OTPrune: Distribution-Aligned Visual Token Pruning via Optimal Transport (Supplementary Material)

Xiwen Chen[1,2*] Wenhui Zhu[3*†] Gen Li[2] Xuanzhao Dong[3] Yujian Xiong[3] Hao Wang[2]
Peijie Qiu[4] Qingquan Song[5] Zhipeng Wang[6†‡] Shao Tang[7] Yalin Wang[3] Abolfazl Razi[2§]
[1] Morgan Stanley, [2] Clemson University, [3] Arizona State University,
[4] Washington University in St. Louis, [5] Texas A&M University,
[6] Rice University [7] Florida State University

A. Proofs

A.1. Proof of Inequality

Recap, for a positive semidefinite square matrix $\boldsymbol{X}$, the operator $\Psi(\cdot)$ is presented as:

$$\Psi(\boldsymbol{X}) = \log\det(\boldsymbol{I} + \gamma\boldsymbol{X}) = \sum_{i=1}^{d} \log(1+\gamma\lambda_i), \qquad (1)$$

where $\{\lambda_i\}$ are the eigenvalues of $\boldsymbol{X}$ and $\gamma > 0$ is a fixed scaling parameter. Since $\boldsymbol{X}$ is positive semidefinite (so $\lambda_i \geq 0$), we have the following inequalities,

$$\Psi(\boldsymbol{X}) \leq \gamma \ \mathrm{tr}(\boldsymbol{X}) \leq \gamma(\mathrm{tr}(\boldsymbol{X}^{1/2}))^2. \qquad (2)$$

Now, we will show its proof as follows.

Proof. Let $\boldsymbol{X} \succeq \boldsymbol{0}$ with non-negative eigenvalues $\{\lambda_i\}$. For the first inequality in (2), use the elementary bound $\log(1+t) \leq t$ for all $t \geq 0$. Since $\gamma\lambda_i \geq 0$,

$$\log(1+\gamma\lambda_i) \leq \gamma\lambda_i \quad \Rightarrow \quad \Psi(\boldsymbol{X}) \leq \gamma \sum_{i=1}^{d} \lambda_i = \gamma \ \mathrm{tr}(\boldsymbol{X}).$$

For the second inequality in (2), note that

$$\begin{aligned}\big(\mathrm{tr}(\boldsymbol{X}^{1/2})\big)^2 &= \Big(\sum_{i=1}^{d} \sqrt{\lambda_i}\Big)^2 \\ &= \sum_{i=1}^{d} \lambda_i + 2 \sum_{1\leq i<j\leq d} \sqrt{\lambda_i\lambda_j} \\ &\geq \sum_{i=1}^{d} \lambda_i = \mathrm{tr}(\boldsymbol{X}),\end{aligned}$$

because each term $\sqrt{\lambda_i\lambda_j}$ is nonnegative. Multiplying both sides by $\gamma > 0$ gives

$$\gamma \ \mathrm{tr}(\boldsymbol{X}) \ \leq \ \gamma\big(\mathrm{tr}(\boldsymbol{X}^{1/2})\big)^2.$$

Combining the two inequalities yields

$$\Psi(\boldsymbol{X}) \ \leq \ \gamma \ \mathrm{tr}(\boldsymbol{X}) \ \leq \ \gamma\big(\mathrm{tr}(\boldsymbol{X}^{1/2})\big)^2,$$

as claimed. □

A.2. Proof of Submodularity

Definition 1 (Submodularity)**.** A set function $h : 2^{[m]} \to \mathbb{R}$ is *submodular* if it satisfies the diminishing-returns property: for any $\mathcal{S} \subseteq \mathcal{T} \subseteq [m]$ and any $i \notin \mathcal{T}$,

$$h(\mathcal{S}\cup\{i\}) - h(\mathcal{S}) \ \geq \ h(\mathcal{T}\cup\{i\}) - h(\mathcal{T}).$$

Lemma 1. *([1–3]) Let $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ be a positive definite matrix, and for any subset $S \subseteq \{1,\ldots,m\}$ let $\boldsymbol{A}_S$ denote the corresponding principal submatrix. Define the set function*

$$h(S) = \log\det(\boldsymbol{A}_S).$$

Then h is a submodular set function.

