R = \partial \mathbf{s}^\star / \partial \mathbf{z}$ and $\mathbf{B} = \partial \boldsymbol{\nu}^\star / \partial \mathbf{z}$, the differentiated KKT conditions in Eq. (102) are

$$\begin{bmatrix} \mathbf{H}_R & \mathbf{F}^\top \\ \mathbf{F} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{J}_R \\ \mathbf{B} \end{bmatrix} = -\begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}. \tag{115}$$

The first block gives $\mathbf{J}_R = -\mathbf{H}_R^{-1} - \mathbf{H}_R^{-1}\mathbf{F}^\top \mathbf{B}$. Substitution into $\mathbf{F}\mathbf{J}_R = \mathbf{0}$ yields

$$\mathbf{B} = -\left(\mathbf{F}\mathbf{H}_R^{-1}\mathbf{F}^\top\right)^{-1} \mathbf{F}\mathbf{H}_R^{-1}. \tag{116}$$

Consequently,

$$\begin{aligned} \mathbf{J}_R &= -\mathbf{K}_R, \\ \mathbf{K}_R &= \mathbf{H}_R^{-1} - \mathbf{H}_R^{-1}\mathbf{F}^\top \left(\mathbf{F}\mathbf{H}_R^{-1}\mathbf{F}^\top\right)^{-1} \mathbf{F}\mathbf{H}_R^{-1} = \Pi_{\mathbf{H}_R}\mathbf{H}_R^{-1}. \end{aligned} \tag{117}$$

The expanded expression shows that $\mathbf{K}_R^\top = \mathbf{K}_R$, while $\mathbf{F}\mathbf{K}_R = \mathbf{0}$. The projector is $\Pi_{\mathbf{H}_R}$; $\mathbf{K}_R$ also includes inverse-curvature scaling and is not generally idempotent. Redundant constraint rows can be retained by replacing the Schur inverse with its Moore–Penrose pseudoinverse, leaving the unique primal sensitivity unchanged. With no constraint rows, $\mathbf{K}_R = \mathbf{H}_R^{-1}$.

From sensitivity to regret gradient. Evaluate regret using the same regularized objective as in the forward problem:

$$\begin{aligned}\ell_{\mathrm{reg}}(\mathbf{z},\mathbf{c}) = \mathbf{c}^\top \mathbf{s}^\star(\mathbf{z}) + R\big(\mathbf{s}^\star(\mathbf{z})\big) \\ - \mathbf{c}^\top \mathbf{s}^\star(\mathbf{c}) - R\big(\mathbf{s}^\star(\mathbf{c})\big).\end{aligned} \tag{118}$$

The true-cost optimum contributes a term constant in $\mathbf{z}$ and need not share the predicted solution's active set. At the predicted solution, stationarity gives $\mathbf{c} + \nabla R(\mathbf{s}^\star) = -\mathbf{e} - \mathbf{F}^\top \boldsymbol{\nu}^\star$. The chain rule and $\mathbf{F}\mathbf{J}_R = \mathbf{0}$ therefore imply

$$\begin{aligned}\mathbf{g}_{\mathrm{PEAR}} := \nabla_{\mathbf{z}} \ell_{\mathrm{reg}} &= \mathbf{J}_R^\top \big(\mathbf{c} + \nabla R(\mathbf{s}^\star)\big) \\ &= -\mathbf{J}_R^\top \mathbf{e} - \left(\mathbf{F}\mathbf{J}_R\right)^\top \boldsymbol{\nu}^\star \\ &= -\mathbf{J}_R^\top \mathbf{e} = \mathbf{K}_R \mathbf{e} = \Pi_{\mathbf{H}_R} \mathbf{H}_R^{-1} \mathbf{e}.\end{aligned} \tag{119}$$

Thus PEAR projects the inverse-curvature-scaled prediction error, not regret itself. Including the same R in Eq. (118) is essential: for a purely linear downstream loss, the gradient is instead $\mathbf{J}_R^\top \mathbf{c}$, which generally differs from the PEAR base rule.

VSJ configuration. Choosing

$$\begin{aligned}(\mathbf{P},\mathbf{M},\boldsymbol{\Omega}) = (-\mathbf{H}_R^{-1}, \mathbf{H}_R, \mathbf{I}), \\ \mathbf{J}^\star = -\Pi_{\mathbf{H}_R} \mathbf{H}_R^{-1} = \mathbf{J}_R, \qquad \mathbf{g}_{\mathrm{PEAR}} = -(\mathbf{J}^\star)^\top \mathbf{e}\end{aligned} \tag{120}$$

recovers the underlying sensitivity and, with the same regret, the base PEAR update. Equivalently, completing the square gives the vector-level characterization

$$\mathbf{g}_{\mathrm{PEAR}} = \arg\min_{\mathbf{u}:\, \mathbf{F}\mathbf{u} = \mathbf{0}} \frac{1}{2} \left(\mathbf{u} - \mathbf{H}_R^{-1}\mathbf{e}\right)^\top \mathbf{H}_R \left(\mathbf{u} - \mathbf{H}_R^{-1}\mathbf{e}\right). \tag{121}$$

This is the same metric projection as strict VSJ, with the prior and metric fixed by forward curvature. VSJ makes those choices and constraint enforcement independently configurable; the recovery does not assert a new projection mechanism.

Quadratically smoothed LPs. For LPs, PEAR adds $R(\mathbf{s}) = \frac{\lambda}{2}\|\mathbf{s}\|_2^2$, $\lambda > 0$, to the forward objective (Lee et al., 2026, Sec. 4.3). Hence

$$\begin{aligned}\mathbf{H}_R = \lambda \mathbf{I}, \qquad (\mathbf{P},\mathbf{M},\boldsymbol{\Omega}) = (-\lambda^{-1}\mathbf{I}, \lambda\mathbf{I}, \mathbf{I}), \\ \mathbf{g}_{\mathrm{PEAR}} = \lambda^{-1}\left[\mathbf{I} - \mathbf{F}^\top (\mathbf{F}\mathbf{F}^\top)^\dagger \mathbf{F}\right] \mathbf{e}.\end{aligned} \tag{122}$$

This recovers the smoothed problem's base gradient, not the derivative of the original unregularized LP. If $\ker(\mathbf{F}) = \{\mathbf{0}\}$, both base PEAR and strict VSJ vanish; choosing a different prior or metric alone cannot prevent this.

Quadratic objectives. For $R(\mathbf{s}) = \frac{1}{2}\mathbf{s}^\top \mathbf{Q}_\theta \mathbf{s}$ with $\mathbf{Q}_\theta \succ \mathbf{0}$ held fixed for cost differentiation, Eq. (120) gives $(-\mathbf{Q}_\theta^{-1}, \mathbf{Q}_\theta, \mathbf{I})$, the quadratic-base PEAR configuration in Tab. 1.

PEAR's mean-variance portfolio problem (Lee et al., 2026, App. C.3), written with negative predicted returns as minimization costs, uses

$$\begin{aligned}R(\mathbf{s}) = \frac{\lambda}{2}\mathbf{s}^\top \boldsymbol{\Sigma}\mathbf{s}, \qquad \mathbf{H}_R = \lambda\boldsymbol{\Sigma}, \\ (\mathbf{P},\mathbf{M},\boldsymbol{\Omega}) = \left(-(\lambda\boldsymbol{\Sigma})^{-1}, \lambda\boldsymbol{\Sigma}, \mathbf{I}\right).\end{aligned} \tag{123}$$

Here $\lambda > 0$ is the risk-aversion parameter, distinct from the LP smoothing parameter, and $\boldsymbol{\Sigma} \succ \mathbf{0}$ is the covariance matrix. Both are held fixed when differentiating the linear costs; derivatives through a predicted covariance are not included. Substitution into Eq. (119) gives $\mathbf{g}_{\mathrm{PEAR}} = \Pi_{\lambda\boldsymbol{\Sigma}}(\lambda\boldsymbol{\Sigma})^{-1}\mathbf{e}$. The single quadratic-base PEAR row covers both cases, with $\mathbf{Q}_\theta = \lambda\mathbf{I}$ or $\lambda\boldsymbol{\Sigma}$.

Scope of recovery. The identities above concern the undamped, unclipped base rule, with normal injection and any additional prediction-error term disabled. PEAR's LP normal injection adds a normal component with an error-dependent scale, so the resulting rule is generally nonlinear in $\mathbf{e}$ and need not satisfy $\mathbf{F}\mathbf{g} = \mathbf{0}$. It is therefore not recovered by this fixed strict Jacobian. Such a native update can instead be supplied to the separate correction in Eq. (11), without applying the prior a second time. For the exact base, however, same-row composition is the identity for every admissible metric and weight choice (Sec. G.3). The derivation also does not assert differentiability at active-set changes or weakly active constraints, nor equivalence of complete forward algorithms or training procedures.

G Further Details on Experiments and Results

G.1 Formulation of the Benchmarks

Following Eq. (1), we collect the formulations and predictor details for the main-paper benchmarks, including the PEAR tasks. Evaluation metrics and protocols are provided in Sec. G.2. The MNIST matching and Pascal VOC-U graph-matching experiments are reported in Secs. G.6 and G.7, including their problem formulations.

Continuous Fractional Knapsack (Mandi et al., 2024).

$$\begin{aligned} \max \quad & \sum_{i=1}^{n} z_i s_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i s_i \le W \\ & 0 \le s_i \le 1 \quad \forall i \in \{1, \dots, n\} \end{aligned} \tag{124}$$

The Knapsack formulation (Equation 124) maximizes the total value $z_{i,j}$ of selected items subject to a strict capacity limit W, where each item has a weight w_i. Unlike the binary version, the decision variables s_i are bounded continuously between 0 and 1 to allow for partial inclusion of items. The predictor is a single linear layer $\mathbb{R}^8 \to \mathbb{R}$, applied independently to each item's eight features.

Binary Knapsack. The PEAR variant (Lee et al., 2026) uses 100 items and five capacity inequalities, $\sum_i w_{ji} s_i \le b_j$ for $j = 1, \dots, 5$, with $b_j = \frac{1}{2} \sum_i w_{ji}$. A linear $5 \to 100$ predictor maps synthetic instance features to item profits under degree-8 data generation. Training uses a continuous surrogate with $0 \le s_i \le 1$; validation and test decisions enforce $s_i \in \{0, 1\}$. This is a separate prediction problem, not a change of integrality on the fractional benchmark's data.

Travelling Salesman Problem (Pogančić et al., 2020).

$$\begin{aligned} \min \quad & \sum_{i \in V} \sum_{j \in V, j \ne i} z_{ij} s_{ij} \\ \text{s.t.} \quad & \sum_{j \in V, j \ne i} s_{ij} = 1 \quad \forall i \in V \\ & \sum_{i \in V, i \ne j} s_{ij} = 1 \quad \forall j \in V \\ & \sum_{i \in S} \sum_{j \in S, j \ne i} s_{ij} \le |S| - 1 \quad S \subsetneq V, 2 \le |S| \le |V| - 1 \\ & s_{ij} \in \{0, 1\} \quad \forall i, j \in V, i \ne j \end{aligned} \tag{125}$$

The Dantzig–Fulkerson–Johnson (Dantzig et al., 1954) formulation in Eq. (125) ensures that each node is visited exactly once and eliminates disconnected subtours through dynamically generated subtour constraints. Here, $s_{i,j}$ denotes edge inclusion and $z_{i,j}$ is the corresponding edge cost. The binary restrictions enforce discrete tours, and solving the resulting TSP is NP-hard. The forward decision is the integer optimum, while VSJ constructs its backward geometry from a continuous relaxation of the final DFJ model; this does not alter the forward tour.

For $k = 40$, the ConvNet processes each $3 \times 20 \times 40$ flag image independently using a 4×4 stride-2 convolution with 20 channels, 2×2 max pooling, a 4×4 stride-2 convolution with 50 channels, 4×4 adaptive average pooling, and fully connected layers $800 \to 500 \to 3$. Pairwise distances between the resulting 3D coordinates define the edge costs. We train with Adam using learning rate 5×10^{-5} and $(\beta_1, \beta_2) = (0.5, 0.999)$.

