LEARNING VARIABLE METRICS FOR CONVERGENT PROXIMAL SPLITTING ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose learned variable metric (LVM), a framework that learns metrics of proximal splitting algorithms adapted to each problem instance, coordinate, and iteration with a convergence guarantee. Proximal splitting algorithms can solve a wide variety of convex optimization problems but often require many iterations to converge. Choosing a suitable metric is an effective remedy, and variable metric theory naturally extends convergence guarantees to metrics that vary over the iterations. Learning is a promising way to design such adaptive metrics beyond the reach of manual design, but a naive implementation leads to (i) an expensive evaluation of a network at every iteration and (ii) difficulty in satisfying the conditions that the theory imposes on the whole infinite sequence of metrics. To resolve these difficulties, we first train a neural network to generate a few representative metrics once per problem instance, and then obtain the metric of every iteration by combining them with basis functions of the normalized iteration index. We present LVM for the proximal gradient method with a quadratic term and the primal-dual hybrid gradient method, and prove their convergence under a mild spectral condition, which is ensured by a rescaling of the representative metrics. In experiments on deblurring and inpainting, LVM reduces the number of iterations to reach a target accuracy in most tested settings compared with several baselines, and the computation time at high accuracies.

1 INTRODUCTION

A vast range of problems in signal processing, imaging, and data science are formulated as the convex minimization problem

$$\underset{\boldsymbol{x}\in\mathcal{X}}{\text{minimize}} \quad f(\boldsymbol{x}) + g(L\boldsymbol{x}), \tag{1}$$

where f and g are proper lower-semicontinuous convex functions on real Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$, respectively, and $L\colon \mathcal{X} \to \mathcal{Y}$ is a bounded linear operator. This formulation has a wide variety of applications, e.g., inverse problems in imaging such as deblurring, inpainting, compressed sensing, and tomographic reconstruction (Afonso et al., 2011; Komodakis & Pesquet, 2015; Chambolle & Pock, 2016), sparse coding (Beck & Teboulle, 2009; Veshki & Vorobyov, 2022), and hyperspectral image restoration (Naganuma & Ono, 2023) and unmixing (Iordache et al., 2012; Naganuma & Ono, 2024). Since (1) typically contains nondifferentiable terms, proximal splitting algorithms, which handle such terms through their proximity operators, have been studied extensively under diverse assumptions on f, g, and L (Condat, 2013; Vũ, 2013; Combettes, 2024). A typical setting assumes that $L = \mathrm{Id}$ and f is differentiable with a Lipschitz continuous gradient, in which case the proximal gradient (PG) method is available (Lions & Mercier, 1979; Chen & Rockafellar, 1997; Bauschke & Combettes, 2017). For a general L, the primal-dual hybrid gradient (PDHG) method is widely used, which avoids an inversion induced by L at each iteration (Chambolle & Pock, 2011).

Despite their flexibility, proximal splitting algorithms often require a large number of iterations to obtain an accurate solution (Pock & Chambolle, 2011; Themelis & Patrinos, 2019; Zhang et al., 2020). Accelerating their convergence has therefore received considerable attention, as we review below.

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.

Table 1: Properties of representative acceleration schemes and ours. Parentheses indicate a dependency of the property on the work.

Scheme	Problem-specific[†]	Coordinate-wise	Iteration-varying	No extra evaluations[‡]	Convergence guarantee
Inertial acceleration (e.g., Beck & Teboulle, 2009)			✓	✓	✓
Preconditioning (e.g., Pock & Chambolle, 2011)	✓	✓		✓	✓
Line search (e.g., Malitsky & Pock, 2018)	✓		✓		✓
Residual balancing (Goldstein et al., 2015)	✓		✓	✓	✓
Curvature estimation (Malitsky & Mishchenko, 2024)	✓		✓	✓	✓
Learned maps (e.g., Banert et al., 2024)	✓	✓	✓		✓
Learned stepsizes (e.g., Sambharya & Stellato, 2026)			✓	✓	(✓)
Learned fixed metrics (e.g., King et al., 2024)	✓	✓		✓	✓
LVM (ours)	✓	✓	✓	✓	✓

[†]Adapted to each problem instance. [‡]No additional evaluation of operators or networks per iteration.

1.1 Related Work

Acceleration for Proximal Splitting Algorithms. Table 1 summarizes representative schemes. One of the most well-established approaches for acceleration introduces an inertial term that does not depend on the problem structure, as in Nesterov's acceleration (Beck & Teboulle, 2009; Parikh & Boyd, 2014; O'Donoghue & Candès, 2015) and the heavy-ball method (Alvarez & Attouch, 2001; Boţ et al., 2015). Preconditioning also speeds up convergence (Liu et al., 2021; Bredies et al., 2022) by replacing the metric of the underlying space with one adapted to the problem, e.g., to set stepsizes coordinate-wise (Pock & Chambolle, 2011) or to exploit the block structure of L (Wen et al., 2017; Naganuma & Ono, 2023). Although preconditioning applies to a wide range of algorithms, its metric is determined by hand-designed rules and does not change along the iterations. Another line of work adapts the stepsizes to the current iterate, to take larger steps by estimating the local Lipschitz constant with line search (Malitsky & Pock, 2018; Chang et al., 2022). However, these schemes may require repeated trial updates at every iteration, which can be expensive in practice. Other schemes balance the multiple stepsizes in PDHG (Goldstein et al., 2015) or estimate the local Lipschitz constant from consecutive iterates (Malitsky & Mishchenko, 2020; 2024), but they adapt only scalar stepsizes.

Learned Solvers. The success of deep learning in recent years opens a new direction for solving (1), as the expressive power of neural networks allows complex rules of algorithms to be learned from data. The dominant approach is unrolling, which truncates the algorithm to a fixed number of iterations and trains the parameters end to end (Gregor & LeCun, 2010; Adler & Öktem, 2017; Cherkaoui et al., 2020; Monga et al., 2021). It is typically trained so that the output well approximates a solution to (1) or directly approximates the desired solution modeled by (1) (Chen et al., 2018), but asymptotic convergence beyond the trained depth falls outside its standard formulation. Another attempt learns a map appearing in the update, e.g., a descent direction, and evaluates the network at every iteration to accelerate the algorithm. These methods ensure convergence by, e.g., confining the output to a certain set (Banert et al., 2024), by a safeguard that falls back on an original update when a criterion fails (Heaton et al., 2023), or by restricting the learned geometry (Tan et al., 2023). Since such an evaluation of a network at every iteration typically costs much more than one iteration of the algorithm, several works instead learn a smaller part of the algorithm. The stepsizes are one such target, adaptive over the iterations but shared by all the coordinates and the problem instances (Ablin et al., 2019; Sambharya & Stellato, 2026), and a similar idea applies to the coefficients of a generalized primal-dual scheme (Banert et al., 2020). Other works learn a metric that adjusts each coordinate but is fixed over the iterations (Li et al., 2023, for linear solvers; King et al., 2024).

Variable Metric Algorithms. Variable metric theory studies algorithms whose metric changes along the iterations (Bonnans et al., 1995; Combettes & Vũ, 2012; Combettes et al., 2014). One of its central topics is their convergence (Combettes & Vũ, 2014; Suzuki et al., 2026), which can be established under several assumptions, including the requirement that the metrics decrease only

by summable amounts between consecutive iterations (detailed in Section 2.1). Under this framework, several works design such adaptive metrics for acceleration with convergence guarantees, by a quadratic majorant of the smooth term (Chouzenoux et al., 2014), a line search (Bonettini et al., 2016; Salzo, 2017), or a quasi-Newton update (Becker et al., 2019). When metrics are restricted to scalar ones or diagonal ones with a block structure, the theory also supplies the analysis behind the adaptive stepsize rules, leading to, e.g., a convergent, spectrally motivated tuning for primal-dual Douglas–Rachford splitting (Lorenz et al., 2025) and a convergent stochastic line-search strategy for PDHG (Suzuki et al., 2026).

1.2 Contributions

As the above discussion suggests, learning the metrics may provide a way to achieve complex adaptation to each problem instance, coordinate, and iteration beyond the reach of manual design, potentially with a convergence guarantee from variable metric theory. However, a naive implementation of this approach faces two difficulties. First, if the metrics are produced by network evaluations during the iterations, the acceleration can be impractical since evaluating a network typically costs much more than a single iteration of these algorithms. Second, variable metric theory imposes conditions on the whole infinite sequence of metrics, which are difficult to ensure by training alone.

These difficulties motivate the central question of this paper. *How can we learn problem-specific, coordinate-wise, and iteration-varying metrics of proximal splitting algorithms without extra evaluations per iteration and with a convergence guarantee?* To answer this question, we propose a framework called learned variable metric (LVM). Our contributions are summarized as follows.

- We propose a method that generates a metric sequence varying over all the iterations by evaluating a neural network only once before the iterations. To this end, we design a few basis functions over the iterations, and introduce representative metrics for each basis function. The metric of every iteration is then obtained by combining the representative metrics with the basis functions. Each metric is also designed to be diagonal with a structure that keeps the proximity operators relative to it efficiently computable. These approaches make it possible to use the problem-specific, coordinate-wise, and iteration-varying metrics without incurring substantial additional cost at each iteration (Section 3.1).
- We develop specializations of LVM to PG with a quadratic term and PDHG. To satisfy the conditions for convergence, which reduce to those on the representative metrics by the construction of the sequence, we rescale them by an efficiently computable spectral factor. We also prove convergence of both algorithms when the rescaling is exact (Section 3.2).
- We train the network with a loss that measures the distance to a reference solution at several iterations of the algorithm. This loss is designed to accelerate convergence over a wide range of iterations by taking the logarithm of the distance (Section 3.3).

Experiments on image deblurring and inpainting show that LVM needs fewer iterations than several baselines in most tested settings, and that its one-time setup cost pays off at high target accuracies. To the best of our knowledge, LVM is the first framework that provides convergence guarantees for proximal splitting algorithms with learned coordinate-wise metrics that are adapted to each problem instance and vary over all the iterations without additional evaluations per iteration.

2 Preliminaries

Throughout, $\mathcal{X}$ and $\mathcal{Y}$ are real Hilbert spaces equipped with an inner product $\langle \cdot \mid \cdot \rangle$ and its induced norm $\| \cdot \|$. We denote by $\mathcal{B}(\mathcal{X}, \mathcal{Y})$ the space of bounded linear operators from $\mathcal{X}$ to $\mathcal{Y}$, by $\mathcal{S}(\mathcal{X})$ the set of self-adjoint operators in $\mathcal{B}(\mathcal{X}, \mathcal{X})$, and by Id the identity operator. Let $\Gamma_0(\mathcal{X})$ be the set of proper lower-semicontinuous convex functions from $\mathcal{X}$ to $\mathbb{R} \cup \{+\infty\}$, and $\ell^1_+(\mathbb{N})$ be the set of summable sequences in $[0, +\infty)$. The Fenchel–Rockafellar conjugate of $f \in \Gamma_0(\mathcal{X})$ is defined as $f^*(\boldsymbol{x}) := \sup_{\boldsymbol{u} \in \mathcal{X}}(\langle \boldsymbol{x} \mid \boldsymbol{u} \rangle - f(\boldsymbol{u}))$. The Löwner partial ordering on $\mathcal{S}(\mathcal{X})$ is

$$U_1 \succeq U_2 \iff (\forall \boldsymbol{x} \in \mathcal{X}) \quad \langle U_1 \boldsymbol{x} \mid \boldsymbol{x} \rangle \geq \langle U_2 \boldsymbol{x} \mid \boldsymbol{x} \rangle, \tag{2}$$

and we set $\mathcal{P}_\alpha(\mathcal{X}) := \{U \in \mathcal{S}(\mathcal{X}) \mid U \succeq \alpha \mathrm{Id}\}$ for $\alpha > 0$. Every $U \in \mathcal{P}_\alpha(\mathcal{X})$ induces the inner product $\langle \boldsymbol{x}_1 \mid \boldsymbol{x}_2 \rangle_U := \langle \boldsymbol{x}_1 \mid U\boldsymbol{x}_2 \rangle$ and its induced norm $\| \cdot \|_U$. The proximity operator of

$f \in \Gamma_0(\mathcal{X})$ relative to the metric induced by U is

$$\operatorname{prox}_f^U(\boldsymbol{x}) := \operatorname*{arg\,min}_{\boldsymbol{u}\in\mathcal{X}} \; f(\boldsymbol{u}) + \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_U^2, \tag{3}$$

and we write $\operatorname{prox}_f := \operatorname{prox}_f^{\mathrm{Id}}$. A function is called proximable when its proximity operator can be computed efficiently. More details are collected in Appendix A.

2.1 Proximal Splitting Algorithms on Variable Metrics

In (1), we assume that $f \in \Gamma_0(\mathcal{X})$, $g \in \Gamma_0(\mathcal{Y})$, $L \in \mathcal{B}(\mathcal{X},\mathcal{Y})$, and that its solution set is nonempty. Both PG and PDHG admit variable metric extensions, as we state below.

PG on variable metrics with a quadratic term. Suppose in (1) that $L = \mathrm{Id}$ and that $f = \frac{1}{2}\|A\cdot-\boldsymbol{b}\|^2$, where $A \in \mathcal{B}(\mathcal{X},\mathcal{W})$ and $\boldsymbol{b} \in \mathcal{W}$ for a real Hilbert space $\mathcal{W}$. With metrics $P_n \in \mathcal{P}_\gamma(\mathcal{X})$, the updates are given as

$$\boldsymbol{x}_{n+1} = \operatorname{prox}_g^{P_n^{-1}}\left(\boldsymbol{x}_n - P_n A^*(A\boldsymbol{x}_n - \boldsymbol{b})\right). \tag{4}$$

The following result states the convergence of (4), and is proved in Appendix C.

Theorem 2.1. *Let $\gamma > 0$ and let $\zeta > 0$. Suppose that the solution set to (1) is nonempty. Assume that the following hold:*

(i) $(\forall n \in \mathbb{N})\ P_n \in \mathcal{P}_\gamma(\mathcal{X})$.

(ii) $(\exists\{\eta_n\} \in \ell_+^1(\mathbb{N}))(\forall n \in \mathbb{N})\ (1+\eta_n)P_{n+1} \succeq P_n$.

(iii) $(\forall n \in \mathbb{N})\ P_n^{-1} - \frac{1}{2}A^*A \succeq \zeta\mathrm{Id}$.

