EXACT NATIVE VERTICES FOR CONTINUOUS ADAPTIVE TOKENIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Choosing where to use fine patches is central to adaptive image tokenization, but each choice changes the length and structure of the token sequence. We introduce Differentiable Virtual-Slot Partitioning (DVSP), which places coarse and fine representations in a shared, fixed-length computation graph. Each image region is represented by four virtual slots whose content, position, and attention mass vary together with a learned resolution coordinate. At the binary endpoints, these slots reproduce either one coarse token or four fine tokens. We prove that every resulting binary partition exactly matches its native variable-length execution under compatible deterministic transformer operations. This allows resolution to be learned through task gradients, with projection evaluated separately from exact execution. On ImageNet, DVSP reduces visual-token counts by 30.4% on ViT-Small with a Top-1 accuracy decrease of 0.53 percentage points. Controlled spatial comparisons show that learned partitions outperform entropy selection at identical per-image token counts and fixed task weights. DVSP makes adaptive input resolution learnable within a continuous representation whose discrete states retain their native meaning.

1 INTRODUCTION

Adaptive coarse-to-fine tokenization assigns fine visual tokens to selected image regions and coarse tokens to the rest. It can reduce the sequence length processed by a Vision Transformer while retaining useful task accuracy, yet deciding where to refine remains difficult to optimize. APT selects patch sizes through entropy thresholds, leaving allocation detached from the downstream task objective (Choudhury et al., 2026). MSViT learns a binary scale gate from task loss, using a straight-through estimator to backpropagate through the hard decision (Havtorn et al., 2023). Figure 1 contrasts these allocation mechanisms with the continuous representation developed here.

This raises a concrete question: *can coarse/fine tokenization be learned through a continuous representation whose every binary state exactly reproduces the corresponding variable-length model?* We call direct processing of the selected coarse and fine tokens *native execution*.

We introduce **Differentiable Virtual-Slot Partitioning (DVSP)**. Each parent region is represented during training by four virtual slots. A real-valued coordinate α_i jointly controls their content, position, and attention mass. An image-only router predicts these unconstrained coordinates and learns through task gradients, with an L_1 penalty encouraging coarse representations. At inference, the signed half-threshold maps each coordinate to its nearest binary value, and the resulting partition determines the variable-length token sequence.

The construction preserves the meaning of both resolution choices. At the coarse endpoint, four identical slots share the attention mass of one token; at the fine endpoint, they become the four ordinary fine tokens. With aligned native embeddings and compatible deterministic transformer operations, this correspondence holds for every mixture of coarse and fine regions. Projection selects a binary partition and can change task predictions; executing that selected partition natively preserves its output. We establish this identity and verify it against APT's independent native tokenizer.

Our ImageNet evaluation follows this representation from continuous learning to discrete execution: how resolution coordinates form, how projection changes predictions, and how learned partitions trade

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.

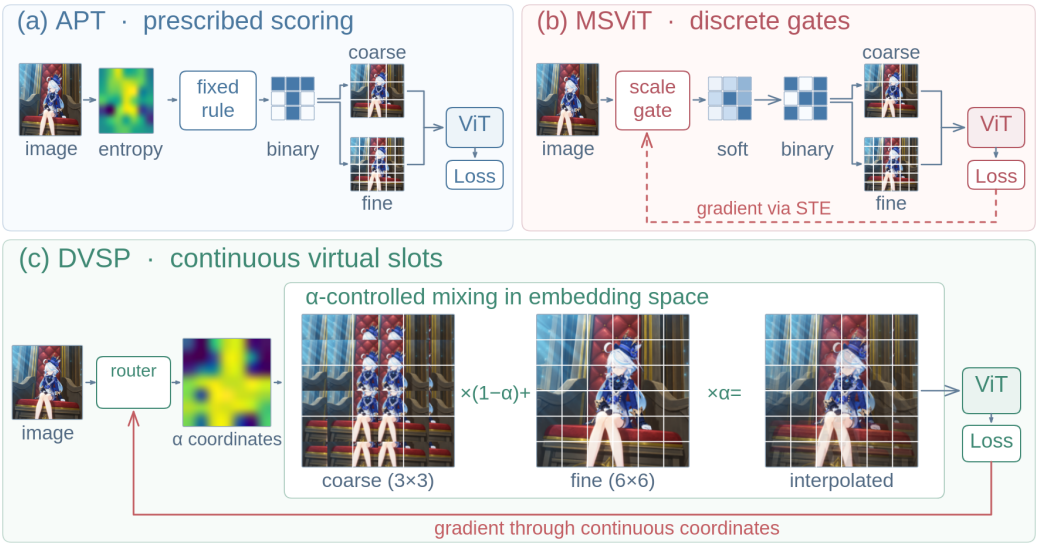


Figure 1: Three routes to coarse/fine image tokenization. (a) APT measures image entropy and applies a fixed rule to produce a binary coarse/fine decision map. (b) MSViT converts soft scale gates into binary decisions with a straight-through estimator. (c) DVSP's image-only router predicts real-valued α coordinates that control continuous virtual slots. The grids and image-space blend are schematic; Figure 2 shows projection and native execution at the four-slot binary endpoints.

accuracy for token reduction. Controlled spatial and gradient probes examine the task information carried by those coordinates.

In summary, our contributions are:

- **Learning variable token structure in a fixed computation graph.** DVSP represents coarse/fine image partitions through continuous virtual slots. Coupling content, position, and attention mass with one resolution coordinate lets task gradients train an image-conditioned router while the training sequence retains a fixed length.
- **Exact execution at every binary partition.** We establish output equivalence between the virtual-slot model and native variable-length execution for arbitrary mixtures of coarse and fine regions under compatible transformer operations. Every binary vertex is therefore an exact execution target, separating the empirical effect of projection from the fidelity of executing the selected partition.
- **Evidence that continuous learning informs discrete spatial decisions.** Matched-count controls establish useful learned region selection. Within the tested ViT-Tiny models, gradients at learned continuous coordinates also rank and select native region changes better than identity-STE gradients at the projected vertices, connecting the learned representation to discrete task utility.

2 Related Work

Adaptive input tokenization. APT assigns patch sizes from image entropy; Quadformer scores regions from pixels or features (Choudhury et al., 2026; Ronen et al., 2023). MSViT learns mixed-scale gates through a straight-through construction, while DART learns differentiable, content-dependent region scores (Havtorn et al., 2023; Yin et al., 2025). DVSP provides a fixed-cardinality representation of the coarse/fine partition family, with task-driven continuous coordinates and exact native computation at its binary states.