Synthetic Portfolio Optimisation (Mandi et al., 2024).

$$\begin{aligned} \max \quad & \sum_{i=1}^{50} z_i s_i \\ \text{s.t.} \quad & \sum_{i=1}^{50}\sum_{j=1}^{50} s_i \sigma_{ij} s_j \le \gamma \\ & \sum_{i=1}^{50} s_i \le 1 \\ & s_i \ge 0 \quad \forall i \in \{1,\dots,50\} \end{aligned} \tag{126}$$

The portfolio problem in Eq. (126) maximizes predicted return subject to a quadratic risk limit γ, a total-allocation budget inequality, and nonnegative allocations. The covariance matrix $(\sigma_{ij})_{i,j=1}^{50}$, risk limit γ, and feasible set are fixed independently of the predicted returns $\mathbf{z}$, which enter only through the linear objective. The predictor is a single linear layer $\mathbb{R}^5 \to \mathbb{R}^{50}$ without batch normalization. The reported VSJ configurations omit the quadratic constraint's Lagrangian curvature, so Portfolio tests an empirical surrogate beyond affine geometry rather than exact nonlinear KKT recovery.

Time-Series Portfolio Optimisation (Hwang et al., 2025).

We use PEAR's evaluation setup (Lee et al., 2026) with daily returns for 89 assets. A DLinear predictor maps 63-day histories to 21-day return forecasts, whose mean gives $\mathbf{z}$. Combining historical and forecast returns gives a sample covariance, shifted to have minimum eigenvalue at least 10^{-4} before computing its Cholesky factor $\mathbf{L}$. The training forward uses $\widehat{\boldsymbol{\Sigma}} = \mathbf{L}\mathbf{L}^\top + 10^{-6}\mathbf{I}$ and solves the long-only mean-variance problem

$$\begin{aligned} \max_{\mathbf{s}\in\mathbb{R}^{89}} \quad & \mathbf{z}^\top \mathbf{s} - \frac{\lambda}{2}\mathbf{s}^\top \widehat{\boldsymbol{\Sigma}}\mathbf{s} \qquad\qquad \lambda = 2. \\ \text{s.t.} \quad & \mathbf{1}^\top \mathbf{s} = 1, \qquad \mathbf{s} \ge \mathbf{0}, \end{aligned} \tag{127}$$

Unlike synthetic Portfolio, risk is penalized in the objective rather than constrained by a fixed upper bound, and the covariance depends on the forecasts. The mean-only backward rules hold this covariance fixed. PEAR's dataset is not publicly released; we rebuild it from the same public daily-return universe following the protocol described in that work. Chronological splits, realized-parameter regret, economic backtesting, and method configurations are specified in Sec. G.2.

G.2 Evaluation Metrics and Protocols

Controlled comparisons and selection. We use official implementations (Pogančić et al. (2020) for TSP; Mandi et al. (2024) for fractional Knapsack and synthetic Portfolio) where available and otherwise faithful reproductions of published formulations. Within each controlled benchmark, all methods share the train/validation/test split, predictor, optimizer, learning-rate candidates, training schedule, and seeds. For these controlled comparisons, learning rates and VSJ configurations minimize mean validation regret; test scores are not used for selection. Ties at the reported validation precision follow the fixed grid order.

PEAR comparison configurations. The recovery control retains PEAR's native numerical conventions, including normal injection on Knapsack; Portfolio uses no normal injection. In Tab. 6, native Adaptive VSJ uses $\mathbf{P} = \mathbf{H}^{-1}$ and $\mathbf{M} = \mathbf{H}$, with the positive prior appropriate to predicted-profit/return maximization. Knapsack uses $\mathbf{H} = 0.1\mathbf{I}$, learning rate 0.003, bound temperature $\mu = 0.01$, and capacity temperature 0.1. Portfolio uses $\mathbf{H} = 2(\mathbf{L}\mathbf{L}^\top + 10^{-6}\mathbf{I})$, where $\mathbf{L}$ is the predicted covariance factor, learning rate 0.001, and $\mu = 0.01$; the forward covariance and local geometry are detached in the mean-only backward rule. The five-seed strict controls retain these priors, metrics, and all non-weight settings without further tuning. They use the same adaptive certificate row selector, omit zero-dual inequalities, and enforce retained rows exactly without Schur ridge or normal injection; a null-space implementation avoids cancellation when the selected rows span the decision space. Adaptive VSJ-PEAR uses $\mathbf{M}_{\text{cap}}$, $\mu = 1$, and learning rate 0.003 on Knapsack, retaining PEAR's source normal injection. To define this metric, unit-normalize

each capacity row $\mathbf{a}_i$ and set $p_j = \max(\sum_i a_{ij}^2, 10^{-12})$, $q_j = p_j^{-1}$, and $\bar{q} = n^{-1}\sum_j q_j$; then $(\mathbf{M}_{\text{cap}}^{-1})_{jj} = 10q_j/\bar{q}$. This uses constraints only and matches the mean inverse-metric scale of $10\mathbf{I}$. The displayed Portfolio correction uses the covariance metric $\mathbf{H}$, $\mu = 0.1$, learning rate 0.001, and no normal injection. In Tab. 6, Knapsack runs train until validation early stopping. Under a matched protocol, in which every arm trains under PEAR's 600 s cap and selects its learning rate from $\{0.003, 0.01, 0.03\}$ (and, for VSJ, $\mu \in \{0.01, 0.1, 1\}$) by mean validation regret over seeds 0–9, PEAR reaches 0.4416 ± 0.0446, Adaptive VSJ 0.3865 ± 0.0320 (-12.5%, paired $p = 0.0004$) and Adaptive VSJ-PEAR 0.4106 ± 0.0286 (-7.0%, $p = 0.005$); both improve on all ten seeds and remain significant after Holm correction.

Relative and absolute regret. For instance i, let $\hat{\mathbf{s}}_i = \mathbf{s}(\hat{\mathbf{c}}_i)$, $\mathbf{s}_i^\star = \mathbf{s}^\star(\mathbf{c}_i)$, and $v_i^\star = \mathbf{c}_i^\top \mathbf{s}_i^\star$. The true-objective gap is $g_i = \sigma[\mathbf{c}_i^\top \hat{\mathbf{s}}_i - v_i^\star]$, with $\sigma = +1$ for minimization and $\sigma = -1$ for maximization. Following the controlled benchmark convention (Mandi et al., 2024), we report

$$\text{Regret}_{\text{rel}} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \frac{g_i}{|v_i^\star|}, \qquad \text{Regret}_{\text{abs}} = \frac{1}{N}\sum_{i=1}^{N} g_i, \qquad \mathcal{I} = \{i : |v_i^\star| > 10^{-10}\}. \tag{128}$$

TSP and fractional Knapsack use relative regret as a ratio, without multiplication by 100. Synthetic Portfolio uses absolute regret because its optimal objective can be zero. The appendix MNIST control also uses relative regret. Binary Knapsack instead follows PEAR's ratio-of-sums convention,

$$\text{Regret}_{\text{binary}} = 100\,\frac{\sum_{i=1}^{N} g_i}{\sum_{i=1}^{N} |v_i^\star| + 10^{-7}}, \tag{129}$$

with binary oracle decisions and maximization gaps. This percentage is not the mean of instance-wise relative regrets; the fractional and binary columns therefore have distinct aggregation and scale.

Time-series portfolio regret. For the task defined in Eq. (127), the PEAR protocol (Lee et al., 2026) uses chronological $70/10/20$ percent splits. Validation uses rolling windows; held-out evaluation takes the first 63 test days as history, then advances in nonoverlapping 21-day investment periods, discarding any incomplete final period. For period k, the predicted portfolio $\hat{\mathbf{s}}_k$ is optimized using forecast returns and covariance, while the oracle $\mathbf{s}_k^\star$ uses realized future returns. Both are scored with the same realized-parameter utility

$$u_k(\mathbf{s}) = \boldsymbol{\mu}_k^\top \mathbf{s} - \frac{\lambda}{2}\mathbf{s}^\top \boldsymbol{\Sigma}_k \mathbf{s}, \qquad \text{Regret}_{\text{portfolio}} = 100\,\frac{\sum_k [u_k(\mathbf{s}_k^\star) - u_k(\hat{\mathbf{s}}_k)]}{\sum_k |u_k(\mathbf{s}_k^\star)| + 10^{-7}}. \tag{130}$$

Here $\lambda = 2$, $\mathbf{s} \geq \mathbf{0}$, and $\mathbf{1}^\top \mathbf{s} = 1$. The mean uses the 21 future days; covariance uses the combined 63-day history and 21-day future window with sample normalization and a diagonal shift $\max(10^{-4} - \lambda_{\min}(\boldsymbol{\Sigma}), 0)\mathbf{I}$. Forecast statistics replace the future window by model predictions. The regret score excludes transaction costs; these enter only the economic backtest. Comparisons concern methods evaluated on the same supplied data described in Sec. G.1.

Economic portfolio metrics. The backtest starts from uniform holdings, rebalances every 21 days, and lets weights drift with daily asset returns between rebalances. Following the released evaluator, nonnegative weights are renormalized and turnover costs $0.001\|\mathbf{s}_{\text{new}} - \mathbf{s}_{\text{old}}\|_1$ multiply the latest stored wealth by one minus this cost. Let $V_0, \ldots, V_T$ be that cost-adjusted wealth series, $r_0 = 0$, $r_t = V_t/V_{t-1} - 1$ for $t \geq 1$, $\bar{r} = (T+1)^{-1}\sum_{t=0}^{T} r_t$, and s_r its sample standard deviation. We report

$$\begin{aligned} R_{\text{cum}} &= 100\left(\frac{V_T}{V_0} - 1\right), \quad R_{\text{ann}} = 100\big[(1+\bar{r})^{252} - 1\big], \\ \text{Sharpe} &= \frac{(1+\bar{r})^{252} - 1}{\sqrt{252}\, s_r}, \quad \text{MDD} = 100 \min_{0\leq t\leq T}\left(\frac{V_t}{\max_{0\leq u\leq t} V_u} - 1\right). \end{aligned} \tag{131}$$

The risk-free rate is zero; Sharpe is set to zero when volatility is zero. This native annualized return is based on the mean daily return, not CAGR. MDD is signed and nonpositive, so closer to zero is better; returns and Sharpe are also higher-is-better. The native evaluator charges each rebalance to the preceding stored wealth value, including V_0; consequently the initial transaction cost cancels from V_T/V_0. These conventions are retained rather than replaced by alternative financial-metric definitions.

G.3 Attribution of the Portfolio PEAR Composition

Exact invariance and implemented geometry. Let $\mathbf{F}_{\mathrm{P}}$ and $\mathbf{F}_{\mathrm{V}}$ denote PEAR's and VSJ's selected rows. The exact PEAR base $\mathbf{g}_0$ in Eq. (119) satisfies $\mathbf{F}_{\mathrm{P}}\mathbf{g}_0 = \mathbf{0}$. Whenever $\mathbf{F}_{\mathrm{V}}\mathbf{g}_0 = \mathbf{0}$, including identical row spaces, Eq. (11) leaves it unchanged for every admissible metric and weighting. Softer enforcement cannot change an already tangent source, and no linear correction revives $\mathbf{g}_0 = \mathbf{0}$. Both implementations retain the budget equality, but at the unrounded OSQP iterate $(\hat{\mathbf{s}}, \mathbf{y})$ PEAR retains lower bound i when $\hat{s}_i < -y_{i+1}$, whereas VSJ retains it when $\lambda_i = \max(-y_{i+1}, 0) > 0$. VSJ uses $d_i = (\max(\hat{s}_i, 0) + \mu)/\lambda_i$, with $\mu = 0.1$ here. Row signs do not change the kernel; differing retained rows can. Portfolio composition adds no normal injection, extra prediction-error gradient, or second prior: the MSE-valued forward is only a carrier whose backward is replaced.