Then $\{\boldsymbol{x}_n\}$ defined by (4) weakly converges to a solution of (1).

PDHG on variable metrics. Consider now the case where f and g in (1) are both proximable and L is arbitrary. We use the primal-dual inclusion associated with (1),

$$\text{find } (\hat{\boldsymbol{x}},\hat{\boldsymbol{y}}) \in \mathcal{X}\times\mathcal{Y} \text{ such that } \begin{pmatrix}0\\0\end{pmatrix} \in \begin{pmatrix}\partial f(\hat{\boldsymbol{x}}) + L^*\hat{\boldsymbol{y}}\\ -L\hat{\boldsymbol{x}} + \partial g^*(\hat{\boldsymbol{y}})\end{pmatrix}, \tag{5}$$

whose solution set consists of the associated primal-dual solution pairs under mild qualification (Chambolle & Pock, 2011; Condat, 2013; Vũ, 2013). With metrics $P_n \in \mathcal{P}_\gamma(\mathcal{X})$ and $Q_n \in \mathcal{P}_\delta(\mathcal{Y})$, the updates are given as

$$\left\lfloor\begin{aligned}\boldsymbol{x}_{n+1} &= \operatorname{prox}_f^{P_n^{-1}}\left(\boldsymbol{x}_n - P_nL^*\boldsymbol{y}_n\right),\\ \boldsymbol{y}_{n+1} &= \operatorname{prox}_{g^*}^{Q_n^{-1}}\left(\boldsymbol{y}_n + Q_nL(2\boldsymbol{x}_{n+1}-\boldsymbol{x}_n)\right).\end{aligned}\right. \tag{6}$$

The following result states the convergence of (6). It is the convergence theorem of Suzuki et al. (2026) with its assumption on $\|P_n\|^{-1}\|Q_n\|^{-1} - \|L\|^2$ replaced by the spectral condition (iii) on the block operator V_n, which is the form our framework satisfies directly. The proof is discussed in Appendix D.

Theorem 2.2. *Let $\gamma > 0$, let $\delta > 0$, and let $\varsigma > 0$. Suppose that the solution set to (5) is nonempty. Assume that the following hold:*

(i) $(\forall n \in \mathbb{N})\ P_n \in \mathcal{P}_\gamma(\mathcal{X})$, $Q_n \in \mathcal{P}_\delta(\mathcal{Y})$.

(ii) $(\exists\{\eta_n^{(1)}\},\{\eta_n^{(2)}\} \in \ell_+^1(\mathbb{N}))(\forall n \in \mathbb{N})\ (1+\eta_n^{(1)})P_{n+1} \succeq P_n$, $(1+\eta_n^{(2)})Q_{n+1} \succeq Q_n$.

(iii) $(\forall n \in \mathbb{N})\ V_n \in \mathcal{P}_\varsigma(\mathcal{X}\times\mathcal{Y})$, *where*

$$V_n := \begin{pmatrix}P_n^{-1} & -L^*\\ -L & Q_n^{-1}\end{pmatrix}.$$

Then $\{(\boldsymbol{x}_n,\boldsymbol{y}_n)\}$ defined by (6) weakly converges to a solution of (5).

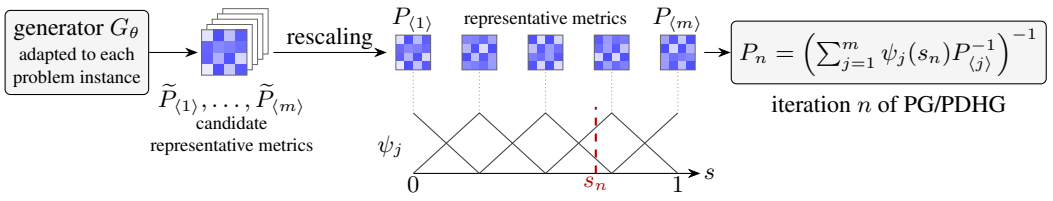


Figure 1: Overview of LVM. Before the iterations, the generator is evaluated once, and its outputs are rescaled into the representative metrics. At iteration n, the metric P_n combines their inverses.

3 Learned Variable Metric (LVM)

LVM constructs the metric sequence in two stages, as shown in Figure 1. Before the iterations, a neural network G_θ with parameters θ, called the *generator*, takes the data and parameters of the problem instance as input. It outputs m metrics $\widetilde{P}_{\langle 1\rangle}, \ldots, \widetilde{P}_{\langle m\rangle}$, whose diagonal entries are made positive by a softplus function, and they are rescaled into $P_{\langle 1\rangle}, \ldots, P_{\langle m\rangle}$ as described in Section 3.2.

In this work, we call $\widetilde{P}_{\langle j\rangle}$ a *candidate representative metric* and $P_{\langle j\rangle}$ a *representative metric*. During the iterations, the metric of each iteration is obtained by combining the representative metrics with coefficients given by fixed functions $\psi_1, \ldots, \psi_m$, termed the *basis* in the following. Section 3.1 describes this construction of the metric sequence from given representative metrics, and Section 3.2 specializes the representative metrics to PG and PDHG to guarantee convergence.

3.1 Metric Sequences from Representative Metrics

We first describe the construction of the metric sequence from representative metrics. Let $\mathcal{H}$ be $\mathcal{X}$ or $\mathcal{Y}$, let $m \geq 2$, and let $P_{\langle 1\rangle}, \ldots, P_{\langle m\rangle} \in \mathcal{P}_\alpha(\mathcal{H})$ be representative metrics for some $\alpha > 0$. For a constant $c > 0$, we map each iteration $n \in \mathbb{N}$ to the normalized time $s_n := n/(n+c) \in [0, 1)$, and set the representative times $\sigma_j := (j-1)/(m-1)$ for $j \in \{1, \ldots, m\}$. The basis consists of the hat functions

$$(\forall s \in [0,1]) \quad \psi_j(s) := \max\{1 - (m-1)|s - \sigma_j|, 0\}, \tag{7}$$

and the metric of iteration n is

$$P_n := \left(\sum_{j=1}^m \psi_j(s_n) P_{\langle j\rangle}^{-1}\right)^{-1}. \tag{8}$$

Since we design the representative metrics to be diagonal as described later, forming (8) requires only elementwise operations on two representative metrics.

The basis has several convenient properties. At $s_n = \sigma_j$, only ψ_j is nonzero and $P_n = P_{\langle j\rangle}$. Between two consecutive representative times, only the two corresponding basis functions are nonzero, and the inverse metric changes linearly in s_n. As s_n increases to 1, the m basis functions cover all the iterations, without fixing the number of iterations in advance or switching to a fixed metric.

The sequence $\{P_n\}$ in (8) has the following basic properties that a metric sequence should satisfy to lead to algorithms with convergence guarantees. More specifically, by (a) and (b) below, the sequence (8) satisfies assumptions (i) and (ii) of Theorems 2.1 and 2.2 for any representative metrics. The proof of the following proposition is given in Appendix B.

Proposition 3.1. *Let $\alpha > 0$, let $c > 0$, let $m \geq 2$, and let $P_{\langle 1\rangle}, \ldots, P_{\langle m\rangle} \in \mathcal{P}_\alpha(\mathcal{H})$. Set $\mu := \max_{1 \leq j \leq m} \|P_{\langle j\rangle}\|$, and let $\{P_n\}$ be defined by (8). Then the following hold.*

(a) $(\forall n \in \mathbb{N})\ \mu \mathrm{Id} \succeq P_n \succeq \alpha \mathrm{Id}$.

(b) $(\forall n \in \mathbb{N})\ (1 + \eta_n) P_{n+1} \succeq P_n$, *where* $\eta_n := \mu \|P_{n+1}^{-1} - P_n^{-1}\|$, *and* $\{\eta_n\} \in \ell_+^1(\mathbb{N})$.

(c) For every $n \in \mathbb{N}$, P_n^{-1} is a convex combination of $P_{\langle 1\rangle}^{-1}, \ldots, P_{\langle m\rangle}^{-1}$ with the coefficients $\psi_1(s_n), \ldots, \psi_m(s_n)$.

Note that Proposition 3.1 relies only on the nonnegativity, the partition of unity, and the bounded variation of the basis along $\{s_n\}$, and other bases with these properties can also be used. Thanks to

the design of (8) that takes a convex combination of the inverse metrics, Proposition 3.1 (c) implies that assumption (iii) needs to be ensured only for the m representative metrics, since (iii) is affine in the inverse metrics.

We design each metric to be diagonal, and choose its structure so that the proximity operators in the updates can still be computed efficiently. For instance, if a function is separable over the coordinates, as the ℓ_1 norm and the indicator of a box are, each coordinate has its own entry, and the proximity operator is computed coordinatewise (Appendix F). If a function acts on groups of coordinates, as the indicator of an ℓ_2 ball does, one entry is shared within each group, and the proximity operator reduces to that of a scaled function on each group.

3.2 Convergence Guarantees of PG/PDHG with LVM

Spectral rescaling of representative metrics. In view of Proposition 3.1, the sequence (8) satisfies assumptions (i) and (ii) of Theorems 2.1 and 2.2 for any representative metrics. To ensure the remaining assumption (iii) for each algorithm, we introduce a spectral rescaling of the representative metrics.

Recall that the candidate representative metrics are denoted by $\widetilde{P}_{\langle j\rangle}$, and we denote by $\widetilde{Q}_{\langle j\rangle}$ those for the dual variable in PDHG. We generate the representative metrics from these candidates by the following process with a margin $\rho \in (0,1)$:

$$P_{\langle j\rangle} := \frac{2\rho}{\beta_j}\widetilde{P}_{\langle j\rangle} \qquad \text{with} \quad \beta_j := \|A\widetilde{P}_{\langle j\rangle}^{1/2}\|^2 \quad \text{for PG,} \tag{9}$$

$$(P_{\langle j\rangle}, Q_{\langle j\rangle}) := \frac{\rho}{\beta_j}(\widetilde{P}_{\langle j\rangle}, \widetilde{Q}_{\langle j\rangle}) \qquad \text{with} \quad \beta_j := \|\widetilde{Q}_{\langle j\rangle}^{1/2} L \widetilde{P}_{\langle j\rangle}^{1/2}\| \quad \text{for PDHG,} \tag{10}$$

and $\{Q_n\}$ for PDHG is defined from $Q_{\langle 1\rangle}, \dots, Q_{\langle m\rangle}$ by (8) analogously to $\{P_n\}$. The following proposition shows that (iii) then holds for every iteration (see Appendix E for the proof).

Proposition 3.2. *Let $\alpha > 0$ and let $\rho \in (0,1)$.*

- *(a) Suppose that $\widetilde{P}_{\langle j\rangle} \in \mathcal{P}_\alpha(\mathcal{X})$ and $P_{\langle j\rangle}$ is given by (9) for every $j \in \{1,\dots,m\}$, and set $\zeta := (1-\rho)\min_{1\le j\le m}\|P_{\langle j\rangle}\|^{-1}$. Then $\{P_n\}$ defined by (8) satisfies assumption (iii) of Theorem 2.1.*
- *(b) Suppose that $\widetilde{P}_{\langle j\rangle} \in \mathcal{P}_\alpha(\mathcal{X})$, $\widetilde{Q}_{\langle j\rangle} \in \mathcal{P}_\alpha(\mathcal{Y})$, and $(P_{\langle j\rangle}, Q_{\langle j\rangle})$ is given by (10) for every $j \in \{1,\dots,m\}$, and set $\varsigma := (1-\rho)\min_{1\le j\le m}\big(\min\{\|P_{\langle j\rangle}\|^{-1}, \|Q_{\langle j\rangle}\|^{-1}\}\big)$. Then $\{P_n\}$ and $\{Q_n\}$ defined by (8) satisfy assumption (iii) of Theorem 2.2.*

In our implementation, we approximate β_j by the power iteration (Appendix G.2), and we verify assumption (iii) for the resulting metrics by the Lanczos method (Kuczyński & Woźniakowski, 1992) in Appendix I.

Convergence of PG and PDHG. By combining Theorems 2.1 and 2.2 with Propositions 3.1 and 3.2, we obtain the convergence guarantees for PG and PDHG. We formalize these guarantees in the following theorems, which are proved in Appendix E.

Theorem 3.3 (PG with a quadratic term)**.** *Consider the setting of Theorem 2.1, i.e., $L = \mathrm{Id}$ and $f = \frac{1}{2}\|A\cdot - \boldsymbol{b}\|^2$ with $A \in \mathcal{B}(\mathcal{X},\mathcal{W})$ and $\boldsymbol{b} \in \mathcal{W}$. Suppose that the solution set to (1) is nonempty. Let $\alpha > 0$ and let $\rho \in (0,1)$. For every $j \in \{1,\dots,m\}$, let $\widetilde{P}_{\langle j\rangle} \in \mathcal{P}_\alpha(\mathcal{X})$, and let $P_{\langle j\rangle}$ be given by (9). Then $\{\boldsymbol{x}_n\}$ defined by (4) with $\{P_n\}$ defined by (8) weakly converges to a solution of (1).*

Theorem 3.4 (PDHG)**.** *Suppose that the solution set to (5) is nonempty. Let $\alpha > 0$ and let $\rho \in (0,1)$. For every $j \in \{1,\dots,m\}$, let $\widetilde{P}_{\langle j\rangle} \in \mathcal{P}_\alpha(\mathcal{X})$ and $\widetilde{Q}_{\langle j\rangle} \in \mathcal{P}_\alpha(\mathcal{Y})$, and let $(P_{\langle j\rangle}, Q_{\langle j\rangle})$ be given by (10). Then $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ defined by (6) with $\{P_n\}$ and $\{Q_n\}$ defined by (8) weakly converges to a solution of (5).*

Importantly, these theorems do not rely on the training of the generator. LVM therefore separates the roles of the rescaling and the training, with the former ensuring convergence and the latter providing acceleration. In particular, the guarantees do not restrict how the candidate representative metrics

Table 2: Mean number of iterations of PG until $\mathcal{G}_n$ first falls below each tolerance on Set10. The best and second-best values among the methods without an inertial term are shown in bold and underlined, respectively, and the last two rows are shown for reference. The notation "> 500" indicates that the criterion was not satisfied within 500 iterations for at least one image.