Reducing tokens inside a transformer. DynamicViT prunes tokens, EViT reorganizes them, ToMe merges hidden states, and DTEM learns embeddings for token merging (Rao et al., 2021; Liang

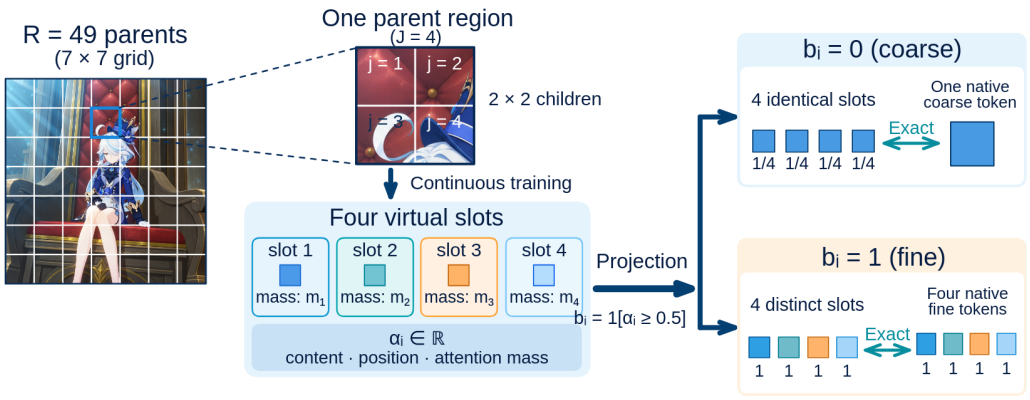


Figure 2: DVSP's four-slot construction at 224×224 resolution. Training uses 196 visual slots; signed half-threshold projection determines the inference partition. *Exact* denotes the binary-to-native output identity in Eq. 5. Colors distinguish the four children.

et al., 2022; Bolya et al., 2023; Lee & Hong, 2024). DVSP decides the input partition before the task transformer. It uses ToMe's proportional-attention form with fractional virtual-slot masses. Coupling these masses to content and position makes a coarse group equivalent to one native token.

Continuous relaxations of discrete choices. Concrete and Gumbel-Softmax relaxations and surrogate-gradient estimators provide general ways to optimize discrete choices (Maddison et al., 2017; Jang et al., 2017; Bengio et al., 2013). Yin et al. (2019) study STE-gradient alignment for a two-layer activation-quantized network with Gaussian inputs. Differentiable Hierarchical Visual Tokenization studies hierarchical region selection (Aasan et al., 2025). DVSP ties its continuous coordinates to the token representation itself, recovering the coarse and fine states at its endpoints. Section 4.6 evaluates how its continuous gradients inform local discrete choices.

3 Differentiable Virtual-Slot Partitioning

We first define the native coarse/fine model, then its fixed-slot training representation, and finally the projection used for inference. Figure 2 shows these stages for one parent region and the resulting image-level token count.

Let an image contain R parent regions, each with $J = 4$ fine children. A binary field $b \in \{0,1\}^R$ selects one native coarse token for parent i when $b_i = 0$, or its four fine children when $b_i = 1$. The resulting task model is $T_\theta(x, b)$, with visual-token count

$$N(b) = R + (J-1)\sum_{i=1}^{R} b_i, \tag{1}$$

excluding CLS and other prefix tokens. We use APT's aligned mixed-size patch embeddings, constituent-patch aggregation, positional treatment, and pretrained-ViT interface (Choudhury et al., 2026). DVSP supplies the learned allocation rule. Task-model parameters θ and router parameters ϕ are disjoint.

3.1 A fixed-cardinality representation

Let $c_i, f_{ij} \in \mathbb{R}^d$ be the native coarse and fine content embeddings for parent i and child j, with corresponding additive positions p_i^c, p_{ij}^f. During training, one unconstrained coordinate $\alpha_i \in \mathbb{R}$ controls the content, position, and attention mass of parent i's J virtual slots:

$$z_{ij}(\alpha_i) = (1-\alpha_i)c_i + \alpha_i f_{ij}, \qquad p_{ij}(\alpha_i) = (1-\alpha_i)p_i^c + \alpha_i p_{ij}^f, \tag{2}$$

$$h_{ij}(\alpha_i) = z_{ij}(\alpha_i) + p_{ij}(\alpha_i), \qquad m_{ij}(\alpha_i) = \max\left\{\frac{1+(J-1)\alpha_i}{J}, \varepsilon\right\}. \tag{3}$$

We use $\varepsilon = 10^{-8}$ to keep mass positive and its logarithm defined for unconstrained α_i. Content and position extrapolate linearly outside $[0, 1]$, while the mass path is piecewise linear. Equation 2 uses additive positions; Appendix A.1 describes compatibility with other positional mechanisms. Training always presents JR visual slots to the task model: 196 for $R = 49$ at 224×224, matching ordinary ViT fine-tuning.

We use the proportional-attention form from ToMe (Bolya et al., 2023), weighting each key by a positive mass. For query a and key t,

$$A_{at} = \frac{m_t \exp(q_a^\top k_t / \sqrt{d_h})}{\sum_u m_u \exp(q_a^\top k_u / \sqrt{d_h})}, \tag{4}$$

Here d_h is the query/key dimension of one attention head. Adding $\log m_t$ before softmax implements this weighting. Prefix-token mass is one. For $0 < \alpha_i < 1$, the four interpolated slots have total mass $1 + (J - 1)\alpha_i$, connecting the unit-mass coarse token to the J unit-mass fine tokens.

3.2 Exactness of every native binary state

At $\alpha_i = 0$, the J slots share the same coarse content and position, each with mass $1/J$. Their combined contribution to attention equals that of one unit-mass coarse token, and identical queries receive identical outputs. At $\alpha_i = 1$, the slots are the native fine tokens with unit masses. These local identities extend to arbitrary mixtures of coarse and fine regions.

Binary-state correspondence. Suppose the native and virtual-slot models have aligned content and positional transformations at both endpoints. If all subsequent operations, including the task readout, are deterministic and preserve duplicate-slot equivalence, then in exact arithmetic

$$\mathcal{F}_\theta(x, b) = T_\theta(x, b) \qquad \text{for every } b \in \{0, 1\}^R. \tag{5}$$

The identity equates task outputs from the full-slot and variable-length sequences. Proportional attention, token-wise normalization, feed-forward layers, residual connections, and CLS readout preserve the required equivalence. Appendix A gives the layerwise proof and positional compatibility conditions.

Numerical verification. All-coarse, all-fine, and mixed-partition comparisons with APT's independent native tokenizer satisfy the 10^{-4} absolute/relative FP32 tolerance for the AugReg models. Appendix B reports the measured discrepancies and additional verification settings.

3.3 Continuous learning and a fixed projection rule

An image-only router predicts $\alpha = q_\phi(x)$. We train the full-slot forward with

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{(x,y)} \left[\ell_{\text{task}}\big(\mathcal{F}_\theta(x, q_\phi(x)), y\big) + \frac{\lambda}{R} \sum_{i=1}^{R} |\alpha_i| \right]. \tag{6}$$

Here ℓ_{task} is the task loss. Backpropagation through the real-valued field trains the shared router, while L_1 encourages coordinates toward the coarse endpoint. All slots remain active during training, and the learned coordinates determine the deployed token count after projection. Section 4.1 specifies the classification objective and trainable parameters; Appendix C gives the additional task protocols.

At inference, we project the router field to the coordinate-wise nearest binary vertex:

$$b_i^\star(x) = \mathbf{1}\{\alpha_i(x) \geq \tfrac{1}{2}\}. \tag{7}$$

This rule minimizes squared Euclidean distance in coordinate space, with ties assigned to fine (Appendix A). It uses signed α, including extrapolated values, and gives an image-dependent token count through Eq. 1. The selected partition is executed as $T_\theta(x, b^\star(x))$. Section 4.3 measures the effect of projection on task predictions.

Table 1: ImageNet validation on all 50,000 images. DVSP executes the native binary partition. Tiny DVSP values are means. Token reduction is relative to 196 dense visual tokens.

Backbone	Method	Top-1 (%)	Visual tokens	Reduction
ViT-Ti/16	Original dense	75.50	196	—
	Adapted entropy	74.20	155.41	20.7%
	DVSP	75.32	153.41	21.7%
ViT-S/16	Original dense	81.39	196	—
	DVSP	80.86	136.45	30.4%

4 Experiments and Analysis

We evaluate classification accuracy and token reduction, examine how continuous coordinates form and project to binary partitions, and test the spatial choices and gradient signals learned by DVSP. ImageNet provides the main evidence; Appendix C gives the additional Cityscapes and COCO studies.