Damping and finite arithmetic. Native PEAR adds $\varepsilon = 10^{-8}$ to every Schur diagonal, including the equality. With $\mathbf{e} = \mathbf{z} - \mathbf{c}$ and $\mathbf{H}$ from Sec. G.2, its exact-arithmetic damped update obeys

$$\begin{aligned} \mathbf{S} &= \mathbf{F}_{\mathrm{P}}\mathbf{H}^{-1}\mathbf{F}_{\mathrm{P}}^{\top}, \\ \mathbf{g}_{\varepsilon} &= \left[\mathbf{I} - \mathbf{H}^{-1}\mathbf{F}_{\mathrm{P}}^{\top}(\mathbf{S} + \varepsilon\mathbf{I})^{-1}\mathbf{F}_{\mathrm{P}}\right]\mathbf{H}^{-1}\mathbf{e}, \\ \mathbf{F}_{\mathrm{P}}\mathbf{g}_{\varepsilon} &= \varepsilon(\mathbf{S} + \varepsilon\mathbf{I})^{-1}\mathbf{F}_{\mathrm{P}}\mathbf{H}^{-1}\mathbf{e}. \end{aligned} \tag{132}$$

The implemented source $\widehat{\mathbf{g}}$ is computed in float32, promoted to float64 for correction, then cast back. Let $\mathbf{A}$ be the fixed-state adaptive correction with $\mathbf{M} = \mathbf{H}$, rows $\mathbf{F}_{\mathrm{V}}$, and exact budget enforcement. We separate geometry, ridge, and finite backward arithmetic, respectively, through

$$\begin{aligned} (\mathbf{A} - \mathbf{I})\widehat{\mathbf{g}} = (\mathbf{A} - \mathbf{I})\mathbf{g}_0 \\ + (\mathbf{A} - \mathbf{I})(\mathbf{g}_{\varepsilon} - \mathbf{g}_0) \\ + (\mathbf{A} - \mathbf{I})(\widehat{\mathbf{g}} - \mathbf{g}_{\varepsilon}). \end{aligned} \tag{133}$$

Component norms are not additive attribution percentages. Holding the represented $\mathbf{H}$ and $\mathbf{e}$ fixed, independent float64 references use $\mathbf{g}_0 = \mathbf{N}(\mathbf{N}^{\top}\mathbf{H}\mathbf{N})^{-1}\mathbf{N}^{\top}\mathbf{e}$, with orthonormal $\mathbf{N}$ spanning $\ker \mathbf{F}_{\mathrm{P}}$ (zero for an empty basis), and a whitened singular-value decomposition for $\mathbf{g}_{\varepsilon}$, avoiding cancellation in an undamped Schur subtraction.

Frozen validation protocol. We replay 32 predetermined, evenly spaced windows from the 285 validation windows at each of five PEAR and five displayed covariance-composition checkpoints: 320 states, on H100, with frozen source and checkpoint hashes verified. All controls share each occurrence's predictions, covariance, and solver iterate; strict/adaptive comparisons also fix the source update. This is a new validation replay, not a reconstruction of training trajectories. No training, configuration selection, regret evaluation, or test rescoring is performed. PEAR and the composition retain the same native OSQP tolerances 10^{-6} and 50-iteration limit; backward rows are constructed from the returned primal-dual iterates.[1] Tab. 8 aggregates windows within seed before reporting five-seed mean and sample SD; overlapping windows are not independent replicates. All absolute norms remove the common batch factor $1/32$; residual row scaling is $\mathbf{1}^{\top}$ for the budget and unit coordinate rows for bounds. Define $r_{\mathrm{eq}} = |\mathbf{1}^{\top}\widehat{\mathbf{g}}|/(\sqrt{89}\|\widehat{\mathbf{g}}\|_2)$ and $r_{\mathrm{P}} = \|\widehat{\mathbf{g}}_{\mathcal{I}_{\mathrm{P}}}\|_2/\|\widehat{\mathbf{g}}\|_2$, with r_{V} defined analogously, where $\mathcal{I}$ indexes retained bounds. The final block reports $100\|\mathbf{T}\widehat{\mathbf{g}} - \widehat{\mathbf{g}}\|_2/\|\widehat{\mathbf{g}}\|_2$ for each correction $\mathbf{T}$; its gap row compares the strict and adaptive corrected vectors. Adaptive controls retain the same certificate strengths at $\mu = 0.1$, restricted to the specified row set.

Findings and limits. Row spaces differ in 16 of 320 states (12 at PEAR checkpoints, four at composition checkpoints), always through one additional VSJ bound. The geometry term dominates the mean absolute correction norm, whereas the ridge term is negligible on these states. In 179 of the 320 states, $\ker \mathbf{F}_{\mathrm{P}} = \{\mathbf{0}\}$ and the exact base update vanishes, $\mathbf{g}_0 = \mathbf{0}$: this is the $\mathrm{rank}(\mathbf{F}) = n$ case of Sec. E.1 occurring in PEAR itself. Across these states the native source norm averages 5.31×10^{-5}, so the 84.9% mean relative correction acts on an already negligible update and must be read alongside absolute magnitudes. Exact-source same-row controls and independent correction checks agree to within $10^{-14}\|\widehat{\mathbf{g}}\|_2$; the final dtype cast changes the corrected vector by at most

[1] Recorded OSQP statuses are iteration limit for 318 replayed states, solved inaccurately for one, and solved for one.

Table 8: **Frozen-state Portfolio composition diagnostics.** Mean ± sample SD of five seed means, with 32 validation states per seed and checkpoint origin. Residuals and correction components use Euclidean norms before the final float32 cast. Relative quantities are per-state ratios, not ratios of aggregate norms. The final block fixes the native source and metric, changing only the row selector or enforcement.

Quantity	PEAR checkpoints	Adaptive VSJ-PEAR checkpoints
Source magnitudes and constraint residuals		
$\|\widehat{\mathbf{g}}\|_2$	1.679 ± 0.250	1.156 ± 0.303
$\|\mathbf{F}_{\mathrm{P}}\widehat{\mathbf{g}}\|_2$	$(4.55 \pm 0.68) \times 10^{-5}$	$(4.69 \pm 0.38) \times 10^{-5}$
$\|\mathbf{F}_{\mathrm{V}}\widehat{\mathbf{g}}\|_2$	0.217 ± 0.151	0.0498 ± 0.0868
Budget residual $100r_{\mathrm{eq}}$	2.85 ± 0.53	2.92 ± 0.48
PEAR-bound residual $100r_{\mathrm{P}}$	43.90 ± 4.42	45.36 ± 5.74
VSJ-bound residual $100r_{\mathrm{V}}$	46.99 ± 3.82	46.07 ± 5.31
Absolute correction and decomposition in Eq. (133)		
Total: $\|(\mathbf{A}-\mathbf{I})\widehat{\mathbf{g}}\|_2$	0.1528 ± 0.0708	0.0444 ± 0.0688
Geometry: $\|(\mathbf{A}-\mathbf{I})\mathbf{g}_0\|_2$	0.1528 ± 0.0708	0.0444 ± 0.0688
Ridge: $\|(\mathbf{A}-\mathbf{I})(\mathbf{g}_\varepsilon-\mathbf{g}_0)\|_2$	$(3.80 \pm 0.85) \times 10^{-9}$	$(4.00 \pm 0.37) \times 10^{-9}$
Finite arithmetic: $\|(\mathbf{A}-\mathbf{I})(\widehat{\mathbf{g}}-\mathbf{g}_\varepsilon)\|_2$	$(4.02 \pm 0.71) \times 10^{-5}$	$(4.11 \pm 0.30) \times 10^{-5}$
Correction size relative to the same native source (%)		
PEAR rows, adaptive	46.80 ± 3.27	48.15 ± 6.73
PEAR rows, strict	55.00 ± 1.71	56.88 ± 8.67
VSJ rows, adaptive	49.04 ± 2.58	48.79 ± 6.39
VSJ rows, strict	59.19 ± 2.47	57.92 ± 7.94
strict-adaptive gap (VSJ rows)	17.95 ± 2.89	17.71 ± 4.22

Table 9: **Consistency of matched VSJ configurations: full comparison.** Each native backward rule is paired with the VSJ configuration in Tab. 1. Test mean ± standard deviation over five seeds. TSP and fractional Knapsack report relative regret; synthetic Portfolio reports absolute regret (Sec. G.2).

Benchmark	Implementation	Identity Update	SparseMAP	OptNet	Entropy-KKT	Log-barrier KKT
TSP	Native	0.2675 ± 0.1313	0.1676 ± 0.0041	0.2619 ± 0.0738	0.1871 ± 0.0134	0.1970 ± 0.0117
	Matched VSJ	0.2514 ± 0.1070	0.1672 ± 0.0024	0.2674 ± 0.0028	0.1771 ± 0.0087	0.2043 ± 0.0116
Fractional Knapsack	Native	0.1634 ± 0.0050	0.0534 ± 0.0099	0.0745 ± 0.0160	0.0503 ± 0.0037	0.1537 ± 0.0849
	Matched VSJ	0.1634 ± 0.0050	0.0534 ± 0.0099	0.0721 ± 0.0153	0.0503 ± 0.0037	0.1494 ± 0.0868
Portfolio	Native	0.0204 ± 0.0001	0.0007 ± 0.0000	0.0007 ± 0.0000	0.0575 ± 0.0010	0.0936 ± 0.0041
	Matched VSJ	0.0204 ± 0.0001	0.0007 ± 0.0000	0.0011 ± 0.0008	0.0575 ± 0.0010	0.0965 ± 0.0040

$4.45 \times 10^{-8}\|\widehat{\mathbf{g}}\|_2$. The last block of Tab. 8 measures strict versus adaptive enforcement on the same implemented source, not a training-performance ablation. Eq. (133) therefore separates the candidate explanations: the damping term contributes 3.80×10^{-9} and finite backward arithmetic 4.02×10^{-5}, against a total correction of 0.1528 on PEAR checkpoints. Numerical treatment accounts for at most 0.1% of the change in either column, and the composition acts through retained rows and enforcement strength, with the last block of Tab. 8 showing enforcement moving the update several times more than row selection.

G.4 End-to-End Consistency of Matched VSJ Configurations

Tab. 9 extends the fractional Knapsack example in Tab. 3 to TSP and synthetic Portfolio. Each native backward rule is replaced by its matched configuration from Tab. 1, under the training and selection protocol in Sec. G.8. On fractional Knapsack, Identity, SparseMAP, and Entropy-KKT agree at the reported precision; OptNet and Log-barrier KKT give nearby means. Across all three benchmarks, every native-matched pair has overlapping mean ± standard-deviation intervals. This is descriptive evidence of end-to-end consistency, not a statistical equivalence test or proof of identical local operators. The fixed-state Jacobian checks in Sec. G.5 address the latter separately; the nonlinear-curvature caveat accompanying synthetic Portfolio in Eq. (126) still applies.

Table 10: **Controlled operator checks.** Each row isolates the local feature that changes the operator. Errors are maxima over the listed regimes; relative errors exclude zero native Jacobians.

Approach	Native implementation	Regime tested	e_{spec}	e_{rel}	$e_{\max}$
Identity	Vendored predopt `STElayer`	One eight-dimensional shortest-path state; the signed-identity Jacobian is cost-independent.	0	0	0
SparseMAP	`entmax.sparsemax`	Six simplex scores covering support sizes $1, \ldots, 6$, plus four generic seeded scores.	8.36×10^{-16}	2.36×10^{-16}	4.44×10^{-16}
OptNet	`qpth`; `cvxpylayers`	The same six-dimensional SPD QP with all box bounds slack and with one lower bound active; both native implementations are tested in both regimes.	5.13×10^{-9}	3.11×10^{-8}	4.13×10^{-9}
Entropy-KKT	`torch.softmax`	One simplex cost at $\mu \in \{1, 0.5, 0.2, 0.1, 10^{-2}\}$, spanning diffuse to nearly concentrated solutions.	1.11×10^{-14}	6.68×10^{-11}	1.11×10^{-14}
Log-barrier KKT	Vendored IntOpt `diffKKT`	The same simplex cost, with the barrier metric read at IntOpt's own central-path iterate.	1.32×10^{-8}	1.02×10^{-8}	9.38×10^{-9}

G.5 Direct Operator Validation

Why compare Jacobians? The recovery results claim that a native backward rule and its matched VSJ implement the same local linear map, not merely that they return similar solutions. The Jacobian $\mathbf{J} = \partial \mathbf{s}^\star / \partial \mathbf{c}$ describes how a small cost perturbation changes the solution, while $\mathbf{J}^\top$ propagates an upstream loss gradient back to the predicted costs. Equality of the full matrices therefore implies agreement for every upstream direction at the tested state.

Why use the same local state? The Jacobian is not generally constant. A change in support or active constraints changes the feasible directions, while curvature, temperature, and the interior-point iterate change their scaling. Comparing operators at different states would confound operator recovery with a forward-solver difference. We therefore give each native-VSJ pair the same problem and cost, verify that their forward solutions agree to the native solver's precision, and match the support, active rows, or local metric used by the operator. We then recover the complete native matrix by calling its backward pass once for each coordinate basis vector; no matrix entry is inferred from a random projection.

Why controlled fixtures? An i.i.d. sample can repeatedly visit the same common regime and miss the cases where the operator changes. The controlled suite instead varies one defining feature at a time: every support size for SparseMAP, slack and binding bounds for OptNet, five temperatures for Entropy-KKT, and IntOpt's own central-path state for Log-barrier KKT. Identity provides a state-independent control. This design makes a failure diagnostic: it reveals which geometric regime breaks the claimed correspondence.