Method	Deblurring				Inpainting			
	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$
LVM (ours)	<u>4</u>	**8**	**23**	**54**	**3**	**7**	**34**	**95**
Fixed scalar (Bauschke & Combettes, 2017)	24	33	49	<u>99</u>	47	93	148	238
Jacobi diag. (Combettes & Vũ, 2014)	7	36	73	119	166	308	> 500	> 500
AdaPGM (Malitsky & Mishchenko, 2024)	**3**	<u>9</u>	<u>26</u>	**54**	<u>33</u>	<u>64</u>	<u>100</u>	<u>151</u>
With an inertial term								
FISTA (Beck & Teboulle, 2009)	3	10	22	35	24	39	55	70
LVM (ours) + FISTA	3	9	16	26	4	10	22	37

are obtained, as long as they belong to $\mathcal{P}_\alpha(\mathcal{X})$ or $\mathcal{P}_\alpha(\mathcal{Y})$ for some $\alpha > 0$ (i.e., their diagonal entries are positive when $\mathcal{X}$ and $\mathcal{Y}$ are finite dimensional). If necessary, we can modify the output of the generator by hand after training or blend it with another preconditioner without losing the convergence guarantees.

3.3 Training of the Metric Generator

We implement the generator G_θ as a neural network with parameters θ that maps the data and parameters of each problem instance to the m candidate representative metrics. We train θ on a set of problem instances whose reference solutions $\boldsymbol{x}^\star$ are computed in advance. Since Theorems 3.3 and 3.4 hold for any θ, the loss can be designed without any penalty or constraint for ensuring the convergence. For each instance, the algorithm runs T iterations with the metrics given by (8), and we minimize the average over the instances of

$$\mathcal{L}(\theta) := \sum_{t \in \mathcal{T}} \omega_t \log \frac{\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2}{\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}, \tag{11}$$

where $\boldsymbol{x}_t$ is the (primal) iterate at iteration t, which depends on θ, $\mathcal{T} \subset \{1, \dots, T\}$ is a set of checkpoint iterations, and $\omega_t \geq 0$ are weights. This loss is designed to learn metrics under which the distance to the reference solution decreases in a balanced manner throughout the iterations. Since the distance decreases by orders of magnitude along the iterations, the sum of the distances would be dominated by the early checkpoints. Hence we balance their contributions by the logarithm, and the weights ω_t determine which checkpoints are emphasized according to their iteration t. The gradient of (11) is computed by backpropagation through the T iterations and the rescaling in Section 3.2. Since the training is performed only once in advance, LVM is particularly useful in applications that solve many problem instances of the same form, e.g., restoring images degraded in the same manner. The detailed settings are specified in Appendices G.3 and G.4.

4 Experiments

We evaluate the proposed framework on two image restoration problems, deblurring and inpainting, each solved by PG and by PDHG. PG is applied to a sparse regularization problem on wavelet coefficients, and PDHG to a total variation problem with a box constraint and a constrained ℓ_2 data fidelity. One generator is trained for each pair of an algorithm and an observation operator on crops of natural images, and is evaluated on standard test images of larger sizes without retraining. We compare the number of iterations and the computation time to reach a target accuracy with fixed stepsizes, classical diagonal preconditioners, and adaptive stepsize schemes. The detailed settings are described in Appendix G, and an ablation study of the learned metrics is given in Appendix J.

4.1 Acceleration of PG

We compare LVM with four methods for PG. Fixed scalar uses one stepsize selected by a grid search on training images and serves as the baseline without adaptation. Jacobi diag. adapts the metric to each coordinate by a rule fixed in advance, using the inverse of an estimate of the diagonal of A^*A (Combettes & Vũ, 2014). AdaPGM (Malitsky & Mishchenko, 2024) updates a scalar stepsize

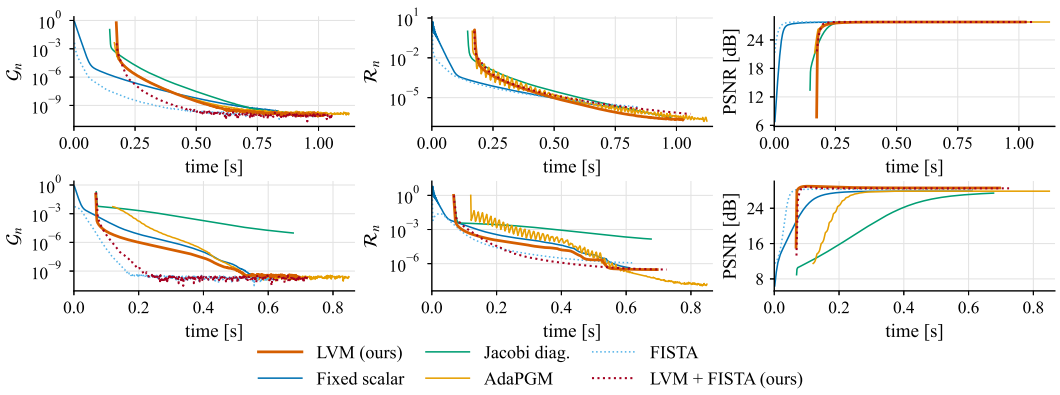


Figure 2: Convergence of PG for deblurring (top) and inpainting (bottom), averaged over Set10 and plotted against the computation time including the setup before the first iteration. The columns show the normalized objective gap $\mathcal{G}_n$, the relative change $\mathcal{R}_n$, and the PSNR [dB].

at every iteration from the curvature estimated with consecutive iterates. FISTA (Beck & Teboulle, 2009) accelerates the iterations by an inertial term instead of adapting the metric, and this term can be added to any choice of the metric, including LVM. We therefore compare the metrics among the methods without an inertial term, and show FISTA and LVM + FISTA, which runs FISTA on the metrics of LVM (Appendix G.6), separately. We evaluate the convergence in terms of the normalized objective gap $\mathcal{G}_n := (F(\boldsymbol{c}_n) - F^\star)/(F(\boldsymbol{c}_0) - F^\star)$, where F is the objective function and $F^\star$ is the smallest value observed. The relative change $\mathcal{R}_n := \|\boldsymbol{c}_n - \boldsymbol{c}_{n-1}\|/\|\boldsymbol{c}_n\|$ is also used as an auxiliary measure, but we note that a decrease of $\mathcal{R}_n$ does not necessarily imply a decrease of the objective value or the convergence of the algorithm in our experimental setting since it can become small even when an overly conservative stepsize or metric is chosen.

Table 2 reports the mean number of iterations until $\mathcal{G}_n$ first falls below each tolerance. LVM needs fewer iterations than Fixed scalar and Jacobi diag. in all cases, and the fewest among the methods without an inertial term for every tolerance in inpainting. Jacobi diag. shares the diagonal structure with LVM but needs more iterations than Fixed scalar in inpainting, which indicates that the acceleration comes from the choice of the entries adapted to each problem instance rather than the diagonal structure alone. We investigate the convergence against the computation time in Figure 2, where the time includes the setup before the first iteration. For LVM, the setup is the construction of the metrics, which takes about 0.17 s in deblurring and 0.07 s in inpainting. Since the construction is performed only once and one iteration of LVM takes only about 3% longer than that of Fixed scalar, this cost is recovered as the iterations proceed, and $\mathcal{G}_n$ of LVM becomes smaller than that of Fixed scalar at around $\mathcal{G}_n = 10^{-7}$ in deblurring and 10^{-3} in inpainting. We also examine the iterates of each method in inpainting in Figure 4 in Appendix H. The iterate of LVM already has no visible artifacts at $n = 5$, which suggests that LVM recovers coarse image structures early.

Although FISTA reaches $\mathcal{G}_n \leq 10^{-6}$ in fewer iterations than LVM in both tasks, LVM + FISTA attains $\mathcal{G}_n \leq 10^{-5}$ and $\mathcal{G}_n \leq 10^{-6}$ in both tasks in the fewest iterations, which suggests that the learned metrics can be combined with an inertial term. Note that the convergence of the combination does not follow directly from Theorem 2.1.

4.2 Acceleration of PDHG

For PDHG, LVM is compared with five methods. Tuned VDP assigns one stepsize to the primal variable and one to each of the two dual variables (Condat et al., 2023) selected by a grid search on training images and fixed over the iterations. P–C diag. (Pock & Chambolle, 2011) and OVDP (Naganuma & Ono, 2023) are diagonal preconditioners given by rules fixed in advance, based on the absolute values of the entries of L and on the operator norms of its blocks, respectively. Adaptive (Goldstein et al., 2015) and PDAL (Malitsky & Pock, 2018) adapt scalar stepsizes along the iterations, the former by balancing the primal and dual residuals and the latter by a line search. Since PDHG solves a saddle point problem, the objective value does not decrease monotonically, and the primal-dual gap is not directly available because the iterates may be infeasible. We instead

Table 3: Mean number of iterations of PDHG until $\mathcal{D}_n$ first falls below each tolerance on Set10. The best and second-best values are shown in bold and underlined, respectively. The notation "> 1000" indicates that the criterion was not satisfied within 1000 iterations for at least one image.

Method	Deblurring				Inpainting			
	$\leq 10^{-2}$	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-2}$	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-5}$
LVM (ours)	**24**	**40**	**62**	**89**	**15**	**26**	**39**	**56**
Tuned VDP (Condat et al., 2023)	<u>47</u>	<u>77</u>	<u>115</u>	<u>172</u>	<u>20</u>	<u>34</u>	<u>56</u>	<u>121</u>
P–C diag. (Pock & Chambolle, 2011)	> 1000	> 1000	> 1000	> 1000	22	57	177	455
OVDP (Naganuma & Ono, 2023)	> 1000	> 1000	> 1000	> 1000	27	66	214	602
Adaptive (Goldstein et al., 2015)	> 1000	> 1000	> 1000	> 1000	170	330	560	> 1000
PDAL (Malitsky & Pock, 2018)	> 1000	> 1000	> 1000	> 1000	32	114	358	> 1000

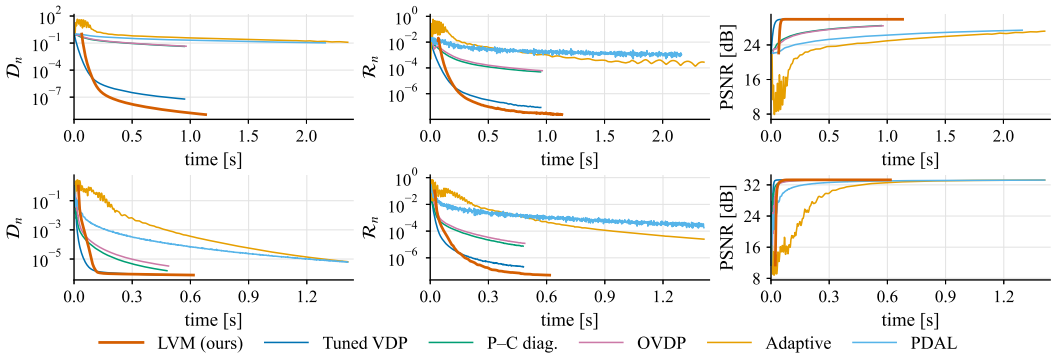


Figure 3: Convergence of PDHG for deblurring (top) and inpainting (bottom), averaged over Set10 and plotted against the computation time including the setup before the first iteration. The columns show the normalized squared distance $\mathcal{D}_n$, the relative change $\mathcal{R}_n$, and the PSNR [dB].

measure the convergence by the normalized squared distance $\mathcal{D}_n := \|\boldsymbol{x}_n - \boldsymbol{x}^\star\|^2/\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2$ to the reference solution $\boldsymbol{x}^\star$. The relative change $\mathcal{R}_n := \|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\|/\|\boldsymbol{x}_n\|$ of the primal variable also serves as an auxiliary measure.

Table 3 reports the mean number of iterations until $\mathcal{D}_n$ first falls below each tolerance. In both tasks, LVM reaches every tolerance in the fewest iterations. Adaptive and PDAL, which use scalar stepsizes instead of diagonal metrics, need the most iterations in inpainting, primarily because they do not exploit the block structure of L in this problem. Tuned VDP, P–C diag., and OVDP adapt to the block coordinates, but need more iterations than LVM. We attribute this to the fact that these methods determine the adaptation by hand, either by general rules or by tuning a few stepsizes, whereas LVM represents through training a complex structure beyond the reach of such modeling.

We investigate the convergence against the computation time in Figure 3, where the time includes the setup before the first iteration. For LVM, the setup is the construction of the metrics, which takes about 0.065 s in deblurring and 0.022 s in inpainting. Since the construction is performed only once and one iteration of LVM takes about 12% longer than that of Tuned VDP in deblurring and 25% longer in inpainting, this cost is recovered as the iterations proceed as in the case of PG, and $\mathcal{D}_n$ of LVM becomes smaller than that of Tuned VDP at around $\mathcal{D}_n = 10^{-5}$ in both tasks.

5 Conclusion

We proposed LVM, a framework that accelerates proximal splitting algorithms with metrics adapted to each problem instance, coordinate, and iteration. The generator produces a few representative metrics once, and the metric of every iteration combines them with a fixed basis over the iterations. With the spectral rescaling of the representative metrics, we showed that PG with a quadratic term and PDHG with LVM converge to a solution for every output of the generator when the rescaling is exact, and the assumptions of the theorems were verified numerically for the metrics used in our experiments (Appendix I). In experiments on deblurring and inpainting solved by PG and PDHG, LVM leads to fewer iterations to reach a target accuracy in most tested settings than several baselines, and less computation time mainly at high accuracies.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models and conceptual frameworks, formulate mathematical claims, assist in the writing of proofs, propose and refine hypotheses, design and provide feedback on research methodology and experiments, implement methods, assist with translation, and interpret results. We have not used generative AI tools to provide critical ingredients for proving mathematical claims, generate synthetic data sets, or clean and reformat datasets, and support of qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to create and modify scientific figures, suggest experimental parameters, create and edit software code, draft parts of the paper, summarize and analyze existing literature, discover research topics and identify research gaps, brainstorm ideas, source and search for information, edit the paper to improve readability, identify relevant literature, format references, suggest a structure for the paper, and propose a title and keywords. We have reviewed all AI-assisted work. In particular, every mathematical claim and proof developed with the aid of generative AI was verified step by step by the authors. AI-generated code was reviewed and tested for correctness by the authors, and all numerical results reported in this paper were obtained by running the code ourselves. Literature and references suggested by generative AI tools were checked against the original publications, and AI-assisted research ideas were examined for overlap with prior work through a manual literature survey. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

P. Ablin, T. Moreau, M. Massias, and A. Gramfort. Learning step sizes for unfolded sparse coding. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Red Hook, NY, USA, 2019. Curran Associates Inc.