Proposition 2. *Let $\boldsymbol{V} \in \mathbb{R}^{m\times d}$ and let $\boldsymbol{V}_{\mathcal{C}}$ denote the submatrix of $\boldsymbol{V}$ consisting of rows indexed by $\mathcal{C} \subseteq [m]$. Define*

$$\Gamma(\mathcal{C}) = \log\det\Big(\boldsymbol{I} + \tilde{\gamma}\,\boldsymbol{V}_{\mathcal{C}}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{V}_{\mathcal{C}}^\top\Big), \qquad \tilde{\gamma} > 0.$$

Then Γ is normalized and submodular.

Proof. To prove this proposition, We need to first prove this: Let $\boldsymbol{V} \in \mathbb{R}^{m\times d}$ be any matrix and $\mathcal{C} \subseteq [m]$ be an index set. Let $\boldsymbol{V}_{\mathcal{C}}$ denote the submatrix of $\boldsymbol{V}$ formed by the rows indexed by $\mathcal{C}$, and let $\tilde{\gamma} \geq 0$. Then the matrix

$$\boldsymbol{M} = \boldsymbol{I}_{|\mathcal{C}|} + \tilde{\gamma}\,\boldsymbol{V}_{\mathcal{C}}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{V}_{\mathcal{C}}^\top \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}$$

*These authors contributed equally to this paper.
†Now at LinkedIn.
‡Corresponding author: zhipeng.wang@alumni.rice.edu
§Corresponding author: arazi@clemson.edu

is symmetric positive definite.

Since $\boldsymbol{V}^\top \boldsymbol{V} \in \mathbb{R}^{d\times d}$ is symmetric positive semidefinite, define

$$\boldsymbol{A} = \boldsymbol{V}_{\mathcal{C}} \boldsymbol{V}^\top \boldsymbol{V} \boldsymbol{V}_{\mathcal{C}}^\top \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{C}|}.$$

We first show that $\boldsymbol{A}$ is positive semidefinite. For any $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{C}|}$,

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{V}_{\mathcal{C}} \boldsymbol{V}^\top \boldsymbol{V} \boldsymbol{V}_{\mathcal{C}}^\top \boldsymbol{x} = \big(\boldsymbol{V}_{\mathcal{C}}^\top \boldsymbol{x}\big)^\top \boldsymbol{V}^\top \boldsymbol{V} \big(\boldsymbol{V}_{\mathcal{C}}^\top \boldsymbol{x}\big).$$

Let $\boldsymbol{y} = \boldsymbol{V}_{\mathcal{C}}^\top \boldsymbol{x} \in \mathbb{R}^d$. Then

$$\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}^\top \boldsymbol{V}^\top \boldsymbol{V}\boldsymbol{y} = \|\boldsymbol{V}\boldsymbol{y}\|_2^2 \geq 0,$$

so $\boldsymbol{A}$ is positive semidefinite.

Now consider

$$\boldsymbol{M} = \boldsymbol{I}_{|\mathcal{C}|} + \tilde{\gamma}\, \boldsymbol{A}.$$

The matrix $\boldsymbol{M}$ is symmetric because both $\boldsymbol{I}_{|\mathcal{C}|}$ and $\boldsymbol{A}$ are symmetric. For any nonzero $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{C}|}$ we have

$$\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x} = \boldsymbol{x}^\top \boldsymbol{I}_{|\mathcal{C}|}\boldsymbol{x} + \tilde{\gamma}\, \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = \|\boldsymbol{x}\|_2^2 + \tilde{\gamma}\, \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}.$$

Since $\boldsymbol{A}$ is positive semidefinite and $\tilde{\gamma} \geq 0$, we have $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \geq 0$, so

$$\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x} \geq \|\boldsymbol{x}\|_2^2 > 0$$

for all $\boldsymbol{x} \neq \boldsymbol{0}$. Therefore $\boldsymbol{M}$ is positive definite.