How are discrepancies measured? Let $\mathbf{E} = \mathbf{J}_{\text{VSJ}} - \mathbf{J}_{\text{native}}$. We report

$$e_{\text{spec}} = \|\mathbf{E}\|_2, \qquad e_{\text{rel}} = \frac{\|\mathbf{E}\|_2}{\|\mathbf{J}_{\text{native}}\|_2}, \qquad e_{\max} = \max_{i,j} |E_{ij}|, \tag{134}$$

where $\|\cdot\|_2$ is the spectral norm. The spectral error is primary because it gives the largest discrepancy over any unit direction and remains defined when the native operator vanishes. The relative error places that discrepancy on the scale of the native operator, while the entrywise error reveals whether any single coefficient disagrees. Reporting all three prevents a nearly vanishing operator from making a small absolute error look misleadingly large or small. At a SparseMAP vertex the native Jacobian can be exactly zero; the absolute errors remain meaningful there, but the relative ratio is undefined and is omitted.

What do the controlled checks show? Tab. 10 shows that SparseMAP and Entropy-KKT agree at float64 roundoff. The larger OptNet residual is explained by `qpth`'s 10^{-8} clamp on inequality duals and slacks: when VSJ is given the effective curvature that `qpth` differentiates, the maximum entrywise error falls to 1.91×10^{-17}. Likewise, the Log-barrier KKT residual comes from IntOpt casting its backward output to float32; applying the same cast to both matrices makes them agree

Table 11: **Broader state coverage.** Maximum errors over 39 deterministic states. Relative errors exclude zero native Jacobians. All constructors and seeds are fixed in the accompanying tests.

Approach	Geometry varied	States	e_{spec}	e_{rel}	$e_{\max}$
Identity	Three costs; the operator should remain unchanged.	3	0	0	0
SparseMAP	Two independently constructed scores at each support size $1, \dots, 6$.	12	8.36×10^{-16}	2.36×10^{-16}	4.44×10^{-16}
OptNet	Active lower bounds in $\{0, 1, 2\}$ crossed with curvature condition numbers $\{3, 10, 30\}$; the native implementation is `qpth`.	9	6.51×10^{-8}	6.85×10^{-8}	3.83×10^{-8}
Entropy-KKT	Temperatures $\{1, 0.2, 0.05\}$ crossed with cost scales $\{0.2, 0.7, 1.5\}$.	9	4.22×10^{-15}	7.32×10^{-8}	4.22×10^{-15}
Log-barrier KKT	Two costs at each scale in $\{0.25, 1, 2\}$, with the metric read at each native central-path state.	6	3.94×10^{-8}	3.70×10^{-8}	3.16×10^{-8}

entrywise. Thus, the observed errors match known numerical choices in the native implementations rather than a different recovered operator.

Reproducibility. These fixtures are diagnostic states, not miniature benchmark samples; their numerical values matter only because they produce the intended local geometry. All controlled problems are six-dimensional except the eight-dimensional Identity case. For SparseMAP, the score $z_i = -2i/[k(k + 1/2)]$, $i = 0, \dots, 5$, produces each support size $k = 1, \dots, 6$; four additional scores use $\mathbf{z} = \alpha\boldsymbol{\xi}$, with $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\alpha \sim \mathcal{U}(0.2, 3.0)$, and NumPy seeds $0, \dots, 3$. The OptNet curvature is $\mathbf{Q} = \mathbf{R}\mathbf{R}^\top + 6\mathbf{I}$ for a standard-normal $\mathbf{R}$ drawn with seed 11, and its two stored costs are chosen to leave all bounds slack or activate one lower bound. Entropy-KKT and Log-barrier KKT reuse a single fixed simplex cost; IntOpt is evaluated with its stopping threshold 0.5. The exact stored costs, generators, and solver settings are included verbatim in the accompanying reference tests.

Why add a broader state sweep? The controlled fixtures make failures easy to diagnose, but they are intentionally small and hand selected. We therefore test whether agreement persists as each operator's defining geometry varies more broadly. The resulting 39 deterministic states are stratified by support, active set, conditioning, temperature, or central-path metric, as appropriate. This varies the optimisation state itself rather than merely probing one state with more gradient directions.

The agreement in Tab. 11 shows that the controlled results are not tied to one convenient active set or metric state. These remain finite implementation checks: recovery beyond the tested states follows from the derivations and their local assumptions, not from the number of samples.

Why test covectors as well? The matrix comparisons exercise the explicit Jacobian interface, whereas training uses vector-Jacobian products through the backward pass and may batch several problem instances. We therefore apply 64 random covectors spanning $\mathbb{R}^n$ and verify both the single-instance and batched paths against the corresponding matrices. This adds no optimisation-state coverage; it checks that the validated operators are applied correctly in the code path used during training. SPO+ and Blackbox are tested separately for preservation of their native updates because they do not define explicit Jacobians.

G.6 Linear Dependence and the Pseudoinverse: MNIST Matching

Problem and protocol. We use the 24×24 MNIST minimum-cost perfect-matching task (Pogančić et al., 2020) as a structural control for dependent equalities. PureConvNet maps each 22×22 grayscale cell independently to a vertex cost using two 50-channel 5×5 convolutions with max pooling. The cost of an edge is $10\, c_u + c_v$, reading vertical edges top to bottom and horizontal edges left to right. On the grid graph $G = (V, E)$ with $|V| = 576$ vertices and $|E| = 1104$ edges joining horizontally and vertically adjacent cells, the integral LP is

$$\begin{aligned} \min \quad & \sum_{e \in E} z_e s_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} s_e = 1 \quad \forall v \in V \\ & 0 \le s_e \le 1 \quad \forall e \in E. \end{aligned} \tag{135}$$

Table 12: **MNIST matching.** Relative test regret, mean ± standard deviation over five seeds. The VSJ configuration is specified above; lower is better.

Method	Relative regret ↓
Blackbox	0.0000 ± 0.0000
Blackbox + Adaptive VSJ	0.0000 ± 0.0000

Here $\delta(v)$ collects the edges incident to v. The grid graph is bipartite under the checkerboard two-coloring, so the perfect-matching polytope is integral and the LP relaxation is exact. We use the published split including 10000/1000/1000 train/validation/test instances, with digits drawn with repetition from a 1000-digit MNIST subset. Both methods use Adam with learning rate 10^{-3}, batch size 70, and 30 epochs, reducing the learning rate by 0.1 at epochs 10 and 20. We report relative test regret, averaged over instances with nonzero oracle cost and then over five seeds; lower is better.

VSJ configuration. We retain the Blackbox forward matching and correct its loss-dependent update using $\mathbf{M} = \mathbf{I}$ and $\mathbf{\Omega}_{\mathrm{adapt}}(10^{-2})$, with strict degree equalities and no additional prior. This is the validation-selected configuration, with ties resolved by the fixed grid order. The 576 incidence equalities $\mathbf{F}_{\mathrm{eq}}\mathbf{s} = \mathbf{b}$ have rank 575: summing either checkerboard color class counts every edge once (Peyré & Cuturi, 2019). The pseudoinverse in Eq. (4) therefore handles the singular Schur system without deleting redundant rows. A feasible-difference update $\mathbf{g} = \mathbf{s}_1 - \mathbf{s}_2$ already satisfies $\mathbf{F}_{\mathrm{eq}}\mathbf{g} = \mathbf{0}$; only the bound correction can change it.

Results. Both methods attain zero regret at the reported precision (Tab. 12), so VSJ preserves Blackbox's performance. This control illustrates VSJ's ability to handle linearly dependent degree constraints: the Moore-Penrose pseudoinverse in Eq. (4) defines a unique metric correction despite the singular Schur system, without discarding redundant rows.

G.7 Pascal VOC-U Graph Matching

Problem. Following BB-GM (Rolínek et al., 2020b), we match keypoints between two images while allowing unmatched keypoints in the unfiltered Pascal VOC-U setting. For keypoint graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, let S_{ij} indicate whether $i \in V_1$ matches $j \in V_2$. BB-GM predicts unary costs z^{v}_{ij} and pairwise costs $z^{\mathrm{e}}_{ii',jj'}$, giving the partial quadratic-assignment problem

$$\begin{aligned} \min_{\mathbf{S}} \quad & \sum_{i\in V_1, j\in V_2} z^{\mathrm{v}}_{ij} S_{ij} + \sum_{(i,i')\in E_1,(j,j')\in E_2} z^{\mathrm{e}}_{ii',jj'} S_{ij} S_{i'j'} \\ \text{s.t.} \quad & \sum_{j\in V_2} S_{ij} \le 1 \quad \forall i \in V_1, \\ & \sum_{i\in V_1} S_{ij} \le 1 \quad \forall j \in V_2, \\ & S_{ij} \in \{0,1\} \quad \forall i \in V_1,\ j \in V_2. \end{aligned} \tag{136}$$

The inequalities permit unmatched keypoints. BB-GM combines visual and graph-based geometric features to predict the costs, and LPMP supplies the discrete forward matching. We report Blackbox baselines using BB-GM's Hamming loss and the squared loss $\|\mathbf{s}-\mathbf{s}_{\mathrm{gt}}\|^2$, then compose VSJ with the squared-loss baseline. The squared-loss runs use an earlier lifted-program construction and a looser conjugate-gradient tolerance than the audited Hamming-loss pipeline; comparisons across the two settings therefore do not isolate the effect of the loss.

VSJ configuration. We retain the squared-loss baseline's predictor, loss, and LPMP forward and correct Blackbox's native loss-dependent update using Eq. (11), without applying a second prior. The reported configuration uses the instance-wise diagonal edge-affinity metric $\mathbf{M} = \mathbf{Q}_\theta$ and adaptive inequality weighting $\mathbf{\Omega}_{\mathrm{adapt}}(10^2)$; equalities in the lifted constraint system remain strict. This is the best observed non-identity VSJ correction in the squared-loss sweep, including the extended weighting grid, rather than an independently validation-selected comparison.

Table 13: **Pascal VOC-U graph matching.** Mean $\pm$ standard deviation over three seeds, using BB-GM's accuracy/F1 evaluation. VSJ is applied to the squared-loss baseline under the same loss and pipeline; Hamming-loss Blackbox is included for reference. Squared-loss runs use the earlier pipeline described above. Bold marks the highest displayed means.

Method	Loss	Accuracy (%) ↑	F1 (%) ↑
Blackbox	Hamming	64.59 ± 0.27	62.21 ± 0.24
Blackbox	Squared	65.21 ± 0.30	62.04 ± 0.44
Blackbox + Adaptive VSJ	Squared	$\mathbf{65.65 \pm 0.33}$	$\mathbf{62.56 \pm 0.40}$

Metrics. Following BB-GM, true-positive, false-positive, and false-negative correspondence counts $(\mathrm{TP}_c, \mathrm{FP}_c, \mathrm{FN}_c)$ are pooled over image pairs within each object class c. With C evaluated classes, the reported percentages are

$$\begin{aligned} \text{Accuracy} &= \frac{100}{C}\sum_{c=1}^{C}\frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FN}_c}, \quad \mathrm{F1} = \frac{100}{C}\sum_{c=1}^{C}\frac{2p_c r_c}{p_c+r_c+\epsilon}, \\ p_c &= \frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FP}_c+\epsilon}, \qquad r_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c+\mathrm{FN}_c+\epsilon}, \qquad \epsilon = 10^{-7}. \end{aligned} \tag{137}$$

Thus accuracy measures recall of ground-truth correspondences, not accuracy over all matching and nonmatching pairs. Class scores are averaged equally; both metrics are higher-is-better.

Results. The squared-loss Blackbox baseline has higher mean accuracy, but slightly lower F1, than its Hamming-loss counterpart (Tab. 13). Incorporating VSJ into this higher-accuracy baseline improves mean accuracy by 0.44 percentage points and F1 by 0.52 points, yielding the highest displayed means on both metrics. This demonstrates an observed improvement from VSJ composition within the squared-loss setting; the three-seed, best-observed comparison remains exploratory and does not establish statistical significance.

G.8 Experimental Validation and Training Details

Tab. 14 records the optimisation settings inherited from each benchmark implementation. We use Adam throughout and perform no method-specific tuning: within each benchmark, all methods share the predictor, data split, batch size, epoch budget, learning-rate candidates and schedule, and random seeds. Learning rates and VSJ configurations are selected by mean validation regret, after which the selected run is evaluated on the held-out test split. Regret results are averaged over five seeds unless a superscript n specifies fewer seeds.