J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE Trans. Med. Imaging*, 37:1322–1332, 2017.

M. Afonso, J. Bioucas-Dias, and M. Figueiredo. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. Image Process.*, 20 (3):681–695, 2011.

F. Alvarez and H. Attouch. An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. *Set-Valued Anal.*, 9:3–11, 2001.

S. Banert, A. Ringh, J. Adler, J. Karlsson, and O. Öktem. Data-driven nonsmooth optimization. *SIAM J. Optim.*, 30(1):102–131, 2020.

S. Banert, J. Rudzusika, O. Öktem, and J. Adler. Accelerated forward-backward optimization using deep learning. *SIAM J. Optim.*, 34(2):1236–1263, 2024.

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Int. Publishing, 2nd edition, 2017.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

S. Becker, J. Fadili, and P. Ochs. On quasi-Newton forward-backward splitting: Proximal calculus and convergence. *SIAM J. Optim.*, 29(4):2445–2481, 2019.

S. Bonettini, I. Loris, F. Porta, and M. Prato. Variable metric inexact line-search-based methods for nonsmooth optimization. *SIAM J. Optim.*, 26(2):891–921, 2016.

J. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. Sagastizabal. A family of variable metric proximal methods. *Math. Program.*, 68:15–47, 1995.

R. I. Boţ, E. R. Csetnek, and C. Hendrich. Inertial Douglas-Rachford splitting for monotone inclusion problems. *Appl. Math. Comput.*, 256:472–487, 2015.

K. Bredies, E. Chenchene, D. A. Lorenz, and E. Naldi. Degenerate preconditioned proximal point algorithms. *SIAM J. Optim.*, 32(3):2376–2401, 2022.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40:120–145, 2011.

A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numer.*, 25:161–319, 2016.

X.-K. Chang, J. Yang, and H. Zhang. Golden ratio primal-dual algorithm with linesearch. *SIAM J. Optim.*, 32(3):1584–1613, 2022.

G. H. Chen and R. T. Rockafellar. Convergence rates in forward-backward splitting. *SIAM J. Optim.*, 7(2):421–444, 1997.

X. Chen, J. Liu, Z. Wang, and W. Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, pp. 9079–9089, 2018.

H. Cherkaoui, J. Sulam, and T. Moreau. Learning to solve TV regularised problems with unrolled algorithms. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, pp. 11513–11524, 2020.

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *J. Optim. Theory Appl.*, 162(1):107–132, 2014.

P. L. Combettes. The geometry of monotone operator splitting methods. *Acta Numer.*, 33:487–632, 2024.

P. L. Combettes and B. C. Vũ. Variable metric quasi-Fejér monotonicity. *Nonlinear Anal. Theory Methods Appl.*, 78, 2012.

P. L. Combettes and B. C. Vũ. Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):1289 – 1318, 2014.

P. L. Combettes, L. Condat, J. C. Pesquet, and B. C. Vũ. A forward-backward view of some primal-dual optimization methods in image recovery. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 4141–4145, 2014.

L. Condat. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2):460–479, 2013.

L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists. *SIAM Rev.*, 65(2):375–435, 2023.

T. Goldstein, M. Li, and X. Yuan. Adaptive primal-dual splitting methods for statistical learning and image processing. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, pp. 2089–2097, 2015.

K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 399–406, 2010.

H. Heaton, X. Chen, Z. Wang, and W. Yin. Safeguarded learned convex optimization. In *Proc. AAAI Conf. Artif. Intell.*, 2023.

S. Hurault, A. Leclaire, and N. Papadakis. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *Proc. Int. Conf. Mach. Learn. (ICML)*, pp. 9483–9505, 2022.

M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 50(11):4484–4502, 2012.

T. Kato. *Perturbation Theory for Linear Operators*. Springer-Verlag, 2nd edition, 1980.

E. King, J. Kotary, F. Fioretto, and J. Drgoňa. Metric learning to accelerate convergence of operator splitting methods. In *Proc. IEEE Conf. Decis. Control.*, pp. 1553–1560, 2024.

N. Komodakis and J. C. Pesquet. Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Process. Mag.*, 32(6):31–54, 2015.

J. Kuczyński and H. Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM J. Matrix Anal. Appl.*, 13(4):1094–1122, 1992. doi: 10.1137/0613066.

A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1964–1971, 2009.

Y. Li, P. Y. Chen, T. Du, and W. Matusik. Learning preconditioners for conjugate gradient PDE solvers. In *Proc. Int. Conf. Mach. Learn. (ICML)*, volume 202, pp. 19425–19439, 2023.

B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pp. 1132–1140, 2017.

P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

Y. Liu, Y. Xu, and W. Yin. Acceleration of primal-dual methods by preconditioning and simple subproblem procedures. *J. Sci. Comput.*, 86(2), 2021.

D. A. Lorenz, J. Marquardt, and E. Naldi. The degenerate variable metric proximal point algorithm and adaptive stepsizes for primal-dual Douglas–Rachford. *Optimization*, 74(6):1355–1381, 2025.

Y. Malitsky and K. Mishchenko. Adaptive gradient descent without descent. In *Proc. Int. Conf. Mach. Learn. (ICML)*, volume 119, pp. 6702–6712, 2020.

Y. Malitsky and K. Mishchenko. Adaptive proximal gradient method for convex optimization. In *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2024.

Y. Malitsky and T. Pock. A first-order primal-dual algorithm with linesearch. *SIAM J. Optim.*, 28 (1):411–432, 2018.

V. Monga, Y. Li, and Y. C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Process. Mag.*, 38(2):18–44, 2021.

K. Naganuma and S. Ono. Variable-wise diagonal preconditioning for primal-dual splitting: Design and applications. *IEEE Trans. Signal Process.*, 71:3281–3295, 2023.

K. Naganuma and S. Ono. Toward robust hyperspectral unmixing: Mixed noise modeling and image-domain regularization. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, 17:8117–8138, 2024.

B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, 15(3):715–732, 2015.

N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127–239, 2014.

J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux. Learning maximally monotone operators for image recovery. *SIAM J. Imaging Sci.*, 14(3):1206–1237, 2021.

T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 1762–1769, 2011.

S. Salzo. The variable metric forward-backward splitting algorithm under mild differentiability assumptions. *SIAM J. Optim.*, 27(4):2153–2181, 2017.

R. Sambharya and B. Stellato. Learning algorithm hyperparameters for fast parametric convex optimization. *SIAM J. Math. Data Sci.*, 8(3):649–676, 2026.

Y. Suzuki, K. Naganuma, and S. Ono. Variable metric stochastic line-search for primal-dual hybrid gradient. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pp. 156–160, 2026.

H. Y. Tan, S. Mukherjee, J. Tang, and C.-B. Schönlieb. Data-driven mirror descent with input-convex neural networks. *SIAM J. Math. Data Sci.*, 5(2):558–587, 2023.

A. Themelis and P. Patrinos. SuperMann: A superlinearly convergent algorithm for finding fixed points of nonexpansive operators. *IEEE Trans. Autom. Control*, 64(12):4875–4890, 2019.

F. G. Veshki and S. A. Vorobyov. Efficient ADMM-based algorithms for convolutional sparse coding. *IEEE Signal Process. Lett.*, 29:389–393, 2022.

B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.*, 38(3):667–681, 2013.

M. Wen, J. Peng, Y. Tang, C. Zhu, and S. Yue. A preconditioning technique for first-order primal-dual splitting method in convex optimization. *Math. Probl. Eng.*, 2017:1–11, 2017.

J. Zhang, B. O'Donoghue, and S. Boyd. Globally convergent Type-I Anderson acceleration for nonsmooth fixed-point iterations. *SIAM J. Optim.*, 30(4):3170–3197, 2020.

A Notation and Additional Preliminaries

Let $M\colon \mathcal{X} \to 2^{\mathcal{X}}$ be a set-valued operator. Its graph is $\operatorname{gra}(M) := \{(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{X} \times \mathcal{X} \mid \boldsymbol{u} \in M\boldsymbol{x}\}$, and M is monotone if $\langle \boldsymbol{u}_1 - \boldsymbol{u}_2 \mid \boldsymbol{x}_1 - \boldsymbol{x}_2\rangle \geq 0$ for every $(\boldsymbol{x}_1, \boldsymbol{u}_1) \in \operatorname{gra}(M)$ and $(\boldsymbol{x}_2, \boldsymbol{u}_2) \in \operatorname{gra}(M)$. If there is no monotone operator B such that $\operatorname{gra}(M) \subsetneq \operatorname{gra}(B)$, then M is called maximally monotone. The set of zeros of M is $\operatorname{zer}(M) := \{\boldsymbol{x} \in \mathcal{X} \mid \boldsymbol{0} \in M\boldsymbol{x}\}$. The subdifferential ∂f of $f \in \Gamma_0(\mathcal{X})$ is maximally monotone, and it is related to the proximity operator by $\operatorname{prox}_f^U = (\mathrm{Id} + U^{-1}\partial f)^{-1}$ for every $U \in \mathcal{P}_\alpha(\mathcal{X})$ (Bauschke & Combettes, 2017). Minimizing f therefore amounts to finding a zero of ∂f.

The product space $\mathcal{X} \times \mathcal{Y}$ is equipped with the inner product $\langle (\boldsymbol{x}_1, \boldsymbol{y}_1) \mid (\boldsymbol{x}_2, \boldsymbol{y}_2)\rangle := \langle \boldsymbol{x}_1 \mid \boldsymbol{x}_2\rangle + \langle \boldsymbol{y}_1 \mid \boldsymbol{y}_2\rangle$. For $L \in \mathcal{B}(\mathcal{X}, \mathcal{Y})$, $\|L\|$ denotes the operator norm and L^* the adjoint operator. The proximity operator of f^* relative to the same metric is obtained from that of f (Bauschke & Combettes, 2017, Proposition 24.24 (ii)), which is how the dual update (6) is implemented.

Inversion reverses the Löwner ordering and gives bounds on inverses, as stated below.

Proposition A.1 (Kato, 1980, Section VI.2.6; see also Combettes & Vũ, 2012, Lemma 2.1)**.** *Let $\mathcal{H}$ be a real Hilbert space, let $\alpha > 0$, let $\mu > 0$, and let U_1 and U_2 be operators in $\mathcal{S}(\mathcal{H})$ such that $\mu\mathrm{Id} \succeq U_1 \succeq U_2 \succeq \alpha\mathrm{Id}$. Then $U_1^{-1}, U_2^{-1} \in \mathcal{S}(\mathcal{H})$, $\alpha^{-1}\mathrm{Id} \succeq U_2^{-1} \succeq U_1^{-1} \succeq \mu^{-1}\mathrm{Id}$, and $\|U_1^{-1}\| \leq \alpha^{-1}$.*

B Properties of the Basis and the Metric Sequence

This section proves Proposition 3.1. We first show the properties of the basis (7) used in the proof, and then derive claims (a)–(c) from them.

Lemma B.1. *Let $m \geq 2$ and let $c > 0$. Then the following hold.*

(a) $(\forall s \in [0,1])\ \psi_1(s), \dots, \psi_m(s) \geq 0$ *and* $\sum_{j=1}^m \psi_j(s) = 1$.

(b) $(\forall j \in \{1, \dots, m\})\ \sum_{n\in\mathbb{N}} |\psi_j(s_{n+1}) - \psi_j(s_n)| \leq 2$.

Proof. (a) The nonnegativity follows from (7). Let $s \in [0,1]$, and let $k \in \{1, \dots, m-1\}$ be such that $s \in [\sigma_k, \sigma_{k+1}]$. Since $\sigma_{k+1} - \sigma_k = 1/(m-1)$, (7) gives $\psi_j(s) = 0$ for every $j \notin \{k, k+1\}$, and

$$\psi_k(s) + \psi_{k+1}(s) = \big(1 - (m-1)(s - \sigma_k)\big) + \big(1 - (m-1)(\sigma_{k+1} - s)\big) = 1.$$

(b) Let $j \in \{1, \dots, m\}$. Since $\{s_n\}$ is increasing in $[0,1)$, and ψ_j is nondecreasing on $[0, \sigma_j]$ and nonincreasing on $[\sigma_j, 1]$, there exists $n_0 \in \mathbb{N} \cup \{+\infty\}$ such that $\{\psi_j(s_n)\}$ is nondecreasing for $n \leq n_0$ and nonincreasing for $n \geq n_0$. Hence, by telescoping, the sum in (b) equals $\big(\sup_{n\in\mathbb{N}} \psi_j(s_n) - \psi_j(s_0)\big) + \big(\sup_{n\in\mathbb{N}} \psi_j(s_n) - \lim_{n\to\infty} \psi_j(s_n)\big)$, which is at most 2. □

Proof of Proposition 3.1. For every $j \in \{1, \dots, m\}$, Proposition A.1 gives

$$\alpha^{-1}\mathrm{Id} \succeq P_{\langle j\rangle}^{-1} \succeq \mu^{-1}\mathrm{Id} \qquad \text{and} \qquad \|P_{\langle j\rangle}^{-1}\| \leq \alpha^{-1}. \tag{12}$$

(a) Let $n \in \mathbb{N}$. By Lemma B.1 (a) and (12), $\alpha^{-1}\mathrm{Id} \succeq \sum_{j=1}^m \psi_j(s_n) P_{\langle j\rangle}^{-1} \succeq \mu^{-1}\mathrm{Id}$. Hence Proposition A.1 shows that P_n in (8) is well defined and $\mu\mathrm{Id} \succeq P_n \succeq \alpha\mathrm{Id}$.

(c) Let $n \in \mathbb{N}$. By (8), $P_n^{-1} = \sum_{j=1}^m \psi_j(s_n) P_{\langle j\rangle}^{-1}$, and the coefficients are nonnegative and sum to one by Lemma B.1 (a).