4.1 Models and training protocol

We use ImageNet-1k at 224×224 resolution (Deng et al., 2009), with AugReg ViT-S/16 and ViT-Ti/16 (Dosovitskiy et al., 2021; Steiner et al., 2022). Both start from public ImageNet-21k-pretrained, ImageNet-1k-finetuned classifiers, including their pretrained heads. DVSP uses the official APT mixed-patch interface (Choudhury et al., 2026). Frozen MobileNetV3-Small features (Howard et al., 2019) feed a newly initialized spatial MLP. We train the MLP, shared/mixed patch embeddings, and classification head; the ViT blocks, normalization, positional parameters, CLS token, and MobileNet features remain frozen.

Main models train for 50 epochs with global batch 1024 and AdamW (Loshchilov & Hutter, 2019), using constant learning rates 10^{-3} for the MLP and 10^{-6} for embedding/head parameters. The objective combines task CE with L_1 regularization: its weight is zero through epoch 0.5, increases linearly to 0.3 at epoch 1.5, and remains constant thereafter. Zero final-layer weights and unit bias initialize an all-fine field. We evaluate the prescribed final epoch using BF16 autocasting, with FP32 coordinates and loss reductions. Appendix B gives the full protocol and training trajectories.

4.2 Classification accuracy and token reduction

Table 1 compares native execution after the projection in Eq. 7 with the original dense classifiers, evaluated without additional training. Tiny and Small reduce mean visual-token counts by 21.7% and 30.4%, with Top-1 decreases of 0.18 and 0.53 percentage points. Counts exclude CLS throughout.

We also independently adapt a Tiny entropy baseline from the same pretrained initialization, training the same task embedding/head modules for 50 epochs. A fixed entropy cutoff is calibrated without labels to match DVSP's mean token count on a training-image probe. DVSP achieves 1.12 percentage points higher mean Top-1 while using 2.00 fewer tokens on average. This compares independently adapted models at similar mean counts; Section 4.5 isolates spatial selection within each trained task model.

The original dense comparison includes both tokenization changes and embedding/head adaptation. The next experiment separates these effects by evaluating all-fine execution with the adapted task weights.

4.3 Separating projection from native execution

The projection and spatial analyses use the same trained model for each backbone. The analyzed Tiny model contributes to the mean in Table 1; Tables 2 and 4 report its individual results.

Table 2: Projection chain for the analyzed models on the complete validation set. Rows B–E share adapted task weights. A–D execute 196 visual slots/tokens; E executes 154.03 tokens for Tiny and 136.45 for Small on average. Values are BF16 evaluations; FP32 independent-tokenizer checks are reported separately.

		ViT-Ti/16		ViT-S/16	
	Forward	CE	Top-1 (%)	CE	Top-1 (%)
A	Original dense	0.94211	75.500	0.67156	81.388
B	Adapted all-fine	0.93357	75.618	0.66606	81.462
C	Continuous full slots	0.95716	75.064	0.69932	80.736
D	Binary full slots	0.94919	75.330	0.69159	80.858
E	Native binary sequence	0.94937	75.356	0.69145	80.856

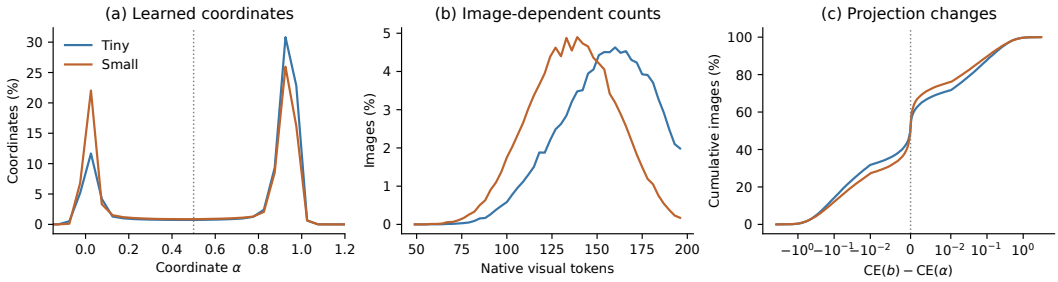


Figure 3: Tiny and Small model diagnostics on 50,000 images. (a) Learned coordinates; the dotted line marks the signed half-threshold. (b) Native visual-token counts. (c) Empirical cumulative distribution of binary-minus-continuous CE, displaying all images on a symmetric logarithmic horizontal scale with a linear neighborhood of zero. Negative values indicate lower CE after projection; the CDF at zero gives the fraction with non-increasing CE.

For image x and label y, define $j_x(a) = \ell_{\text{task}}(\mathcal{F}_\theta(x, a), y)$ and let b be the projected field. Deployment has two distinct transitions:

$$\ell_{\text{task}}(T_\theta(x, b), y) - j_x(\alpha) = \underbrace{j_x(b) - j_x(\alpha)}_{\text{projection change}} + \underbrace{\ell_{\text{task}}(T_\theta(x, b), y) - j_x(b)}_{\text{native-execution change}=0}. \tag{8}$$

Equation 5 makes the native-execution term zero in exact arithmetic. We measure the projection term below; Appendix A.3 relates it to the continuous path.

Table 2 traces the deployment chain. Adapted all-fine execution slightly improves on the original dense reference (A→B). Learned continuous coordinates change task accuracy while retaining all slots (B→C); binary projection recovers part of that accuracy change (C→D), and native execution realizes the reduced token count (D→E).

Projection lowers mean CE by 0.00796 for Tiny and 0.00773 for Small. The paired image-bootstrap 95% interval for binary-minus-continuous CE is $[-0.00991, -0.00613]$ for Tiny and $[-0.00965, -0.00585]$ for Small. Figure 3 shows the learned fields, token counts, and per-image loss changes. Projection improves aggregate CE in these models, with individual changes spanning both signs. Small BF16 differences between binary full-slot and native outputs arise from finite-precision arithmetic; the FP32 verification protocol is given in Appendix B.

4.4 Formation of coarse and fine modes

How does router optimization affect the emergence of a bimodal coordinate distribution? We compare ViT-S/16 models with MLP learning rates 10^{-4} and 10^{-3}, holding initialization, global batch 1024, embedding/head learning rate, and the CE+L_1 objective fixed. Both are evaluated after six epochs on the same 5,000-image validation monitor.

Table 3: Router learning-rate comparison at epoch 6 on ViT-S/16, with global batch 1024. Coordinate fractions pool all 49 parents per image. Tokens and Top-1 use native signed-half-threshold execution on the same 5,000 images.

MLP LR	$\lvert\alpha\rvert < 0.1$ (%)	$0.4 \le \alpha \le 0.6$ (%)	Visual tokens	Top-1 (%)
10^{-4}	6.42	7.87	161.05	81.10
10^{-3}	22.86	4.58	147.02	81.22

Table 4: Spatial selection for the analyzed models at identical per-image token counts. All entries use the complete validation set and the same task weights within each backbone.