Every VSJ configuration is a parameterization of the same layer implementation through $(\mathbf{P}, \mathbf{M}, \mathbf{\Omega})$. Native baselines use the corresponding benchmark implementation where available, with `qpth` for OptNet, LP-SparseMAP for SparseMAP, and the released IntOpt solver for Log-barrier KKT. Thus comparisons within a benchmark change the backward operator while holding the predictor, optimizer, training schedule, data, and seeds fixed.

Collapse of native Blackbox differentiation and SPO+ on TSP. For native Blackbox differentiation and SPO+, training collapses on some TSP seeds: the fraction of test instances decoded to the optimal tour stays at 0.000 for Blackbox (test regret 2.26) and at 0.18 and 0.12 for SPO+ (test regret 0.02), against at least 0.96 on the remaining seeds. Among the five seeds, Blackbox differentiation collapses on one and SPO+ on two. Collapse of Blackbox differentiation on this task is also reported by Sahoo et al. (2023). To align with the published results, we exclude the collapsed seeds and report the native approaches over the remaining seeds. We did not observe this collapse for their VSJ-corrected variants, whose results are reported over all five seeds.

G.9 Mechanism and Axis-Ablation Details

We first document the binary Knapsack diagnostics and a weight-assignment training ablation, then separate fixed-state checks of adaptive enforcement from component ablations on the controlled benchmarks. Tab. 15 summarizes the latter checks and ablations.

Table 14: **Training protocol inherited from the benchmark implementations.** The Globe TSP reference makes two settings depend on the instance size k; since we run $k = 40$, we set learning rate 5×10^{-5} and repellent regularization constant $C_{40} = 20.0$, applied between epochs 15 and 30. TSP uses the experimental setup of Pogančić et al. (2020, App. A.3.3, Table 7) for $k = 40$: learning rate 5×10^{-5}, batch size 50, 100 epochs with the learning rate divided by 10 at epochs 80 and 90, Adam with $\beta_1 = 0.5$ and $\epsilon = 10^{-3}$, $\lambda = 20$, and a repellent force between epochs 15 and 30. For fractional Knapsack, native SPO+ (Elmachtoub & Grigas, 2022) and Blackbox differentiation (Pogančić et al., 2020) use the learning rate 1.0 and $\lambda = 1$ selected as optimal for this instance by the validation grid search of Mandi et al. (2024, Sec. 5.2); the curvature-based natives use a common regularization strength $\mu = 10$.

Benchmark	Learning rate	Batch	Epochs	Schedule
TSP	5×10^{-5}	50	100	milestones 80, 90 ($\gamma = 0.100$)
Knapsack	1.000	128	30	plateau (0.200, patience 2)
Matching	0.001	70	30	milestones 10, 20 ($\gamma = 0.100$)
Portfolio	0.050	128	30	plateau (0.200, patience 2)

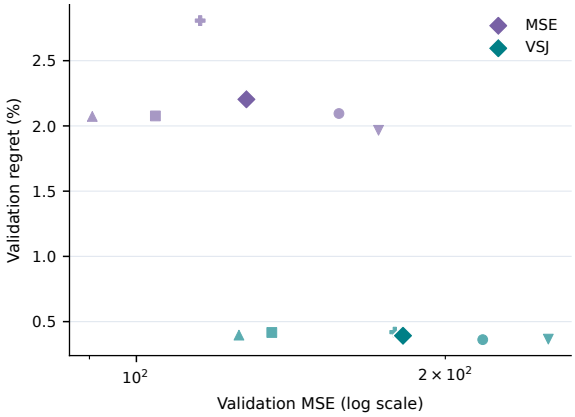


Figure 4: **Prediction accuracy versus decision quality on binary Knapsack.** MSE-trained and Adaptive VSJ-trained predictors on all 500 validation examples per seed. Symbols denote seeds and diamonds denote means. Lower prediction error need not imply lower regret.

G.9.1 Binary Knapsack Diagnostics and Weight Assignment

Prediction accuracy versus decision quality. The analyses in Figs. 3 and 4 use the separate 100-item, degree-8 binary Knapsack benchmark with five seeds. In Fig. 4, MSE-trained and VSJ-trained predictors are evaluated on all 500 validation examples per seed; symbols denote seeds and diamonds denote arithmetic means. MSE attains lower validation prediction error yet substantially higher regret than Adaptive VSJ, illustrating that prediction accuracy alone is insufficient for decision quality. These are historical checkpoints trained on H100 and A100 GPUs, respectively, not a hardware-matched training comparison.

Checkpoint sampling and common starting predictions. The intent of Fig. 3a is to compare update directions near trained-model predictions while holding the starting point fixed. We select 32 fixed, evenly spaced examples from each seed's 500-example validation split. For each seed, the same 32 examples are evaluated using one trained checkpoint from each of PEAR, pure Adaptive VSJ, and Adaptive VSJ-PEAR, giving $32 \times 3 \times 5 = 480$ example-checkpoint pairs, not 480 distinct validation examples. These are frozen trained models, not random initializations. Each pair supplies a saved predicted-profit vector $\mathbf{z} \in \mathbb{R}^{100}$ and its forward solver solution and dual multipliers. Both updates are evaluated at every such prediction and solver state, using the same true validation profits, regardless of the checkpoint origin. Thus, a checkpoint supplies a common starting point rather than determining which update is tested. Using three checkpoint origins broadens the sampled predictions beyond those produced by one training procedure.

Equal-norm local-step diagnostics. For a nonzero PEAR or Adaptive VSJ update $\mathbf{g}$, each probe applies $\Delta\mathbf{z} = -\alpha\|\mathbf{z}\|_2\mathbf{g}/\|\mathbf{g}\|_2$. We solve the binary decision problem at $\mathbf{z} + \Delta\mathbf{z}$ and score regret using the true validation profits; an exactly zero update leaves the prediction unchanged. Thus, $\alpha = 0.01$ gives a step of length 1% of the starting prediction norm, not a 1% change in every coordinate. For each of the two updates and each α, the probe restarts from the original $\mathbf{z}$; the steps are not cumulative, and predictor parameters are never updated. We display only the original local range $0 \le \alpha \le 0.1$.

Aggregation and scope. At each step size, regret is aggregated as a ratio of sums over the 96 example-checkpoint pairs within each seed, with 32 pairs from each checkpoint origin. We then compute the five-seed mean and sample standard deviation; the latter defines the shaded bands. The 480 pairs are not treated as independent experimental replicates. These label-informed probes compare local directions at sampled trained-model predictions, not behavior from random initialization, training trajectories, or held-out test performance. They separate direction from raw update magnitude at these starting points, but do not by themselves explain the final trained-model performance gap.

Signal and normal-injection controls. Replacing VSJ's decision-loss signal by prediction error nearly preserves its local gain, whereas removing PEAR's normal injection alone does not improve its log-step-averaged regret. These suggest that backward constraint treatment, rather than the input signal alone, contributes to the local difference; they do not isolate adaptive weighting or PEAR composition.

Does the assignment of adaptive weights matter? To distinguish the role of constraint-specific weighting from the weight magnitudes alone, we compare native Adaptive VSJ with a shuffled-weight control during training. This experiment uses fresh paired H100 runs, rather than the historical checkpoints in Fig. 4. Within each of the five seeds, the pair shares data and initialization, the forward solver and decision loss, $\mathbf{P} = 10\mathbf{I}$, $\mathbf{M} = 0.1\mathbf{I}$, and the same constraint-row selection rule. The bound temperature is $\mu = 0.01$ and the capacity temperature is 0.1. Both arms use learning rate 0.003, batch size 32, validation every 10 epochs, the same checkpoint-selection and early-stopping rules, and caps of 10,000 epochs and 600 seconds. Checkpoints are selected by validation regret; no test scoring is used in this ablation.

Shuffling intervention. At each backward evaluation, we first construct the native row matrix $\mathbf{F}$ and finite relaxation coefficients $d_k = (r_k + \mu_k)/\lambda_k$ for the retained positive-dual inequalities. The control leaves $\mathbf{F}$ unchanged and permutes these coefficients separately within the capacity, lower-bound, and upper-bound groups. Since $w_k = 1/(1 + d_k)$, this preserves the multiset of adaptive weights within each group at a given state while changing their association with constraint normals; zero-dual rows remain omitted. Each seed uses one fixed random-priority map over row identities, restricted to the currently eligible rows. Thus the same eligible set receives the same permutation throughout training, rather than newly sampled noise at every step; neither the permutation nor the hyperparameters are selected in a new search. The intervention changes only the assignment of the backward relaxation coefficients, not the metric, prior, or forward problem.

Results and interpretation. In Fig. 3b, native assignments yield validation regret of $0.3913 \pm 0.0290\%$, compared with $2.4596 \pm 0.5067\%$ under Shuffling (mean $\pm$ sample standard deviation across five seeds). The paired increase is 2.068 ± 0.479 percentage points, with deterioration in every seed. This supports aligning enforcement strengths with their corresponding constraints rather than using the same weight distribution with arbitrary assignments. The result concerns validation-selected training outcomes, not held-out test gains or an equal-step causal comparison: early stopping and the wall-time cap allow different numbers of optimizer steps and different selected epochs. It also does not isolate the entire PEAR-VSJ difference, which involves additional choices of row selection, normalization, relaxation, and training settings.

G.9.2 Fixed-State Enforcement and Component Ablations

Does adaptive weighting realize the intended geometry? On a fixed five-dimensional simplex state with $\mathbf{P} = -\mathbf{I}$ and $\mathbf{M} = \mathbf{I}$, we vary only $\mathbf{\Omega}$. The optimum is supported on the first two

Table 15: **Adaptive enforcement and component ablations. A:** With the solution, $\mathbf{F}$, $\mathbf{P} = -\mathbf{I}$, and $\mathbf{M} = \mathbf{I}$ fixed, only $\mathbf{\Omega}$ varies; active-bound columns have zero slack. Smaller residuals indicate stronger enforcement. **B:** Component ablations on fractional Knapsack from the displayed reference configuration. Entries are test regret (mean $\pm$ standard deviation over five seeds); lower is better.

A. Enforcement mechanism

Correction	$\|\mathbf{F}\mathbf{J}^\star\|_F$	ρ_{eq}	$\rho_{\lambda=.2}$	$\rho_{\lambda=.4}$	$\rho_{\lambda=.6}$
None ($\mathbf{\Omega} = \mathbf{0}$)	2.828	2.236	1.000	1.000	1.000
Adapt. ($\mu = 1$)	1.164	$< 10^{-12}$	0.749	0.664	0.595
Adapt. ($\mu = .1$)	0.433	$< 10^{-12}$	0.342	0.215	0.156
Adapt. ($\mu = .01$)	0.067	$< 10^{-12}$	0.057	0.029	0.020
Strict ($\mathbf{\Omega} = \mathbf{I}$)	$< 10^{-12}$	$< 10^{-12}$	$< 10^{-12}$	$< 10^{-12}$	$< 10^{-12}$
$\mathbf{D} = \text{diag}(r_k/\lambda_k)$	$< 10^{-12}$	$< 10^{-12}$	$< 10^{-12}$	$< 10^{-12}$	$< 10^{-12}$
Uniform $\gamma = 0.378$	1.164	$< 10^{-12}$	0.672	0.672	0.672
Uniform $\gamma = 3.233$	0.433	$< 10^{-12}$	0.250	0.250	0.250
Uniform $\gamma = 30.00$	0.067	$< 10^{-12}$	0.039	0.039	0.039

B. Complementary effects

Configuration	Axis	Regret
Reference ($-\mathbf{H}_{\text{Log}}^{-1}, \mathbf{I}, \mathbf{I}$)	–	0.0908 ± 0.0041
$\mathbf{P} \to -\mathbf{I}$	$\mathbf{P}$	0.0534 ± 0.0099
$\mathbf{M} \to \mathbf{H}_{\text{Log}}$	$\mathbf{M}$	0.1494 ± 0.0868
$\mathbf{\Omega} \to \mathbf{\Omega}_{10^{-1}}$	$\mathbf{\Omega}$	0.0920 ± 0.0045
Combined ($\mathbf{M} \to \mathbf{H}_{\text{Log}}$, $\mathbf{\Omega} \to \mathbf{\Omega}_{10^{-1}}$)	$\mathbf{M}, \mathbf{\Omega}$	$\mathbf{0.0527 \pm 0.0058}$

Table 16: **Full adaptive-weighting check at $\mu = 0.100$.** Panel A reports the equality and active-bound rows at the simplex optimum; Panel B isolates two boundary cases of the weighting rule.