(b) By (c) and (12), for every $n \in \mathbb{N}$,

$$\|P_{n+1}^{-1} - P_n^{-1}\| = \Big\|\sum_{j=1}^m \big(\psi_j(s_{n+1}) - \psi_j(s_n)\big) P_{\langle j\rangle}^{-1}\Big\| \leq \alpha^{-1} \sum_{j=1}^m |\psi_j(s_{n+1}) - \psi_j(s_n)|.$$

Summing over n and using Lemma B.1 (b), we obtain $\sum_{n\in\mathbb{N}} \eta_n \leq 2m\mu/\alpha < +\infty$, and hence $\{\eta_n\} \in \ell^1_+(\mathbb{N})$. Next, let $n \in \mathbb{N}$. Then

$$\begin{aligned} P_{n+1}^{-1} &\preceq P_n^{-1} + \|P_{n+1}^{-1} - P_n^{-1}\|\mathrm{Id} \\ &\preceq P_n^{-1} + \mu\|P_{n+1}^{-1} - P_n^{-1}\|P_n^{-1} \\ &= (1+\eta_n)P_n^{-1}, \end{aligned}$$

where the first inequality follows from the Cauchy–Schwarz inequality, and the second from $P_n^{-1} \succeq \mu^{-1}\mathrm{Id}$, which holds by (a) and Proposition A.1. Hence, by (a) and Proposition A.1, $(1+\eta_n)P_{n+1} \succeq P_n$. □

C Convergence of Variable Metric PG

This section proves Theorem 2.1. The proof applies the weak convergence result of Combettes & Vũ (2012) for sequences that are quasi-Fejér monotone relative to variable metrics. We show that the sequence generated by (4) is such a sequence with respect to the solution set, where the metric at iteration n is P_n^{-1}, and that its weak sequential cluster points are solutions.

Notation. Throughout this section, $\{\boldsymbol{x}_n\}$ denotes the sequence generated by (4), and we set $F := f + g$ with $f = \frac{1}{2}\|A\cdot - \boldsymbol{b}\|^2$ as in Section 2.1. We write Z for the solution set of (1) with $L = \mathrm{Id}$, that is, the set of minimizers of F, which is nonempty by assumption. The gradient of f is $\nabla f = A^*(A\cdot - \boldsymbol{b})$, and hence

$$(\forall \boldsymbol{x} \in \mathcal{X})(\forall \boldsymbol{y} \in \mathcal{X}) \qquad \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) = A^*A(\boldsymbol{x} - \boldsymbol{y}). \tag{13}$$

For every $n \in \mathbb{N}$, we set $R_n := P_n^{-1}$, which is the metric used in (4) at iteration n.

Known results. We recall the weak convergence of quasi-Fejér monotone sequences relative to variable metrics, which is the basis of the proof. The statement below is the "if" part of Theorem 3.3 of Combettes & Vũ (2012), where we set $\phi = |\cdot|^2$ and $\varepsilon_n \equiv 0$ in Definition 3.1 (i) there so that the definition reduces to (15). The pointwise convergence of $\{W_n\}$ to an operator in $\mathcal{P}_\alpha(\mathcal{X})$, which Theorem 3.3 requires, follows from (14) by Lemma 2.3 (i) there.

Proposition C.1 (Combettes & Vũ, 2012, Lemma 2.3 (i) and Theorem 3.3)**.** *Let $\alpha > 0$, let $\{\eta_n\} \in \ell^1_+(\mathbb{N})$, and let $\{W_n\}$ be a sequence in $\mathcal{P}_\alpha(\mathcal{X})$ such that*

$$\sup_{n\in\mathbb{N}} \|W_n\| < +\infty \qquad \textit{and} \qquad (\forall n \in \mathbb{N}) \quad (1+\eta_n)W_n \succeq W_{n+1}. \tag{14}$$

Let C be a nonempty subset of $\mathcal{X}$, and let $\{\boldsymbol{x}_n\}$ be a sequence in $\mathcal{X}$ such that

$$(\forall \boldsymbol{z} \in C)(\forall n \in \mathbb{N}) \qquad \|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|^2_{W_{n+1}} \leq (1+\eta_n)\|\boldsymbol{x}_n - \boldsymbol{z}\|^2_{W_n}. \tag{15}$$

Then $\{\boldsymbol{x}_n\}$ weakly converges to a point in C if every weak sequential cluster point of $\{\boldsymbol{x}_n\}$ belongs to C.

We also use Proposition 3.2 (i) of Combettes & Vũ (2012), which shows that $\{\|\boldsymbol{x}_n - \boldsymbol{z}\|_{W_n}\}$ converges for every $\boldsymbol{z} \in C$ under the assumptions of Proposition C.1, and the bounds on inverses in Proposition A.1 of Appendix A.

Auxiliary lemmas. For every $n \in \mathbb{N}$, we denote by $\mathcal{X}_n$ the real Hilbert space obtained by endowing $\mathcal{X}$ with $\langle\cdot \mid \cdot\rangle_{R_n}$, which is well defined by Proposition A.1 and assumption (i). In $\mathcal{X}_n$, the gradient of f is $P_n\nabla f$ and the proximity operator of g is $\mathrm{prox}_g^{R_n}$. The first lemma is the main step. It gives (15) with $W_n = R_n$ and $C = Z$, and it is where assumption (iii) of Theorem 2.1 is used.

Lemma C.2. *Let $\gamma > 0$ and let $\zeta > 0$. Assume that assumptions (i)–(iii) of Theorem 2.1 hold. Then, for every $\boldsymbol{z} \in Z$ and every $n \in \mathbb{N}$,*

$$\|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|^2_{R_{n+1}} \leq (1+\eta_n)\|\boldsymbol{x}_n - \boldsymbol{z}\|^2_{R_n} - \zeta\|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|^2. \tag{16}$$

Proof. Let $\boldsymbol{z} \in Z$ and let $n \in \mathbb{N}$. Since f is real-valued, Bauschke & Combettes (2017, Corollary 27.3) applied in $\mathcal{X}_n$ with unit stepsize gives $\boldsymbol{z} = \operatorname{prox}_g^{R_n}(\boldsymbol{z} - P_n \nabla f(\boldsymbol{z}))$. Hence

$$\begin{aligned}
\|\boldsymbol{x}_n - \boldsymbol{z}\|_{R_n}^2 - \|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|_{R_n}^2 &= \|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|_{R_n}^2 + 2\langle \boldsymbol{x}_{n+1} - \boldsymbol{z} \mid \boldsymbol{x}_n - \boldsymbol{x}_{n+1}\rangle_{R_n} \\
&\geq \|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|_{R_n}^2 + 2\langle \boldsymbol{x}_{n+1} - \boldsymbol{z} \mid P_n(\nabla f(\boldsymbol{x}_n) - \nabla f(\boldsymbol{z}))\rangle_{R_n} \\
&= \|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|_{R_n}^2 + 2\langle \boldsymbol{x}_{n+1} - \boldsymbol{z} \mid \nabla f(\boldsymbol{x}_n) - \nabla f(\boldsymbol{z})\rangle \\
&= \|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|_{R_n}^2 + 2\langle A(\boldsymbol{x}_{n+1} - \boldsymbol{z}) \mid A(\boldsymbol{x}_n - \boldsymbol{z})\rangle \\
&\geq \|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|_{R_n}^2 - \tfrac{1}{2}\|A(\boldsymbol{x}_n - \boldsymbol{x}_{n+1})\|^2 \\
&\geq \zeta \|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|^2,
\end{aligned}$$

where the first inequality follows from the firm nonexpansiveness of $\operatorname{prox}_g^{R_n}$ in $\mathcal{X}_n$ (Bauschke & Combettes, 2017, Proposition 12.28), applied to $\boldsymbol{x}_n - P_n \nabla f(\boldsymbol{x}_n)$ and $\boldsymbol{z} - P_n \nabla f(\boldsymbol{z})$, the second and third equalities from $R_n = P_n^{-1}$ and (13), respectively, the second inequality from $\|A(\boldsymbol{x}_{n+1} + \boldsymbol{x}_n - 2\boldsymbol{z})\|^2 \geq 0$, and the third from assumption (iii) of Theorem 2.1. Hence

$$\begin{aligned}
\|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|_{R_{n+1}}^2 &\leq (1 + \eta_n)\|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|_{R_n}^2 \\
&\leq (1 + \eta_n)\left(\|\boldsymbol{x}_n - \boldsymbol{z}\|_{R_n}^2 - \zeta\|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|^2\right) \\
&\leq (1 + \eta_n)\|\boldsymbol{x}_n - \boldsymbol{z}\|_{R_n}^2 - \zeta\|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|^2,
\end{aligned}$$

where the first inequality follows from $(1 + \eta_n)R_n \succeq R_{n+1}$, which holds by Proposition A.1 and assumptions (i) and (ii). □

The second lemma shows that the weak sequential cluster points of $\{\boldsymbol{x}_n\}$ are solutions.

Lemma C.3. *Let $\gamma > 0$ and let $\zeta > 0$. Assume that assumptions (i)–(iii) of Theorem 2.1 hold. Then every weak sequential cluster point of $\{\boldsymbol{x}_n\}$ belongs to Z.*

Proof. We first show that $\boldsymbol{x}_n - \boldsymbol{x}_{n+1} \to \boldsymbol{0}$. Fix $\boldsymbol{z} \in Z$. By Proposition A.1, assumptions (i)–(iii), and Lemma C.2, the assumptions of Proposition C.1 hold with $W_n = R_n$ and $C = Z$. Hence Proposition 3.2 (i) of Combettes & Vũ (2012) shows that $\{\|\boldsymbol{x}_n - \boldsymbol{z}\|_{R_n}\}$ converges. Since $\eta_n \to 0$, Lemma C.2 yields

$$\zeta\|\boldsymbol{x}_n - \boldsymbol{x}_{n+1}\|^2 \leq (1 + \eta_n)\|\boldsymbol{x}_n - \boldsymbol{z}\|_{R_n}^2 - \|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|_{R_{n+1}}^2 \to 0.$$

Now let $\boldsymbol{x}$ be a weak sequential cluster point of $\{\boldsymbol{x}_n\}$, and take a subsequence $\{\boldsymbol{x}_{k_n}\}$ such that $\boldsymbol{x}_{k_n} \rightharpoonup \boldsymbol{x}$. Then $\boldsymbol{x}_{k_n+1} \rightharpoonup \boldsymbol{x}$ since $\boldsymbol{x}_{k_n} - \boldsymbol{x}_{k_n+1} \to \boldsymbol{0}$. For every $n \in \mathbb{N}$ and every $\boldsymbol{y} \in \mathcal{X}$,

$$\begin{aligned}
F(\boldsymbol{y}) &\geq F(\boldsymbol{x}_{n+1}) + \langle \boldsymbol{x}_n - P_n \nabla f(\boldsymbol{x}_n) - \boldsymbol{x}_{n+1} \mid \boldsymbol{y} - \boldsymbol{x}_{n+1}\rangle_{R_n} + \langle \nabla f(\boldsymbol{x}_{n+1}) \mid \boldsymbol{y} - \boldsymbol{x}_{n+1}\rangle \\
&= F(\boldsymbol{x}_{n+1}) + \langle (R_n - A^*A)(\boldsymbol{x}_n - \boldsymbol{x}_{n+1}) \mid \boldsymbol{y} - \boldsymbol{x}_{n+1}\rangle,
\end{aligned}$$

where the inequality follows from Bauschke & Combettes (2017, Proposition 12.26) in $\mathcal{X}_n$, applied to $\boldsymbol{x}_n - P_n \nabla f(\boldsymbol{x}_n)$, and the equality from $R_n = P_n^{-1}$ and (13). Along $\{k_n\}$, the inner product tends to 0, since $(R_{k_n} - A^*A)(\boldsymbol{x}_{k_n} - \boldsymbol{x}_{k_n+1}) \to \boldsymbol{0}$ by Proposition A.1, assumption (i), and $\boldsymbol{x}_n - \boldsymbol{x}_{n+1} \to \boldsymbol{0}$, and $\{\boldsymbol{x}_{k_n+1}\}$ is bounded (Bauschke & Combettes, 2017, Lemma 2.46). Since F is convex and lower semicontinuous, it is weakly lower semicontinuous (Bauschke & Combettes, 2017, Theorem 9.1), and hence $F(\boldsymbol{x}) \leq \liminf_{n\to\infty} F(\boldsymbol{x}_{k_n+1}) \leq F(\boldsymbol{y})$ for every $\boldsymbol{y} \in \mathcal{X}$, that is, $\boldsymbol{x} \in Z$. □

Proof of the theorem. We are now ready to prove Theorem 2.1.

Proof of Theorem 2.1. We apply Proposition C.1 with $W_n = R_n$ and $C = Z$, which is nonempty by assumption. In view of Proposition C.1, it remains to verify the following.

(a) $(\exists \alpha > 0)(\forall n \in \mathbb{N})\ R_n \in \mathcal{P}_\alpha(\mathcal{X})$.

(b) $\sup_{n\in\mathbb{N}} \|R_n\| < +\infty$ and $(\forall n \in \mathbb{N})\ (1 + \eta_n)R_n \succeq R_{n+1}$.

(c) $(\forall \boldsymbol{z} \in Z)(\forall n \in \mathbb{N})\ \|\boldsymbol{x}_{n+1} - \boldsymbol{z}\|_{R_{n+1}}^2 \leq (1 + \eta_n)\|\boldsymbol{x}_n - \boldsymbol{z}\|_{R_n}^2$.

(d) Every weak sequential cluster point of $\{\boldsymbol{x}_n\}$ belongs to Z.

Condition (a) follows from assumption (iii), (b) from Proposition A.1 and assumptions (i) and (ii), (c) from Lemma C.2, and (d) is Lemma C.3. Hence Proposition C.1 shows that $\{\boldsymbol{x}_n\}$ weakly converges to a point in Z, which is a solution of (1). □

Remark C.4. The results of Combettes & Vũ (2014) also cover (4), but not under assumption (iii) of Theorem 2.1. Their Example 4.3 with $U_n = P_n$ requires $\sup_{n\in\mathbb{N}} \|P_n\| < 2/\|A\|^2$, which assumption (iii) does not imply for nonscalar P_n. Their Theorem 4.1 applied to ∂F alone with $U_n = (P_n^{-1} - A^*A)^{-1}$ requires $P_n^{-1} - A^*A \succeq \zeta\mathrm{Id}$, which is stronger than assumption (iii).