	ViT-Ti/16		ViT-S/16	
Spatial selection	Top-1 (%)	CE	Top-1 (%)	CE
Random	72.616	1.06418	78.568	0.78977
Entropy	74.326	0.98940	79.826	0.73482
DVSP	**75.356**	**0.94937**	**80.856**	**0.69145**

At 10^{-3}, a pronounced mode near zero develops alongside the fine mode near one. Table 3 quantifies this separation: the near-zero fraction rises from 6.42% to 22.86%, while the fraction around the deployment boundary falls from 7.87% to 4.58%. Mean native token count decreases from 161.05 to 147.02 at similar Top-1 accuracy. This controlled comparison supports the chosen router learning rate for forming a more separated coarse/fine field within the same training horizon. The field remains continuously valued during optimization and is projected only for deployment.

4.5 Do learned partitions select useful spatial locations?

We replace DVSP's spatial assignment with entropy or random selection, giving each control exactly the same fine-parent count on every image and holding adapted task weights fixed. All three use the same native executor. This intervention isolates spatial membership within the trained task model.

In Table 4, DVSP exceeds entropy by 1.03 percentage points on both backbones. Paired 95% intervals are $[0.848, 1.218]$ for Tiny and $[0.836, 1.222]$ for Small. The corresponding CE advantages are 0.04002 and 0.04336. These interventions show that the learned field selects useful regions within the adapted task model.

Qualitative spatial choices. Figure 4 illustrates how different partitions preserve different evidence for classification. The examples come from the frozen DeiT-S study in Appendix C, with identical per-image counts across selectors. They show successful DVSP cases; Table 5 provides the aggregate comparison, including the feature scorer.

4.6 Do continuous gradients inform native discrete decisions?

To examine the learning signal behind spatial choices, we compare continuous and identity-STE endpoint gradients within two continuously trained ViT-Ti models, with MLP learning rates 10^{-4} and 10^{-3}. Evaluation uses 1,024 images, holding the model and image fixed within each comparison. Appendix B.1 specifies the protocol.

Using the task loss j_x defined in Section 4.3, we evaluate coordinate gradients at the learned field α and its projected vertex b:

$$g^{\mathrm{cont}} = \nabla_a j_x(a)|_{a=\alpha}\,, \qquad g^{\mathrm{STE}} = \nabla_a j_x(a)|_{a=b}\,. \tag{9}$$

Let $H(\alpha) = b$ denote the signed half-threshold. The endpoint signal uses the hard forward $j_x(H(\alpha))$ with the backward substitution $DH = I$ in the same DVSP model.

The signals agree at a shared differentiable binary vertex and otherwise evaluate different representations. For each region, we measure the native CE change from flipping b_i. With direction $d_i = 1 - 2b_i$, the directional score $s_i = d_i g_i$ ranks these changes against the measured difference

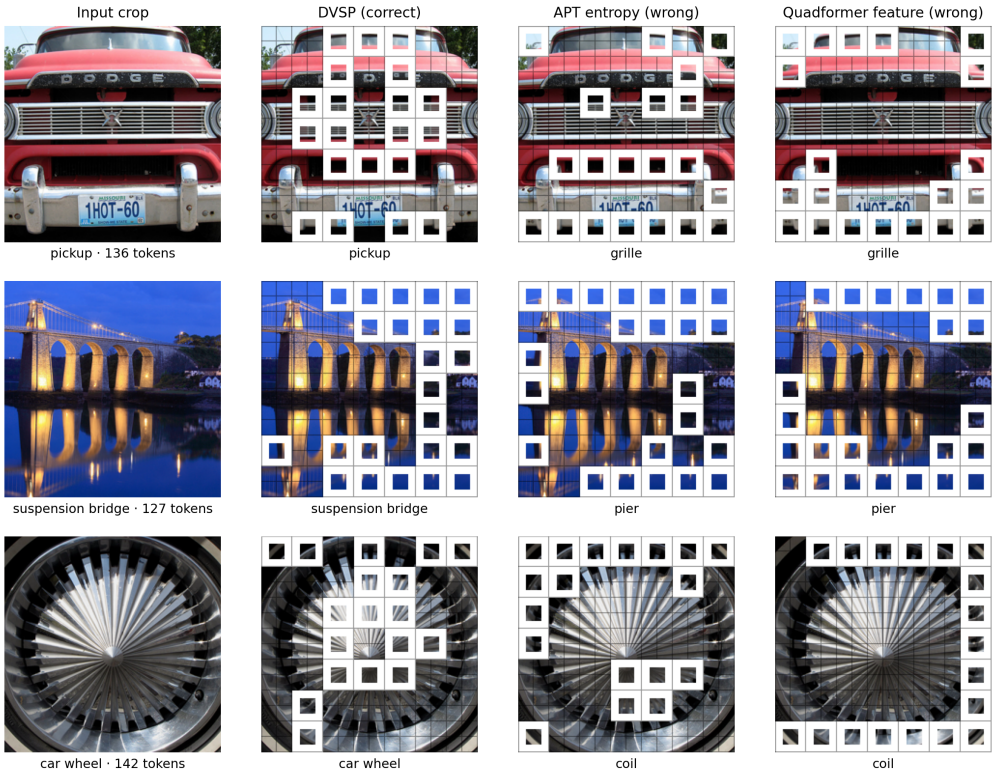


Figure 4: **Spatial choices in a frozen DeiT-S model.** Illustrative DVSP successes from Table 5. Columns show the input, signed-half-threshold DVSP, entropy, and fixed Quadformer feature partitions at identical per-image counts and task weights. Labels give ground truth and predictions. Fine regions occupy four cells; coarse regions appear as centered thumbnails.

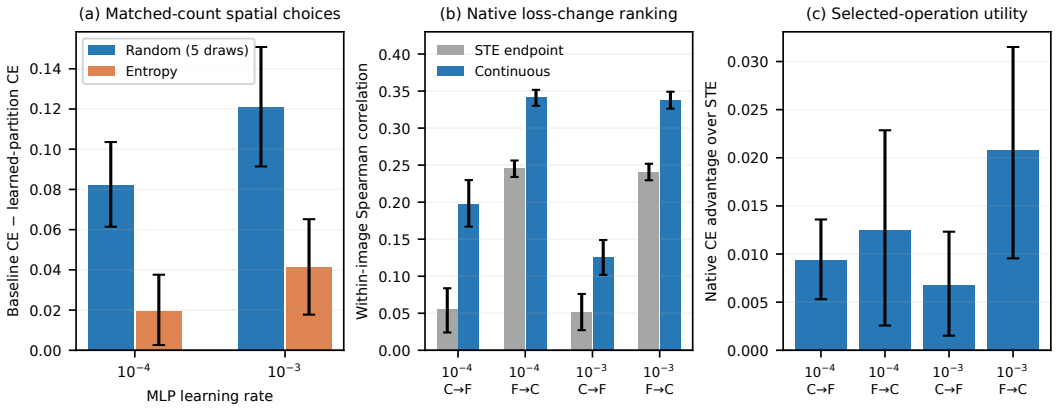


Figure 5: **Continuous gradients and native decisions.** Two ViT-Ti models, identified by MLP learning rate, on 1,024 images. (a) Matched-count spatial controls. (b) Correlation with native single-region CE changes. (c) Native CE advantage of the operation selected by the continuous gradient over the STE-selected operation. C→F and F→C denote refinement and coarsening. Bars show means and paired image-bootstrap 95% intervals. The gradient probes use labels offline; routing uses images alone.

between $b + d_i e_i$ and b. The continuous gradient is evaluated at α, while the discrete flip starts at b; this probe tests their empirical relationship. Refinement and coarsening are evaluated separately to keep candidate token costs identical.