Constraint row	λ_k	r_k	w_k	γ_k	d_k	$\|\mathbf{f}_k^\top \mathbf{J}^\star\|_2$
A. Rows at the simplex optimum						
Equality $\mathbf{1}^\top \mathbf{s} = 1$	0.300	0.000	1.000	∞	0.000	6.632×10^{-16}
Active bound $s_4 = 0$	0.200	0.000	0.667	2.000	0.500	0.342
Active bound $s_5 = 0$	0.400	0.000	0.800	4.000	0.250	0.215
Active bound $s_3 = 0$	0.600	0.000	0.857	6.000	0.167	0.156
B. Boundary behavior of the weighting rule						
Zero-dual limit	0.000	0.000	0.000	0.000	∞	omitted
Positive-slack probe	0.600	0.300	0.600	1.500	0.667	–

coordinates; the lower bounds on s_3, s_4, and s_5 are active with multipliers 0.600, 0.200, and 0.400, respectively. Panel A of Tab. 15 measures global tangency by $\|\mathbf{F}\mathbf{J}^\star\|_F$ and rowwise enforcement by $\rho_k = \|\mathbf{f}_k^\top \mathbf{J}^\star\|_2$. The global residual falls from 2.828 without correction to below 10^{-12} under strict VSJ, and from 1.164 to 0.067 as adaptive μ decreases from 1 to 0.01; equalities remain strict under adaptive weighting. For the three zero-slack bounds, $\rho_{\lambda=.2} > \rho_{\lambda=.4} > \rho_{\lambda=.6}$ at every tested μ. These results confirm stronger enforcement for larger multipliers in this fixed geometry. Decreasing μ suppresses inequality-normal motion while preserving equality tangency. Two closest alternatives are included. Setting $\mathbf{D} = \text{diag}(r_k/\lambda_k)$ removes μ; at an exact KKT point every active row has $r_k = 0$, so this rule returns the strict projector and is not a distinct finite-weight alternative. A single stiffness shared by all inequalities, as in a penalty construction with one inequality coefficient, is tuned here to match the adaptive global residual exactly; it then distributes enforcement uniformly and departs from the adaptive row profile by 9.3%, 31.4% and 41.6% at $\mu = 1$, 0.1 and 0.01.

Dual and slack controls. Tab. 16 reports the complete adaptive-weight calculation at $\mu = 0.100$, together with zero-dual and positive-slack controls. The positive-slack row is an algebraic probe, not an additional constraint at the simplex optimum.

Across the active-bound rows, increasing the multiplier from $\lambda_k = 0.200$ to 0.600 decreases d_k from 0.500 to 0.167 and the normal-direction residual from 0.342 to 0.156. At fixed $\lambda_k = 0.600$, increasing slack from 0 to 0.300 changes (w_k, γ_k, d_k) from $(0.857, 6.000, 0.167)$ to $(0.600, 1.500, 0.667)$ and therefore weakens enforcement. When $\lambda_k = 0$, the row has zero weight and stiffness and is omitted.

Do the components contribute independently and jointly? Panel B of Tab. 15 examines the controlled fractional Knapsack benchmark. Starting from $(\mathbf{P}, \mathbf{M}, \mathbf{\Omega}) = (-\mathbf{H}_{\text{Log}}^{-1}, \mathbf{I}, \mathbf{I})$, changing only $\mathbf{P}$ to $-\mathbf{I}$ lowers mean regret from 0.0908 to 0.0534. Changing $\mathbf{M}$ to $\mathbf{H}_{\text{Log}}$ or $\mathbf{\Omega}$ to $\mathbf{\Omega}_{10^{-1}}$ alone does not help, whereas changing both lowers regret to 0.0527. Thus $\mathbf{P}$ has an independent effect, while $\mathbf{M}$ and $\mathbf{\Omega}$ exhibit complementarity in this slice.

Table 17: **One-axis-at-a-time ablations.** Panel A fixes $(\mathbf{M}, \mathbf{\Omega}) = (\mathbf{Q}_\theta, \mathbf{\Omega}_{10^{-2}})$. Panel B fixes $(\mathbf{P}, \mathbf{\Omega}) = (-\mathbf{Q}_\theta^{-1}, \mathbf{\Omega}_{10^{-2}})$. Panel C fixes $(\mathbf{P}, \mathbf{M}) = (-\mathbf{Q}_\theta^{-1}, \mathbf{Q}_\theta)$. Entries are held-out test regret (mean $\pm$ standard deviation over five seeds). Knapsack denotes fractional Knapsack and Matching denotes MNIST matching; both report relative regret, while synthetic Portfolio reports absolute regret. Lower is better.

Varied axis	Setting	Knapsack	Matching	Portfolio
A. Prior	$\mathbf{M} = \mathbf{Q}_\theta, \quad \mathbf{\Omega} = \mathbf{\Omega}_{10^{-2}}$			
$\mathbf{P}$	$-\mathbf{I}$	0.0501 ± 0.0035	0.2587 ± 0.0017	0.0157 ± 0.0017
	$-\mathbf{Q}_\theta^{-1}$	0.0619 ± 0.0075	0.0046 ± 0.0012	0.0375 ± 0.0024
	$-\mathbf{H}_{\mathrm{Ent}}^{-1}$	0.0776 ± 0.0277	0.0442 ± 0.0043	0.0411 ± 0.0049
	$-\mathbf{H}_{\mathrm{Log}}^{-1}$	0.3125 ± 0.1337	0.0505 ± 0.0031	0.0891 ± 0.0049
B. Metric	$\mathbf{P} = -\mathbf{Q}_\theta^{-1}, \ \mathbf{\Omega} = \mathbf{\Omega}_{10^{-2}}$			
$\mathbf{M}$	$\mathbf{I}$	0.3954 ± 0.0103	0.0454 ± 0.0336	0.0589 ± 0.0123
	$\mathbf{Q}_\theta$	0.0619 ± 0.0075	0.0046 ± 0.0012	0.0375 ± 0.0024
	$\mathbf{H}_{\mathrm{Ent}}$	0.3810 ± 0.0130	0.2436 ± 0.0378	0.0473 ± 0.0049
	$\mathbf{H}_{\mathrm{Log}}$	0.3841 ± 0.0156	0.2038 ± 0.0173	0.0469 ± 0.0022
C. Constraint weighting	$\mathbf{P} = -\mathbf{Q}_\theta^{-1}, \ \mathbf{M} = \mathbf{Q}_\theta$			
$\mathbf{\Omega}$	$\mathbf{0}$	0.4591 ± 0.0181	0.1900 ± 0.0250	0.0495 ± 0.0068
	$\mathbf{I}$	0.0721 ± 0.0153	0.0031 ± 0.0004	0.0011 ± 0.0008
	$\mathbf{\Omega}_{10^{-3}}$	0.0737 ± 0.0190	0.0031 ± 0.0008	0.0295 ± 0.0006
	$\mathbf{\Omega}_{10^{-2}}$	0.0619 ± 0.0075	0.0046 ± 0.0012	0.0375 ± 0.0024
	$\mathbf{\Omega}_{10^{-1}}$	0.0540 ± 0.0021	0.0109 ± 0.0015	0.0530 ± 0.0004
	$\mathbf{\Omega}_1$	0.0554 ± 0.0077	0.0177 ± 0.0013	0.0516 ± 0.0022

Further one-axis comparisons. Tab. 17 varies one component at a time while fixing the other two, extending the analysis to MNIST matching and synthetic Portfolio. The theory predicts distinct operators, not a universal downstream ranking.

The prior choice materially changes all three tasks in this fixed slice. In Panel B, the metric $\mathbf{Q}_\theta$ is strongest on all three tasks. Removing correction is worst on Knapsack and Matching. Within the displayed weighting slice, Knapsack favors $\mathbf{\Omega}_{10^{-1}}$, while strict enforcement is strongest on Portfolio and ties for best on Matching.

G.10 Computational Efficiency

Benchmark choice. We use Knapsack and MNIST matching as contrasting predictor workloads. Knapsack couples a single linear layer to 48 item decisions, making decision-layer overhead visible. Matching couples PureConvNet to a 4×4 cost map, providing a vision workload in which model computation can amortize correction. Because correction cost also depends on the retained active rows and certificate path, this pair probes contrasting workloads rather than isolating modality.

Comparisons. Panel A of Table 18 compares every native explicit-Jacobian backward rule with its mathematically matched VSJ realization, holding the predictor, batch, forward problem, and resulting backward operator fixed. Panel B compares each native loss-dependent update with a representative VSJ composition from the configuration sweep; here the comparison measures the cost of adding geometric correction rather than the cost of the validation-selected winner. The timed settings use $(\mathbf{Q}_\theta, \mathbf{\Omega}_{10^{-2}})$ for SPO+ and $(\mathbf{I}, \mathbf{\Omega}_{10^{-1}})$ for Blackbox.

Protocol. After warm-up, timings synchronize accelerator operations at each measurement boundary and report mean $\pm$ standard deviation over the three training seeds. Backward time covers the complete decision-layer backward path, including certificate acquisition or an auxiliary relaxation and the Schur-complement correction for VSJ. Step time additionally includes predictor forward and backward propagation and the optimizer update; peak memory is the maximum allocated memory during that complete step. All paired rows use identical hardware, software, batches, and forward solves.

Table 18: **Training cost across backward operators.** Panel A compares native explicit-Jacobian backward rules with mathematically matched VSJ realizations; Panel B compares native loss-dependent updates with representative VSJ compositions. Times are milliseconds per batch and values are mean ± standard deviation over three seeds. VSJ backward includes certificate acquisition and correction.

		Backward (ms)		Complete step (ms)		Peak memory (GB)	
Benchmark	Operator/update	Native	VSJ	Native	VSJ	Native	VSJ
A. Explicit-Jacobian operators: native implementation vs matched VSJ							
Knapsack (tabular)	Identity Update ($-\mathbf{I}$)	0.27±0.06	98.7±1.9	58.4±8.2	248±6	0.38±0.01	0.37±0.01
	SparseMAP ($-\mathbf{I}$)	130±10	103±3	526±8	219±9	1.04±0.03	0.41±0.01
	OptNet ($-\mathbf{Q}_\theta^{-1}$)	16.3±1.3	196±11	468±2	582±20	0.52±0.03	0.42±0.01
	Entropy-KKT ($-\mathbf{H}_{\mathrm{Ent}}^{-1}$)	166±10	114±4	962±11	536±11	2.36±0.02	0.38±0.01
	Log-barrier KKT ($-\mathbf{H}_{\mathrm{Log}}^{-1}$)	178±17	110±3	3387±527	506±10	2.43±0.04	0.38±0.00
MNIST matching (vision)	Identity Update ($-\mathbf{I}$)	34.8±0.1	182±6	70.7±1.7	247±9	1.19±0.00	1.19±0.00
	SparseMAP ($-\mathbf{I}$)	47.4±1.2	221±6	287±24	345±8	1.19±0.01	1.18±0.00
	OptNet ($-\mathbf{Q}_\theta^{-1}$)	35.7±1.4	305±24	146±1	460±33	1.21±0.00	1.19±0.00
	Entropy-KKT ($-\mathbf{H}_{\mathrm{Ent}}^{-1}$)	203±2	176±2	809±23	945±5	2.07±0.01	1.19±0.01
	Log-barrier KKT ($-\mathbf{H}_{\mathrm{Log}}^{-1}$)	201±17	170±2	1073±6	772±6	4.09±0.05	1.19±0.01
B. Loss-dependent updates: native update vs representative VSJ composition							
Knapsack (tabular)	SPO+	5.23±0.42	200±12	52.2±10.1	266±15	0.38±0.01	0.38±0.01
	Blackbox	8.73±0.35	160±5	51.3±7.2	224±12	0.37±0.01	0.38±0.00
MNIST matching (vision)	SPO+	34.4±0.5	57.7±2.8	73.2±1.0	95.9±4.6	1.19±0.01	1.18±0.01
	Blackbox	36.5±1.6	60.1±0.2	70.9±2.8	94.3±2.4	1.18±0.01	1.19±0.01

Results. Complete-step ratios account for the different native workloads. In Panel B, SPO+ and Blackbox with VSJ take respectively $5.10\times$ and $4.37\times$ their native Knapsack step times, but $1.31\times$ and $1.33\times$ on MNIST matching. Peak memory in this panel differs by at most 0.01 GB. In Panel A, matched VSJ has a faster complete step for four of ten pairs (Knapsack SparseMAP, Entropy-KKT, and Log-barrier KKT, and matching Log-barrier KKT), and its reported mean peak memory never exceeds native.