D Convergence of Variable Metric PDHG

This section proves Theorem 2.2. The work of Suzuki et al. (2026) assumes that $\|P_n\|^{-1}\|Q_n\|^{-1} - \|L\|^2 \geq \zeta$ for some $\zeta > 0$, and uses this assumption only to establish assumption (iii) of Theorem 2.2 for some $\varsigma > 0$ (Suzuki et al., 2026, Lemma 3.2). Theorem 2.2 assumes the latter directly, so it covers every metric sequence covered by the original statement, and the rest of the argument is unchanged. As in Suzuki et al. (2026), the proof reduces the updates (6) to the proximal point algorithm on variable metrics of Combettes & Vũ (2014).

Notation. Throughout this section, we set $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ and write $\boldsymbol{z} := (\boldsymbol{x}, \boldsymbol{y})$ for its elements, $\boldsymbol{z}_n := (\boldsymbol{x}_n, \boldsymbol{y}_n)$ for the sequence generated by (6), and Z for the solution set of (5). The space $\mathcal{Z}$ is equipped with the product inner product of Appendix A. Each V_n belongs to $\mathcal{S}(\mathcal{Z})$ since P_n^{-1} and Q_n^{-1} are self-adjoint, and assumption (iii) of Theorem 2.2 means that $\langle \boldsymbol{z} \mid V_n\boldsymbol{z}\rangle \geq \varsigma\|\boldsymbol{z}\|^2$ for every $\boldsymbol{z} \in \mathcal{Z}$. We define the set-valued operator $M\colon \mathcal{Z} \to 2^{\mathcal{Z}}$ by

$$M\boldsymbol{z} := \begin{pmatrix} \partial f(\boldsymbol{x}) + L^*\boldsymbol{y} \\ -L\boldsymbol{x} + \partial g^*(\boldsymbol{y}) \end{pmatrix}, \tag{17}$$

so that $Z = \mathrm{zer}(M)$.

Known results. We recall the proximal point algorithm on variable metrics, which is the basis of the proof. The statement below is obtained from Theorem 4.1 (i) of Combettes & Vũ (2014) by setting $B = 0$, $a_n \equiv b_n \equiv \mathbf{0}$, and $\lambda_n \equiv \gamma_n \equiv 1$ there, which is admissible since $B = 0$ is β-cocoercive for every $\beta > 0$.

Proposition D.1 (Combettes & Vũ, 2014, Theorem 4.1 (i))**.** *Let $M\colon \mathcal{X} \to 2^{\mathcal{X}}$ be maximally monotone such that $\mathrm{zer}(M) \neq \emptyset$, let $\alpha > 0$, let $\{\eta_n\} \in \ell^1_+(\mathbb{N})$, and let $\{U_n\}$ be a sequence in $\mathcal{P}_\alpha(\mathcal{X})$ such that*

$$\sup_{n\in\mathbb{N}} \|U_n\| < +\infty \qquad \text{and} \qquad (\forall n \in \mathbb{N}) \quad (1+\eta_n)U_{n+1} \succeq U_n. \tag{18}$$

Let $\{\boldsymbol{x}_n\}$ be a sequence defined by

$$\boldsymbol{x}_{n+1} = (\mathrm{Id} + U_nM)^{-1}(\boldsymbol{x}_n). \tag{19}$$

Then $\{\boldsymbol{x}_n\}$ weakly converges to a point in $\mathrm{zer}(M)$.

We also use the bounds on inverses in Proposition A.1 of Appendix A.

Auxiliary lemmas. The first lemma collects the bounds implied by assumptions (i) and (iii).

Lemma D.2. *Let $\gamma > 0$, let $\delta > 0$, and let $\varsigma > 0$. Assume that assumptions (i) and (iii) of Theorem 2.2 hold, and set $\kappa := \max(\gamma^{-2}, \delta^{-2})$ and $\mu := \sqrt{2(\kappa + \|L\|^2)}$. Then the following hold for every $n \in \mathbb{N}$.*

(a) $P_n^{-1} \preceq \gamma^{-1}\mathrm{Id}$, $Q_n^{-1} \preceq \delta^{-1}\mathrm{Id}$, $\|P_n^{-1}\| \leq \gamma^{-1}$, and $\|Q_n^{-1}\| \leq \delta^{-1}$.

(b) $\|V_n\| \leq \mu$.

(c) V_n is invertible, $V_n^{-1} \in \mathcal{P}_{1/\mu}(\mathcal{Z})$, and $\|V_n^{-1}\| \leq \varsigma^{-1}$.

Proof. (a) By Proposition A.1, assumption (i) gives the claims.

(b) Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{Z}$. Then

$$\begin{aligned}\|V_n\boldsymbol{z}\|^2 &= \|P_n^{-1}\boldsymbol{x} - L^*\boldsymbol{y}\|^2 + \|-L\boldsymbol{x} + Q_n^{-1}\boldsymbol{y}\|^2\\ &\le 2\left(\|P_n^{-1}\boldsymbol{x}\|^2 + \|L^*\boldsymbol{y}\|^2\right) + 2\left(\|L\boldsymbol{x}\|^2 + \|Q_n^{-1}\boldsymbol{y}\|^2\right)\\ &\le 2\left\{\left(\gamma^{-2} + \|L\|^2\right)\|\boldsymbol{x}\|^2 + \left(\delta^{-2} + \|L\|^2\right)\|\boldsymbol{y}\|^2\right\}\\ &\le 2\left(\kappa + \|L\|^2\right)\|\boldsymbol{z}\|^2 = \mu^2\|\boldsymbol{z}\|^2,\end{aligned}$$

where the first inequality uses $(a+b)^2 \le 2a^2 + 2b^2$ for $a, b \in \mathbb{R}$, and the second uses (a) and $\|L^*\| = \|L\|$.

(c) By Proposition A.1, (b) and assumption (iii) give the claims. □

The second lemma builds the summable sequence in the second condition of (18) from those in assumption (ii).

Lemma D.3. *Let $\gamma > 0$, let $\delta > 0$, and let $\varsigma > 0$. Assume that assumptions (i)–(iii) of Theorem 2.2 hold, and set*

$$(\forall n \in \mathbb{N}) \qquad \eta_n := \max\left\{\frac{\eta_n^{(1)}}{\varsigma\gamma}, \frac{\eta_n^{(2)}}{\varsigma\delta}\right\}. \tag{20}$$

Then $\{\eta_n\} \in \ell_+^1(\mathbb{N})$ and

$$(\forall n \in \mathbb{N}) \qquad (1+\eta_n)V_n \succeq V_{n+1} \qquad \textit{and} \qquad (1+\eta_n)V_{n+1}^{-1} \succeq V_n^{-1}. \tag{21}$$

Proof. Since $0 \le \eta_n \le \eta_n^{(1)}/(\varsigma\gamma) + \eta_n^{(2)}/(\varsigma\delta)$ for every $n \in \mathbb{N}$, the sequence $\{\eta_n\}$ is summable. Let $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{Z}$ and $n \in \mathbb{N}$. By the definition of V_n,

$$\langle \boldsymbol{z} \mid V_n\boldsymbol{z}\rangle = \langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle - \langle \boldsymbol{x} \mid L^*\boldsymbol{y}\rangle - \langle \boldsymbol{y} \mid L\boldsymbol{x}\rangle + \langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle, \tag{22}$$

and assumption (iii) gives $\langle \boldsymbol{z} \mid V_n\boldsymbol{z}\rangle \ge \varsigma\|\boldsymbol{x}\|^2 + \varsigma\|\boldsymbol{y}\|^2$. Hence

$$\langle \boldsymbol{x} \mid L^*\boldsymbol{y}\rangle + \langle \boldsymbol{y} \mid L\boldsymbol{x}\rangle \le \langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle + \langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle - \varsigma\|\boldsymbol{x}\|^2 - \varsigma\|\boldsymbol{y}\|^2. \tag{23}$$

Using (22) for V_n and for V_{n+1}, and then (23) with $\eta_n \ge 0$, we obtain

$$\begin{aligned}&(1+\eta_n)\langle \boldsymbol{z} \mid V_n\boldsymbol{z}\rangle - \langle \boldsymbol{z} \mid V_{n+1}\boldsymbol{z}\rangle\\ &\quad= (1+\eta_n)\langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle - \langle \boldsymbol{x} \mid P_{n+1}^{-1}\boldsymbol{x}\rangle + (1+\eta_n)\langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle - \langle \boldsymbol{y} \mid Q_{n+1}^{-1}\boldsymbol{y}\rangle\\ &\qquad- \eta_n\left(\langle \boldsymbol{x} \mid L^*\boldsymbol{y}\rangle + \langle \boldsymbol{y} \mid L\boldsymbol{x}\rangle\right)\\ &\quad\ge \langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle - \langle \boldsymbol{x} \mid P_{n+1}^{-1}\boldsymbol{x}\rangle + \langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle - \langle \boldsymbol{y} \mid Q_{n+1}^{-1}\boldsymbol{y}\rangle + \eta_n\varsigma\left(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2\right)\\ &\quad\ge (1+\eta_n\varsigma\gamma)\langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle - \langle \boldsymbol{x} \mid P_{n+1}^{-1}\boldsymbol{x}\rangle + (1+\eta_n\varsigma\delta)\langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle - \langle \boldsymbol{y} \mid Q_{n+1}^{-1}\boldsymbol{y}\rangle\\ &\quad\ge (1+\eta_n^{(1)})\langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle - \langle \boldsymbol{x} \mid P_{n+1}^{-1}\boldsymbol{x}\rangle + (1+\eta_n^{(2)})\langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle - \langle \boldsymbol{y} \mid Q_{n+1}^{-1}\boldsymbol{y}\rangle\\ &\quad\ge 0,\end{aligned}$$

where the second inequality uses $\|\boldsymbol{x}\|^2 \ge \gamma\langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle$ and $\|\boldsymbol{y}\|^2 \ge \delta\langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle$ from Lemma D.2 (a), the third uses (20) together with $\langle \boldsymbol{x} \mid P_n^{-1}\boldsymbol{x}\rangle \ge 0$ and $\langle \boldsymbol{y} \mid Q_n^{-1}\boldsymbol{y}\rangle \ge 0$, and the last one uses $P_{n+1}^{-1} \preceq (1+\eta_n^{(1)})P_n^{-1}$ and $Q_{n+1}^{-1} \preceq (1+\eta_n^{(2)})Q_n^{-1}$, which follow from Proposition A.1 and assumptions (i) and (ii). This proves the first relation in (21), and the second follows from it by Proposition A.1 and assumption (iii). □

Proof of the theorem. We are now ready to prove Theorem 2.2.

Proof of Theorem 2.2. By Lemma D.2 (c), each V_n is invertible. The operator M is maximally monotone, and the updates (6) are equivalently rewritten as $\boldsymbol{z}_{n+1} = (\mathrm{Id} + V_n^{-1}M)^{-1}(\boldsymbol{z}_n)$ (see Condat, 2013). This is the iteration (19) with this M and $U_n = V_n^{-1}$, and $\mathrm{zer}(M) = Z$ is nonempty by assumption. In view of Proposition D.1, it remains to verify the following.

(a) $(\exists\alpha > 0)(\forall n \in \mathbb{N})\ V_n^{-1} \in \mathcal{P}_\alpha(\mathcal{Z})$.

(b) $\sup_{n\in\mathbb{N}} \|V_n^{-1}\| < +\infty$.

(c) $(\exists\{\eta_n\} \in \ell^1_+(\mathbb{N}))(\forall n \in \mathbb{N})\ (1+\eta_n)V_{n+1}^{-1} \succeq V_n^{-1}$.

Conditions (a) and (b) follow from Lemma D.2 (c) with $\alpha := 1/\mu$ and $\sup_{n\in\mathbb{N}} \|V_n^{-1}\| \le \varsigma^{-1}$, and (c) follows from Lemma D.3 with $\{\eta_n\}$ defined by (20). Hence Proposition D.1 shows that $\{\boldsymbol{z}_n\}$ weakly converges to a point in $\operatorname{zer}(M) = Z$. □

E Spectral Rescaling and Convergence with LVM

This section proves Proposition 3.2 and Theorems 3.3 and 3.4. In each part of Proposition 3.2, we first show the condition for each representative metric, and then extend it to every iteration by Proposition 3.1 (c). The theorems follow from Theorems 2.1 and 2.2 by verifying their assumptions (i)–(iii).

Proof of Proposition 3.2. (a) Let $j \in \{1,\dots,m\}$. By (9), $P_{\langle j\rangle}^{1/2} = (2\rho/\beta_j)^{1/2}\widetilde{P}_{\langle j\rangle}^{1/2}$, and hence

$$\|AP_{\langle j\rangle}^{1/2}\|^2 = \frac{2\rho}{\beta_j}\|A\widetilde{P}_{\langle j\rangle}^{1/2}\|^2 = 2\rho. \tag{24}$$

Let $\boldsymbol{x} \in \mathcal{X}$. Then

$$\begin{aligned}
\langle \boldsymbol{x} \mid (P_{\langle j\rangle}^{-1} - \tfrac{1}{2}A^*A)\boldsymbol{x}\rangle &= \|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 - \tfrac{1}{2}\|AP_{\langle j\rangle}^{1/2}P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 \\
&\ge \|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 - \tfrac{1}{2}\|AP_{\langle j\rangle}^{1/2}\|^2\|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 \\
&= (1-\rho)\|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 \\
&\ge (1-\rho)\|P_{\langle j\rangle}\|^{-1}\|\boldsymbol{x}\|^2,
\end{aligned}$$

where the second equality follows from (24). Hence $P_{\langle j\rangle}^{-1} - \frac{1}{2}A^*A \succeq \zeta\mathrm{Id}$.