Figure 5 shows higher within-image rank correlation for the continuous gradient in both directions and models. At MLP learning rate 10^{-3}, correlation increases from 0.0515 to 0.1252 for refinement and from 0.2408 to 0.3380 for coarsening. Selecting the lowest-scoring operation with the continuous gradient also yields lower native CE, with mean advantages of 0.00678 and 0.02083, respectively; both paired intervals exclude zero. The other model exhibits the same direction of improvement.

The continuous signal improves both local operation ranking and the task loss of the selected operation in these fixed-model probes, complementing the spatial-selection results in Section 4.5.

5 Conclusion

DVSP represents coarse/fine image partitions in one continuous, fixed-cardinality model. Coupled content, position, and attention mass give every compatible binary state an exact native variable-length execution. This correspondence provides a discrete execution target for task-driven learning while keeping the effect of projection measurable.

ImageNet experiments show accurate classification with fewer visual tokens, useful learned spatial choices, and continuous gradients that improve local discrete-operation ranking and selection in the tested models. Together, these results support virtual slots as a representation for learning adaptive input resolution.

AI use statement

Generative AI tools were used to assist literature discovery, code generation, experiment orchestration, visualization, and language editing. The authors are responsible for the final content and its verification.

References

Marius Aasan, Martine Hjelkrem Tan, Nico Catalano, Changkyu Choi, and Adín Ramírez Rivera. Differentiable hierarchical visual tokenization. In *Advances in Neural Information Processing Systems*, volume 38, pp. 97391–97420, 2025. doi: 10.52202/085713-2930. URL https://papers.nips.cc/paper_files/paper/2025/hash/7efb80373ab158f5ee06fb8aac6b28a0-Abstract-Conference.html.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. URL https://arxiv.org/abs/1308.3432.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

Rohan Choudhury, JungEun Kim, Jinhyung Park, Eunho Yang, László A. Jeni, and Kris M. Kitani. Faster vision transformers with adaptive patches. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10009361.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Cordts_The_Cityscapes_Dataset_CVPR_2016_paper.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit,

and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Jakob Drachmann Havtorn, Amélie Royer, Tijmen Blankevoort, and Babak Ehteshami Bejnordi. MSViT: Dynamic mixed-scale tokenization for vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 838–848, 2023. URL https://openaccess.thecvf.com/content/ICCV2023W/NIVT/html/Havtorn_MSViT_Dynamic_Mixed-Scale_Tokenization_for_Vision_Transformers_ICCVW_2023_paper.html.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1314–1324, 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Howard_Searching_for_MobileNetV3_ICCV_2019_paper.html.

Shihua Huang, Yongjie Hou, Longfei Liu, Xuanlong Yu, and Xi Shen. Real-time object detection meets DINOv3. *arXiv preprint arXiv:2509.20787*, 2025. URL https://arxiv.org/abs/2509.20787.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.01144.

Dong Hoon Lee and Seunghoon Hong. Learning to merge tokens via decoupled embedding for efficient vision transformers. In *Advances in Neural Information Processing Systems*, volume 37, pp. 54079–54104, 2024. doi: 10.52202/079017-1713. URL https://arxiv.org/abs/2412.10569.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=E9GCubxR9s.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In *European Conference on Computer Vision*, pp. 116–131, 2018. URL https://arxiv.org/abs/1807.11164.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.00712.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13937–13949, 2021. URL https://arxiv.org/abs/2106.02034.

Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4613–4622, 2023. doi: 10.1109/CVPRW59228.2023.00486. URL https://arxiv.org/abs/2304.00287.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022. URL https://arxiv.org/abs/2106.10270.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Strudel_Segmenter_Transformer_for_Semantic_Segmentation_ICCV_2021_paper.html.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357, 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1903.05662.

Shicheng Yin, Kaixuan Yin, Yang Liu, Weixing Chen, and Liang Lin. DART: Differentiable dynamic adaptive region tokenizer for vision foundation models. *arXiv preprint arXiv:2506.10390*, 2025. doi: 10.48550/arXiv.2506.10390. URL https://arxiv.org/abs/2506.10390.

A Exact Vertices and Projection

Appendix A derives the representation identities used in Section 3. Appendix B gives the classification and gradient-probe protocols, and Appendix C reports the additional frozen-task studies.

A.1 Exactness of every native binary vertex

For a query a, let $\ell_{ai} = q_a^\top k_i/\sqrt{d_h}$. Consider a coarse group whose J identical keys share this logit and value vector v_i, each with mass $1/J$. Its contributions to the attention denominator and numerator are

$$\sum_{j=1}^{J} \frac{1}{J} e^{\ell_{ai}} = e^{\ell_{ai}}, \qquad \sum_{j=1}^{J} \frac{1}{J} e^{\ell_{ai}} v_i = e^{\ell_{ai}} v_i. \tag{10}$$

Both equal the contributions of one native unit-mass coarse token. Other keys are unchanged. Duplicate queries also yield identical outputs. Token-wise normalization, feed-forward maps, and residual connections preserve duplicate equality; repeating this argument proves equality at every block and at an output operator that respects duplicate equivalence, including CLS readout. A normalized mass-weighted pooling operator also respects this equivalence; an unweighted mean over all virtual slots generally does not. Fine groups already equal the native children with unit masses. The local argument holds for any mixture of coarse and fine parents, giving Eq. 5. It assumes deterministic computation and no slot-index-specific operation beyond the encoded states.

If b contains K fine parents, the native sequence has $JK + (R - K) = R + (J - 1)K$ visual tokens, each with unit attention mass. Equation 5 identifies this sequence's task output with the full-slot binary forward at the same b.

Compatibility of positional operations. The duplicate argument requires identical positional transformations for every copy of a coarse token and the native child transformations at the fine endpoint. For additive positions this follows from Eq. 2. With rotary positions, coarse copies must use the same coarse coordinate in every affected query/key rotation; fine slots must use their native child coordinates. Slot-index-dependent biases or decoder operations require the same compatibility condition. Interpolating spatial coordinates before rotary encoding is one endpoint-compatible construction.

A.2 Nearest binary vertex

For any $\alpha \in \mathbb{R}^R$, squared Euclidean distance is $\sum_i (\alpha_i - b_i)^2$. Comparing $b_i = 1$ with $b_i = 0$ gives $1 - 2\alpha_i$. Thus $b_i = 0$ is the unique minimizer below one half and $b_i = 1$ is the unique minimizer above it. At equality both minimize the distance; our rule assigns the tie to fine. Independence of the summands proves global minimality of coordinate distance over all binary assignments, including for extrapolated coordinates. The minimum squared distance is

$$d^2_{\mathrm{vert}}(\alpha) = \sum_i \min\{\alpha_i^2, (\alpha_i - 1)^2\}. \tag{11}$$

The selected membership is unchanged under coordinate perturbations of magnitude at most η if $|\alpha_i - \frac{1}{2}| > \eta$.

A.3 Loss change under endpoint projection

Fix the image, task-model parameters, and selected binary field b. Write $j_x(a) = \ell_{\mathrm{task}}(\mathcal{F}_\theta(x, a), y)$ and $q(t) = j_x(\alpha + t(b - \alpha))$. Assume q is absolutely continuous and the chain rule holds almost everywhere. Then $q(1) - q(0) = \int_0^1 q'(t)\, dt$. The chain rule yields

$$q'(t) = \nabla_a j_x(\alpha + t(b - \alpha))^\top (b - \alpha) \tag{12}$$

almost everywhere. Substituting $q(0) = j_x(\alpha)$ and $q(1) = \ell_{\mathrm{task}}(T_\theta(x, b), y)$ gives

$$\ell_{\mathrm{task}}(T_\theta(x, b), y) - j_x(\alpha) = \int_0^1 \nabla_a j_x(\alpha + t(b - \alpha))^\top (b - \alpha)\, dt. \tag{13}$$

Equation 13 expresses the task-loss projection term of Eq. 8 as a path integral with fixed model parameters. The positive mass floor keeps the forward defined even for extrapolated coordinates; derivatives at isolated knot crossings do not affect the integral. The loss change depends on gradients along the entire path, which motivates measuring it directly in Section 4.3.