Interpretation. Training applies VSJ to the incoming covector rather than forming an $n \times n$ Jacobian. With k retained rows and diagonal $\mathbf{M}^{-1}$ and $\mathbf{P}$, dense $\mathbf{F}$ requires $O(k^2n + k^3)$ time and $O(kn+k^2)$ memory through a $k \times k$ Schur complement. The timings also include certificate acquisition, which is solver dependent and may require an auxiliary relaxation. Matching's lower relative loss-dependent overhead reflects both its smaller decision dimension and amortization by PureConvNet inference and backpropagation; because k and certificate acquisition are problem dependent, this contrast does not establish a modality-wide runtime advantage. Panel A fixes the mathematical operator, so its speed differences reflect implementation paths rather than a universal algorithmic advantage. Overall, VSJ is memory-efficient in these runs, while its runtime depends on predictor cost, active-system size, and certificate acquisition.

G.11 Validation Results for Configuration Selection

Tabs. 19 to 21 report the validation sweeps for native backward rules (and their VSJ reproductions) for explicit Jacobians, new VSJ configurations for explicit Jacobians and loss-dependent update compositions, respectively. We include them to document the configuration selection underlying the held-out test comparisons in Tabs. 4 and 5.

Configuration choices. For explicit Jacobians, we evaluate combinations drawn from $\mathbf{P} \in \{-\mathbf{I}, -\mathbf{Q}_\theta^{-1}, -\mathbf{H}_{\mathrm{Ent}}^{-1}, -\mathbf{H}_{\mathrm{Log}}^{-1}\}$, $\mathbf{M} \in \{\mathbf{I}, \mathbf{Q}_\theta, \mathbf{H}_{\mathrm{Ent}}, \mathbf{H}_{\mathrm{Log}}\}$, and $\mathbf{\Omega} \in \{\mathbf{0}, \mathbf{I}, \mathbf{\Omega}_{\mathrm{adapt}}(\mu)\}$ for $\mu \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. Here $\mathbf{0}$ gives no correction, $\mathbf{I}$ strict correction, and $\mathbf{\Omega}_\mu := \mathbf{\Omega}_{\mathrm{adapt}}(\mu)$ adaptive correction.

Table 19: **Validation regret of established backward rules and their matched VSJ reproductions.** Values are mean ± standard deviation over available seeds (up to five), with the benchmarks and regret conventions of Tab. 20; lower is better. Together with the configuration of that table, VSJ reproductions are considered for validating the per-prior VSJ configuration for Tab. 4; bold marks the selected entries.

ID	Approach	$\mathbf{P}$	$\mathbf{M}$	$\mathbf{\Omega}$	Knapsack	TSP	Matching	Portfolio
1	Identity Update	$-\mathbf{I}$	$\mathbf{I}$	$\mathbf{0}$	0.1730 ± 0.0295	0.2690 ± 0.1276	0.2781 ± 0.0000	0.0202 ± 0.0001
2	↪ *VSJ reproduction*	$-\mathbf{I}$	$\mathbf{I}$	$\mathbf{0}$	0.1730 ± 0.0295	0.3029 ± 0.1148	0.2781 ± 0.0000	0.0202 ± 0.0001
3	SparseMAP	$-\tau^{-1}\mathbf{I}$	$\tau\mathbf{I}$	$\mathbf{I}$	0.0595 ± 0.0093	0.1668 ± 0.0046	0.0030 ± 0.0012	0.0006 ± 0.0000
4	↪ *VSJ reproduction*	$-\tau^{-1}\mathbf{I}$	$\tau\mathbf{I}$	$\mathbf{I}$	0.0595 ± 0.0093	0.1655 ± 0.0018	0.0027 ± 0.0010	$\mathbf{0.0006} \pm 0.0000$
5	OptNet	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.0759 ± 0.0148	0.2605 ± 0.0731	0.0029 ± 0.0005	0.0007 ± 0.0001
6	↪ *VSJ reproduction*	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.0775 ± 0.0242	0.2685 ± 0.0033	0.0029 ± 0.0004	$\mathbf{0.0010} \pm 0.0007$
7	Entropy-KKT	$-\mathbf{H}_{\mathrm{Ent}}^{-1}$	$\mathbf{H}_{\mathrm{Ent}}$	$\mathbf{I}$	0.0554 ± 0.0058	0.1873 ± 0.0130	0.0061 ± 0.0017	0.0561 ± 0.0009
8	↪ *VSJ reproduction*	$-\mathbf{H}_{\mathrm{Ent}}^{-1}$	$\mathbf{H}_{\mathrm{Ent}}$	$\mathbf{I}$	0.0554 ± 0.0058	0.1767 ± 0.0053	0.0060 ± 0.0022	0.0561 ± 0.0009
9	Log-barrier KKT	$-\mathbf{H}_{\mathrm{Log}}^{-1}$	$\mathbf{H}_{\mathrm{Log}}$	$\mathbf{I}$	0.1641 ± 0.0755	0.1960 ± 0.0108	0.0122 ± 0.0022	0.0896 ± 0.0037
10	↪ *VSJ reproduction*	$-\mathbf{H}_{\mathrm{Log}}^{-1}$	$\mathbf{H}_{\mathrm{Log}}$	$\mathbf{I}$	0.1594 ± 0.0780	0.2115 ± 0.0097	0.0133 ± 0.0015	0.0923 ± 0.0032

Table 20: **Explicit-Jacobian validation sweep.** Panel A reports bare priors; Panel B reports geometry-corrected configurations. Values are mean ± standard deviation over available seeds (up to five). Knapsack denotes fractional Knapsack and Matching denotes MNIST matching. These and TSP report relative regret, while synthetic Portfolio reports absolute regret; lower is better. Selection for Tab. 4 minimizes mean validation regret per prior and benchmark, including its VSJ reproduction (Tab. 19) and bare prior; ties follow fixed-grid order. Bold marks the selected configurations.

ID	Approach	$\mathbf{P}$	$\mathbf{M}$	$\mathbf{\Omega}$	Knapsack	TSP	Matching	Portfolio
A. Uncorrected VSJ priors ($\mathbf{\Omega} = \mathbf{0}$; $\mathbf{M}$ immaterial)								
1	VSJ	$-\mathbf{Q}_\theta^{-1}$	–	$\mathbf{0}$	0.4560 ± 0.0147	0.1839 ± 0.0137	0.1896 ± 0.0281	0.0476 ± 0.0062
2	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	–	$\mathbf{0}$	0.1887 ± 0.0290	$\mathbf{0.1690} \pm 0.0651$	0.2781 ± 0.0000	$\mathbf{0.0006} \pm 0.0000$
3	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	–	$\mathbf{0}$	0.1773 ± 0.0276	0.3409 ± 0.1937	0.2781 ± 0.0000	$\mathbf{0.0727} \pm 0.0029$
B. Geometry-corrected VSJ configurations								
Prior: $\mathbf{P} = -\mathbf{I}$								
4	VSJ	$-\mathbf{I}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0651 ± 0.0098	0.4795 ± 0.1168	0.0182 ± 0.0013	0.0374 ± 0.0015
5	VSJ	$-\mathbf{I}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0559 ± 0.0059	$\mathbf{0.1181} \pm 0.0193$	0.0160 ± 0.0011	0.0315 ± 0.0014
6	VSJ	$-\mathbf{I}$	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.4832 ± 0.0988	0.7058 ± 0.7333	0.3161 ± 0.0257	0.0996 ± 0.0084
7	VSJ	$-\mathbf{I}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	$\mathbf{0.0550} \pm 0.0051$	0.2336 ± 0.0289	0.2761 ± 0.0012	0.0177 ± 0.0071
8	VSJ	$-\mathbf{I}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0552 ± 0.0061	0.2296 ± 0.0168	0.2748 ± 0.0023	0.0171 ± 0.0022
9	VSJ	$-\mathbf{I}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0561 ± 0.0060	0.3354 ± 0.0782	0.2761 ± 0.0015	0.0166 ± 0.0008
10	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0607 ± 0.0076	0.4586 ± 0.1511	0.2781 ± 0.0000	0.0207 ± 0.0002
11	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0559 ± 0.0064	0.6064 ± 0.3016	0.2781 ± 0.0000	0.0202 ± 0.0001
12	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10)$	0.0576 ± 0.0068	0.4115 ± 0.2272	0.2781 ± 0.0000	0.0202 ± 0.0001
13	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0651 ± 0.0096	0.5138 ± 0.3291	0.2781 ± 0.0000	0.0321 ± 0.0003
14	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0605 ± 0.0074	0.4861 ± 0.2051	0.2781 ± 0.0000	0.0207 ± 0.0002
15	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0562 ± 0.0060	0.3699 ± 0.1042	0.2781 ± 0.0000	0.0202 ± 0.0001
16	VSJ	$-\mathbf{I}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10)$	0.0572 ± 0.0067	0.4291 ± 0.1135	0.2781 ± 0.0000	0.0202 ± 0.0001
Prior: $\mathbf{P} = -\mathbf{Q}_\theta^{-1}$								
17	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{I}$	$\mathbf{I}$	0.3882 ± 0.0115	0.1224 ± 0.0207	0.0996 ± 0.0694	0.0788 ± 0.0090
18	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.3943 ± 0.0090	0.1842 ± 0.1140	0.0785 ± 0.0728	0.0695 ± 0.0055
19	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.3982 ± 0.0101	0.1233 ± 0.0128	0.0455 ± 0.0298	0.0559 ± 0.0118
20	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.3867 ± 0.0083	0.1962 ± 0.0584	0.0287 ± 0.0035	0.0539 ± 0.0189
21	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.3882 ± 0.0201	0.3138 ± 0.0906	0.0273 ± 0.0022	0.0381 ± 0.0024
22	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0787 ± 0.0275	0.1124 ± 0.0064	0.0035 ± 0.0005	0.0290 ± 0.0011
23	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0640 ± 0.0047	$\mathbf{0.1001} \pm 0.0065$	0.0047 ± 0.0014	0.0369 ± 0.0023
24	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	$\mathbf{0.0576} \pm 0.0077$	0.2095 ± 0.0556	0.0115 ± 0.0015	0.0521 ± 0.0007
25	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0623 ± 0.0016	0.2534 ± 0.0084	0.0182 ± 0.0020	0.0502 ± 0.0021
26	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{I}$	0.3716 ± 0.0352	0.2472 ± 0.0002	0.1887 ± 0.0302	0.0491 ± 0.0057
27	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.3829 ± 0.0133	0.2296 ± 0.0351	0.2542 ± 0.0441	0.0454 ± 0.0046
28	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.3553 ± 0.0939	0.3017 ± 0.0122	0.2525 ± 0.0320	0.0468 ± 0.0053
29	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.3937 ± 0.0056	0.2186 ± 0.0389	0.2734 ± 0.0167	0.0552 ± 0.0075
30	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10)$	0.3964 ± 0.0038	0.2119 ± 0.0084	0.2485 ± 0.0681	0.0446 ± 0.0016
31	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{I}$	0.3256 ± 0.1472	0.5245 ± 0.2020	0.2590 ± 0.0324	0.0792 ± 0.0141
32	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.3860 ± 0.0064	0.2154 ± 0.0022	0.2092 ± 0.0207	0.0456 ± 0.0019
33	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.3952 ± 0.0088	0.1850 ± 0.0000	0.2094 ± 0.0145	0.0466 ± 0.0060
34	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.3937 ± 0.0071	0.3837 ± 0.0464	0.2054 ± 0.0223	0.0499 ± 0.0064
35	VSJ	$-\mathbf{Q}_\theta^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10)$	0.3948 ± 0.0062	0.1883 ± 0.0037	0.2097 ± 0.0198	0.0444 ± 0.0047
Prior: $\mathbf{P} = -\mathbf{H}_{\text{Ent}}^{-1}$								
36	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{I}$	$\mathbf{I}$	0.0589 ± 0.0065	0.2058 ± 0.0047	0.0469 ± 0.0033	0.0725 ± 0.0000
37	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0589 ± 0.0065	0.2118 ± 0.0176	0.0469 ± 0.0033	0.0725 ± 0.0000
38	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0594 ± 0.0065	0.2058 ± 0.0047	0.0469 ± 0.0033	0.0725 ± 0.0000
39	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0591 ± 0.0066	0.2166 ± 0.0096	0.0469 ± 0.0033	0.0725 ± 0.0000
40	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0607 ± 0.0068	0.2118 ± 0.0176	0.0469 ± 0.0033	0.0725 ± 0.0000
41	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.0684 ± 0.0134	0.2800 ± 0.0140	0.0451 ± 0.0039	0.0420 ± 0.0043
42	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0788 ± 0.0287	0.3004 ± 0.0216	0.0448 ± 0.0034	0.0417 ± 0.0041
43	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0952 ± 0.0333	0.2734 ± 0.0272	0.0451 ± 0.0039	0.0411 ± 0.0049
44	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0917 ± 0.0688	0.3590 ± 0.0630	0.0448 ± 0.0033	0.0177 ± 0.0096
45	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0708 ± 0.0118	0.3074 ± 0.1231	0.0456 ± 0.0049	0.0097 ± 0.0203
46	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0555 ± 0.0058	0.1851 ± 0.0061	0.0060 ± 0.0015	0.0561 ± 0.0007
47	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	$\mathbf{0.0553} \pm 0.0059$	0.1826 ± 0.0083	0.0061 ± 0.0018	0.0543 ± 0.0007
48	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0561 ± 0.0062	0.1853 ± 0.0103	0.0061 ± 0.0016	0.0006 ± 0.0000
49	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0581 ± 0.0071	0.1853 ± 0.0103	0.0061 ± 0.0017	0.0006 ± 0.0000
50	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{I}$	0.0560 ± 0.0064	0.2232 ± 0.0828	0.0882 ± 0.0212	0.0006 ± 0.0000
51	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0555 ± 0.0059	0.2232 ± 0.0828	0.0804 ± 0.0072	0.0006 ± 0.0000
52	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0556 ± 0.0058	0.2232 ± 0.0828	0.0838 ± 0.0074	0.0006 ± 0.0000
53	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0576 ± 0.0069	0.2232 ± 0.0828	0.0830 ± 0.0147	0.0006 ± 0.0000
54	VSJ	$-\mathbf{H}_{\text{Ent}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10)$	0.0621 ± 0.0097	0.1746 ± 0.0015	0.0772 ± 0.0061	0.0006 ± 0.0000
Prior: $\mathbf{P} = -\mathbf{H}_{\text{Log}}^{-1}$								
55	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{I}$	$\mathbf{I}$	0.0961 ± 0.0117	0.2128 ± 0.0175	0.0483 ± 0.0023	0.0921 ± 0.0062
56	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0960 ± 0.0117	0.2185 ± 0.0213	0.0483 ± 0.0023	0.0907 ± 0.0042
57	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0964 ± 0.0119	0.2128 ± 0.0175	0.0483 ± 0.0023	0.0902 ± 0.0050
58	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0974 ± 0.0127	0.2128 ± 0.0175	0.0483 ± 0.0023	0.0906 ± 0.0026
59	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0900 ± 0.0190	0.2128 ± 0.0175	0.0483 ± 0.0023	0.0950 ± 0.0041
60	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.3428 ± 0.1407	$\mathbf{0.1146} \pm 0.0338$	0.0507 ± 0.0034	0.0827 ± 0.0026
61	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.3438 ± 0.1405	0.1446 ± 0.1012	0.0507 ± 0.0034	0.0813 ± 0.0037
62	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.3286 ± 0.1470	0.2370 ± 0.0870	0.0507 ± 0.0034	0.0845 ± 0.0048
63	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.3512 ± 0.1453	0.3441 ± 0.0427	0.0507 ± 0.0034	0.0820 ± 0.0023
64	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.1469 ± 0.0676	0.2538 ± 0.0109	0.0507 ± 0.0034	0.0781 ± 0.0069
65	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{I}$	0.0705 ± 0.0063	0.2194 ± 0.0195	0.0423 ± 0.0024	0.0898 ± 0.0057
66	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-3})$	0.0705 ± 0.0065	0.2168 ± 0.0179	0.0423 ± 0.0024	0.0891 ± 0.0028
67	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0703 ± 0.0039	0.2194 ± 0.0195	0.0423 ± 0.0024	0.0903 ± 0.0047
68	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0662 ± 0.0085	0.2194 ± 0.0195	0.0423 ± 0.0024	0.0916 ± 0.0040
69	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0610 ± 0.0061	0.2194 ± 0.0195	0.0423 ± 0.0024	0.0889 ± 0.0044
70	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.2341 ± 0.0954	0.2091 ± 0.0071	0.0131 ± 0.0014	0.0924 ± 0.0023
71	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	$\mathbf{0.0588} \pm 0.0077$	0.2091 ± 0.0071	0.0132 ± 0.0015	0.0834 ± 0.0042
72	VSJ	$-\mathbf{H}_{\text{Log}}^{-1}$	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0595 ± 0.0070	0.2091 ± 0.0071	0.0132 ± 0.0017	0.0779 ± 0.0042