Let $n \in \mathbb{N}$. Since $P_{\langle j\rangle} \succeq (2\rho\alpha/\beta_j)\mathrm{Id}$ for every j, Proposition 3.1 (c) applies, and

$$P_n^{-1} - \tfrac{1}{2}A^*A = \sum_{j=1}^m \psi_j(s_n)\big(P_{\langle j\rangle}^{-1} - \tfrac{1}{2}A^*A\big) \succeq \sum_{j=1}^m \psi_j(s_n)\zeta\mathrm{Id} = \zeta\mathrm{Id}.$$

(b) Let $j \in \{1,\dots,m\}$. By (10), $P_{\langle j\rangle}^{1/2} = (\rho/\beta_j)^{1/2}\widetilde{P}_{\langle j\rangle}^{1/2}$ and $Q_{\langle j\rangle}^{1/2} = (\rho/\beta_j)^{1/2}\widetilde{Q}_{\langle j\rangle}^{1/2}$, and hence

$$\|Q_{\langle j\rangle}^{1/2}LP_{\langle j\rangle}^{1/2}\| = \frac{\rho}{\beta_j}\|\widetilde{Q}_{\langle j\rangle}^{1/2}L\widetilde{P}_{\langle j\rangle}^{1/2}\| = \rho. \tag{25}$$

Let $V_{\langle j\rangle}$ be defined as V_n in Theorem 2.2 with $P_{\langle j\rangle}$ and $Q_{\langle j\rangle}$ in place of P_n and Q_n, and let $\boldsymbol{z} = (\boldsymbol{x},\boldsymbol{y}) \in \mathcal{X}\times\mathcal{Y}$. Since $\langle \boldsymbol{x} \mid L^*\boldsymbol{y}\rangle = \langle Q_{\langle j\rangle}^{1/2}LP_{\langle j\rangle}^{1/2}P_{\langle j\rangle}^{-1/2}\boldsymbol{x} \mid Q_{\langle j\rangle}^{-1/2}\boldsymbol{y}\rangle$, we have

$$\begin{aligned}
\langle \boldsymbol{z} \mid V_{\langle j\rangle}\boldsymbol{z}\rangle &= \|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 - 2\langle Q_{\langle j\rangle}^{1/2}LP_{\langle j\rangle}^{1/2}P_{\langle j\rangle}^{-1/2}\boldsymbol{x} \mid Q_{\langle j\rangle}^{-1/2}\boldsymbol{y}\rangle + \|Q_{\langle j\rangle}^{-1/2}\boldsymbol{y}\|^2 \\
&\ge \|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 - 2\rho\|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|\|Q_{\langle j\rangle}^{-1/2}\boldsymbol{y}\| + \|Q_{\langle j\rangle}^{-1/2}\boldsymbol{y}\|^2 \\
&\ge (1-\rho)\big(\|P_{\langle j\rangle}^{-1/2}\boldsymbol{x}\|^2 + \|Q_{\langle j\rangle}^{-1/2}\boldsymbol{y}\|^2\big) \\
&\ge (1-\rho)\big(\|P_{\langle j\rangle}\|^{-1}\|\boldsymbol{x}\|^2 + \|Q_{\langle j\rangle}\|^{-1}\|\boldsymbol{y}\|^2\big) \\
&\ge \varsigma\|\boldsymbol{z}\|^2,
\end{aligned}$$

where the first inequality follows from the Cauchy–Schwarz inequality and (25), and the last from the definition of ς in Proposition 3.2 (b). Hence $V_{\langle j\rangle} \in \mathcal{P}_\varsigma(\mathcal{X}\times\mathcal{Y})$.

Let $n \in \mathbb{N}$. Since $P_{\langle j\rangle} \succeq (\rho\alpha/\beta_j)\mathrm{Id}$ and $Q_{\langle j\rangle} \succeq (\rho\alpha/\beta_j)\mathrm{Id}$ for every j, Proposition 3.1 (c) applies to $\{P_n\}$ and $\{Q_n\}$ with the same coefficients $\psi_j(s_n)$. Since these coefficients sum to one and the off-diagonal blocks of $V_{\langle j\rangle}$ do not depend on j, we obtain $V_n = \sum_{j=1}^m \psi_j(s_n)V_{\langle j\rangle}$, and hence $V_n \in \mathcal{P}_\varsigma(\mathcal{X}\times\mathcal{Y})$ by the nonnegativity of the coefficients. □

Theorems 3.3 and 3.4 state the convergence of PG and PDHG with LVM, and follow immediately by combining Theorems 2.1 and 2.2 with Propositions 3.1 and 3.2.

Proof of Theorem 3.3. Set $\nu := 2\rho\alpha \min_{1\le j\le m} \beta_j^{-1}$. By (9), $P_{\langle j\rangle} \in \mathcal{P}_\nu(\mathcal{X})$ for every $j \in \{1,\dots,m\}$. In view of Theorem 2.1, it remains to verify its assumptions (i)–(iii).

(i) This follows from Proposition 3.1 (a) with ν in place of α.

(ii) This follows from Proposition 3.1 (b).

(iii) This follows from Proposition 3.2 (a).

Hence Theorem 2.1 shows that $\{\boldsymbol{x}_n\}$ weakly converges to a solution of (1). □

Proof of Theorem 3.4. Set $\nu := \rho\alpha \min_{1\le j\le m} \beta_j^{-1}$. By (10), $P_{\langle j\rangle} \in \mathcal{P}_\nu(\mathcal{X})$ and $Q_{\langle j\rangle} \in \mathcal{P}_\nu(\mathcal{Y})$ for every $j \in \{1,\dots,m\}$. In view of Theorem 2.2, it remains to verify its assumptions (i)–(iii).

(i) This follows from Proposition 3.1 (a) applied to $\{P_n\}$ and $\{Q_n\}$ with ν in place of α.

(ii) This follows from Proposition 3.1 (b) applied to $\{P_n\}$ and $\{Q_n\}$.

(iii) This follows from Proposition 3.2 (b).

Hence Theorem 2.2 shows that $\{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}$ weakly converges to a solution of (5). □

F Proximity Operators Relative to Diagonal Metrics

Let $P = \mathrm{Diag}(\boldsymbol{p})$ with $\boldsymbol{p} = (p_i)_{1\le i\le N} \in (0,+\infty)^N$, and let $h(\boldsymbol{x}) = \sum_{i=1}^N h_i(x_i)$ with $h_i \in \Gamma_0(\mathbb{R})$. The objective in (3) with $U = P^{-1}$ is $\sum_{i=1}^N \big(h_i(u_i) + (x_i - u_i)^2/(2p_i)\big)$, and hence it is minimized coordinatewise, that is,

$$(\forall \boldsymbol{x} \in \mathbb{R}^N)(\forall i \in \{1,\dots,N\}) \qquad \big(\mathrm{prox}_h^{P^{-1}}(\boldsymbol{x})\big)_i = \mathrm{prox}_{p_i h_i}(x_i). \tag{26}$$

For $h = \lambda\|\cdot\|_1$ with $\lambda > 0$, (26) is the soft thresholding with the threshold λp_i (Bauschke & Combettes, 2017, Example 24.20),

$$\big(\mathrm{prox}_{\lambda\|\cdot\|_1}^{P^{-1}}(\boldsymbol{x})\big)_i = \mathrm{sign}(x_i)\max\{|x_i| - \lambda p_i, 0\}.$$

For the indicator h of $[0,1]^N$, each $p_i h_i$ is the indicator of $[0,1]$, and (26) is the clipping $\min\{\max\{x_i, 0\}, 1\}$, which does not depend on P.

The dual update in (6) evaluates the proximity operator of g^* relative to Q_n^{-1}, which is obtained from that of g relative to Q_n by Bauschke & Combettes (2017, Proposition 24.24 (ii)) as

$$\mathrm{prox}_{g^*}^{Q_n^{-1}} = Q_n\big(\mathrm{Id} - \mathrm{prox}_g^{Q_n}\big)Q_n^{-1}. \tag{27}$$

Hence the structure of Q_n is chosen according to g.

G Details of the Experimental Setup

G.1 Problems and Datasets

Let $\boldsymbol{x} \in \mathbb{R}^N$ be a grayscale image with pixel values in $[0,1]$, and let $\boldsymbol{b} = \Phi\boldsymbol{x} + \boldsymbol{n}$ be its observation, where $\boldsymbol{n}$ is additive white Gaussian noise with standard deviation $\sigma = 0.01$. For deblurring, Φ is a circular convolution with the 19×19 motion blur kernel of Levin et al. (2009). For inpainting, Φ masks 25% of the pixels chosen uniformly at random, and the noise is added before masking. PG solves

$$\underset{\boldsymbol{c}\in\mathbb{R}^N}{\text{minimize}} \quad \tfrac{1}{2}\|\Phi\Psi^\top\boldsymbol{c} - \boldsymbol{b}\|^2 + \lambda\|\boldsymbol{c}\|_1, \tag{28}$$

where Ψ is the three-level orthonormal Haar transform and $\boldsymbol{x} = \Psi^\top \boldsymbol{c}$, so that (28) is (1) with $L = \mathrm{Id}$ and $A = \Phi\Psi^\top$ in Section 2.1, and $\|A^*A\| \le 1$ holds. We set $\lambda = 0.002$ for deblurring and $\lambda = 0.005$ for inpainting, which are tuned manually. PDHG solves

$$\underset{\boldsymbol{x}\in[0,1]^N}{\text{minimize}} \quad \|D\boldsymbol{x}\|_{2,1} \quad \text{subject to} \quad \|\Phi\boldsymbol{x} - \boldsymbol{b}\| \le \varepsilon, \tag{29}$$

where D is the forward difference operator, $\|\cdot\|_{2,1}$ is the isotropic total variation, and $\varepsilon = 0.95\sigma\sqrt{N}$. This is (1) with $L\boldsymbol{x} = (D\boldsymbol{x}, \Phi\boldsymbol{x})$, and $\|L\| \le 3$ holds.

The generators are trained on grayscale random 128×128 crops of the Flickr2K dataset (Lim et al., 2017), where λ for PG is drawn at random per instance. The test set, referred to as Set10, consists of ten standard grayscale images, seven of size 256×256 and three of size 512×512, each with one noise realization shared by PG and PDHG.

G.2 Metrics and Updates

PG solves (28) with the metric $P_n = \mathrm{Diag}(\boldsymbol{p}_n)$, which has one entry for each wavelet coefficient. By (26), the update (4) reads

$$\boldsymbol{c}_{n+1} = \mathrm{soft}_{\lambda\boldsymbol{p}_n}\big(\boldsymbol{c}_n - \boldsymbol{p}_n \odot A^*(A\boldsymbol{c}_n - \boldsymbol{b})\big),$$

where $\odot$ is the elementwise product and $\mathrm{soft}_{\lambda\boldsymbol{p}_n}$ is the soft thresholding with the threshold $\lambda(\boldsymbol{p}_n)_i$ for the ith coordinate.

PDHG solves (29) as (1) with $f = \iota_{[0,1]^N}$ and $g(\boldsymbol{z}_1, \boldsymbol{z}_2) = \|\boldsymbol{z}_1\|_{2,1} + \iota_{\mathcal{C}}(\boldsymbol{z}_2)$, where $\mathcal{C} := \{\boldsymbol{z} \in \mathbb{R}^N \mid \|\boldsymbol{z} - \boldsymbol{b}\| \le \varepsilon\}$, and its dual variable is $\boldsymbol{y} = (\boldsymbol{u}, \boldsymbol{v}) \in \mathbb{R}^{2N} \times \mathbb{R}^N$. The primal metric $P_n = \mathrm{Diag}(\boldsymbol{p}_n)$ has one entry for each pixel. The dual metric Q_n multiplies the two components of $\boldsymbol{u}$ at the ith pixel by $(\boldsymbol{q}_n)_i$ and multiplies $\boldsymbol{v}$ by a scalar $r_n > 0$. Then the updates (6) read

$$\left\lfloor \begin{aligned} \boldsymbol{x}_{n+1} &= \mathrm{clip}\big(\boldsymbol{x}_n - \boldsymbol{p}_n \odot (D^*\boldsymbol{u}_n + \Phi^*\boldsymbol{v}_n)\big), \\ \boldsymbol{u}_{n+1} &= \mathrm{proj}\big(\boldsymbol{u}_n + \boldsymbol{q}_n \odot D(2\boldsymbol{x}_{n+1} - \boldsymbol{x}_n)\big), \\ \boldsymbol{v}_{n+1} &= \mathrm{shrink}_{r_n\varepsilon}\big(\boldsymbol{v}_n + r_n(\Phi(2\boldsymbol{x}_{n+1} - \boldsymbol{x}_n) - \boldsymbol{b})\big), \end{aligned} \right.$$

where clip clips each pixel to $[0, 1]$, $\boldsymbol{q}_n\odot$ multiplies both components at each pixel by the same entry, proj projects the two components at each pixel onto the unit ℓ_2 ball, and $\mathrm{shrink}_t(\boldsymbol{w}) := (1 - t/\max\{\|\boldsymbol{w}\|, t\})\boldsymbol{w}$. The first line follows from (26), and the others from (27) and the proximity operators of $\|\cdot\|_{2,1}$ and $\iota_{\mathcal{C}}$ (see Parikh & Boyd, 2014), since Q_n is a multiple of the identity on each pixel of $\boldsymbol{u}$ and on $\boldsymbol{v}$.

The quantity β_j in (9) and (10) is approximated by 50 steps for PG and 20 steps for PDHG of the power iteration applied to $\widetilde{P}_{\langle j\rangle}^{1/2} A^* A \widetilde{P}_{\langle j\rangle}^{1/2}$ and to $\widetilde{P}_{\langle j\rangle}^{1/2} L^* \widetilde{Q}_{\langle j\rangle} L \widetilde{P}_{\langle j\rangle}^{1/2}$, whose largest eigenvalues are β_j and β_j^2, respectively. We multiply the estimate of β_j by 1.05 and set $\rho = 0.95$.

G.3 Generator

The generator is a convolutional network with about 10^4 parameters. For PG, it acts on the wavelet coefficients arranged as an image, and its input consists of seven channels, namely $A^*\boldsymbol{b}$, λ as a constant map, the level of the subband of each coefficient, and four binary maps that indicate the type of the subband. For PDHG, its input consists of two channels, $\boldsymbol{x}_0$ and $\log(\varepsilon/\sqrt{N})$ as a constant map. A 3×3 convolution with 32 channels and a ReLU are followed by six residual blocks, each consisting of a 3×3 depthwise convolution with dilation 1, 2, 4, 1, 2, and 4 in this order, a ReLU, and a 1×1 convolution. For PG, a 1×1 convolution outputs the m candidates as m channels. For PDHG, two 1×1 convolutions output the m candidates of $\boldsymbol{p}_n$ and $\boldsymbol{q}_n$ in Appendix G.2, and global average pooling followed by two linear layers outputs those of r_n. The softplus function is applied to each output and 10^{-6} is added to avoid zero entries due to underflow in single precision, although the analysis in Section 3.2 does not require this constant.