B Classification Experimental Details

Initialization and trainable parameters. Both ViT-S/16 and ViT-Ti/16 use AugReg checkpoints pretrained on ImageNet-21k and finetuned on ImageNet-1k. We retain the pretrained 1000-class head and initialize the official APT mixed-patch interface from the same checkpoint. An initial all-fine comparison checks the conversion against the original timm forward. The router uses torchvision MobileNetV3-Small's first eleven feature blocks and a newly initialized $96 \to 96 \to 1$ MLP with GELU. Its final weight is zero and final bias one. MobileNet features remain in evaluation mode. The task's shared patch projection, mixed-patch attention/aggregation modules, and head are trainable; its transformer blocks, final normalization, positional parameters, CLS token are frozen. The frozen transformer remains in the task-gradient path.

Optimizer and image processing. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay 0.05 except for bias and one-dimensional parameters. We clip the joint trainable gradient norm at 1.0 after any accumulation. Learning rates remain constant. The L_1 coefficient follows

$$\lambda(e) = 0.3 \, \min\{1, \max\{0, e - 0.5\}\}, \tag{14}$$

where e is training progress in epochs. Thus the regularization ramp spans epochs 0.5–1.5. No Mixup, CutMix, label smoothing, auxiliary teacher, or detached utility is used.

Validation follows the public checkpoint's bicubic interpolation, crop ratio 0.9, and channel-wise mean/std 0.5. The router receives the same augmented image after applying its pretrained normalization. BF16 autocast is used for all main training and validation, with FP32 coordinates and loss reductions where required by the implementation.

Batching and endpoint. Both backbones in the main experiments train on eight RTX 4090 GPUs with 128 images per GPU and no gradient accumulation, giving global batch 1024. Incomplete batches are normalized by their actual sample count. The reported endpoint is epoch 50.

Monitoring and paired evaluation. Training records augmented online CE, Top-1, and projected counts at every epoch. A fixed 5,000-image validation monitor is evaluated every two epochs before the final epoch and is not used to select the endpoint. Final evaluations use all 50,000 validation images. Projection and spatial comparisons retain predictions and CE for the same images. We use 2,000 paired image-bootstrap resamples to summarize mean CE and accuracy differences conditional on each trained checkpoint. These intervals are unadjusted, pointwise intervals for the stated comparisons.

Router learning-rate comparison. Section 4.4 uses separate ViT-S/16 runs initialized from the same pretrained weights and all-fine router. Only the MLP learning rate changes; the trainable modules, optimizer, task learning rate, and L_1 ramp match the main protocol. Both runs use four RTX 4090 GPUs, 128 images per GPU, and two accumulation steps, retaining global batch 1024. The comparison evaluates both models at epoch 6 on the fixed validation monitor.

Independently adapted entropy baseline. The Tiny baseline starts independently from the same AugReg pretrained checkpoint and trains the shared/mixed patch embeddings and classification head for 50 epochs with the same task learning rate, optimizer, global batch, and preprocessing as DVSP. The router network is frozen and entropy determines binary partitions. Before training, a fixed entropy cutoff is calibrated on 10,000 training images without labels to match the analyzed Tiny model's total fine-parent count (Table 2). The cutoff remains fixed during training and validation. Both methods average 155.23 visual tokens on this calibration probe; the entropy baseline averages 155.41 on the full validation set. This calibration matches the probe mean, while each method determines its own per-image validation count. We evaluate the prescribed final epoch.

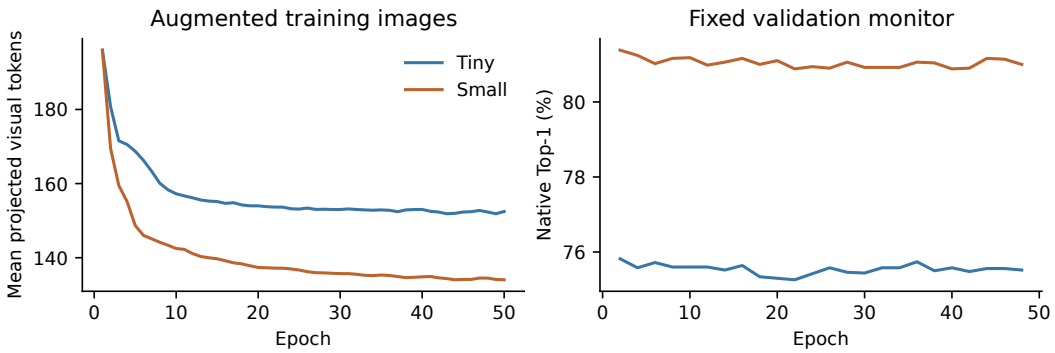


Figure 6: Measured training trajectories for the analyzed models. Left: projected counts on augmented training images. Right: native accuracy on the fixed validation monitor. Full-validation final results are reported separately in Table 1.

Spatial controls. Entropy ranks parent regions using APT's grayscale entropy score. Random selection uses a deterministic generator indexed by validation-image order. Each control receives exactly the learned number of fine parents for that image, and uses the same adapted task weights and native executor. This full-validation control uses one fixed random partition per image; the gradient study below averages five random partitions. Neither control uses labels to choose its regions.

Coordinate and projection statistics. The analyzed Tiny and Small fields have 21.56% and 32.30% of coordinates with $|\alpha| < 0.1$. Their native-count 10th/50th/90th percentiles are 121/157/184 and 106/136/166. Median binary-minus-continuous CE is -0.000072 and -0.000061; CE increases on 47.66% and 47.24% of images. Projection changes the predicted class on 2,485 Tiny images and 2,045 Small images. These paired distributions distinguish an aggregate CE improvement from the effect on an individual image.

Numerical verification. We compare the fixed-slot binary forward with APT's independently tokenized native sequence in FP32, using all-coarse, all-fine, and predetermined mixed masks. Absolute and relative tolerances are 10^{-4}. On the analyzed epoch-50 models, the largest absolute logit discrepancies across these checks are 2.93×10^{-5} for Tiny and 8.30×10^{-5} for Small. The initial all-fine conversion is also compared with the original dense classifier. Separate FP64 checks on DeiT-S/16 and high-resolution ViT-B/16 give maximum relative logit discrepancies of 8.6×10^{-15} and 2.7×10^{-14}, respectively; these checks use different checkpoints from the AugReg study.

B.1 Learned-field native utility analysis

Models and evaluation. The gradient study uses a separate controlled learning-rate comparison of ViT-Ti fields with MLP learning rates 10^{-4} and 10^{-3}. Both use global batch 1024, embedding/head learning rate 10^{-6}, and the L_1 schedule in Eq. 14. The models differ in router MLP learning rate. Every signal comparison holds the model and image fixed. Figure 5 evaluates 1,024 images. Intervals use 3,000 paired image-bootstrap resamples and quantify image-sampling uncertainty.

Routing interventions. For each image, all assignments use exactly the learned partition's number of fine parents and the same trained task weights. Random routing averages five fixed random partitions; entropy ranks the parent regions at the same count. These interventions evaluate spatial assignments within the adapted task model. The learned router and both replacement rules use no evaluation labels to choose their partitions.