Then we can apply Lemma 1, and we can conclude Γ is sumdoular.

□

B. Detail of Algorithm

In this appendix we describe how our pruning objective naturally leads to the OTPrune algorithm. The derivation begins directly with the token–token interaction matrix and proceeds to the incremental update rules used in Algorithm 1, without introducing additional notation beyond what is necessary.

Given the vision token matrix $\boldsymbol{V} \in \mathbb{R}^{m\times d}$, we first construct a token–token interaction matrix

$$\tilde{\boldsymbol{V}} = \boldsymbol{V}\boldsymbol{V}^\top \in \mathbb{R}^{m\times m}.$$

The i-th row of this matrix,

$$\boldsymbol{w}_i^\top = \tilde{\boldsymbol{V}}[i,:],$$

captures how token i interacts with all other tokens. For a subset $\mathcal{C} \subseteq [m]$, we use

$$\tilde{\boldsymbol{V}}_{\mathcal{C}} = \tilde{\boldsymbol{V}}[\mathcal{C},:]$$

to denote the corresponding row submatrix.

Our pruning objective is the log-determinant

$$\Gamma(\mathcal{C}) = \log\det\Big(\boldsymbol{I} + \tilde{\gamma}\, \tilde{\boldsymbol{V}}_{\mathcal{C}} \tilde{\boldsymbol{V}}_{\mathcal{C}}^\top\Big), \qquad \tilde{\gamma} > 0.$$

This quantity becomes large when the selected interaction vectors $\{\boldsymbol{w}_i : i \in \mathcal{C}\}$ span a diverse subspace. It is convenient to define the kernel

$$K = \boldsymbol{I} + \tilde{\gamma}\, \tilde{\boldsymbol{V}}\, \tilde{\boldsymbol{V}}^\top, \quad \text{so that} \quad K_{ij} = \delta_{ij} + \tilde{\gamma}\, \langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle.$$

Here, the Kronecker delta δ_{ij} equals 1 if $i = j$ and 0 otherwise. Then $\Gamma(\mathcal{C}) = \log\det(K_{\mathcal{C}})$, where $K_{\mathcal{C}}$ is the principal submatrix indexed by $\mathcal{C}$.

Greedy marginal gain. When considering a new token $j \notin \mathcal{C}$, the determinant of the enlarged matrix can be written using the classical one-step determinant identity:

$$\det(K_{\mathcal{C}\cup\{j\}}) = \det(K_{\mathcal{C}}) \cdot \Big(K_{jj} - K_{\mathcal{C}j}^\top K_{\mathcal{C}}^{-1} K_{\mathcal{C}j}\Big).$$

Thus the improvement in the objective is

$$\Gamma(\mathcal{C}\cup\{j\}) - \Gamma(\mathcal{C}) = \log d_j^2, \qquad d_j^2 = K_{jj} - K_{\mathcal{C}j}^\top K_{\mathcal{C}}^{-1} K_{\mathcal{C}j}.$$

The greedy algorithm simply selects the token with the largest d_j^2.

Cholesky representation. Let the Cholesky factor of the current kernel submatrix be

$$K_{\mathcal{C}} = LL^\top.$$

Define the coefficient vector

$$\boldsymbol{c}_j = L^{-1} K_{\mathcal{C}j}.$$

Because $K_{\mathcal{C}}^{-1} = (L^\top)^{-1} L^{-1}$, the quadratic form simplifies to

$$K_{\mathcal{C}j}^\top K_{\mathcal{C}}^{-1} K_{\mathcal{C}j} = \|\boldsymbol{c}_j\|^2,$$

so the greedy score becomes

$$d_j^2 = K_{jj} - \|\boldsymbol{c}_j\|^2.$$

Thus maintaining the greedy scores reduces to tracking $\boldsymbol{c}_i$ and d_i^2 for all unselected tokens.