Table 21: **Loss-dependent update validation sweep.** Native SPO+ and Blackbox and their VSJ compositions are evaluated across four metrics and five constraint weightings on four benchmarks. Values are mean ± standard deviation over available seeds (up to five); benchmark identities and regret conventions follow Tab. 20; lower is better. Selection for Tab. 5 minimizes mean validation regret per native update and benchmark over $\mathbf{\Omega} \neq \mathbf{0}$; ties follow fixed-grid order. The TSP $\mathbf{M} = \mathbf{Q}_\theta$ blocks use a quadratic forward program, so their $\mathbf{\Omega} = \mathbf{0}$ rows do not reproduce the native forward. Bold marks the selected compositions.

ID	Configuration	VJP update	M	Ω	Knapsack	TSP	Matching	Portfolio
A. SPO+								
1	Native SPO+	native SPO+	–	–	0.0820 ± 0.0581	0.0072 ± 0.0125	0.0000 ± 0.0000	0.0014 ± 0.0000
Metric: $\mathbf{M} = \mathbf{I}$								
2	SPO+ + VSJ	native SPO+	$\mathbf{I}$	$\mathbf{0}$	0.0820 ± 0.0581	0.0072 ± 0.0125	0.0000 ± 0.0000	0.0014 ± 0.0000
3	SPO+ + VSJ	native SPO+	$\mathbf{I}$	$\mathbf{I}$	0.4822 ± 0.0990	3.4892 ± 0.0476	0.2946 ± 0.0314	0.0996 ± 0.0084
4	SPO+ + VSJ	native SPO+	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0551 ± 0.0060	$\mathbf{0.0000} \pm 0.0000$	0.0000 ± 0.0000	$\mathbf{0.0007} \pm 0.0001$
5	SPO+ + VSJ	native SPO+	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	$\mathbf{0.0546} \pm 0.0059$	0.0294 ± 0.0288	0.0000 ± 0.0000	0.0012 ± 0.0001
6	SPO+ + VSJ	native SPO+	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0558 ± 0.0071	0.0072 ± 0.0125	0.0000 ± 0.0000	0.0015 ± 0.0000
Metric: $\mathbf{M} = \mathbf{Q}_\theta$								
7	SPO+ + VSJ	native SPO+	$\mathbf{Q}_\theta$	$\mathbf{0}$	0.0820 ± 0.0581	0.2309 ± 0.0068	0.0000 ± 0.0000	0.0014 ± 0.0000
8	SPO+ + VSJ	native SPO+	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.0948 ± 0.0198	0.5718 ± 0.1007	0.0011 ± 0.0007	0.0367 ± 0.0318
9	SPO+ + VSJ	native SPO+	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0805 ± 0.0110	0.0369 ± 0.0005	0.0009 ± 0.0006	0.0313 ± 0.0292
10	SPO+ + VSJ	native SPO+	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0700 ± 0.0091	0.0349 ± 0.0002	0.0002 ± 0.0002	0.0015 ± 0.0000
11	SPO+ + VSJ	native SPO+	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0616 ± 0.0043	0.0331 ± 0.0005	0.0000 ± 0.0000	0.0015 ± 0.0001
Metric: $\mathbf{M} = \mathbf{H}_{\text{Ent}}$								
12	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Ent}}$	$\mathbf{0}$	0.0820 ± 0.0581	0.0072 ± 0.0125	0.0000 ± 0.0000	0.0014 ± 0.0000
13	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Ent}}$	$\mathbf{I}$	0.1352 ± 0.0082	3.4892 ± 0.0476	0.0003 ± 0.0001	0.0279 ± 0.0033
14	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.1102 ± 0.0063	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0016 ± 0.0001
15	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.1022 ± 0.0044	0.0192 ± 0.0333	0.0000 ± 0.0000	0.0015 ± 0.0000
16	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0916 ± 0.0024	0.0265 ± 0.0291	0.0000 ± 0.0000	0.0015 ± 0.0001
Metric: $\mathbf{M} = \mathbf{H}_{\text{Log}}$								
17	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Log}}$	$\mathbf{0}$	0.0820 ± 0.0581	0.0072 ± 0.0125	0.0000 ± 0.0000	0.0014 ± 0.0000
18	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Log}}$	$\mathbf{I}$	0.1352 ± 0.0082	3.4892 ± 0.0476	0.0003 ± 0.0001	0.0279 ± 0.0033
19	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.1074 ± 0.0096	0.0000 ± 0.0000	0.0000 ± 0.0000	0.0016 ± 0.0001
20	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.1016 ± 0.0044	0.0192 ± 0.0333	0.0000 ± 0.0000	0.0015 ± 0.0000
21	SPO+ + VSJ	native SPO+	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0903 ± 0.0039	0.0265 ± 0.0291	0.0000 ± 0.0000	0.0015 ± 0.0001
B. Blackbox differentiation								
22	Native Blackbox	native Blackbox	–	–	0.0552 ± 0.0055	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
Metric: $\mathbf{M} = \mathbf{I}$								
23	Blackbox + VSJ	native Blackbox	$\mathbf{I}$	$\mathbf{0}$	0.0552 ± 0.0055	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
24	Blackbox + VSJ	native Blackbox	$\mathbf{I}$	$\mathbf{I}$	0.4825 ± 0.0988	3.4892 ± 0.0476	0.2946 ± 0.0314	$\mathbf{0.0996} \pm 0.0084$
25	Blackbox + VSJ	native Blackbox	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0566 ± 0.0070	0.0012 ± 0.0016	0.0000 ± 0.0000	0.1214 ± 0.0134
26	Blackbox + VSJ	native Blackbox	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0551 ± 0.0064	0.0008 ± 0.0014	0.0000 ± 0.0000	0.1675 ± 0.0003
27	Blackbox + VSJ	native Blackbox	$\mathbf{I}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	$\mathbf{0.0550} \pm 0.0061$	$\mathbf{0.0000} \pm 0.0000$	0.0000 ± 0.0000	0.1675 ± 0.0004
Metric: $\mathbf{M} = \mathbf{Q}_\theta$								
28	Blackbox + VSJ	native Blackbox	$\mathbf{Q}_\theta$	$\mathbf{0}$	0.0552 ± 0.0055	0.2834 ± 0.0075	0.0000 ± 0.0000	0.1678 ± 0.0000
29	Blackbox + VSJ	native Blackbox	$\mathbf{Q}_\theta$	$\mathbf{I}$	0.0557 ± 0.0063	0.2679 ± 0.0169	0.0028 ± 0.0007	0.1119 ± 0.0094
30	Blackbox + VSJ	native Blackbox	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0557 ± 0.0057	0.0835 ± 0.0033	0.0023 ± 0.0002	0.1660 ± 0.0004
31	Blackbox + VSJ	native Blackbox	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0555 ± 0.0055	0.0671 ± 0.0022	0.0021 ± 0.0010	0.1676 ± 0.0002
32	Blackbox + VSJ	native Blackbox	$\mathbf{Q}_\theta$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0552 ± 0.0055	0.0703 ± 0.0043	0.0019 ± 0.0007	0.1678 ± 0.0000
Metric: $\mathbf{M} = \mathbf{H}_{\text{Ent}}$								
33	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Ent}}$	$\mathbf{0}$	0.0552 ± 0.0055	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
34	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Ent}}$	$\mathbf{I}$	0.0585 ± 0.0057	3.4892 ± 0.0476	0.1791 ± 0.0322	0.1670 ± 0.0008
35	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0556 ± 0.0042	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
36	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0555 ± 0.0052	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
37	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Ent}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0553 ± 0.0053	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
Metric: $\mathbf{M} = \mathbf{H}_{\text{Log}}$								
38	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Log}}$	$\mathbf{0}$	0.0552 ± 0.0055	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
39	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Log}}$	$\mathbf{I}$	0.0585 ± 0.0057	3.4892 ± 0.0476	0.1791 ± 0.0322	0.1670 ± 0.0008
40	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-2})$	0.0556 ± 0.0043	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
41	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(10^{-1})$	0.0554 ± 0.0052	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000
42	Blackbox + VSJ	native Blackbox	$\mathbf{H}_{\text{Log}}$	$\mathbf{\Omega}_{\text{adapt}}(1)$	0.0555 ± 0.0052	0.0000 ± 0.0000	0.0000 ± 0.0000	0.1678 ± 0.0000