Gradient and native references. We evaluate all 49 single-parent flips, separating coarse-to-fine and fine-to-coarse operations. Within each direction, we compute image-wise Spearman correlation and select the operation with the lowest predicted CE change. Selection uses every image with at least one eligible region. Ranking requires at least two candidates and nonconstant signals. Gradients are task-CE derivatives, excluding L_1, and are computed offline using image labels.

Main signals use BF16 autocast with FP32 coordinates and loss reductions. Native CE changes are measured in FP32 with the official APT tokenizer and standard attention; FP32 gradient comparisons are also retained. Binary full-slot and native logits are checked at the base partition and every flipped partition. The sign-statistic threshold is the larger of 10^{-4} and twice the measured maximum logit discrepancy; changes below it are excluded. This keeps small native changes distinguishable from execution error.

C Supplementary Task Evaluations

These studies examine spatial selection with frozen task networks. Their protocols differ from the main AugReg study and use validation-dependent development. Scorer comparisons change the selected regions while holding the task model and each image's or crop's fine-parent count fixed; they do not compare complete native tokenization systems.

C.1 Frozen-task ImageNet classification

We use an APT-finetuned DeiT-S/16 at resolution 224 (Touvron et al., 2021; Choudhury et al., 2026). All task parameters are frozen. The router consists of the first eleven feature blocks of ImageNet-pretrained MobileNetV3-Small and a fresh $96 \to 96 \to 1$ GELU MLP. Unlike the AugReg study, the router feature extractor is trained, with batch normalization kept in evaluation mode. AdamW uses feature/head learning rates $10^{-4}/10^{-3}$, weight decay 0.05 with bias and normalization exclusions, and batch 512. A one-epoch warm-up precedes cosine decay over ten epochs; augmentation uses random resized crops and horizontal flips. The objective is CE plus $0.3\,\mathrm{mean}(|\alpha|)$.

The signed nearest-vertex partition is evaluated on all 50,000 validation images.

Table 5: Frozen-task ImageNet evaluation. Binary spatial controls use the same fine-parent count on every image and the same native executor. The continuous row executes all 196 slots.

Forward or spatial score	Mean tokens	CE	Top-1 (%)
Native all-fine	196	0.84489	79.734
Continuous full slots	196	0.88671	78.806
Binary: random	123.19	0.96356	77.132
Binary: APT entropy	123.19	0.91022	78.298
Binary: Quadformer pixel	123.19	0.92556	77.914
Binary: Quadformer feature	123.19	0.88544	78.932
Binary: DVSP	123.19	0.88286	78.864

The binary-minus-continuous Top-1 difference is $+0.058$ percentage points (two-sided exact McNemar $p = 0.266$), while the mean CE difference is -0.00385 (paired 95% interval $[-0.00481, -0.00289]$). Projection does not establish a Top-1 improvement. DVSP exceeds random, entropy, and pixel selection, while the paired comparison with the feature scorer detects no significant accuracy difference ($p = 0.470$). These are exploratory, unadjusted validation comparisons. CE intervals use the standard error of paired per-image differences.

Projection component interventions. We replace just one of the content, position, or attention-mass paths by its binary value while retaining real coordinates in the others. These interventions use the frozen task weights and router field from Table 5, the same 50,000 images, and 196 slots. Table 6 shows these interventions; they are not an additive decomposition of the full projection effect.

Projecting content alone lowers CE, whereas projecting mass alone slightly raises it. The full projection therefore cannot be attributed solely to removing fractional attention masses; these paths interact in the frozen model.

Additional spatial visualizations. Figure 7 shows further examples with the frozen DeiT-S task model, identical per-image token counts, and the same binary endpoint evaluator. Aggregate comparisons appear in Table 5.

Table 6: Full-slot component interventions on the frozen ImageNet task model. All rows execute 196 slots; binary means that all three paths use projected coordinates.

Coordinate paths	CE
Continuous	0.88671
Content projected	0.88313
Position projected	0.88661
Mass projected	0.88733
All binary	0.88286

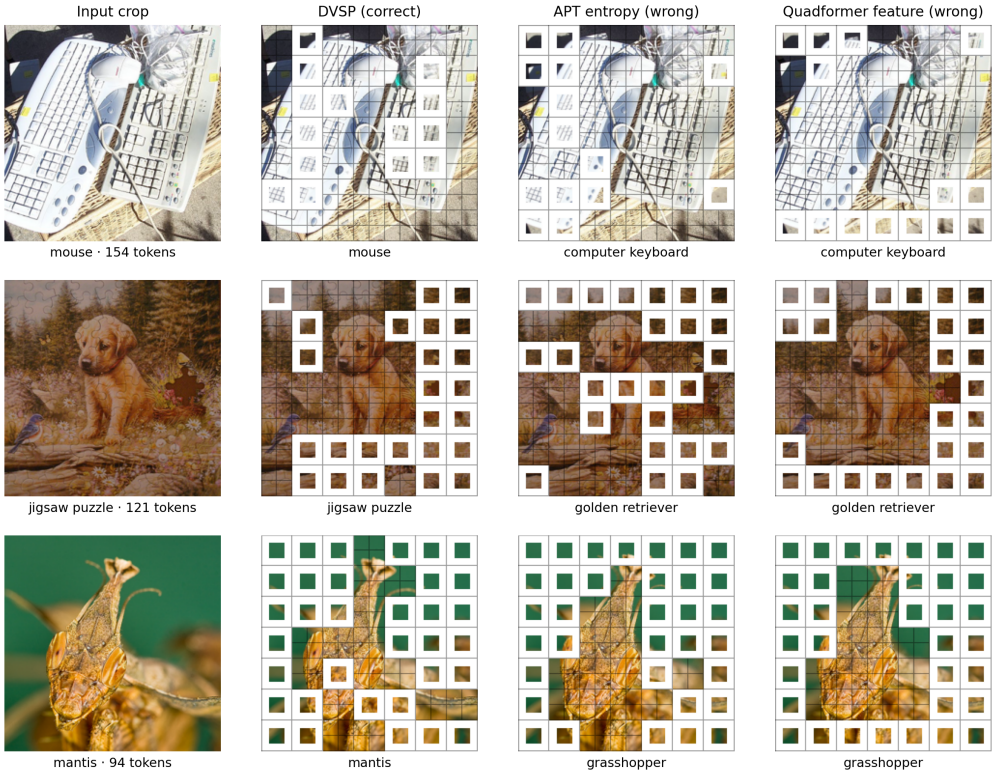


Figure 7: Additional illustrative classification examples under the protocol of Figure 4. Columns show input, DVSP, entropy, and fixed Quadformer feature selection, respectively. Each row uses an identical token count across the three partitions; text identifies ground truth and predicted classes.

C.2 Frozen-task semantic segmentation

We use the published Segmenter ViT-S/16 Cityscapes model (Strudel et al., 2021; Cordts et al., 2016), freezing both encoder and mask-transformer decoder. The MobileNet router backbone and MLP train for ten epochs with AdamW, learning rates 10^{-4} and 10^{-3}, weight decay 0.05, and a uniform L_1 penalty. The final MLP weight is initially zero and its bias one. Training uses random 768×768 crops, four GPUs, FP16 autocasting, and FP32 attention masses. Evaluation covers 500 images using 768-pixel windows and stride 512.