Adding a token. Suppose token j is selected. The Cholesky factor of the enlarged set $\mathcal{C} \cup \{j\}$ takes the form

$$L' = \begin{pmatrix} L & 0 \\ \boldsymbol{c}_j^\top & d_j \end{pmatrix}.$$

For any remaining token $i \notin \mathcal{C} \cup \{j\}$, let the updated coefficient vector be

$$\boldsymbol{c}_i' = \begin{bmatrix} \boldsymbol{c}_i \\ e_i \end{bmatrix}.$$

Plugging this into $L'\boldsymbol{c}'_i = K_{\mathcal{C}\cup\{j\},\,i}$ yields

$$e_i = \frac{K_{ji} - \langle \boldsymbol{c}_j, \boldsymbol{c}_i \rangle}{d_j}.$$

Once this new coordinate is known, the updated greedy score becomes

$$d_i'^2 = K_{ii} - \|\boldsymbol{c}'_i\|^2 = d_i^2 - e_i^2.$$

In general, the update takes the form

$$e_i = \frac{K_{ji} - \langle \boldsymbol{c}_j, \boldsymbol{c}_i \rangle}{d_j}.$$

For our kernel $K_{ji} = \delta_{ji} + \tilde{\gamma}\langle \boldsymbol{w}_j, \boldsymbol{w}_i \rangle$. Since we only update $i \neq j$ (token j has just been selected), the Kronecker delta term vanishes and we obtain

$$e_i = \frac{\tilde{\gamma}\langle \boldsymbol{w}_j, \boldsymbol{w}_i \rangle - \langle \boldsymbol{c}_j, \boldsymbol{c}_i \rangle}{d_j}.$$

This expression exactly the operations executed in Algorithm 1.

Initialization and complexity. With $\mathcal{C} = \emptyset$, all coefficient vectors are empty and the initial scores reduce to

$$d_i^2 = K_{ii} = 1 + \tilde{\gamma}\,\|\boldsymbol{w}_i\|^2.$$

At iteration t, each of the $(m-t)$ remaining items requires an $O(t)$ inner product $\langle \boldsymbol{c}_j, \boldsymbol{c}_i \rangle$, so the overall cost of the iteration is

$$O\big((m-t)t\big).$$

Summing over $t = 1, \ldots, k$ gives

$$\sum_{t=1}^{k} O\big((m-t)t\big) = O(mk^2),$$

which is the total complexity of selecting k tokens.

This completes the derivation of OTPrune directly from the log-determinant objective.

C. Discussion on Non-uniform Weighting Strategies

In our current framework, we adopt a *uniform weight assumption* for the source distribution P, where each token $v_i \in \boldsymbol{V}$ is assigned an equal mass of $1/M$. While this simplification allows for a highly efficient and training-free selection process, it treats all visual regions as equally significant, which may not fully capture the semantic variance inherent in complex images.

Potential for Adaptive Importance. Theoretically, the optimal transport formulation in *OTPrune* can be generalized to accommodate *non-uniform weights* by assigning an importance score w_i to each token, such that $\sum w_i = 1$. As suggested during the rebuttal, such weights could be derived from internal model signals to better reflect semantic density:

- *Attention-based Priors*: Leveraging the cumulative attention weights from preceding transformer layers to identify patches that the model naturally prioritizes.
- *Task-specific Significance*: Utilizing cross-modal alignment scores to assign higher weights to visual tokens that are most relevant to the input text prompt.

Future Challenges. Transitioning to a non-uniform weighting strategy transforms *OTPrune* into a more complex joint optimization problem involving both subset selection and mass allocation. We consider the exploration of *soft-weighted OTPrune*, where token weights are dynamically adjusted during the pruning process—as a promising direction for further improving performance at extremely low pruning ratios.

References

[1] Francis Bach. Learning with submodular functions: A convex optimization perspective. *arXiv preprint arXiv:1111.6453*, 2011. 1

[2] Andreas Krause and Carlos Guestrin. Near-optimal observation selection using submodular functions. In *AAAI*, pages 1650–1654, 2007.

[3] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286, 2012. 1