G.4 Training

Table 4 lists the settings of training. We use $m = 5$, the Adam optimizer with the learning rate 10^{-3}, and gradient clipping at norm 1, and keep the parameters of the epoch with the smallest

average training loss. The reference solutions of the training instances are computed by the solvers in Appendix G.5, with the maximum number of iterations and the tolerance in Table 4. In (11), 10^{-12} is added to the denominator and inside the logarithm, and the gradient passes through the Rayleigh quotient at the last step of the power iteration. The last layers of the generator are initialized so that the initial metrics equal the scalar stepsize given by the rescaling for PG and are proportional to the stepsizes of Tuned VDP in Section 4.2 for PDHG. Each generator is trained on one NVIDIA GeForce RTX 5090 GPU.

Table 4: Settings of training for each pair of an algorithm and a task.

	PG		PDHG	
	Deblurring	Inpainting	Deblurring	Inpainting
Training instances	1000	10600	1000	1072
Iterations and tolerance for $\boldsymbol{x}^\star$	5000, 10^{-8}	1000, 10^{-6}	10000, 10^{-8}	10000, 10^{-8}
T	40	100	250	250
$\mathcal{T}$	$\{5, 10, 20, 40\}$	$\{10, 25, 50, 100\}$	$\{8, 15, 31, 63, 125, 250\}$	
ω_t	proportional to t		uniform	
c	$40/9$	$50/9$	$250/9$	$250/9$
Batch size	8	32	8	16
Epochs	100	30	100	200
Training time [h]	10.5	4.8	37.8	27.1

G.5 Reference Solutions and Accuracy Measures

A reference solution of each instance is computed by FISTA for (28) and by PDHG with scalar stepsizes for (29), with at most 20000 iterations and the tolerance 10^{-10} on the relative change of the iterate. For each tolerance, we record the first iteration at which each measure falls below it, and report the average over the ten images only when every image reaches the tolerance within the budget of 500 iterations for PG and 1000 iterations for PDHG. All methods start from $\boldsymbol{c}_0 = \boldsymbol{0}$ for PG and from $\boldsymbol{x}_0 = \boldsymbol{b}$ with zero dual variables for PDHG, and run the full budget in single precision on an NVIDIA GeForce RTX 5090 GPU. The elapsed time up to each iteration is measured by synchronizing the GPU after every iteration and is taken as the minimum over three runs that do not compute the measures. The computation before the first iteration, such as the construction of the metrics and the power iteration, is measured separately. Its time is included in Figures 2 and 3. For LVM, this computation consists of one evaluation of the generator and the power iterations, and its average time is 0.17 and 0.066 seconds for PG and 0.065 and 0.022 seconds for PDHG in deblurring and inpainting, respectively.

G.6 Compared Methods

For PG, the compared methods are ISTA with a fixed scalar stepsize selected by a grid search, the Jacobi preconditioner built from an estimate of the diagonal of A^*A, the adaptive proximal gradient method of Malitsky & Mishchenko (2024), and FISTA with the stepsize set from $\|A^*A\| \leq 1$. For PDHG, they are a variable-wise diagonal preconditioner whose primal stepsize and two dual stepsizes (Condat et al., 2023) are selected by a grid search, the diagonal preconditioner of Pock & Chambolle (2011), the residual balancing scheme of Goldstein et al. (2015), the line search of Malitsky & Pock (2018), and the variable-wise diagonal preconditioners of Naganuma & Ono (2023). The grid searches score each candidate by the loss (11) on eight crops of Flickr2K, with $T = 500$ and $\mathcal{T} = \{62, 125, 250, 500\}$ for PG and $T = 250$ and $\mathcal{T} = \{31, 62, 125, 250\}$ for PDHG. The initial stepsizes of the adaptive schemes are set from the power iteration for PG and from $\|L\| \leq 3$ for PDHG, and the remaining parameters follow the original papers. When the learned metrics are combined with FISTA, they are scaled by a constant so that $A^*A \preceq P_n^{-1}$ holds, and the generator is not retrained.

H Details of the Experimental Results

This section examines the early iterations of PG in inpainting with the PSNR in Figure 2 and the iterates in Figure 4. The average PSNR of LVM exceeds 28 dB at about 0.07 s, right after the construction of the metrics, whereas Fixed scalar, Jacobi diag., and AdaPGM do not reach 28 dB within 500 iterations. In Figure 4, the iterate of LVM at $n = 2$ contains no visible missing pixels and resembles a coarse approximation of the image, while the missing pixels remain visible at $n = 50$ for Fixed scalar, Jacobi diag., and AdaPGM. This suggests that the learned metrics promote the recovery of coarse image structures in the early iterations.

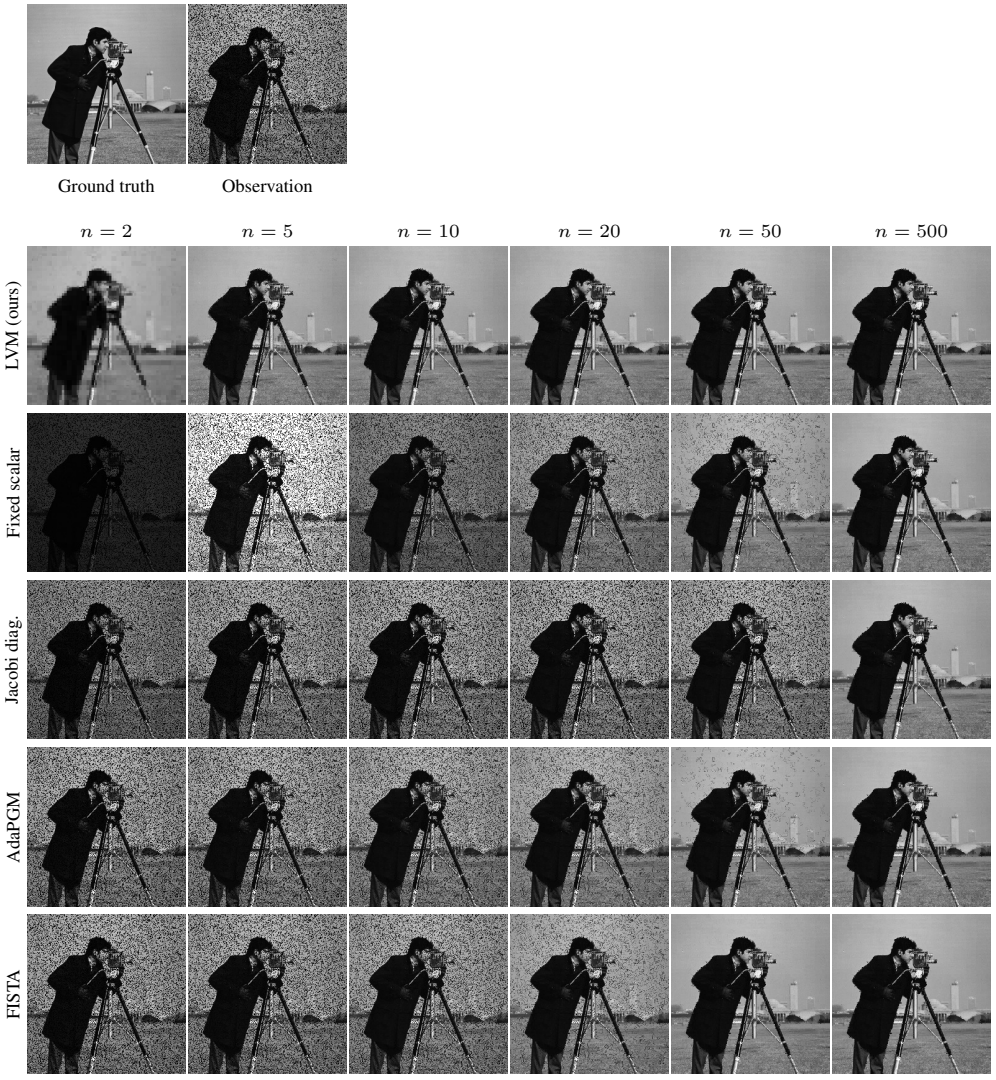


Figure 4: Iterates of PG on inpainting at $n = 2, 5, 10, 20, 50$ and the output $n = 500$.

I Spectral Rescaling in the Implementation

Proposition 3.2 assumes the exact value of β_j, whereas our implementation replaces it by $1.05\hat{\beta}_j$, where $\hat{\beta}_j$ is obtained from the Rayleigh quotient after K steps of the power iteration, with $K = 50$ for PG and $K = 20$ for PDHG (Appendix G.2). Since the Rayleigh quotient never exceeds the largest eigenvalue, $\hat{\beta}_j \leq \beta_j$ holds, and the representative metrics actually used satisfy $\xi_j = (\rho/1.05)(\beta_j/\hat{\beta}_j)$, where $\xi_j := \frac{1}{2}\|AP_{\langle j\rangle}^{1/2}\|^2$ for PG and $\xi_j := \|Q_{\langle j\rangle}^{1/2} L P_{\langle j\rangle}^{1/2}\|$ for PDHG. The

Table 5: Largest value of ξ_j on Set10 for each representative metric j. Theorems 3.3 and 3.4 apply if $\xi_j < 1$ for every j.

	PG		PDHG	
	Deblurring	Inpainting	Deblurring	Inpainting
$j = 1$	0.937	0.943	0.942	0.958
$j = 2$	0.931	0.909	0.936	0.951
$j = 3$	0.934	0.952	0.961	0.981
$j = 4$	0.932	0.943	0.972	0.974
$j = 5$	0.931	0.961	0.949	0.941

Table 6: Ablation of the learned metric for PG on Set10. Mean number of iterations until $\mathcal{G}_n$ first falls below each tolerance. The best and second-best values are shown in bold and underlined, respectively. The notation "> 500" indicates that the criterion was not satisfied within 500 iterations for at least one image.

Method	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-5}$	$\leq 10^{-6}$
Deblurring				
LVM (ours)	**4**	**8**	**23**	<u>54</u>
LVM (ours, matched training)	<u>5</u>	<u>9</u>	<u>24</u>	**53**
− temporal variation	7	13	30	60
Inpainting				
LVM (ours)	**3**	**7**	**34**	**95**
− temporal variation	<u>13</u>	<u>66</u>	> 500	> 500

proof of Proposition 3.2 remains valid with ρ replaced by $\max_j \xi_j$, and hence Theorems 3.3 and 3.4 apply to our implementation whenever $\max_j \xi_j < 1$, which extends to every iteration by Proposition 3.1 (c). For every problem of Set10 used in Sections 4.1 and 4.2, we evaluated ξ_j numerically for the representative metrics actually generated, as similarly done in Pesquet et al. (2021) and Hurault et al. (2022). In our evaluation, the value of β_j is computed by the Lanczos method (Kuczyński & Woźniakowski, 1992) in double precision up to a relative residual of 10^{-10}.

In Table 5, the computed values of ξ_j are smaller than 1 for every setting and every representative metric, and the largest one is 0.981. These values suggest that the metrics used in the experiments of the main text satisfy the assumptions of Theorems 3.3 and 3.4.

J Ablation Study

LVM adapts its metrics to each problem instance, coordinate, and iteration. This section examines their contributions to the acceleration by removing the variation of the metrics over the iterations or over the coordinates and the instances.

For PG, the variant without the variation over the iterations, labeled − temporal variation, shares one candidate among all the representative times, and its metric still varies over the coordinates and the instances. It follows King et al. (2024) in learning a metric fixed over the iterations. For deblurring, it is trained with $T = 200$ and $\mathcal{T} = \{5, 10, 20, 40, 100, 200\}$, as is LVM (ours, matched training) in Table 6. We do not remove the variation over the coordinates for PG, since the rescaling (9) maps every scalar candidate to the same fixed stepsize (Section 3.2), whose tuned version is Fixed scalar in Section 4.1. For PDHG, − temporal variation is defined in the same way, and the variant without the variation over the coordinates, labeled − spatial variation, learns one constant for each of $\boldsymbol{p}_n$, $\boldsymbol{q}_n$, and r_n in Appendix G.2 at each representative time, which is shared by all the pixels and all the instances. The two variants therefore have the properties of learned fixed metrics and learned stepsizes in Table 1, respectively. The other settings are the same as those of LVM (Appendix G).

At the tightest tolerance, removing either variation increases the number of iterations in every setting of Tables 6 and 7, where LVM trained with the same settings is used for PG deblurring. The variation over the iterations matters most in PG inpainting, where the variant without it does not reach $\mathcal{G}_n \leq 10^{-5}$ within 500 iterations, whereas LVM reaches $\mathcal{G}_n \leq 10^{-6}$ in 95 iterations. In PDHG inpainting,

Table 7: Ablation of the learned metric for PDHG on Set10. Mean number of iterations until $\mathcal{D}_n$ first falls below each tolerance. The best and second-best values are shown in bold and underlined, respectively. The notation "> 1000" indicates that the criterion was not satisfied within 1000 iterations for at least one image.

Method	$\leq 10^{-2}$	$\leq 10^{-3}$	$\leq 10^{-4}$	$\leq 10^{-5}$
Deblurring				
LVM (ours)	**24**	**40**	**62**	**89**
− temporal variation	29	50	<u>73</u>	<u>101</u>
− spatial variation	<u>28</u>	<u>48</u>	75	113
Inpainting				
LVM (ours)	<u>15</u>	<u>26</u>	<u>39</u>	**56**
− temporal variation	**10**	**22**	**35**	<u>65</u>
− spatial variation	16	28	48	> 1000

this variant reaches the three loosest tolerances in fewer iterations than LVM but falls behind at the tightest one, which suggests that a single metric cannot suit both the early and the late iterations. Without the variation over the coordinates, PDHG does not reach $\mathcal{D}_n \leq 10^{-5}$ within 1000 iterations in inpainting and needs 113 instead of 89 iterations to reach $\mathcal{D}_n \leq 10^{-5}$ in deblurring. Since the constants of this variant are shared by all the instances, this gap reflects the adaptation both to the coordinates and to the instance. For PG, removing the variation over the coordinates leaves only a fixed scalar stepsize, and Fixed scalar in Table 2 needs more iterations than LVM in every setting. In PDHG deblurring, both variants reach every tolerance in fewer iterations than Tuned VDP in Table 3, and LVM, which combines their degrees of freedom, needs fewer iterations than both. These comparisons indicate that both the variation over the iterations and the adaptation to the coordinates and the instances contribute to the acceleration.