The dense reference gives 76.491% mIoU with 2,304 tokens per crop; continuous full slots give 74.374%. Table 7 compares binary spatial selection. Segmentation evaluates all full-slot binary endpoints. Reported counts are the partition-implied token counts $R + 3\sum_i b_i$, describing the corresponding coarse/fine partitions; these experiments assess spatial selection quality. Entropy uses 512-bin grayscale statistics in each 32×32 parent. The fixed feature score uses ImageNet-pretrained ShuffleNetV2-$0.5\times$ (Ma et al., 2018) and the Quadformer semantic-blur criterion (Ronen et al., 2023).

Table 7: Frozen-task Cityscapes at the signed half-threshold. All scores have the same count per crop (1044.90 mean partition-implied tokens).

Spatial score	mIoU (%)
APT entropy	72.379
Quadformer feature	72.487
DVSP	74.770

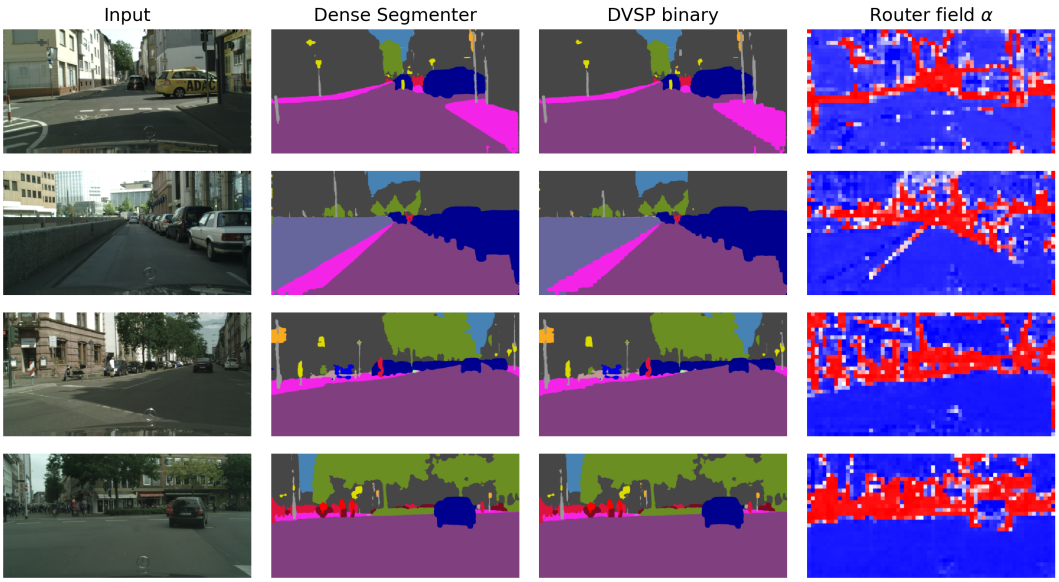


Figure 8: **Alternative-threshold segmentation examples ($|\alpha_i| \geq 0.1$).** Rows show distinct validation images; columns show the input, dense Segmenter prediction, DVSP binary full-slot prediction, and router field. Dense and DVSP predictions use the same frozen task weights and sliding-window evaluation. The illustrated DVSP partitions use $|\alpha_i| \geq 0.1$. Field colors are normalized separately within each scene between its second and 98th percentiles, with red indicating larger coordinates and blue smaller coordinates.

Alternative-threshold segmentation examples. Figure 8 compares dense and DVSP predictions on four Cityscapes validation scenes using the same frozen Segmenter. The visualized binary predictions use $|\alpha_i| \geq 0.1$; Table 7 uses the signed half-threshold.

C.3 Object Detection

The detection study uses DEIMv2-S at 640×640 with a DINOv3-distilled ViT-Tiny backbone (Huang et al., 2025; Lin et al., 2014). The entire task network is frozen. Starting from a router trained for 12 epochs with a frozen task model, we train its MobileNet features and MLP for three further FP32 epochs. AdamW uses feature/head learning rates $10^{-4}/10^{-3}$, weight decay 0.05, batch 128, and L_1 coefficient 0.1. Task gradients come from the final decoder's bounding-box, generalized-IoU, fine-grained localization, and MAL losses; auxiliary, denoising, encoder, and pre-decoder losses are detached from the router.

Evaluation covers all 5,000 COCO val2017 images. Each scorer receives the same per-image count as signed half-threshold DVSP and uses compact binary execution (Table 8). All-fine execution of the same task checkpoint gives 50.764 AP with 1,600 visual tokens. The detector interpolates coarse and fine spatial coordinates before rotary position encoding, rather than additive positional embeddings. Both endpoints must recover their native positional transformations; the independent FP64 tokenizer checks reported in this paper cover classification interfaces.

Table 8: COCO val2017 box AP at matched fine-parent counts per image. All scores use the same compact binary executor and 1017.5 mean visual tokens.

Spatial score	AP	AP_{50}	AP_{75}
APT entropy	49.362	67.204	53.414
Quadformer feature	49.939	67.808	53.944
DVSP coordinate	**50.389**	**68.150**	**54.445**

Qualitative detection results. Figure 9 shows selected COCO examples from the same evaluation. DVSP identifies the partially visible keyboard and boat, and recovers both chairs and the refrigerator in the street scene. The spatial controls miss or misclassify some of these objects at the shared display threshold.

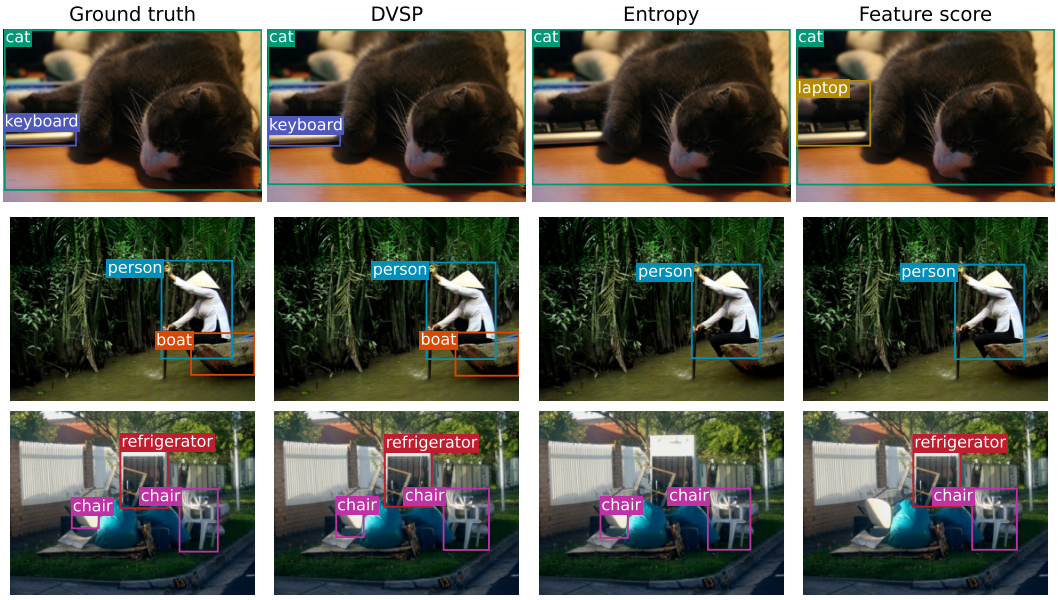


Figure 9: **COCO detection examples.** Selected DVSP successes under the protocol of Table 8. Columns show ground-truth boxes, DVSP, entropy selection, and the fixed Quadformer feature scorer. All predictions use the same frozen detector and identical per-image fine-parent counts. Every prediction with confidence ≥ 0.5 is displayed; colors denote object classes.